Packet Switching

- The network layer is responsible for **host-to-host delivery** and for **routing the packets** through the routers.
- A connecting device such as a router acts as a switch. When a packet arrives from one of its ports (interface), the packet is forwarded through another port to the next switch (or final destination).
- A process called switching occurs at the connecting device.

SWITCHING

The passage of a message from a source to a destination **involves many decisions.** When a message reaches a connecting device, a decision needs to be made to select one of the output ports through which the packet needs to be send out.

Circuit Switching

- A physical circuit (or channel) is established between the source and destination of the message before the delivery of the message. After the circuit is established, the entire message, is transformed from the source to the destination.
- The circuit switching was never implemented at the network layer; it is mostly used at the physical layer.
- In circuit switching, the whole message is sent from the source to the destination without being divided into packets.

Packet Switching

- The network layer in the Internet today is a packet-switched network.
- In packet switching, the message is first divided into manageable packets at the source (normally called **Datagrams** in the network layer) before being transmitted. The packets are assembled at the destination.
- The connecting devices in a packet-switching network still need to decide how to route the packets to the final destination.
- Today, a packet-switched network can use two different approaches to route the packets: the datagram approach and the virtual circuit approach. We discuss both approaches in the next section.

Packet Switching at Network Layer

- The network layer is designed as a packet-switched network.
- The packet-switched network layer of the Internet was originally designed as a ***Connectionless service,*** but recently there is a tendency to change this *to a* ***Connectionoriented service.*** We first discuss the dominant trend and then briefly discuss the new one.

Datagram Approach: Connectionless Service

- When the Internet started, the network layer was designed to provide a **connectionless service.**
- The network layer protocol **treats each packet independently**.
- The packets in a message **may or may not travel the same path to their destination.**
- The switches in this type of network are called **routers**.
- Each packet is routed based on the information contained in its header.
- In a connectionless packet-switched network, the forwarding decision is based on the **destination address of the packet**.

Figure 18.3 *A connectionless packet-switched network*

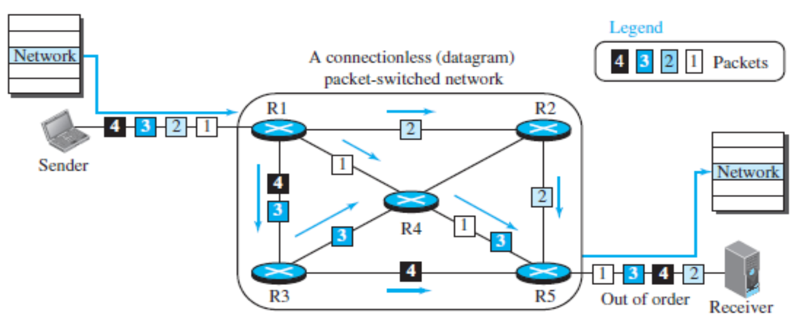
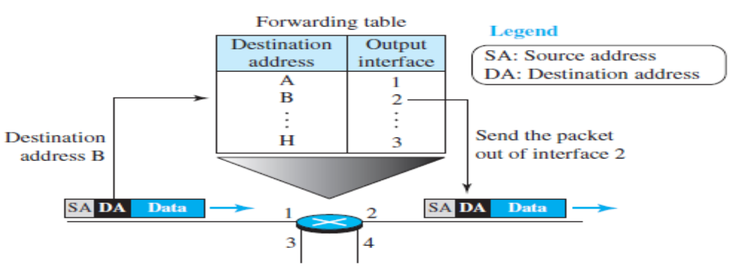


Figure 18.4 *Forwarding process in a router when used in a connectionless network*

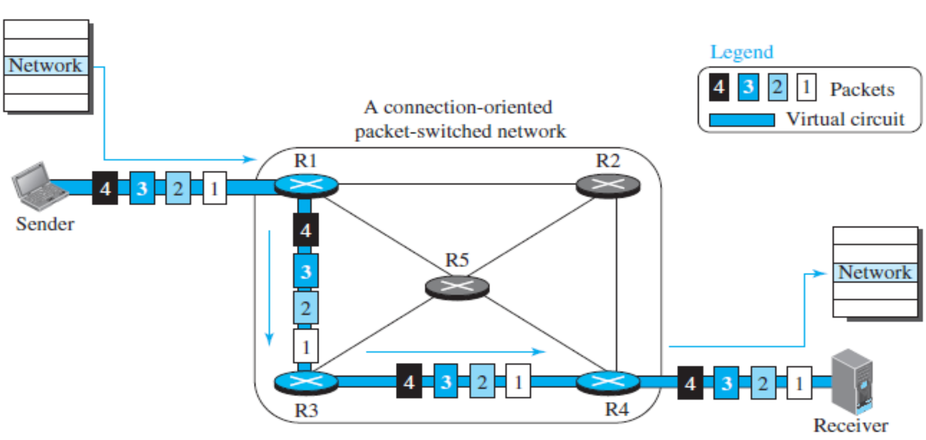


In the datagram approach, the forwarding decision is based on the destination address of the packet.

Virtual-Circuit Approach: Connection-Oriented Service

- In a **connection-oriented service, there is a relation between all packets belonging to a message.**
- Before all datagrams in a message can be sent, **a virtual connection should be set up to define the path for the datagrams.**
- After connection setup, the datagrams can follow the same path.
- In this type of service, not only must the packet contain the source and destination addresses, it must also contain ***a flow label, a virtual circuit identifier*** that defines the virtual path the packet should follow.
- In a connection-oriented packet switched network, **the forwarding decision** is based on the ***label*** of the packet.

Figure 18.5 *A virtual-circuit packet-switched network*



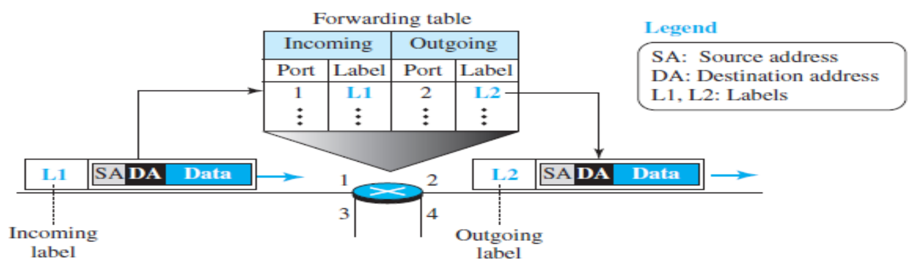
To create a connection-oriented service, **a three-phase process is used**: ***setup, data transfer*, and *teardown*:**

- In the **setup phase**, ***the source and destination addresses*** of the sender and receiver is used to make table entries for the connection-oriented service.
- In the **teardown phase**, the source and destination **inform** the router to delete the corresponding entries.
- **Data transfer** occurs between these two phases.

Setup Phase

In the **setup phase**, **a router creates an entry for a virtual circuit**. For example, suppose source A needs to create a virtual circuit to destination B. ***Two auxiliary packets*** need to be exchanged between the sender and the receiver: the request packet and the acknowledgment packet.

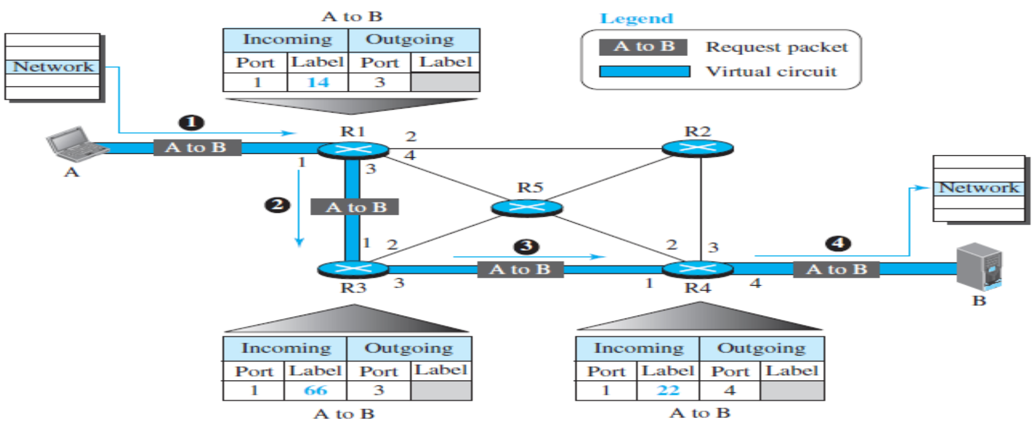
Figure 18.6 *Forwarding process in a router when used in a virtual-circuit network*



In the virtual-circuit approach, the forwarding decision is based on the label of the packet.

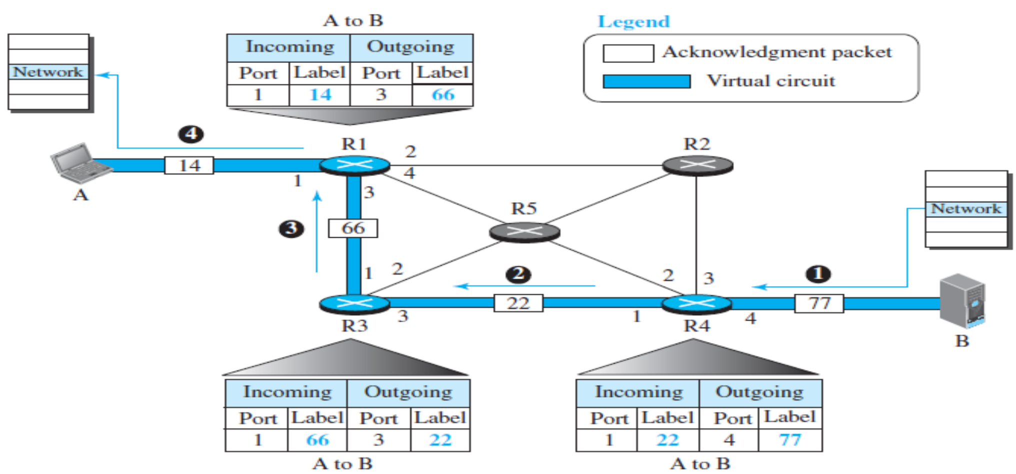
<u>Request packet</u> :A request packet is sent from the source to the destination. This auxiliary packet **carries the source and destination addresses**. The following Figure shows the process.

Figure 18.7 *Sending request packet in a virtual-circuit network*



<u>Acknowledgment Packet</u>: A special packet, called the **acknowledgment packet**, **completes the entries in the switching tables.** The following Figure shows the process.

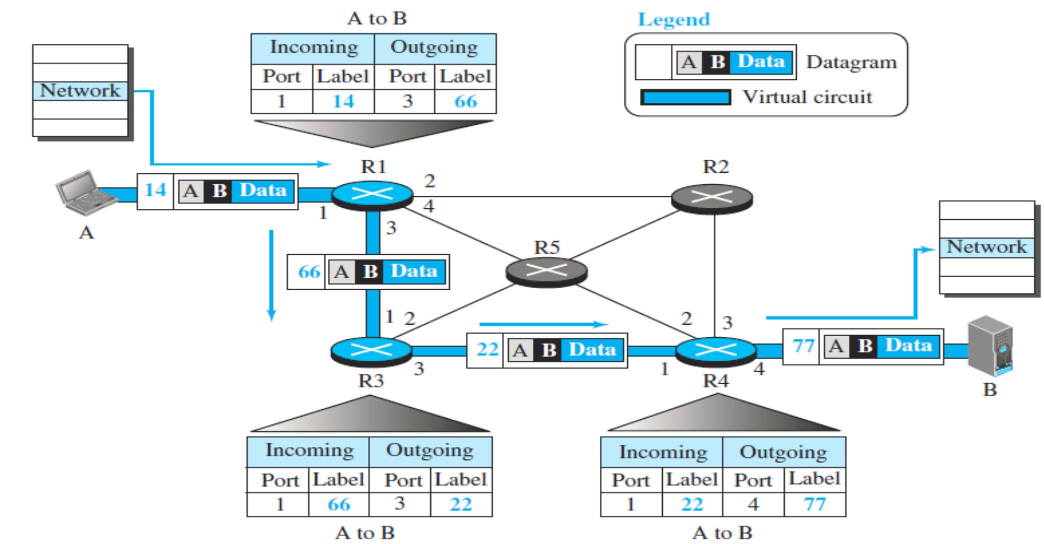
Figure 18.8 *Sending acknowledgments in a virtual-circuit network*



Data Transfer Phase

- After all routers have created their routing table for a specific virtual circuit, then the network layer packets belonging to one message can be sent one after another.
- All the packets in the message follow the same sequence of labels to reach their destination.
- **The packet arrives in order at the destination.**

Figure 18.9 *Flow of one packet in an established virtual circuit*

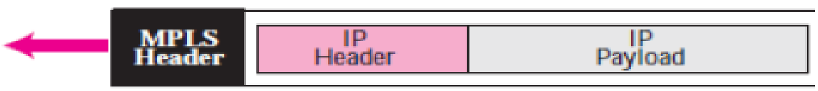


Teardown Phase

In the teardown phase, source A, after sending all packets to B, sends a special packet called a teardown packet. Destination B responds with a confirmation packet. All routers delete the corresponding entry from their tables.

MPLS (Multi-Protocol Label Switching)

some conventional routers in the Internet can be replaced by MPLS routers that can behave like a router and a switch. To simulate connection-oriented switching using a protocol like IP, the first thing that is needed is to add a field to the packet that carry the label. The IPv4 packet format does not allow this extension (this field is provided in IPv6 packet format). The solution is to encapsulate the IPv4 packet in an MPLS packet (as though MPLS is a layer between the data link layer and the network layer). The whole IP packet is encapsulated as the payload in an MPLS packet and MPLS header is added.



NETWORK-LAYER PERFORMANCE

The upper-layer protocols that use the service of the network layer expect to receive an **ideal service**, **but the network layer is not perfect.** The performance of a network can be measured in terms of **delay, throughput, and packet loss**. Congestion control is an issue that can improve the performance.

Delay

All of us expect instantaneous response from a network, but a packet, from its source to its destination, encounters delays. The delays in a network can be divided into four types: ***transmission delay, propagation delay, processing delay, and queuing delay***. Let us first discuss each of these delay types and then show how to calculate a packet delay from the source to the destination.

Transmission Delay

A source host or a router **cannot send a packet instantaneously**. A sender needs to put the bits in a packet on the line one by one. If the first bit of the packet is put on the line at time **t1** and the last bit is put on the line at time **t2,** transmission delay of the packet is **(t2 − t1).** Definitely, the transmission delay is longer for a longer packet and shorter if the sender can transmit faster. In other words, the transmission delay is:

$$Delay_{tr} = (\text{Packet length}) / (\text{Transmission rate})$$

For example, in a Fast Ethernet LAN with the transmission rate of 100 million bits per second and a packet of 10,000 bits, it takes (10,000)/(100,000,000) or 100 ***microseconds*** for all bits of the packet to be put on the line.

Propagation Delay

Propagation delay is the time it takes for a bit to travel from point A to point B in the transmission media. The propagation delay for a packet-switched network depends on the propagation delay of each network (LAN or WAN). The propagation delay depends on the propagation speed of the media, which is ***3×10^8 meters/second*** in a vacuum and normally much less in a wired medium; it also depends on the distance of the link. In other words, propagation delay is:

$$Delay_{pg} = (\text{Distance}) / (\text{Propagation speed}).$$

For example, if the distance of a cable link in a point-to-point WAN is 2000 meters and the propagation speed of the bits in the cable is **2×10^8 meters/second**, then the propagation delay is 10 microseconds.

Processing Delay

The processing delay *is the time* required for a router or a destination host to **receive a packet from its input port, remove the header, perform an error detection procedure, and deliver the packet to the output port** (in the case of a router) or deliver the packet to the upper-layer protocol (in the case of the destination host). **The processing delay may be different for each packet, but normally is calculated as an average.**

$$\text{Delay}_{pr} = \text{Time required to process a packet in a router or a destination host}$$

Queuing Delay

Queuing delay can normally happen in a router. A router has an input queue connected to each of its input ports to store packets waiting to be processed; the router also has an output queue connected to each of its output ports to store packets waiting to be transmitted. **The queuing delay for a packet in a router is measured as the time a packet waits in the input queue and output queue of a router.**

$$\text{Delay}_{qu} = \text{The time a packet waits in input and output queues in a router}$$

Total Delay

Assuming equal delays for the sender, routers, and receiver, the total delay (source-to destination delay) a packet encounters can be calculated if we know the number of routers, *n*, in the whole path.

$$\text{Total delay} = (n + 1)\ (\text{Delay}_{tr} + \text{Delay}_{pg} + \text{Delay}_{pr}) + (n)\ (\text{Delay}_{qu})$$

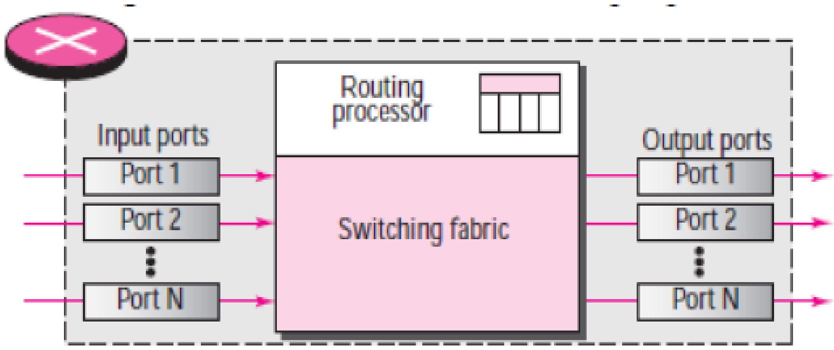
Note that if we have **n** routers, we have **(n + 1)** links.

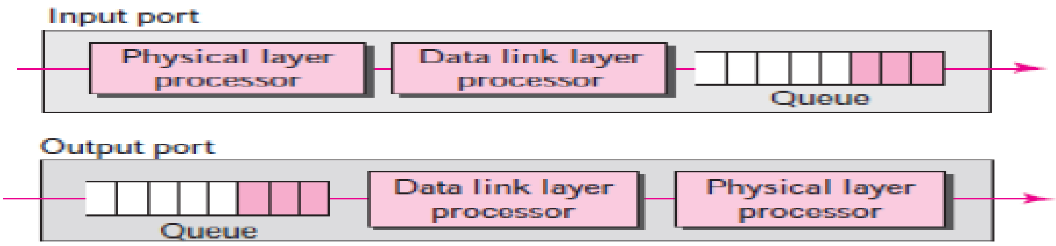
- we have **(n + 1)** transmission delays related to **n** routers and the source,
- **(n + 1)** propagation delays related to **(n + 1)** links,
- **(n + 1)** processing delays related to **n** routers and the destination,
- and only **n** queuing delays related to **n** routers.

STRUCTURE OF A ROUTER

Components

We can say that a router has four components: input ports, output ports, the routing processor, and the switching fabric.





Routing Processor

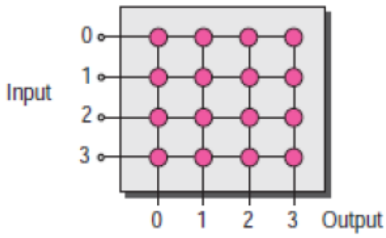
The routing processor performs the functions of the network layer. This activity is sometimes referred to as table lookup because the routing processor searches the routing table.

Switching Fabrics

The most difficult task in a router is to move the packet from the input queue to the output queue. Routers use a variety of switching fabrics. We briefly discuss some of these fabrics here.

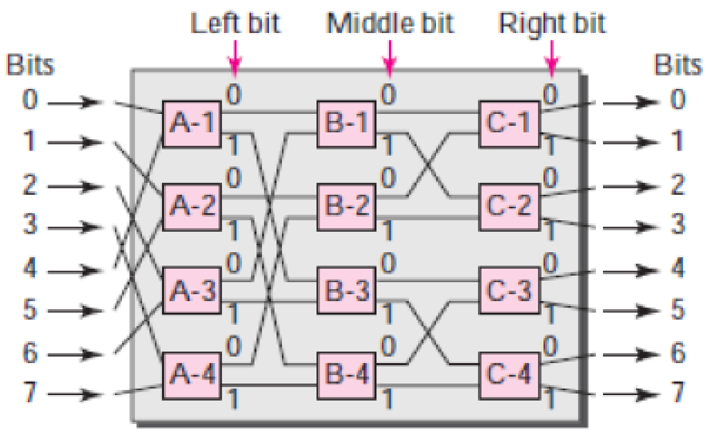
Crossbar Switch

The simplest type of switching fabric is the crossbar switch. A crossbar switch connects n inputs to n outputs in a grid, using electronic microswitches at each crosspoint**.**

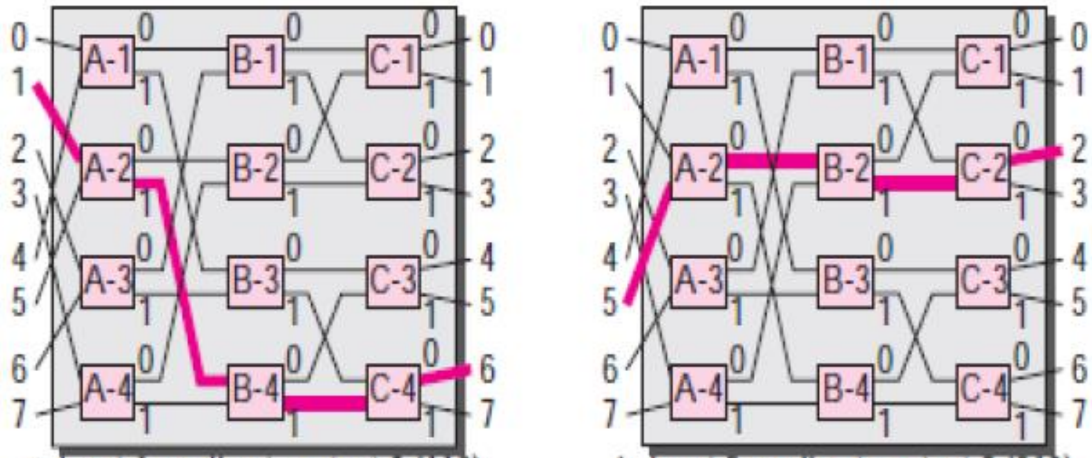


Banyan Switch

More realistic than the crossbar switch is the banyan switch**. A banyan switch is a multistage switch with microswitches at each stage that route the packets based on the output port represented as a binary string.** For **n** inputs and **n** outputs, we have **$\log_2(n)$ stages** with **n/2 microswitches** at each stage. The first stage routes the packet based on the highest order bit of the binary string. The second stage routes the packets based on the second highest order bit, and so on.



For example:



a. Input 1 sending to output 6 (110)

b. Input 5 sending to output 2 (010)