

Plan

1. Computer Networks and the Internet

- What is the Internet?
- The network edge
- **The network core**
- Network access and physical media
- ISPs and Internet backbones
- Delay and loss in packet-switched networks
- Protocol layers and their service models

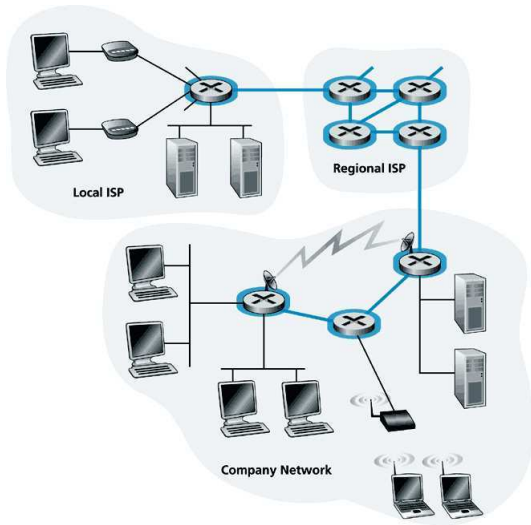
The network core

The network core is a mesh of interconnected routers.

They are of two kinds:

- **Circuit-switching** networks: the resources (buffers, link bandwidth) needed along a path (or **circuit**) to provide a communication between hosts are *reserved* for the duration of the **session**.
- **Packet-switching** networks: the resources are *not reserved* and thus the data may have to wait and are cut into packets.

The network core (cont)



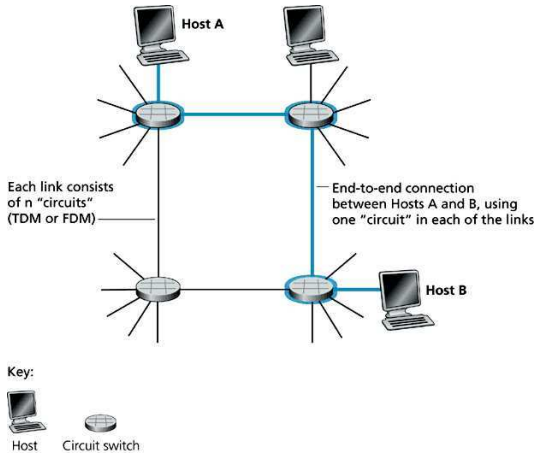
Circuit-switched networks

The **telephone network** is of this kind: even if no one talks on the phone, the resources are still reserved until one hangs up.

Since bandwidth has been reserved, the data (voice) rate is guaranteed.

Circuits are not shared and must be setup first: the routers (called here **circuit switches**) along the path “know” that they are part of a given circuit (i.e., keep a state in memory).

Circuit-switched networks (cont)

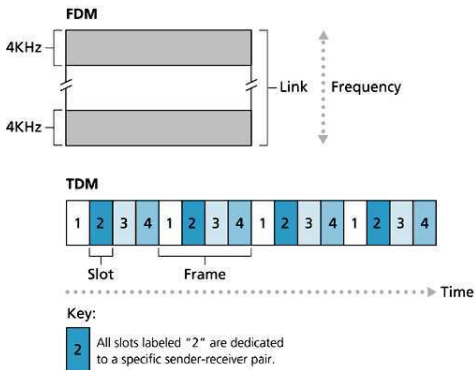


Circuit-switched networks/Multiplexing

Each link divides its data rate equally between all the active circuits it supports. This is called **multiplexing**. There are two kinds of multiplexing.

- **Frequency-division multiplexing (FDM)** divides and *shares the frequency spectrum of the link* among all the connections established across it. In telephony, the bandwidth allocated to a circuit is 4 kHz. FM radio also use FDM.
- **Time-division multiplexing (TDM)** defines a period of time, called **frame**, divides it into equal **slots** which are reserved to a given circuit. In other words, each time slot is periodically repeated and, *during each slot, the full bandwidth is available to the current circuit only*. Modern trend in telephony is to replace FDM with TDM.

Circuit-switched networks/Multiplexing/Figures



Circuit-switched networks/Multiplexing/TDM examples

For example, if a TDM link transmits 8,000 frames per second and each slot allows sending 8 bits, then the transmission rate of a circuit on this link is 64 kb/s (also noted 64 **Kbps**).

Now, suppose

- we want to send a file of size 640,000 bits;
- all links in the network use TDM with 24 slots/frame and have a bit rate of 1.536 Mbps (mega-bits per second);
- it takes 500 ms (milliseconds, also noted **msec**) to establish an end-to-end circuit.

How long does it take to send the file?

Circuit-switched networks/Multiplexing/Solution

1. 24 slots means there are 24 circuits along each link.
2. TDM implies that each circuit has the full bandwidth in turn.
Therefore, each circuit has the full bandwidth $1/24$ of the time.
3. Thus the rate for each circuit is the link rate by its time using it:
 $1.536/24 = 64 \text{ kb/s} = 64,000 \text{ b/s}$
4. The file is transferred in $640,000/64,000 = 10$ seconds.
5. But we have to add the time to setup the circuit, so the total transfer time is
 $10 + 0.5 = 10.5$ seconds.

Note that **the transfer time is independent of the number of links.**

Packet-switched networks

The source breaks the messages into smaller pieces known as **packets**. The packets are directed towards their destination by **routers**, also called **packet switches**.

Packets are transmitted over each link at the *full* transmission rate: *only one packet travels a link at a time*.

Most packet-switches use **store-and-forward transmission**: the switch must receive the entire packet before it starts to retransmit the first bit of it.

Packet-switched networks/Queuing

Each router has multiple links attached to it. For each link, the router has an **output buffer** (also called **output queue**), which stores packets that are about to be transmitted.

If an incoming packet finds the next link busy, it is stored (queued) in the corresponding output buffer until the link is available: this delay is called **queuing delay**.

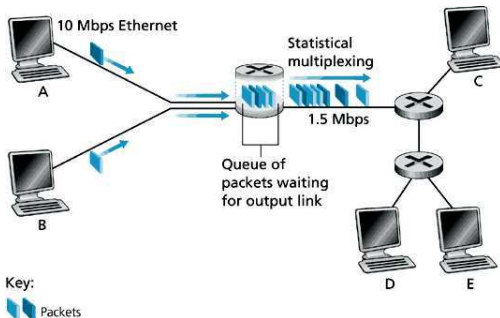
If the queue is already full (because of network congestion), the incoming packet is discarded (dropped): this is a **packet loss**.

Packet-switched networks/Statistical multiplexing

The packets are retransmitted whenever they are available and the output link is free. This leads to a random ordering, called **statistical multiplexing**.

This is very different from TDM in circuit switching, where each host gets the same time slot in a revolving frame.

Packet-switched networks/Statistical multiplexing (cont)



Warning! In this figure, several packets for the same connection are drawn on the same link, but, in fact, there is only one at a time.

Packet-switched networks/Delay

Let us consider on a very simple case how long it takes to send a message of L bits on a series of Q links whose rate is R bits/sec.

Assume that there is no queuing delay and no connection establishment.

The packet must be transmitted on the first link: this takes L/R seconds.

Then it must be retransmitted on the following $Q - 1$ links, that is to say $Q - 1$ times.

Thus the total delay is simply QL/R .

Packet switching versus circuit switching

Let us compare packet switching and circuit switching.

- Opponents to packet switching say that packet switching is not suitable for **real-time services** (e.g., telephone calls and video-conference) because of its variable and unpredictable delays (mainly queuing delays).
- Proponents of packet switching argue that
 - it offers better sharing of the bandwidth than circuit switching,
 - it is simpler, less costly to implement and more efficient.

Packet switching versus circuit switching (cont)

Why is packet switching more efficient?

Let us consider a simple example. Suppose

- users share a 1 Mbps link;
- each user alternates periods of activity and of inactivity:
 - when he is active, he generates data at the constant rate of 100 Kbps;
 - when he is idle, he produces no data.
- each user is active during 10% of the time.

With circuit switching, 100 Kbps must be reserved for *each* user. Thus the link can only support 10 (1 Mbps/100 Kbps) users simultaneously.

Packet switching versus circuit switching (cont)

If there are 35 users, the probability that there are 11 or more simultaneously active users is around 0.0004 (we don't show the computation).

- When there are 10 or less active users at the same time, the total used bandwidth is less or equal to 1 Mbps. This is similar to the circuit switching situation discussed before.
- When there are 11 or more active users, the output queue of the routers will start to grow (this is congestion).

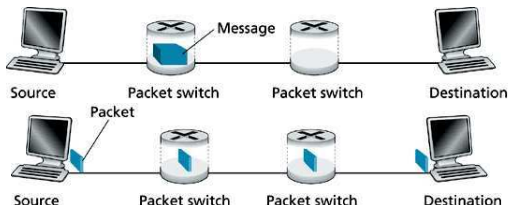
Because the probability of having more than 11 active users is very low, packet switching is almost as good as circuit switching, *while allowing more than three times the number of users* (35 users versus 10).

Message segmenting

We have seen some advantages of packet-switched networks. Now, what about the size of the packets?

If an application message is not cut into smaller pieces (packets), i.e., the message itself is a big packet, this is called **message switching**.

Contrast this with small packets in the following figures:



Message segmenting (cont)

When a message is segmented, the packet-switched network **pipelines** the transmission, i.e., portions of the message are transmitted in parallel by the source and the two packet switches.

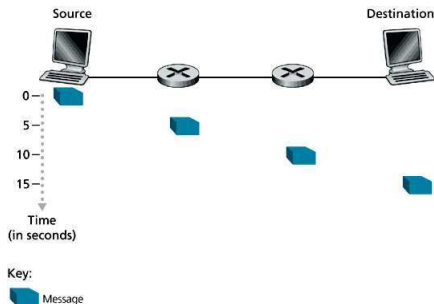
The advantage of message segmentation with packet switching is that the end-to-end delay is lower than with message switching. Let us see why.

Consider a message of size $7.5 \cdot 10^6$ bits, three links of rate 1.5 Mbps connecting two hosts and assume that there is no congestion.

How much time is required to send and receive the message with message switching?

Message segmenting (cont)

This is what happens:



It takes 15 seconds ($3 \text{ links} \times 7.5 \cdot 10^6 \text{ bits}/1.5 \text{ Mbps}$). Even a bit more, in fact, due to store-and-forward delays.

Message segmenting (cont)

Now suppose that the source breaks the message into 5,000 packets.

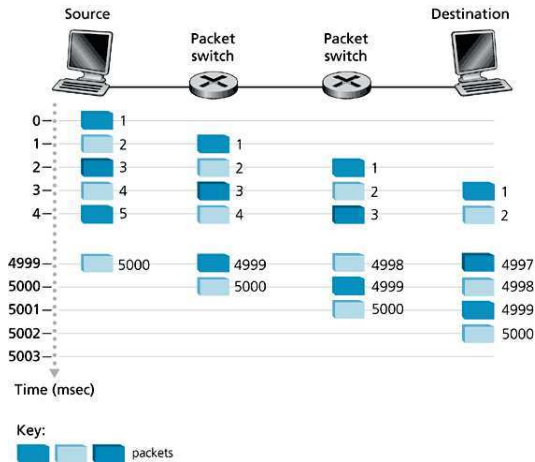
Then each packet is 1,500 ($7.5 \cdot 10^6 / 5,000$) bits long.

Assume that there is no congestion.

How long does it takes to move the 5,000 packets to destination?

The next figure shows what happens.

Message segmenting (cont)



Message segmenting (cont)

As shown in the previous figure, the first packet reached the first switch after 1 ms. The second packet reaches it after 2 ms etc.

But, while the first packet is moving from the first switch to the second, the second packet is *at the same time* traveling from the source to the first switch.

The last packet reaches then the first switch after 5,000 ms = 5 seconds. But it still has to cross two more links, therefore the total delay is 5,002 ms (in fact, a little bit more due to store-and-forward delays).

This is much less than with message switching!

The reason is pipelining: once the first packet reaches the destination, the source and the two switches are transmitting simultaneously.

Message segmenting (cont)

There is another interest in message segmentation.

Bit errors can be introduced into the packets, when moving along the networks.

When a switch detects a bit error, it discards the whole packet.

- If the entire message is a packet (i.e., we do message segmenting), then the entire message is discarded.
- If the packets are small, only a small part of the message, corresponding to the erroneous packets, has to be retransmitted.

Message segmenting (cont)

One drawback of segmentation is that the packets do not contain only the application data but also some control information, called **header**, that helps routing.

This situation is comparable to sending mail: the envelope contains (written on itself) the control information for routing and the letter contains the application data.

If we cut our message into pieces much smaller than the envelopes, then the postal service has to carry a heavier load of envelopes than of messages!

Consequently, the price would not be the same.

Packet forwarding in computer networks

Packet-switched networks can be divided in two kinds:

- **Datagram networks** routers forward packets according to host destination **address**. Similar to a home address, a network address is a unique number on the network. The Internet is a datagram network.
- **Virtual circuit networks** routers forward packets according to virtual circuit numbers. X.25 and ATM (Asynchronous Transfer Mode) use such numbers.

Virtual circuit networks

A **virtual circuit (VC)** consists of

- a path, i.e., a series of links and switches;
- virtual circuit numbers (VC numbers, in short), one for each link along the path;
- a VC number translation table in each switch along the path.

Once a VC is established, the source can send packets into the VC with the appropriate VC number.

Because a VC has a different number on each link, the switches must replace the VC number of each incoming packet with a new one.

This new number is in the translation table of the switches.

Virtual circuit networks (cont)

Example. The numbers in the figure at the next slide are **interface numbers**.

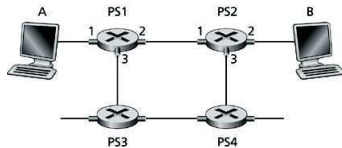
They are local to each switch.

Assume a packet starts from host A and must arrive to B: it has to cross three links.

It begins with the VC number of the first link, say 12, and enters PS1 at the interface 1.

There, its VC number has to be replaced by another one, say 22, and the packet must go out through interface 2 (towards PS2).

Virtual circuit networks (cont)



Virtual circuit networks (cont)

The VC number translation table at switch PS1 looks like

In interface	In VC#	Out interface	Out VC#
1	12	2	22
...

The network (actually, the switches) must maintain a **state information** about the VC.

Even if the VC numbers are all the same, the switches still must associate interfaces and VC numbers.

Each time a new circuit is created, a new entry must be added to the table of the switches along the path. Dually, each time a connection is released, the entries must be removed.

Datagram networks

Datagram networks are similar to postal services.

Each host has a unique address in the network.

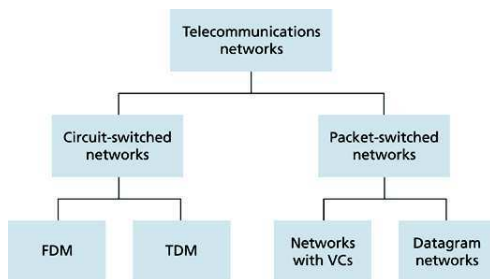
Elements in the address are hierarchical, in order to simplify the routing. This is like putting on an envelope country, city, town, district, street name, building number, apartment number.

We shall discuss later the packet structure.

For now on, remember that a datagram network do not maintain connection-related information in the routers: these do not know which connection they support.

Network taxonomy

Here is the taxonomy of the networking concepts we reviewed:



Warning! *This picture does not include the service view of the networks.* A datagram network is neither a connectionless nor a connection-oriented network. It can provide both services to different applications, through TCP or UDP.