COMPUTER NETWORKS

CHAPTER 1

Packet Switching

In a network application, end systems exchange messages with each other. Messages can contain anything the application designer wants. To send a message from a source end system to a destination end system, the source breaks long messages into smaller chunks of data known as packets. Between source and destination, each packet travels through communication links and packet switches (for which there are two predominant types, routers and link-layer switches). Packets are transmitted over each communication link at a rate equal to the full transmission rate of the link. So, if a source end system or a packet switch is sending a packet of L bits over a link with transmission rate R bits/sec, then the time to transmit the packet is L / R seconds.

Store And Forward Transmission:

Most packet switches use store-and-forward transmission at the inputs to the links. Store-and-forward transmission means that the packet switch must receive the entire packet before it can begin to transmit the first bit of the packet onto the outbound link.

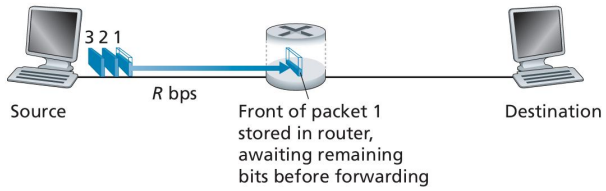


Figure 1.11 ♦ Store-and-forward packet switching

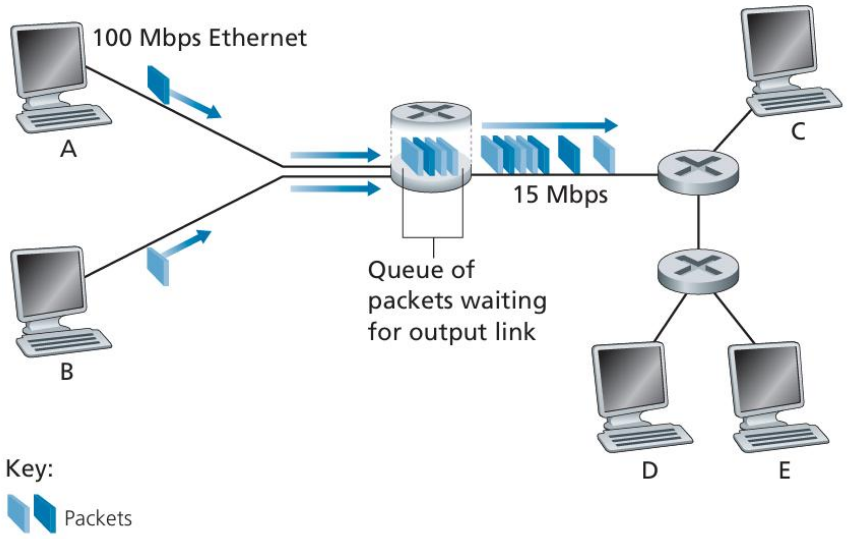
CALCULATION OF DELAY:

the amount of time that elapses from when the source begins to send the packet until the destination has received the entire packet. (Here we will ignore propagation delay—the time it takes for the bits to travel across the wire at near the speed of light The source begins to transmit at time 0; at time L/R seconds, the source has transmitted the entire packet, and the entire packet has been received and stored at the router (since there is no propagation delay). At time L/R seconds, since the router has just received the entire packet, it can begin to transmit the packet onto the outbound link towards the destination; at time 2L/R, the router has transmitted the entire packet, and the entire packet has been received by the destination. Thus, the total delay is 2L/R.

$$d_{\text{end-to-end}} = N\frac{L}{R}$$

QUEUEING DELAYS AND PACKET LOSS:

Each packet switch has multiple links attached to it. For each attached link, the packet switch has an output buffer (also called an output queue), which stores packets that the router is about to send into that link. The output buffers play a key role in packet switching. If an arriving packet needs to be transmitted onto a link but finds the link busy with the transmission of another packet, the arriving packet must wait in the output buffer. Thus, in addition to the store-and-forward delays, packets suffer output buffer queuing delays.Since the amount of buffer space is finite, an arriving packet may find that the buffer is completely full with other packets waiting for transmission. In this case, packet loss will occur—either the arriving packet or one of the already-queued packets will be dropped.



How does the router determine which link it should forward the packet onto?

In the Internet, every end system has an address called an IP address. When a source end system wants to send a packet to a destination end system, the source includes the destination's IP address in the packet's header. As with postal addresses, this address has a hierarchical structure. When a packet arrives at a router in the network, the router examines a portion of the packet's destination address and forwards the packet to an adjacent router. More specifically, each router has a forwarding table that maps destination addresses (or portions of the destination addresses) to that router's outbound links. When a packet arrives at a router, the router examines the address and searches its forwarding table, using this destination address, to find the appropriate outbound link. The router then directs the packet to this outbound link.

- We just learned that a router uses a packet's destination address to index a for warding table and determine the appropriate outbound link. But this statement begs yet another question: How do forwarding tables get set?

Internet has a number of special routing protocols that are used to automatically set the forwarding tables. A routing protocol may, for example, determine the shortest path from each router to each destination and use the shortest path results to configure the forwarding tables in the routers.

°
\

CIRCUIT SWITCHING

There are two fundamental approaches to moving data through a network of links and switches: circuit switching and packet switching. In circuit-switched networks, the resources needed along a path (buffers, link transmission rate) to provide for communication between the end systems are reserved for the duration of the communication session between the end systems.In packet-switched networks, these resources are not reserved; a session's messages use the resources on demand and, as a consequence, may have to wait (that is, queue) for access to a communication link. Traditional telephone networks are examples of circuit-switched networks.

As a simple analogy, consider two restaurants, one that requires reservations and another that neither requires reservations nor accepts them. For the restaurant that requires reservations, we have to go through the hassle of calling before we leave home. But when we arrive at the restaurant we can, in principle, immediately be seated and order our meal. For the restaurant that does not require reservations, we don't need to bother to reserve a table. But when we arrive at the restaurant, we may have to wait for a table before we can be seated.

WORKING:

The network must establish a connection between the sender and the receiver. This is a bona fide connection for which the switches on the path between the sender and receiver maintain connection state for that connection. In the jargon of telephony, this connection is called a circuit. When the network establishes the circuit, it also reserves a constant transmission rate in the network's links (representing a fraction of each link's transmission capacity) for the duration of the connection. Since a given transmission rate has been reserved for this sender-to-receiver connection, the sender can transfer the data to the receiver at the guaranteed constant rate.

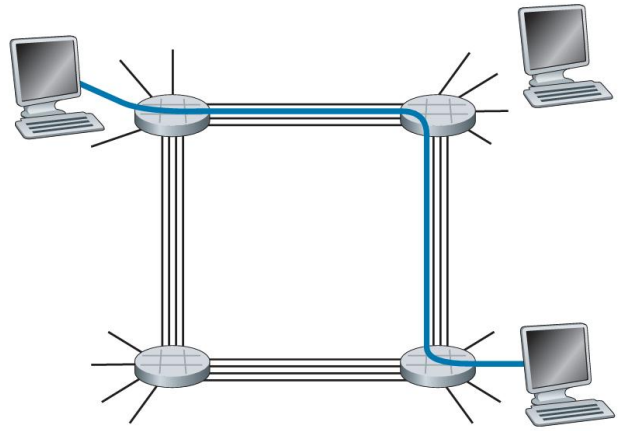


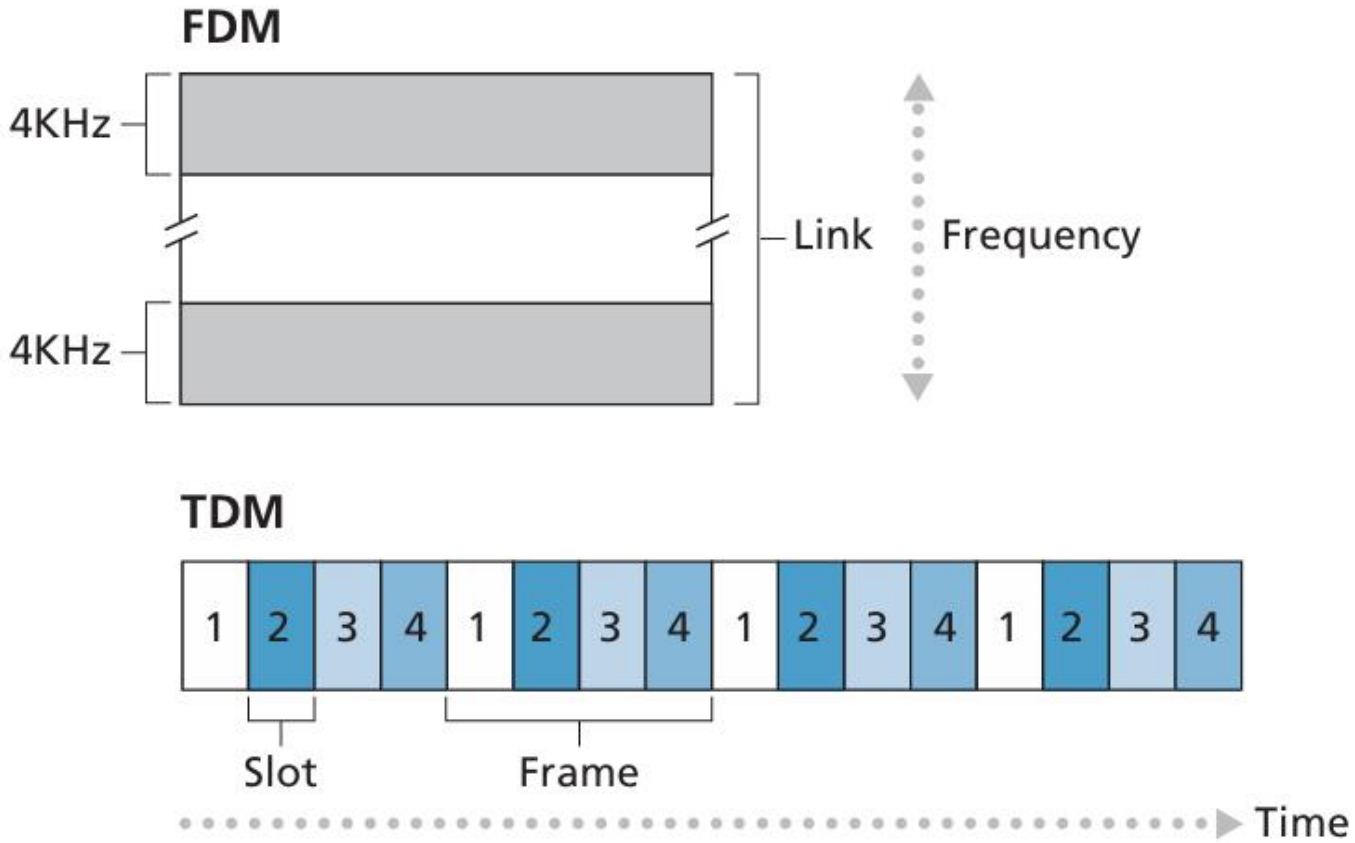
Figure 1.13 ♦ A simple circuit-switched network consisting of four switches and four links

Multiplexing in Circuit-Switched Networks

With **FDM**, the frequency spectrum of a link is divided up among the connections established across the link. Specifically, the link dedicates a frequency band to each connection for the duration of the connection. In telephone networks, this frequency band typically has a width of 4kHz (that is, 4,000 hertz or 4,000 cycles per second). The width of the band is called, not surprisingly, the bandwidth. FM radio stations also use FDM to share the frequency spectrum between 88 MHz and 108 MHz, with each station being allocated a specific frequency band.

For a **TDM** link, time is divided into frames of fixed duration, and each frame is divided into a fixed number of time slots. When the network establishes a connection across a link, the network dedicates one time slot in every frame to this connection. These slots are dedicated for the sole use of that connection, with one time slot available for use (in every frame) to transmit the connection's data.

VISUAL REPRESENTATION:



For TDM, the transmission rate of a circuit is equal to the frame rate multiplied by the number of bits in a slot. For example, if the link trans
mits 8,000 frames per second and each slot consists of 8 bits, then the transmission rate of each circuit is 64 kbps.

Proponents of packet switching have always argued that circuit switching is waste
ful because the dedicated circuits are idle during **silent periods.**

IMPORTANT EXAMPLE:

Let us consider how long it takes to send a file of 640,000 bits from Host A to Host B over a circuit-switched network. Suppose that all links in the network use TDM with 24 slots and have a bit rate of 1.536 Mbps. Also suppose that it takes 500 msec to establish an end-to-end circuit before Host A can begin to transmit the file. How long does it take to send the file? Each circuit has a transmission rate of (1.536 Mbps)/24 = 64 kbps, so it takes (640,000 bits)/(64 kbps) = 10 seconds to transmit the file. To this 10 seconds we add the circuit establishment time, giving 10.5 seconds to send the file.

PACKET SWITCHING VS CIRCUIT SWITCHING

Critics of packet switching have often argued that packet switching is not suitable for real-time services (for example, telephone calls and video conference calls) because of its variable and unpredictable end-to-end delays (due primarily to variable and unpredictable queuing delays). Proponents of packet switching argue that (1) it offers better sharing of transmission capacity than circuit switching and (2) it is simpler, more efficient, and less costly to implement than circuit switching.

Why is packet switching more efficient? Let's look at a simple example. Suppose users share a 1 Mbps link. Also suppose that each user alternates between periods of activity, when a user generates data at a constant rate of 100 kbps, and periods of inactivity, when a user generates no data. Suppose further that a user is active only10 percent of the time (and is idly drinking coffee during the remaining 90 percent of the time). With circuit switching, 100 kbps must be reserved for each user at all times. For example, with circuit-switched TDM, if a one-second frame is divided into 10 time slots of 100 ms each, then each user would be allocated one time slot per frame.

A NETWORK OF NETWORKS

End systems (PCs, smartphones, Web servers, mail servers, and so on) connect into the Internet via an access ISP. The access ISP can provide either wired or wireless connectivity, using an array of access technologies including DSL, cable, FTTH, Wi-Fi, and cellular. Note that the access ISP does not have to be a telco or a cable company; instead it can be, for example, a university (providing Internet access to students, staff, and faculty), or a company (providing access for its employees). But connecting end users and content providers into an access ISP is only a small piece of solving the puzzle of connecting the billions of end systems that make up the Internet. To complete this puzzle, the access ISPs themselves must be interconnected.This is done by creating a **network of networks**

Interconnect the access ISPs so that all end systems can send packets to each other. One naive approach would be to have each access ISP directly connect with every other access ISP. Such a mesh design is, of course, much too costly for the access ISPs, as it would require each access ISP to have a separate communication link to each of the hundreds of thousands of other access ISPs all over the world.

FIRST NETWORK STRUCTURE

Our first network structure, Network Structure 1, interconnects all of the access ISPs with a single global transit ISP. Our (imaginary) global transit ISP is a network of routers and communication links that not only spans the globe, but also has at least one router near each of the hundreds of thousands of access ISPs. Of course, it would be very costly for the global ISP to build such an extensive network.Since the access ISP pays the global transit ISP, the access ISP is said to be a **custome**r and the global transit ISP is said to be a **provider.** Now if some company builds and operates a global transit ISP that is profitable, then it is natural for other companies to build their own global transit ISPs and compete with the original global transit ISP. This leads to Network Structure 2, which consists of the hundreds of thousands of access ISPs and multiple global transit ISPs. The access ISPs certainly prefer Network Structure 2 over Network Structure 1 since they can now choose among the competing global transit providers as a function of their pricing and services. Note, however, that the global transit ISPs themselves must interconnect: Otherwise access ISPs connected to one of the global transit providers would not be able to communicate with access ISPs connected to the other global transit providers.

SECOND NETWORK STRUCTURE

Network Structure 2, just described, is a two-tier hierarchy with global transit providers residing at the top tier and access ISPs at the bottom tier. This assumes that global transit ISPs are not only capable of getting close to each and every access ISP, but also find it economically desirable to do so. In reality, although some ISPs do have impressive global coverage and do directly connect with many access ISPs, no ISP has presence in each and every city in the world. Instead, in any given region, there may be a regional ISP to which the access ISPs in the region connect. Each regional ISP then connects to tier-1 ISPs. Tier-1 ISPs are similar to our (imaginary) global transit ISP; but tier-1 ISPs, which actually do exist, do not have a presence in every city in the world.

NETWORK STRUCTURE THREE

Returning to this network of networks, not only are there multiple competing tier-1 ISPs, there may be multiple competing regional ISPs in a region. In such a hierarchy, each access ISP pays the regional ISP to which it connects, and each regional ISP pays the tier-1 ISP to which it connects. (An access ISP can also connect directly to a tier-1 ISP, in which case it pays the tier-1 ISP). Thus, there is customerprovider relationship at each level of the hierarchy. Note that the tier-1 ISPs do not pay anyone as they are at the top of the hierarchy.

NETWORK STRUCTURE 4

To build a network that more closely resembles today's Internet, we must add **points of presence (PoPs), multi-homing, peering, and Internet exchange points (IXPs)** to the hierarchical Network Structure 3. PoPs exist in all levels of the hierarchy, except for the bottom (access ISP) level. A PoP is simply a group of one or more routers (at the same location) in the provider's network where customer ISPs can connect into the provider ISP. For a customer network to connect to a provider's PoP, it can lease a high-speed link from a third-party telecommunications provider to directly connect one of its routers to a router at the PoP. Any ISP (except for tier-1 ISPs) may choose to multi-home, that is, to connect to two or more provider ISPs. So, for example, an access ISP may multi-home with two regional ISPs, or it may multihome with two regional ISPs and also with a tier-1 ISP. Similarly, a regional ISP may multi-home with multiple tier-1 ISPs. When an ISP multi-homes, it can continue to send and receive packets into the Internet even if one of its providers has a failure.customer ISPs pay their provider ISPs to obtain global Internet interconnectivity. The amount that a customer ISP pays a provider ISP reflects the amount of traffic it exchanges with the provider. To reduce these costs, a pair of nearby ISPs at the same level of the hierarchy can **peer**, that is, they can directly connect their networks together so that all the traffic between them passes over the direct connection rather than through upstream intermediaries. When two ISPs peer, it is typically settlement-free, that is, neither ISP pays the other. As noted earlier, tier-1 ISPs also peer with one another, settlement-free.a third-party company can create an Internet Exchange Point (IXP), which is a meeting point where multiple ISPs can peer together. An IXP is typically in a stand-alone building with its own switches [Ager 2012]. There are over 600 IXPs in the Internet today [PeeringDB 2020]. We refer to this ecosystem—consisting of access ISPs, regional ISPs, tier-1 ISPs, PoPs, multi-homing, peering, and IXPs—as Network Structure 4

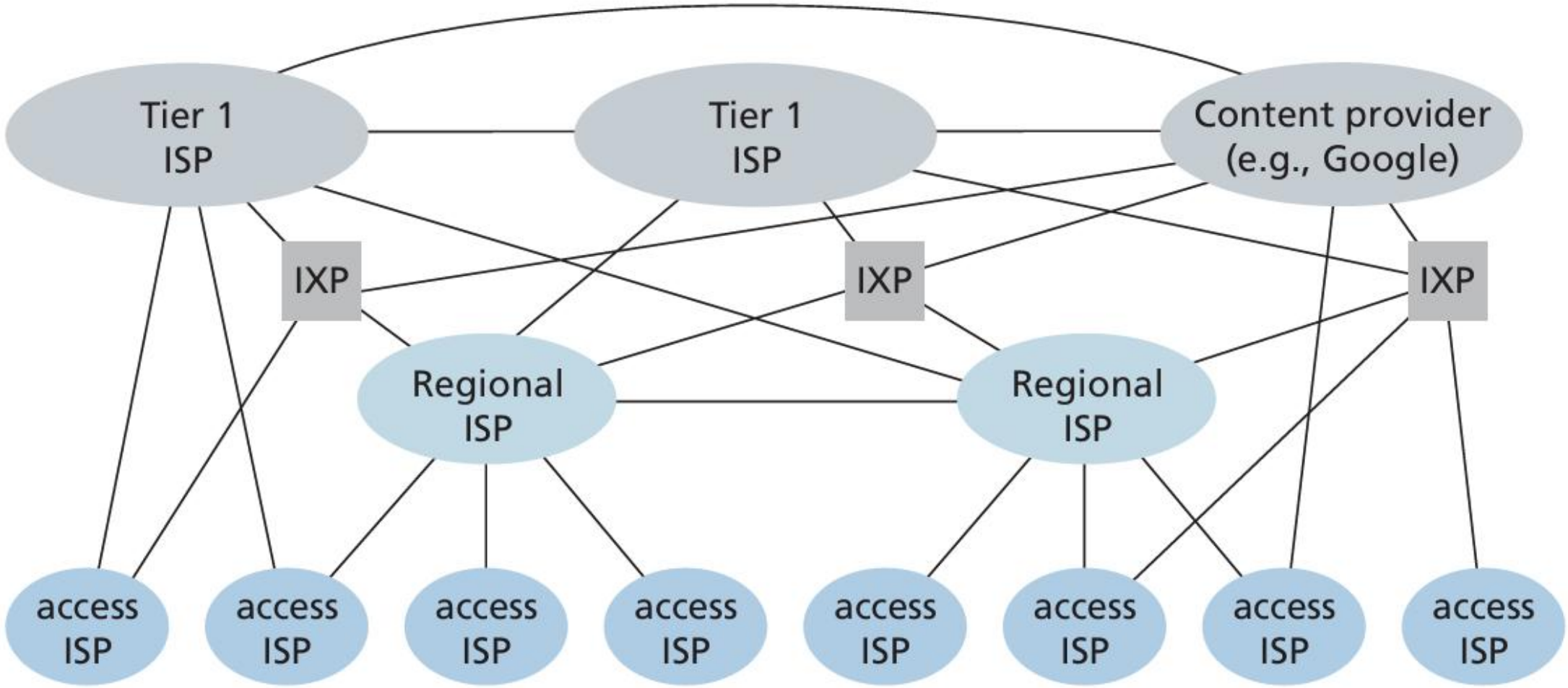


Figure 1.15 ♦ Interconnection of ISPs

TYPES OF DELAY:

- **PROCESSING DELAY:**

The time required to examine the packet's header and determine where to direct

the packet is part of the processing delay. The processing delay can also include

other factors, such as the time needed to check for bit-level errors in the packet

that occurred in transmitting the packet's bits from the upstream node to router A.

Processing delays in high-speed routers are typically on the order of microseconds

or less.

- **Queuing Delay:**

At the queue, the packet experiences a queuing delay as it waits to be transmitted

onto the link. The length of the queuing delay of a specific packet will depend on the

number of earlier-arriving packets that are queued and waiting for transmission onto

the link. If the queue is empty and no other packet is currently being transmitted, then

our packet's queuing delay will be zero. On the other hand, if the traffic is heavy and

many other packets are also waiting to be transmitted, the queuing delay will be long.

TYPES OF DELAY:

- **Transmission Delay:**

The transmission delay is

L/R. This is the amount of time required to push (that is, transmit) all of the packet's

bits into the link. Transmission delays are typically on the order of microseconds to

milliseconds in practice.

- **PROPAGATION DELAY:**

Once a bit is pushed into the link, it needs to propagate to router B. The time required to propagate from the beginning of the link to router B is the propagation delay. The bit propagates at the propagation speed of the link. The propagation speed depends on the physical medium of the link (that is, fiber optics, twisted-pair copper wire, and so on) The propagation delay is the distance between two routers divided by the propagation speed. That is, the propagation delay is d/s, where d is the distance between router A and router B and s is the propagation speed of the link. Once the last bit of the packet propagates to node B, it and all the preceding bits of the packet are stored in router B. The whole process then continues with router B now performing the forwarding. In wide-area networks, propagation delays are on the order of milliseconds.

COMPARISON:

The transmission delay is the amount of time required
for the router to push out the packet; it is a function of the packet's length and the transmission rate of the link, but has nothing to do with the distance between the two routers. The propagation delay, on the other hand, is the time it takes a bit to propa gate from one router to the next; it is a function of the distance between the two rout ers, but has nothing to do with the packet's length or the transmission rate of the link.

If we let d_{proc}, d_{queue}, d_{trans}, and d_{prop} denote the processing, queuing, transmission, and propagation delays, then the total nodal delay is given by

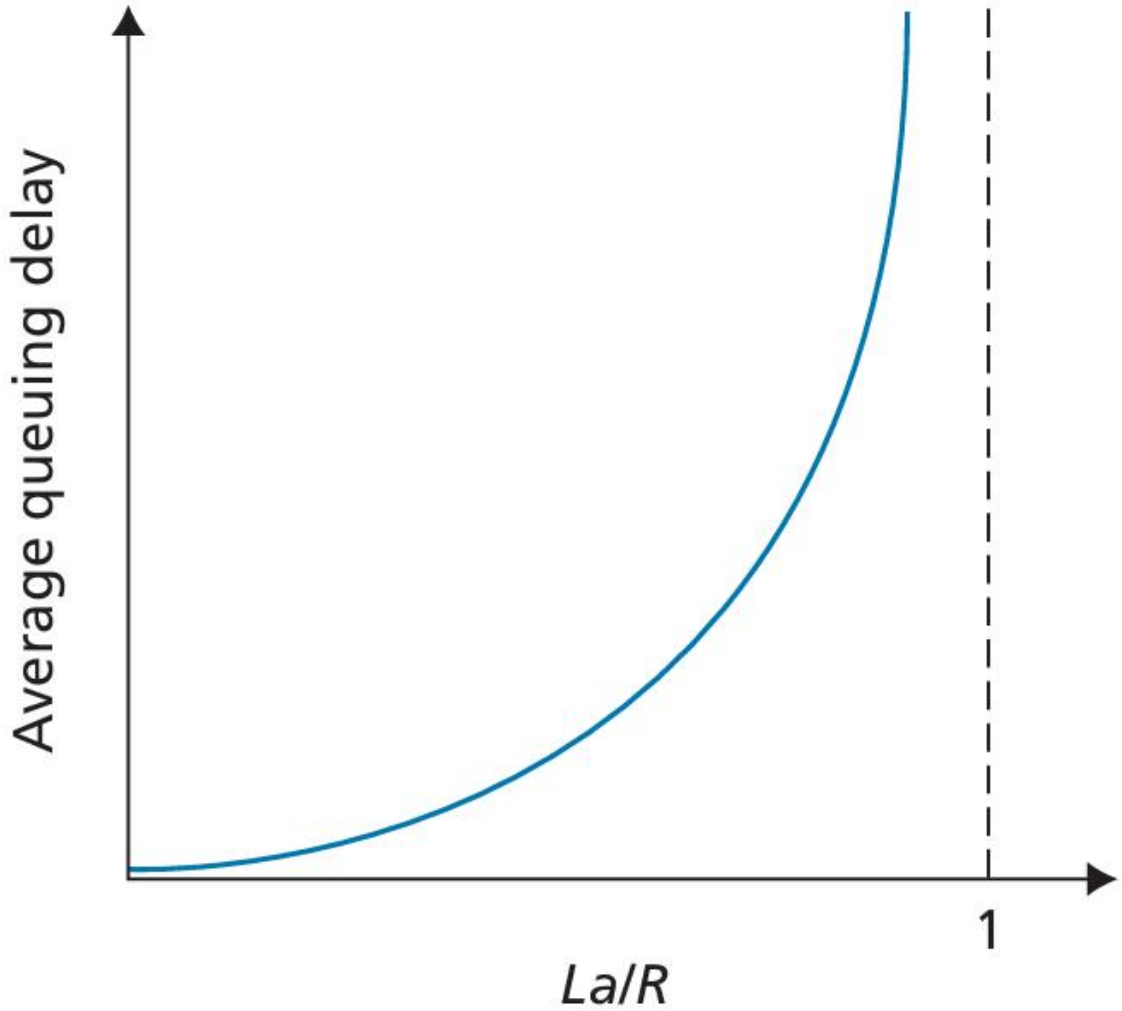
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

QUEUEING DELAY DETAIL:

When is the queuing delay large and when is it insignificant? The answer to this question depends on the rate at which traffic arrives at the queue, the transmission rate of the link, and the nature of the arriving traffic, that is, whether the traffic arrives periodically or arrives in bursts. To gain some insight here, let a denote the average rate at which packets arrive at the queue (a is in units of packets/sec). Recall that R is the transmission rate; that is, it is the rate (in bits/sec) at which bits are pushed out of the queue. Also suppose, for simplicity, that all packets consist of bits. Then the average rate at which bits arrive at the queue is La bits/sec. Finally, assume that the queue is very big, so that it can hold essentially an infinite number of bits. The ratio La/R, called the traffic intensity, often plays an important role in estimating the extent of the queuing delay.

- **CASES:**

If La/R > 1, then the average rate at which bits arrive at the queue exceeds the rate at which the bits can be transmitted from the queue. In this unfortunate situation, the queue will tend to increase without bound and the queuing delay will approach infinity! Therefore, one of the golden rules in traffic engineering is: Design your system so that the traffic intensity is no greater than 1.Now consider the case La/R . Here, the nature of the arriving traffic impacts the queuing delay. For example, if packets arrive periodically—that is, one packet arrives every L/R seconds—then every packet will arrive at an empty queue and there will be no queuing delay. On the other hand, if packets arrive in bursts but periodically, there can be a significant average queuing delay. For example, suppose N packets arrive simultaneously every (L/R)N seconds.



1.18 ♦ Dependence of average queuing delay on traffic intensity

PACKET LOSS:

In our discussions above, we have assumed that the queue is capable of holding an infinite number of packets. In reality a queue preceding a link has finite capacity, although the queuing capacity greatly depends on the router design and cost. Because the queue capacity is finite, packet delays do not really approach infinity as the traf fic intensity approaches 1. Instead, a packet can arrive to find a full queue. With no place to store such a packet, a router will drop that packet; that is, the packet will be lost. This overflow at a queue can again be seen in the interactive animation when the traffic intensity is greater than 1.. The fraction of lost packets increases as the traffic intensity increases. Therefore, performance at a node is often measured not only in terms of delay, but also in terms of the probability of packet loss.

.

VOIP AND THROUGHPUT

In VoIP, the sending side must first fill a packet with

encoded digitized speech before passing the packet to the Internet. This time to fill a

packet—called the packetization delay—can be significant and can impact the user

perceived quality of a VoIP call.

- **THROUGHPUT**

To define throughput, consider transferring a large file from Host A to Host B across a computer network. This transfer might be, for example, a large video clip from one computer to another. The instantaneous throughput at any instant of time is the rate (in bits/sec) at which Host B is receiving the file.If the file consists of F bits and the transfer takes T seconds for Host B to receive all F bits, then the average throughput of the file transfer is F/T bits/sec.

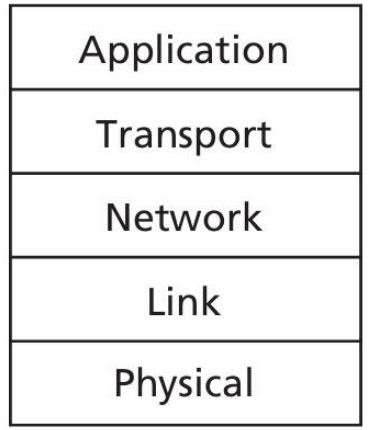
THROUGHPUT

Let Rs denote the rate of the link between the server and the router; and Rc denote the rate of the link between the router and the client. Suppose that the only bits being sent in the entire network are those from the server to the client. We now ask, in this ideal scenario, what is the server- to-client throughput? To answer this question, we may think of bits as fluid and communication links as pipes. Clearly, the server cannot pump bits through its link at a rate faster than Rs bps; and the router cannot forward bits at a rate faster than Rc bps. If Rs 6 Rc, then the bits pumped by the server will “flow” right through the router and arrive at the client at a rate of Rs bps, giving a throughput of Rs bps. If, on the other hand, Rc 6 Rs, then the router will not be able to forward bits as quickly as it receives them. In this case, bits will only leave the router at rate Rc, giving an end-to-end throughput of Rc.Thus,for this simple two-link network, the throughput is min{Rc, Rs}, that is, it isthe transmission rate of the bottleneck link. Having determined the throughput, we can now approximate the time it takes to transfer a large file of F bits from server to client as F/min{Rs, Rc}. For a specific example, suppose that you are downloading an MP3 file of F = 32 million bits, the server has a transmission rate of Rs = 2 Mbps, and you have an access link of Rc = 1 Mbps. Thetime needed to transfer the file is Then 32 seconds.

- .

PROTOCOL LAYERS And THEIR MODELS:

Protocol layering has conceptual and structural advantages [RFC 3439]. As we have seen, layering provides a structured way to discuss system components. Modularity makes it easier to update system components. We mention, however, that some researchers and networking engineers are vehemently opposed to layering [Wakeman 1992]. One potential drawback of layering is that one layer may duplicate lower-layer functionality. For example, many protocol stacks provide error recovery on both a per-link basis and an end-to-end basis. A second potential drawback is tha functionality at one layer may need information (for example, a timestamp value) that is present only in another layer; this violates the goal of separation of layers When taken together, the protocols of the various layers are called the protocol stack.



Five-layer Internet protocol stack

APPLICATION LAYER:

The application layer is where network applications and their application-layer pro

tocols reside. The Internet's application layer includes many protocols, such as the

HTTP protocol (which provides for Web document request and transfer), SMTP

(which provides for the transfer of e-mail messages), and FTP (which provides for

the transfer of files between two end systems).

An application-layer protocol is distributed over multiple end systems, with the

application in one end system using the protocol to exchange packets of information

with the application in another end system. We'll refer to this packet of information

at the application layer as a message

TRANSPORT LAYER:

The Internet's transport layer transports application-layer messages between application

endpoints. In the Internet, there are two transport protocols, TCP and UDP, either of

which can transport application-layer messages. TCP provides a connection-oriented

service to its applications. This service includes guaranteed delivery of application-layer messages to the destination and flow control (that is, sender/receiver speed matching)

TCP also breaks long messages into shorter segments and provides a congestion-control

mechanism, so that a source throttles its transmission rate when the network is con

gested. The UDP protocol provides a connectionless service to its applications. This is a

no-frills service that provides no reliability, no flow control, and no congestion control.

In this book, we'll refer to a transport-layer packet as a segment.

NETWORK LAYER:

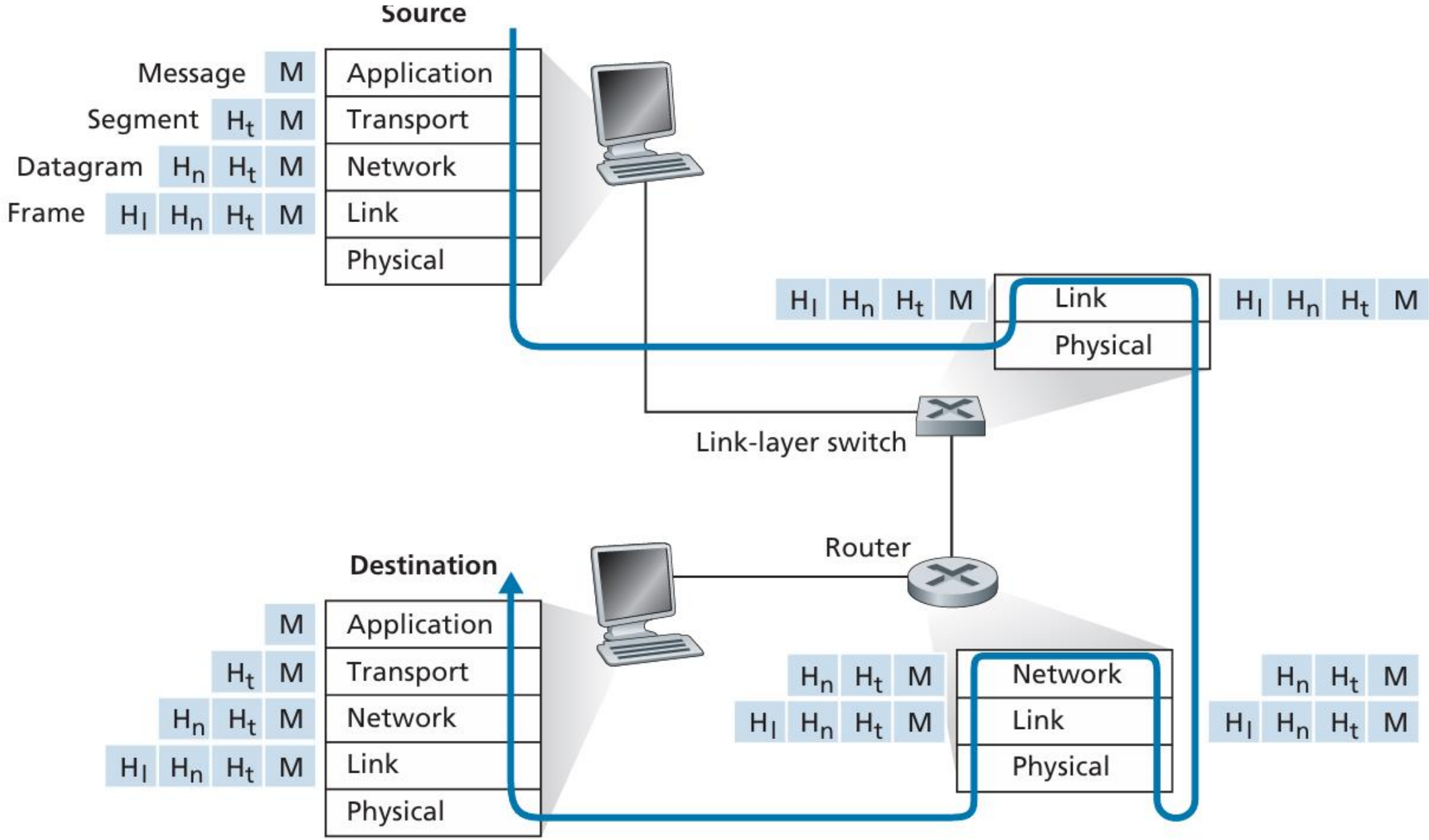
The Internet's network layer is responsible for moving network-layer packets known as datagrams from one host to another. The Internet transport-layer protocol (TCP or UDP) in a source host passes a transport-layer segment and a destination address to the network layer, just as you would give the postal service a letter with a destination address. The network layer then provides the service of delivering the segment to the transport layer in the destination host.The Internet's network layer includes the celebrated IP protocol, which defines the fields in the datagram as well as how the end systems and routers act on these fields. There is only one IP protocol, and all Internet components that have a network layer must run the IP protocol. The Internet's network layer also contains routing protocols that determine the routes that datagrams take between sources and destinations.

LINK LAYER:

The Internet's network layer routes a datagram through a series of routers between the source and destination. To move a packet from one node (host or router) to the next node in the route, the network layer relies on the services of the link layer. In particular, at each node, the network layer passes the datagram down to the link layer, which delivers the datagram to the next node along the route. At this next node, the link layer passes the datagram up to the network layer.Examples of link-layer pro tocols include Ethernet, WiFi, and the cable access network's DOCSIS protocol.

PHYSICAL LAYER:

While the job of the link layer is to move entire frames from one network element to an adjacent network element, the job of the physical layer is to move the individual bits within the frame from one node to the next. The protocols in this layer are again link dependent and further depend on the actual transmission medium of the link (for example, twisted-pair copper wire, single-mode fiber optics). For example, Ether net has many physical-layer protocols: one for twisted-pair copper wire, another for coaxial cable, another for fiber, and so on. In each case, a bit is moved across the link in a different way.


Source
Message
Segment
Datagram
Frame
M
H_t M
H_n H_t M
H_l H_n H_t M
Application
Transport
Network
Link
Physical
Link-layer switch
Router
Destination
Application
Transport
Network
Link
Physical
Link
Physical
Network
Link
Physical

ENCAPSULATION:

At the sending host, an application-layer message (M in Figure 1.24) is passed to the transport layer. In the simplest case, the transport layer takes the message and appends additional information (so-called transport-layer header information, Ht in Figure 1.24) that will be used by the receiver-side transport layer. The application-layer message and the transport-layer header information together constitute the transport-layer segment. The transport-layer segment thus encapsulates the application-layer message. The added information might include information allowing the receiver-side transport layer to deliver the message up to the appropriate application, and error-detection bits that allow the receiver to determine whether bits in the message have been changed in route. The transport layer then passes the segment to the network layer, which adds network-layer header information (Hn in Figure 1.24) such as source and destination end system addresses, creating a network-layer datagram. The datagram is then passed to the link layer, which (of course!) will add its own link-layer header information and create a link-layer frame. Thus, we see that at each layer, a packet has two types of fields: header fields and a payload field. The payload is typically a packet from the layer above.

THREATS TO COMPUTER:

MALWARE:

But, unfortunately, along with all that good stuff comes malicious stuff—

collectively known as malware—that can also enter and infect our devices. Once

malware infects our device it can do all kinds of devious things, including delet

ing our files and installing spyware that collects our private information, such

associal security numbers, passwords, and keystrokes, and then sends this (over

the Internet, of course!) back to the bad guys.

Our compromised host may also beenrolled in a network of thousands of similarly compromised devices, collectively known as a botnet, which the bad guys control and leverage for spam e-mail distribution or distributed denial-of-service attacks (soon to be discussed) against targeted hosts. Much of the malware out there today is self-replicating: once it infects one host, from that host it seeks entry into other hosts over the Internet, and from the newly infected hosts, it seeks entry into yet more hosts. In this manner, self- replicating malware can spread exponentially fast.

ATTACK ON SERVERS:

Another broad class of security threats are known as denial-of-service (DoS)

attacks. As the name suggests, a DoS attack renders a network, host, or other piece

of infrastructure unusable by legitimate users.

Vulnerability attack. This involves sending a few well-crafted messages to a vulnerable application or operating system running on a targeted host. If the right sequence of packets is sent to a vulnerable application or operating system, the service can stop or, worse, the host can crash. • **Bandwidth flooding.** The attacker sends a deluge of packets to the targeted host—so many packets that the target's access link becomes clogged, preventing Legitimate packets from reaching the server.• **Connection flooding.** The attacker establishes a large number of half-open or fully open TCP connections (TCP connections are discussed in Chapter 3) at the target host. The host can become so bogged down with these bogus connections that it stops accepting legitimate connections

PACKET SNIFFER AND IP SPOOFING:

While ubiquitous Internet access is extremely convenient and enables marvelous new applications for mobile users, it also creates a major security vulnerability—by placing a passive receiver in the vicinity of the wireless transmitter, that receiver can obtain a copy of every packet that is transmitted! These packets can contain all kinds of sensitive information, including passwords, social security numbers, trade secrets, and private personal messages. A passive receiver that records a copy of every packet that flies by is called a packet sniffer.

It is surprisingly easy (you will have the knowledge to do so shortly as you proceed through this text!) to create a packet with an arbitrary source address, packet content, and destination address and then transmit this hand-crafted packet into the Internet, which will dutifully forward the packet to its destination. Imagine the unsuspecting receiver (say an Internet router) who receives such a packet, takes the (false) source address as being truthful, and then performs some command embedded in the packet's contents (say modifies its forwarding table). The ability to inject packets into the Internet with a false source address is known as IP spoofing, and is but one of many ways in which one user can masquerade as another user.

Internet history

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational

- 1972:
  - ARPAnet public demo
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes

INTERNET HISTORY:

- 1970: ALOHAnet satellite network in Hawaii
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- late70's: proprietary architectures: DECnet, SNA, XNA
- 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best-effort service model
- stateless routing
- decentralized control

define today's Internet architecture

INTERNET HISTORY:

- 1983: deployment of TCP/IP
- 1982: smtp e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: ftp protocol defined
- 1988: TCP congestion control

- new national networks: CSnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

INTERNET HISTORY:

- early 1990s: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990s: commercialization of the Web

late 1990s – 2000s:

- more killer apps: instant messaging, P2P file sharing
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gbps

INTERNET HISTORY:

- aggressive deployment of broadband home access (10-100's Mbps)
- 2008: software-defined networking (SDN)
- increasing ubiquity of high-speed wireless access: 4G/5G, WiFi
- service providers (Google, FB, Microsoft) create their own networks
  - bypass commercial Internet to connect "close" to end user, providing "instantaneous" access to social media, search, video content, …
- enterprises run their services in "cloud" (e.g., Amazon Web Services, Microsoft Azure)
- rise of smartphones: more mobile than fixed devices on Internet (2017)
- ~15B devices attached to Internet (2023, statista.com)

REVIEW QUESTIONS:

R1. What is the difference between a host and an end system? List several different types of end systems. Is a Web server an end system?

ANS:A host and an end system are terms often used interchangeably in computer networking, but they have subtle distinctions:**HOST**: A host is any device that can send or receive data over a network. It typically refers to a computer or server that hosts services or applications.Examples: Personal computers, servers, virtual machines.

End System: An end system is any device that is at the edge of the network and interacts directly with users or other end systems. It is a term used to describe devices that are endpoints in a network.Examples: PCs, smartphones, tablets, IoT devices, smart TVs.

Types of End Systems

- Personal Computers (PCs): Desktops, laptops.
- Mobile Devices: Smartphones, tablets.
- IoT Devices: Smart thermostats, smart refrigerators, connected home assistants.
- Gaming Consoles: PlayStation, Xbox.
- Smart TVs: Internet-connected televisions.
- Is a Web Server an End System?
- Yes, a web server is considered an end system. Although it serves web pages to clients (such as browsers on personal computers or mobile devices), it is still an endpoint in the network, interacting directly with users or other servers. It does not forward data like a router but instead processes and responds to requests directly.

R2:The word protocol is often used to describe diplomatic relations. How does Wikipedia describe diplomatic protocol?

Diplomatic protocol refers to the set of rules and guidelines governing the behavior and interactions of diplomats and state officials in various formal and ceremonial situations. This includes managing official ceremonies, state visits, and other diplomatic events. Key aspects of diplomatic protocol involve ensuring appropriate etiquette, precedence, and the correct use of titles and forms of address.

R3. Why are standards important for protocols?

Standards are important for protocols in computer networking because they ensure interoperability, allowing devices from different manufacturers to work together. They provide consistency and reliability, which aids in troubleshooting and network maintenance. Standards enable scalability, allowing networks to grow efficiently, and they enhance security by defining protocols that protect data. Additionally, they foster innovation by providing a stable foundation for new technologies and reduce costs by eliminating the need for proprietary solutions

R4. Four Access Technologies

DSL (Digital Subscriber Line)

Classification: Home access
Description: Uses existing telephone lines to provide internet connectivity to homes.

Ethernet

Classification: Enterprise access
Description: A wired networking technology commonly used in businesses and large organizations to connect computers within a local area network (LAN).

Fiber to the Home (FTTH)

Classification: Home access
Description: Provides high-speed internet by running optical fiber directly to residences.

3G/4G/5G Wireless

Classification: Wide-area wireless access
Description: Mobile network technologies that offer internet access over large geographic areas via cellular networks.

R5. HFC Transmission Rate and Collisions
Transmission Rate: Shared among users

Explanation: In Hybrid Fiber-Coaxial (HFC) networks, the transmission rate is shared among users within a particular segment of the network. This means that the available bandwidth is divided among multiple households connected to the same coaxial cable segment.
Collisions in Downstream HFC Channel: Not possible

Explanation: Collisions are not possible in the downstream HFC channel because data is transmitted from the central office (headend) to users in a unidirectional manner. The headend controls the flow of data, ensuring that each user receives their allocated data without interference from others. This differs from the upstream channel, where multiple users can send data back to the headend, potentially leading to collisions if not properly managed.

R7. Transmission Rate of Ethernet LANs
Common Transmission Rates:
10 Mbps (Ethernet)
100 Mbps (Fast Ethernet)
1 Gbps (Gigabit Ethernet)
10 Gbps (10 Gigabit Ethernet)
40 Gbps and 100 Gbps (Higher-speed Ethernet standards for data centers)

R8. Physical Media for Ethernet
Twisted Pair Cable (Cat5e, Cat6, Cat6a, Cat7, Cat8)
Coaxial Cable
Fiber Optic Cable (Single-mode and Multi-mode)
Wireless (IEEE 802.11 standards for Wi-Fi)
R9. Transmission Rates and Sharing for Residential Access Technologies
HFC (Hybrid Fiber-Coaxial)

Transmission Rates: 10 Mbps to 1 Gbps (downstream), 1 Mbps to 50 Mbps (upstream)
Rate Type: Shared among users in the same segment
DSL (Digital Subscriber Line)
Transmission Rates: 1 Mbps to 100 Mbps (downstream), 0.5 Mbps to 10 Mbps (upstream)
Rate Type: Dedicated per user line from the home to the DSLAM (Digital Subscriber Line Access Multiplexer)
FTTH (Fiber to the Home)

Transmission Rates: 100 Mbps to 10 Gbps (downstream and upstream)
Rate Type: Typically dedicated, providing consistent speeds

R10. Popular Wireless Internet Access Technologies

Wi-Fi (IEEE 802.11 Standards)

Description: Commonly used for local area networks in homes, offices, and public spaces.
Speeds: Vary from 54 Mbps (802.11g) to several Gbps (802.11ax - Wi-Fi 6).
Range: Typically 100-200 feet indoors.
Use Cases: Home networks, hotspots, enterprise networks.

Cellular Networks (3G, 4G LTE, 5G)

Description: Provides wide-area coverage through cellular towers, supporting mobile devices.
Speeds:
3G: Up to 2 Mbps.
4G LTE: Up to 1 Gbps.
5G: Up to 10 Gbps.
Range: Wide coverage area, often miles from the cell tower.
Use Cases: Mobile internet, remote access, IoT devices.

Comparison:

Wi-Fi is typically used for short-range, high-speed connections within a confined area, such as a home or office. It relies on a fixed infrastructure of routers and access points.
Cellular Networks provide broader coverage and mobility, making them suitable for outdoor and wide-area usage, but they may have higher latency and variable speeds depending on network congestion and coverage.

R11. Suppose there is exactly one packet switch between a sending host and a receiving host. The transmission rates between the sending host and the switch and between the switch and the receiving host are R1 and R2, respectively. Assuming that the switch uses store-and-forward packet switching, what is the total end-to-end delay to send a packet of length L? (Ignore queuing, propagation delay, and processing delay.)

$$\text{Total End-to-End Delay} = \text{Transmission Delay}_1 + \text{Transmission Delay}_2$$

$$\text{Total End-to-End Delay} = \frac{L}{R_1} + \frac{L}{R_2}$$

R12. What advantage does a circuit-switched network have over a packet-switched net work? What advantages does TDM have over FDM in a circuit-switched network?

Advantages of a Circuit-Switched Network over a Packet-Switched Network

Guaranteed Bandwidth and Quality of Service (QoS):

In a circuit-switched network, a dedicated circuit or path is established between the sender and receiver for the duration of the communication session. This guarantees a fixed bandwidth and consistent quality of service, which is crucial for real-time applications like voice calls.

Predictable Latency:

Since the path is dedicated and the resources are reserved, there is minimal variation in latency, providing a more predictable and stable communication experience.

No Congestion:

The dedicated nature of the connection ensures that there is no competition for bandwidth from other users, thus avoiding congestion and delays that can occur in packet-switched networks during periods of high traffic.

Simplicity:

Circuit-switched networks can be simpler to design and manage for certain types of communication, particularly for point-to-point connections that require continuous and predictable transmission rates.

Advantages of TDM over FDM in a Circuit-Switched Network

Efficient Use of Bandwidth:

Time Division Multiplexing (TDM) allocates time slots to different communication channels on the same frequency, allowing multiple signals to share the same transmission medium sequentially. This can be more efficient than Frequency Division Multiplexing (FDM), where each channel requires a separate frequency band, potentially leading to underutilization if some channels are not always in use.

Easier Implementation and Management:

TDM systems can be easier to implement and manage because they do not require the allocation and management of separate frequency bands for each communication channel. This makes it simpler to add or remove channels without the need for complex frequency coordination.

Reduced Interference:

TDM reduces the potential for interference between channels since only one channel transmits at any given time on the same frequency. In contrast, FDM channels, which operate simultaneously on different frequencies, can suffer from crosstalk and other forms of interference.

Scalability:

TDM systems can be more scalable in terms of adding more users or channels. Since time slots can be dynamically allocated, it's easier to accommodate varying traffic loads compared to FDM, where the number of available frequency bands is fixed and adding more channels can be more complex.

R13. Suppose users share a 2 Mbps link. Also suppose each user transmits continuously at 1 Mbps when transmitting, but each user transmits only 20 percent of the time. (See the discussion of statistical multiplexing in Section 1.3.)

a. When circuit switching is used, how many users can be supported?

b. For the remainder of this problem, suppose packet switching is used. Why will there be essentially no queuing delay before the link if two or fewer users transmit at the same time? Why will there be a queuing delay if three users transmit at the same time?

c. Find the probability that a given user is transmitting.

d. Suppose now there are three users. Find the probability that at any given time, all three users are transmitting simultaneously. Find the fraction of time during which the queue grows.

When using circuit switching, each user requires a dedicated 1 Mbps link when transmitting. Therefore, the number of users that can be supported is simply the total link capacity divided by the per-user capacity:

$$\text{Number of users} = \frac{\text{Total link capacity}}{\text{Per-user capacity}} = \frac{2\ \text{Mbps}}{1\ \text{Mbps}} = 2\ \text{users}$$

In packet switching, the link is shared among all users. Queuing delay occurs when more users transmit simultaneously than the link can handle.

No Queuing Delay with Two or Fewer Users: If two or fewer users transmit at the same time, the total transmission rate will be at most 2 Mbps, which the link can handle without delay.

Queuing Delay with Three Users: If three users transmit simultaneously, the total transmission rate would be 3 Mbps, exceeding the link capacity. This leads to queuing delay as packets wait to be transmitted.

c. Probability of a Given User Transmitting

Each user transmits 20% of the time, so the probability that a given user is transmitting at any given time is:

$$P(\text{user transmitting}) = 0.20$$

d. Probability of All Three Users Transmitting Simultaneously

The probability that all three users are transmitting at the same time can be found by multiplying the individual probabilities (since the events are independent):

$$P(\text{all three transmitting}) = (0.20)^3 = 0.008$$

The fraction of time during which the queue grows is the probability that the total transmission rate exceeds the link capacity (which happens when all three users transmit simultaneously):

$$\text{Fraction of time queue grows} = 0.008$$

R14. Why will two ISPs at the same level of the hierarchy often peer with each other? How does an IXP earn money?

- By peering directly, ISPs can reduce the costs associated with sending traffic through third-party transit providers.
- Peering typically results in lower latency and higher bandwidth, improving the overall performance and speed of data transmission between the networks of the peering ISPs.
- Peering helps balance the traffic load between networks, optimizing the use of their infrastructure and improving network reliability.

How an IXP Earns Money

An Internet Exchange Point (IXP) generates revenue through several means:

Membership Fees:

IXPs charge participating ISPs and other network operators membership or port fees. These fees can be based on the speed and number of ports used by the members.

Cross-Connect Fees:

Fees are charged for the physical connections between the members' equipment and the IXP infrastructure.

Value-Added Services:

IXPs may offer additional services such as data center space, technical support, traffic analysis, and DDoS mitigation, for which they charge extra fees.

Sponsorships and Partnerships:

IXPs might also receive funds through sponsorships from network equipment vendors, telecom operators, or other entities interested in promoting robust internet infrastructure.

R15. Some content providers have created their own networks. Describe Google's network. What motivates content providers to create these networks?

Google's Network

Google has developed an extensive private network infrastructure known as Google Global Cache (GGC). Here are some key features and aspects of Google's network:

Private Fiber Optic Network:

Google has invested heavily in its own fiber optic infrastructure, spanning across the globe. This network includes undersea cables, terrestrial fiber, and data centers strategically located in various regions. This allows Google to have greater control over the data flow and reduce reliance on third-party networks.

Data Centers:

Google operates numerous data centers worldwide, ensuring that content is close to users. This reduces latency and improves the speed of data delivery. These data centers are highly advanced, featuring state-of-the-art technology for efficiency and security.

Content Delivery Network (CDN):

GGC is part of Google's CDN, which caches content closer to end users. By placing popular content on servers that are geographically nearer to users, Google can significantly reduce the time it takes for data to travel over the internet.

Peering and Interconnects:

Google extensively peers with other networks at Internet Exchange Points (IXPs) and private peering locations. This helps in reducing transit costs and improving performance by minimizing the number of hops data must traverse.

Infrastructure as a Service (IaaS):

Google's network infrastructure supports its cloud services, such as Google Cloud Platform (GCP), which offers robust and scalable cloud computing solutions to businesses and developers.

- By building their own networks, content providers like Google can reduce latency and improve the speed at which content is delivered to users. This is particularly important for services requiring high bandwidth and low latency, such as video streaming, gaming, and real-time communication.
- Operating a private network can be more cost-effective in the long run compared to paying transit fees to third-party ISPs.
- Owning the infrastructure allows content providers to ensure higher reliability and better control over their data flow.
- Content providers can design their networks to scale according to their needs, supporting future growth and increasing traffic demands without relying on external networks' capacity limitations

R16. Consider sending a packet from a source host to a destination host over a fixed route. List the delay components in the end-to-end delay. Which of these delays are constant and which are variable?

Processing Delay: This is the time required to process the packet headers, check for bit-level errors, and determine the packet's destination. This delay occurs at each router along the path.

Constant or Variable: Generally considered constant but can vary slightly with the router's current load.

Queuing Delay: This is the time a packet spends waiting in the queue until it can be processed by the router. This delay occurs at each router along the path.**Constant or Variable:** Variable, as it depends on the amount of traffic and congestion at the router at any given time.**Transmission Delay:** This is the time required to push all the packet's bits onto the link. It depends on the packet's length and the transmission rate of the link.

Constant or Variable: Generally considered constant for a given packet size and link, but can vary if the packet size or link characteristics change.**Propagation Delay:** This is the time it takes for a signal to propagate from the sender to the receiver. It is a function of the distance between the two hosts and the propagation speed of the signal in the medium (e.g., fiber optic cable, copper wire).**Constant or Variable:** Constant, as it depends on the physical distance and the propagation speed, both of which are typically fixed for a given route.

R18. How long does it take a packet of length 1,000 bytes to propagate over a link of distance 2,500 km, propagation speed 2.5 # 108 m/s, and transmission rate 2 Mbps? More generally, how long does it take a packet of length L to propagate over a link of distance d, propagation speed s, and transmission rate R bps? Does this delay depend on packet length? Does this delay depend on transmission rate?

Transmission Delay:

$$\text{Transmission Delay} = \frac{8,000 \text{ bits}}{2 \times 10^6 \text{ bits/second}} = 0.004 \text{ seconds} = 4 \text{ milliseco}$$

Propagation Delay:

$$\text{Propagation Delay} = \frac{2,500,000 \text{ meters}}{2.5 \times 10^8 \text{ meters/second}} = 0.01 \text{ seconds} = 10 \text{ milli}$$

Total Delay:

$$\text{Total Delay} = \text{Transmission Delay} + \text{Propagation Delay} = 4 \text{ milliseconds} +$$

For a packet of length L to propagate over a link of distance d with propagation speed s and transmission rate R:

$$\text{Total Delay} = \frac{L}{R} + \frac{d}{s}$$

Dependence on Packet Length and Transmission Rate:

Propagation Delay does not depend on packet length or transmission rate ❓ R. It only depends on the distance d and the propagation speed s. Transmission Delay depends directly on the packet length L and the transmission rate R.A larger packet length or a lower transmission rate will increase the transmission

R19. Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates R1 = 500 kbps, R2 = 2 Mbps, and R3 = 1 Mbps.

a. Assuming no other traffic in the network, what is the throughput for the file transfer?

b. Suppose the file is 4 million bytes. Dividing the file size by the through put, roughly how long will it take to transfer the file to Host B?

c. Repeat (a) and (b), but now with R2 reduced to 100 kbps.

The throughput is determined by the smallest link rate, which is R1=500kbps

To convert the file size to bits:

$$\text{File size} = 4,000,000 \text{ bytes} \times 8 = 32,000,000 \text{ bits}$$

Time to transfer the file:

$$\text{Transfer time} = \frac{\text{File size}}{\text{Throughput}} = \frac{32,000,000 \text{ bits}}{500,000 \text{ bits/second}} = 64 \text{ seconds}$$

The throughput is determined by the smallest link rate, which is R2=100kbps

2. **Calculate the transfer time:**

$$\text{Transfer time} = \frac{\text{File size}}{\text{Throughput}} = \frac{32,000,000 \text{ bits}}{100,000 \text{ bits/second}} = 320 \text{ seconds}$$

R20. Suppose end system A wants to send a large file to end system B. At a very high level, describe how end system A creates packets from the file. When one of these packets arrives to a router, what information in the packet does the router use to determine the link onto which the packet is forwarded? Why is packet switching in the Internet analogous to driving from one city to another and asking directions along the way?

End system A segments a large file into smaller packets.
Each packet is constructed with a header (containing source and destination IP addresses, sequence numbers, etc.) and a payload (the file segment).
When a packet arrives at a router, the router uses the destination IP address in the packet header to look up its routing table.
The routing table determines the next link or next-hop router for forwarding the packet.

ANALOGY:
Packet switching is like driving from one city to another and asking for directions along the way. Just as you might need to adjust your route based on local advice, packets can take varying paths through the network based on current conditions and router decisions

R22: Tasks Performed by a Layer

Layers in a network protocol stack can perform various tasks, including:

Encapsulation: Wrapping data with the appropriate headers and trailers.

Error Detection and Correction: Identifying and correcting errors in data transmission.

Flow Control: Managing the rate of data transmission to prevent overload.

Routing and Forwarding: Determining the path and sending data to the next destination.

Session Management: Establishing, maintaining, and terminating connections between applications.

Overlap Across Layers:

Encapsulation: Can be performed at multiple layers (e.g., data is encapsulated in segments, packets, and frames).

Error Detection: Can occur at several layers (e.g., link-layer frames often have checksums, and transport-layer segments also include error-checking mechanisms).

R23: Five Layers in the Internet Protocol Stack

Application Layer:

Responsibilities: Provides network services directly to applications (e.g., HTTP, FTP, SMTP).

Transport Layer:

Responsibilities: Manages end-to-end communication, provides reliable data transfer (e.g., TCP, UDP).

Network Layer:

Responsibilities: Handles logical addressing and routing (e.g., IP).

Link Layer:

Responsibilities: Manages data transfer between directly connected nodes and error detection/correction (e.g., Ethernet, Wi-Fi).

Physical Layer:

Responsibilities: Transmits raw bit streams over physical media (e.g., cables, switches).

R24: Network Message Types

Application-Layer Message: Data exchanged between applications (e.g., HTTP request/response).

Transport-Layer Segment: A unit of data with transport-layer headers (e.g., TCP segment, UDP segment).

Network-Layer Datagram: A packet with network-layer headers (e.g., IP packet).

Link-Layer Frame: A packet with link-layer headers and trailers (e.g., Ethernet frame).

R25: Layer Processing by Different Devices

Router:

Processes: Network Layer (routing and forwarding), sometimes the Link Layer (handling data frames on each interface).

Link-Layer Switch:

Processes: Link Layer (frame switching and filtering).

Host:

Processes: All layers (Application, Transport, Network, Link, and Physical), as hosts must handle end-to-end communication, including application interactions and network communication.

R26: WHAT IS SELF-REPLICATING MALWARE?Self-replicating malware is a type of malicious software that has the capability to duplicate itself and spread to other systems without user intervention.

Viruses: Malicious programs that attach themselves to legitimate files or programs. They replicate by infecting other files and are often activated when the infected file is executed.

Worms: Standalone malware that spreads independently by exploiting vulnerabilities in systems or networks. Unlike viruses, worms do not need to attach themselves to a file; they can self-replicate and propagate across networks.

R27: Botnet Creation and DDoS Attacks

Creating a Botnet:

Infection: Botnets are created by infecting a large number of computers (bots) with malware. This is typically achieved through phishing emails, malicious downloads, or exploiting vulnerabilities.

Control: Once infected, these computers connect to a central command-and-control (C&C) server or peer-to-peer network controlled by the attacker.

Propagation: The botnet malware may include functionality to self-replicate and infect other systems, further expanding the botnet.

Using a Botnet for DDoS Attacks:

Command Issuance: The attacker sends a command from the C&C server to the bots, instructing them to initiate a coordinated attack.

Attack Execution: The bots flood the target with a high volume of traffic or requests, overwhelming its resources and causing service disruptions or downtime.

Attack Types: Common DDoS attack types include flooding the target with data (e.g., SYN flood, UDP flood) or exploiting application vulnerabilities.

R28: Malicious Actions by Trudy

Trudy, positioned as a man-in-the-middle (MITM), can perform several malicious activities:

Interception and Eavesdropping:

Capture Sensitive Data: Trudy can intercept packets containing sensitive information (e.g., passwords, personal data).

Data Injection:

Inject Malicious Content: Trudy can modify or insert malicious data into packets sent by Alice or Bob, potentially causing harm or spreading malware.

Session Hijacking:

Take Over Sessions: By injecting packets, Trudy can hijack ongoing sessions and impersonate Alice or Bob, gaining unauthorized access to accounts or systems.

Spoofing:

Send False Information: Trudy can spoof messages, misleading Alice or Bob with false information or instructions.

Man-in-the-Middle Attacks:

Alter Communication: Trudy can alter the communication between Alice and Bob, redirecting or tampering with their messages.