Computer Networks

# Lecture 01: Introduction

**Prof. Dr. Sahar M. Ghanem**

Associate Professor

Computer and Systems Engineering Department

Faculty of Engineering, Alexandria University

# Course Outline

- Textbook: Computer Networking: A Top-Down Approach, 8th ed., Kurose & Ross
- Grading:
  - attendance & participation: 5-7
  - assignments & quizzes: 40
  - midterm: 15
  - final: 40
- Join with code: **l42tcab**
- Course materials and discussions will be on MS Teams.
- TA: Eng. Mohamed Essam

# Chapter 1

# Computer Networks and the Internet

# Outline

- What Is the Internet?
- The Network Edge
- The Network Core
- Delay, Loss, and Throughput in Packet-Switched Networks
- Protocol Layers and Their Service Models

# What Is the Internet?

# Overview

- We'll learn that the Internet is a **network of networks**, and we'll learn how these networks connect with each other.
- We'll use **the public Internet**, a specific computer network, as our principal vehicle for discussing computer networks and their protocols.

# A Nuts-and-Bolts Description (1/3)

- The Internet is a computer network that interconnects **billions of computing devices** throughout the world.
- All of these devices are called **hosts** or **end systems**.
- By some estimates, there were about 18 billion devices connected to the Internet in 2017, and the number will reach **28.5 billion by 2022**.
- End systems are connected together by a network of **communication links** and **packet switches**.
- A **packet switch** takes a packet arriving on one of its incoming communication links and forwards that packet on one of its outgoing communication links.

# A Nuts-and-Bolts Description (2/3)

- The **transmission rate** of a link measured in bits/second (bps).
- The two most prominent types of packet switches in today's Internet are **routers** and **link-layer switches**.
- The sequence of communication links and packet switches traversed by a packet from the sending end system to the receiving end system is known as a **route** or **path** through the network.
- End systems access the Internet through **Internet Service Providers (ISPs)**.
- Each **ISP** is in itself a **network** of packet switches and communication links.

# A Nuts-and-Bolts Description (3/3)

- End systems, packet switches, and other pieces of the Internet run **protocols**.
- The **Transmission Control Protocol (TCP)** and the **Internet Protocol (IP)** are two of the most important protocols in the Internet.
    - The Internet's principal protocols are collectively known as **TCP/IP**.
- **Internet standards** are developed by the **Internet Engineering Task Force (IETF)**.
- The IETF standards documents are called **requests for comments (RFCs)**.
    - There are currently nearly 9000 RFCs.
    - Other bodies also specify standards for network components, e.g. the **IEEE 802** LAN Standards Committee.

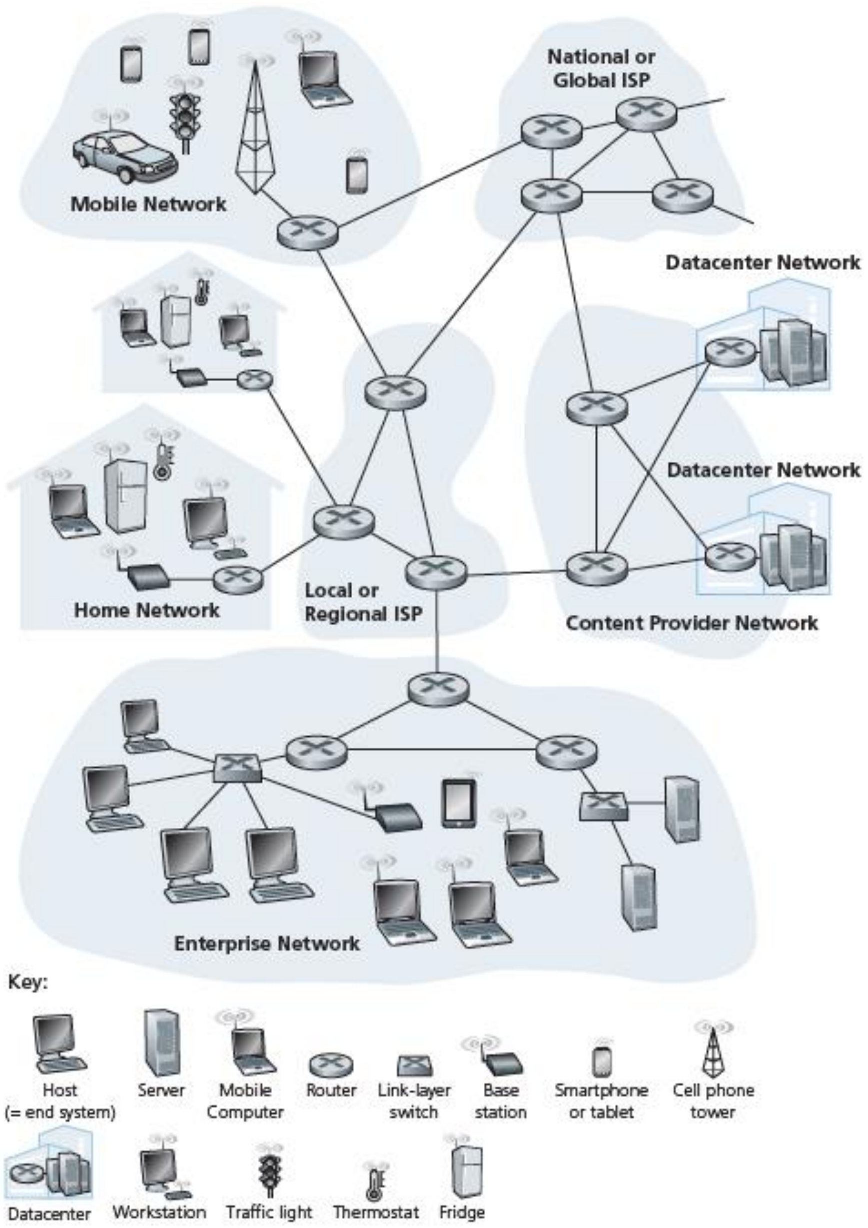


**Figure 1.1** ♦ Some pieces of the Internet

# A Services Description

- *The Internet is an infrastructure that provides services to **distributed applications**.*
- Internet applications run on end systems—they **do not run** in the packet switches in the network core.
- End systems attached to the Internet provide a **socket interface** that specifies how a program asks the Internet infrastructure to deliver data to another end system.
- The Internet provides **multiple services** to its applications.

# What Is a Protocol?

- It takes two (or more) communicating entities running the same **protocol** in order to accomplish a task.
- In a **human protocol**, *there are specific messages we send, and specific actions we take in response to the received reply messages or other events* (such as no reply within some given amount of time).
- Much of this course is about computer network protocols.
- *A **protocol** defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event.*

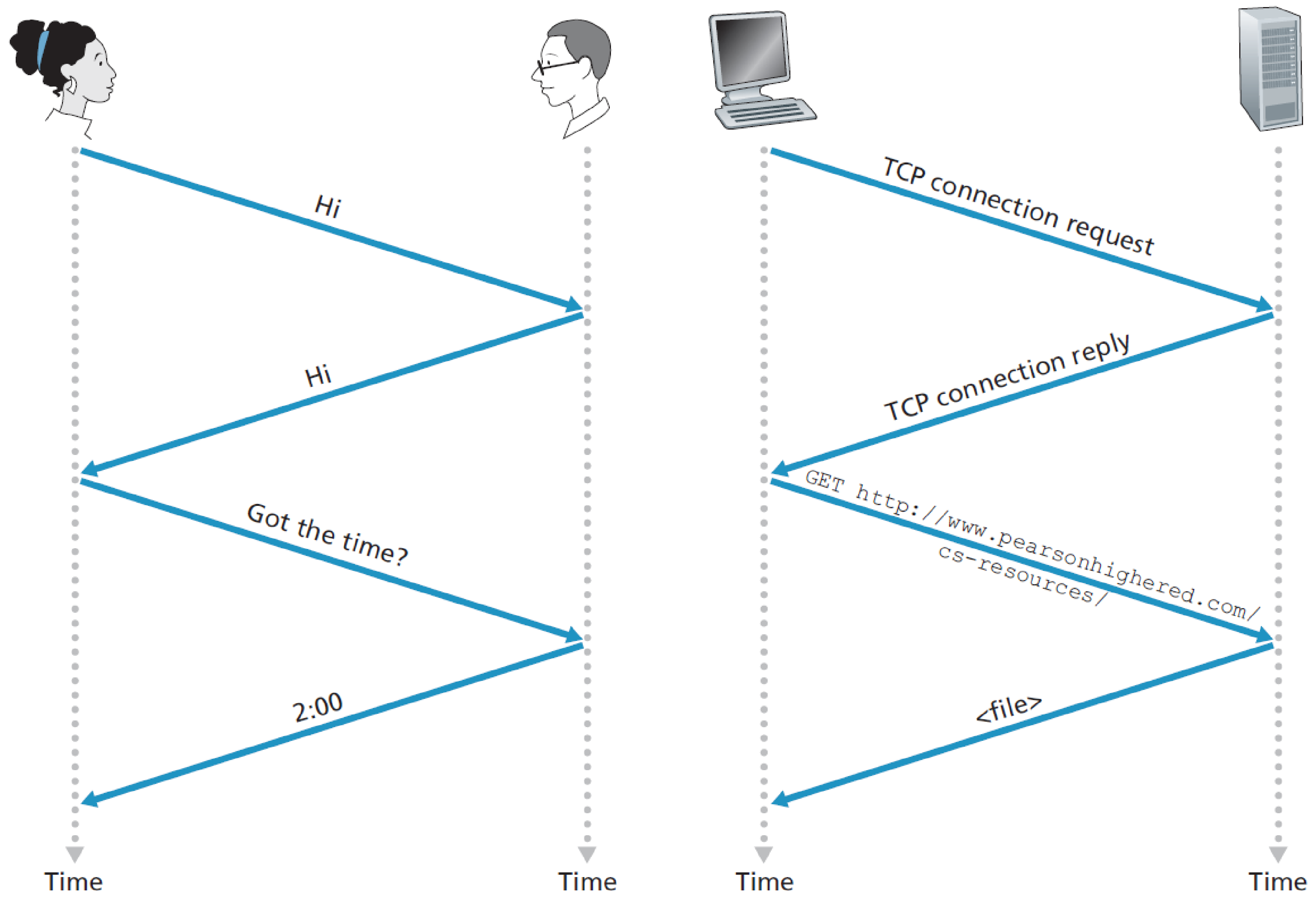


**Figure 1.2** ♦ A human protocol and a computer network protocol

# The Network Edge

# End Systems

- The Internet's **end systems** include desktop computers (e.g., desktop PCs, Macs, and Linux boxes), servers (e.g., Web and e-mail servers), and mobile devices (e.g., laptops, smartphones, and tablets). Furthermore, an increasing number of non-traditional **"things"** are being attached to the Internet as end systems.
- End systems are also referred to as ***hosts*** because they host (that is, run) application programs.
- Hosts are sometimes further divided into two categories: **clients** and **servers**.
- Most of the servers reside in large **data centers**.
  - For example, as of 2020, Google has **19 data centers** on four continents, collectively containing **several million servers**.

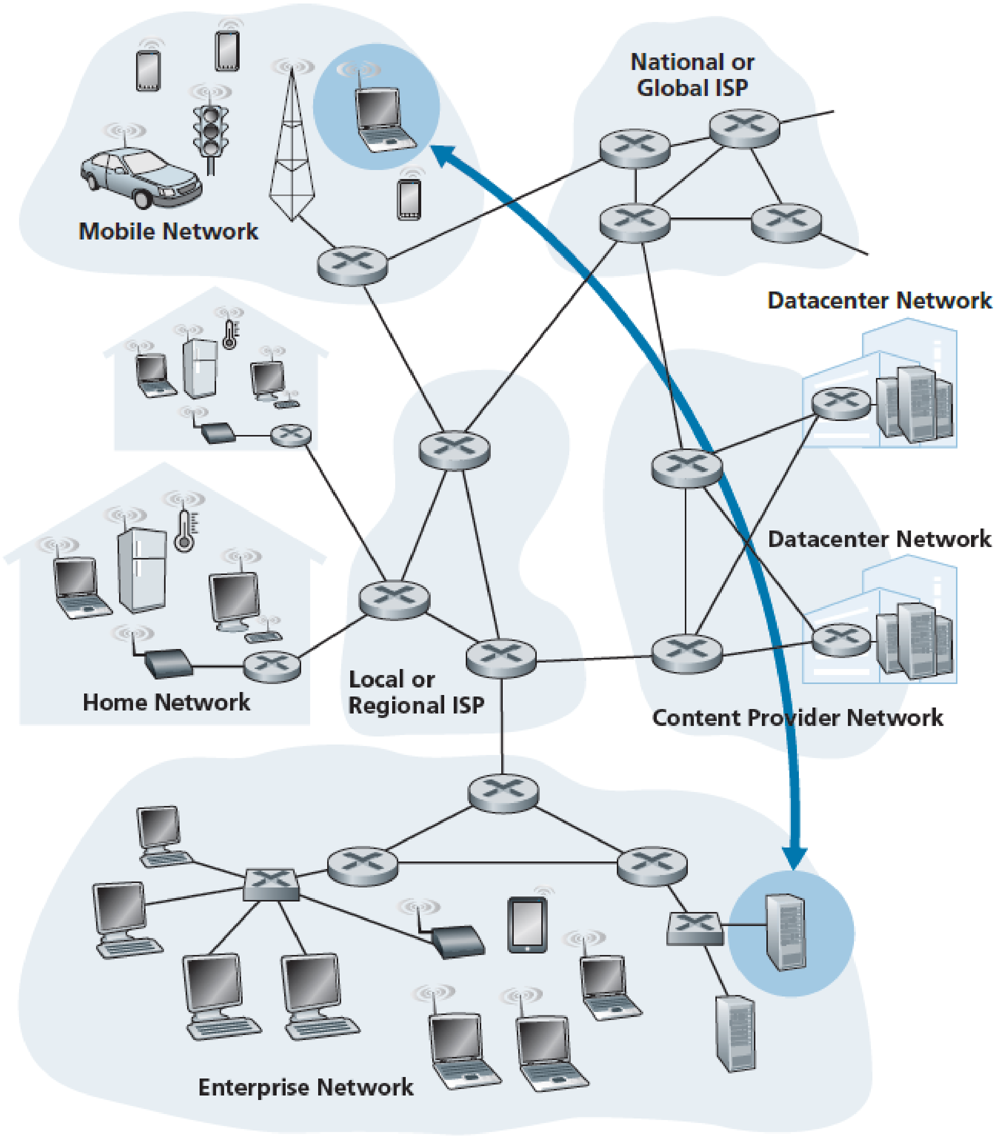


**Figure 1.3** ♦ End-system interaction

# Access Networks

- Home Access: DSL, Cable, FTTH, and 5G Fixed Wireless
- Access in the Enterprise (and the Home): Ethernet and WiFi
- Wide-Area Wireless Access: 3G and LTE 4G and 5G

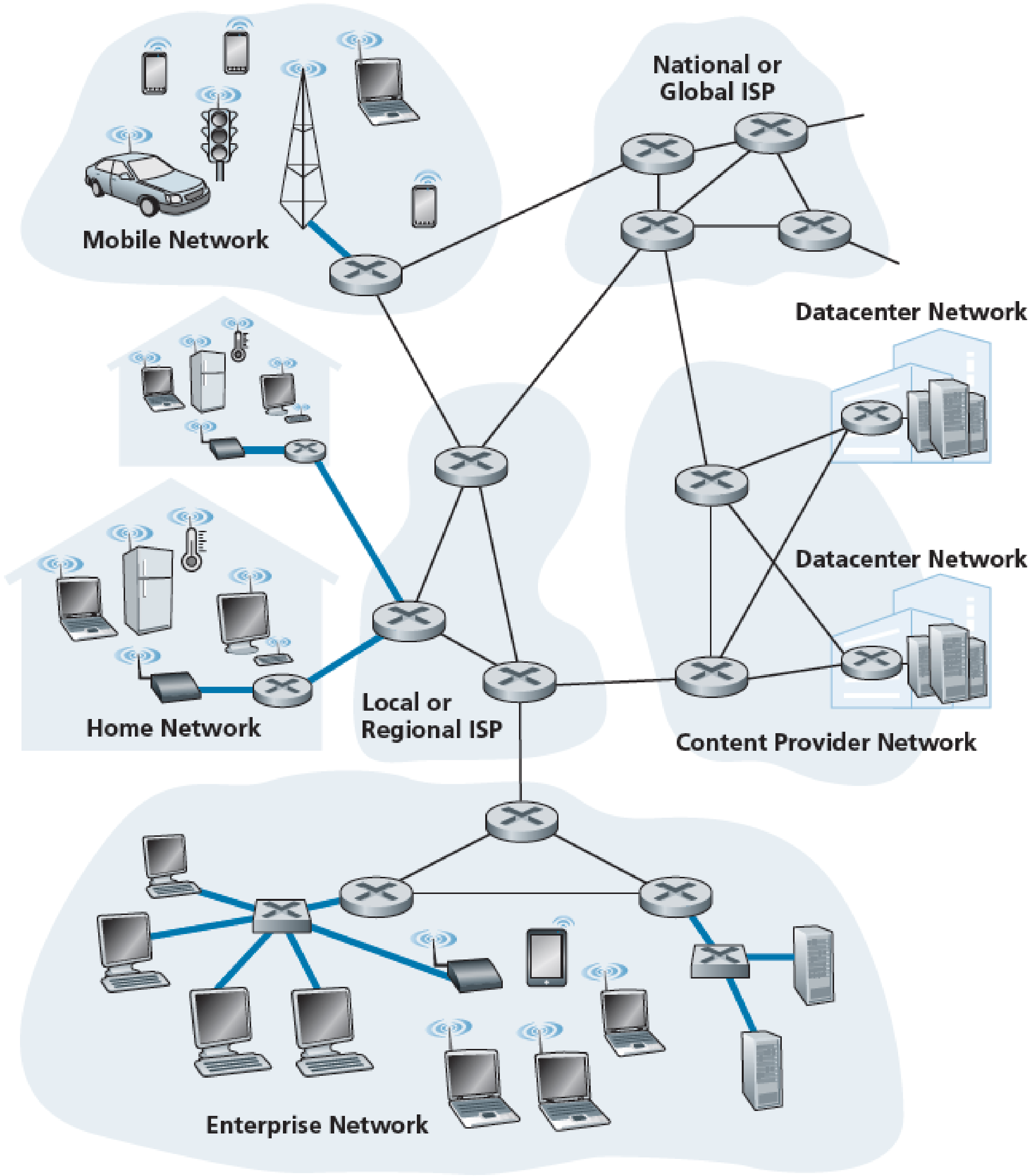


**Figure 1.4** ♦ Access networks

# Home Access: DSL (1/2)

- When **digital subscriber line (DSL)** is used, a customer's telco is also its ISP.
- A DSL modem uses the existing telephone line to exchange data with a **digital subscriber line access multiplexer (DSLAM)** located in the telco's local central office (CO).
- The residential telephone line carries both data and traditional telephone signals simultaneously, which are encoded at different **frequencies**:
  - A high-speed downstream channel, in the **50 kHz to 1 MHz** band
  - A medium-speed upstream channel, in the **4 kHz to 50 kHz** band
  - An ordinary two-way telephone channel, in the **0 to 4 kHz** band

# Home Access: DSL (2/2)

- On the customer side, a **splitter** separates the data and telephone signals arriving to the home and forwards the data signal to the DSL modem.
- On the telco side, in the CO, the **DSLAM** separates the data and phone signals and sends the data into the Internet.
  - Hundreds or even thousands of households connect to a single DSLAM.
  - Downstream transmission rates of **24 Mbs and 52 Mbs**
  - upstream rates of **3.5 Mbps and 16 Mbps**
  - the newest standard provides for aggregate upstream plus downstream rates of **1 Gbps**
- DSL is designed for short distances between the home and the CO.
  - located within **5 to 10 miles** of the CO. (1 mile=1.6 km)

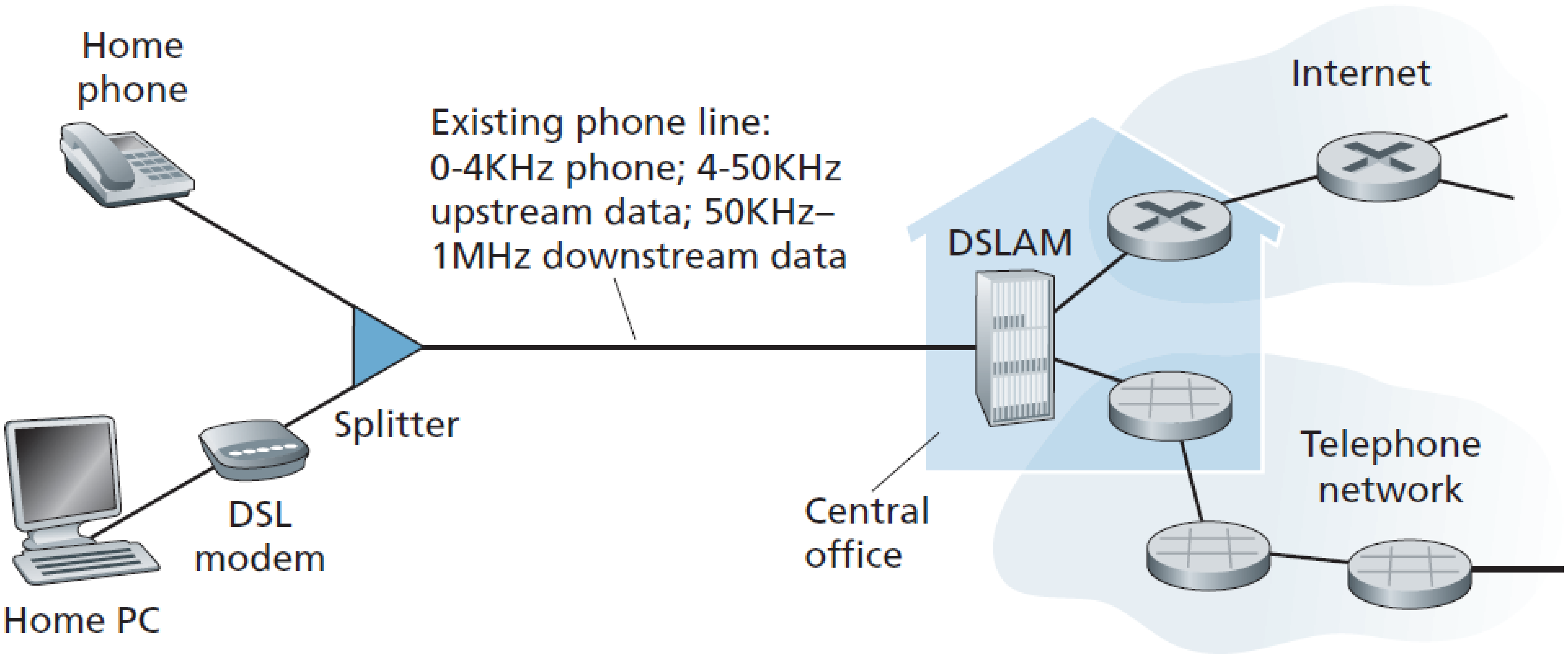


**Figure 1.5** ♦ DSL Internet access

# Other Home Access

- **Cable Internet access** makes use of the cable television company's existing cable television infrastructure.
  - It is often referred to as hybrid fiber coax (HFC) and is a shared broadcast medium.
  - downstream bitrates of 40 Mbps and 1.2 Gbps, and upstream rates of 30 Mbps and 100 Mbps.
- **Fiber to the home (FTTH)** provides even higher speeds is that can potentially provide Internet access rates in the gigabits per second range.
- **5G fixed wireless** promises high-speed residential access, without installing costly and failure-prone cabling from the telco's CO to the home.

# Access in the Enterprise/Home: Ethernet

- **A local area network (LAN)** is used to connect an end system to the edge router. **Ethernet** users use twisted-pair copper wire to connect to an Ethernet **switch**.
- With **Ethernet** access:
  - users typically have **100 Mbps to tens of Gbps** access to the Ethernet switch
  - servers may have **1 Gbps to 10 Gbps** access

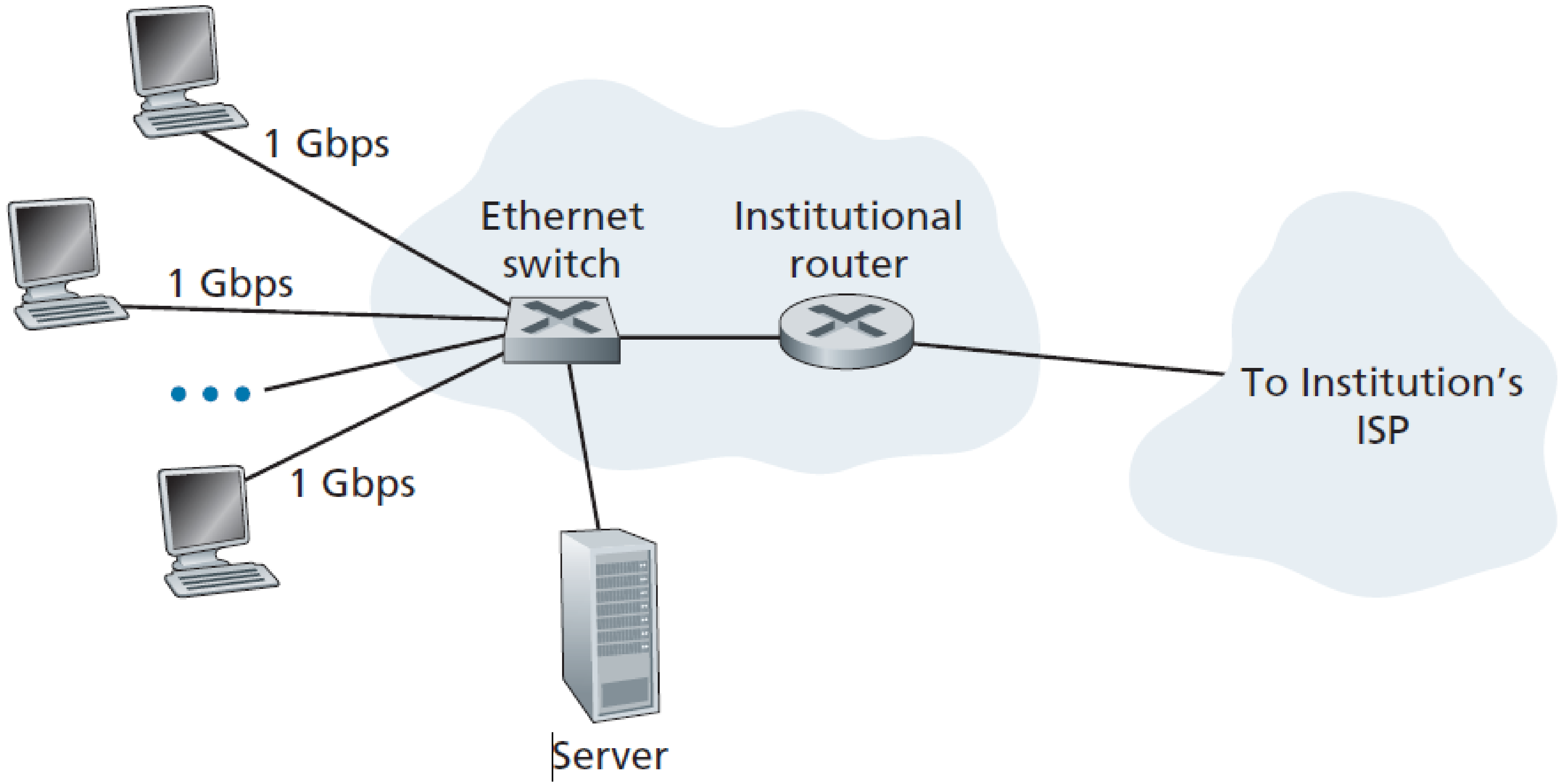


**Figure 1.8** ♦ Ethernet Internet access

# Access in the Enterprise/Home: WiFi

- Wireless LAN access based on IEEE 802.11 technology (**WiFi**) is now just about everywhere.
- A wireless LAN user must typically be within a **few tens of meters** of the access point.
- 802.11 today provides a shared transmission rate of up to more than **100 Mbps**.
- e.g. home network
  - a roaming laptop, multiple home appliances, as well as a wired PC
  - a base station (**WiFi access point**) that communicates with the wireless PC and other wireless devices in the home
  - a **home router** that connects the wireless access point, and any other wired home devices, to the Internet.

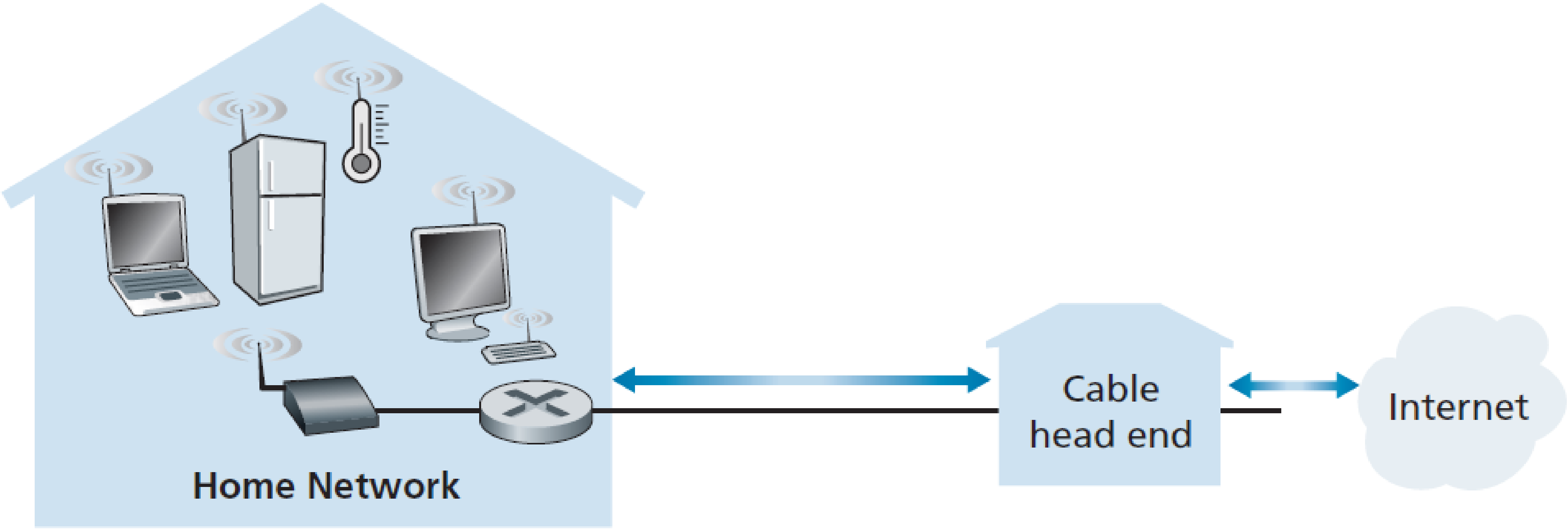


**Figure 1.9** ♦ A typical home network

# Wide-Area Wireless Access: 3G and LTE 4G and 5G

- Mobile devices employ the same wireless infrastructure used for **cellular telephony** to send/receive packets through a base station that is operated by the cellular network provider.
- A user need only be within a few **tens of kilometers** (as opposed to a few **tens of meters**) of the base station.
- **4G** wireless provides real-world download speeds of up to **60 Mbps**.
- **5G** will provide even higher-speed.

# Physical Media

- A  bit traveling from source to destination, passes through a series of transmitter-receiver pairs and it is sent by propagating electromagnetic waves or optical pulses across a **physical medium**.
- Physical media fall into two categories: **guided media** and **unguided media**.
- With **guided media**, the waves are guided along a solid medium, such as a fiber-optic cable, a twisted-pair copper wire, or a coaxial cable.
- With **unguided media**, the waves propagate in the atmosphere and in outer space, such as in a wireless LAN or a digital satellite channel.

# The Network Core

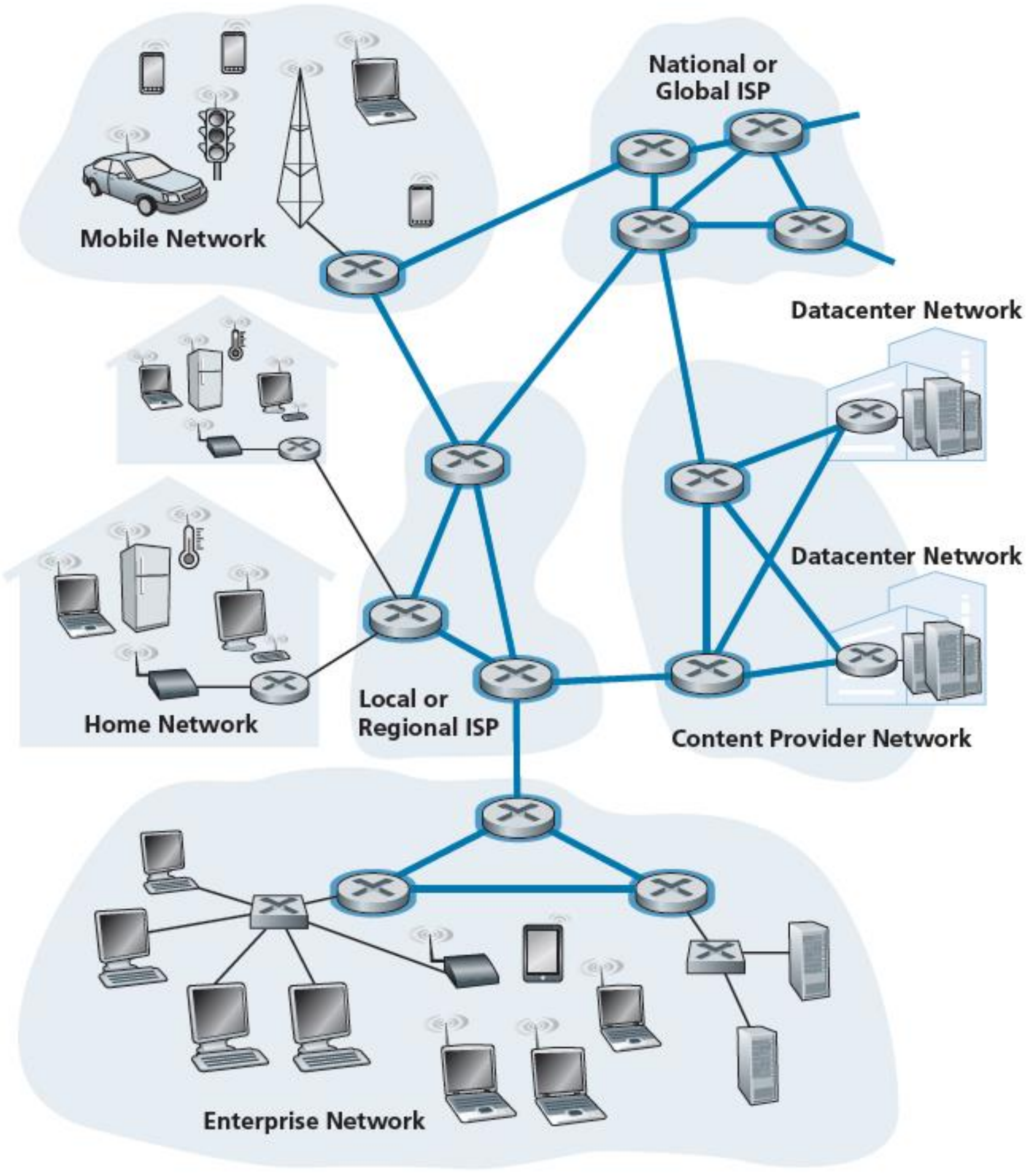


**Figure 1.10** ♦ The network core

# Packet Switching (1/3)

- In a network application, end systems exchange **messages** with each other.
- The source breaks long messages into smaller chunks of data known as **packets**.
- Each packet travels through communication links and **packet switches**.
    - **routers** and **link-layer switches**
    - a router will typically have many incident links
- Most packet switches use **store-and-forward transmission** at the inputs to the links. That is it must receive the entire packet before it can begin to transmit the first bit of the packet onto the outbound link.

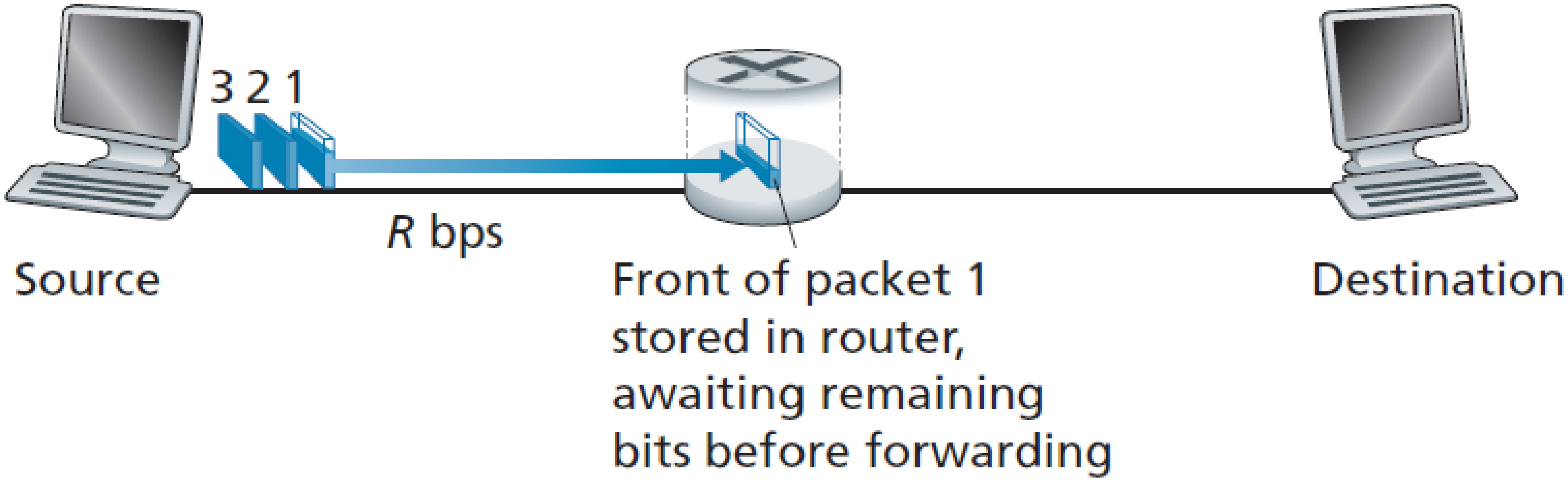


**Figure 1.11** ♦ Store-and-forward packet switching

# Packet Switching (2/3)

- Each packet consisting of $\boldsymbol{L}$ bits; Transmission rate is $\boldsymbol{R}$ bits/sec.
- Sending one packet from source to destination over a path consisting of $\boldsymbol{N}$ links ($\boldsymbol{N-1}$ routers) each of rate $\boldsymbol{R}$, the delay

$$\boldsymbol{d_{end-to-end} = N\,L/R}$$

  - ignoring propagation delay
- For each attached link, the packet switch has an **output buffer/queue**, which stores packets that the router is about to send into that link.
- In addition to the store-and-forward delays, packets suffer output buffer **queuing delays**
  - that depend on the level of congestion in the network

# Packet Switching (3/3)

- The amount of buffer space is finite, therefore **packet loss** will occur—either the arriving packet or one of the already-queued packets will be dropped

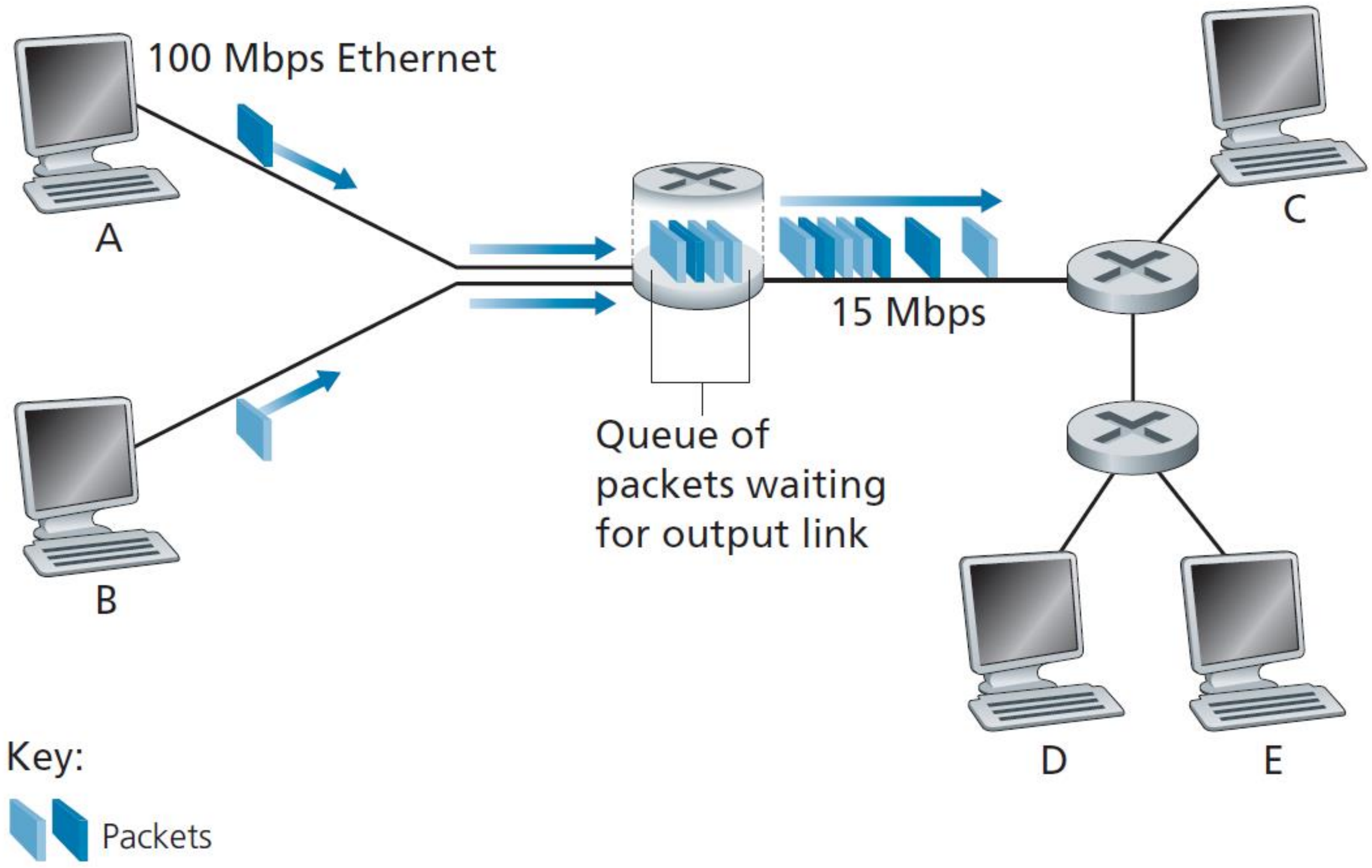


**Figure 1.12** ♦ Packet switching

# Forwarding Tables and Routing Protocols

- How does the router determine which link it should forward the packet onto?
- Every end system has an address called an **IP address** that has a hierarchical structure.
- The **destination's IP address** is in the packet's header.
- Each router has a **forwarding table**.
- A router uses a packet's destination address to index a forwarding table and determine the appropriate outbound link.
- the Internet has a number of special **routing protocols** that are used to automatically set the forwarding tables.

# Circuit Switching

- Traditional **telephone networks** are examples of circuit-switched networks.
- In **circuit-switched** networks, the resources needed along a path (buffers, link transmission rate) are ***reserved*** for the duration of the communication session between the end systems.
- When two hosts want to communicate, the network establishes a dedicated **end-to-end connection** between the two hosts.
    - The sender can transfer the data to the receiver at the ***guaranteed*** constant rate.
- The Internet makes its **best effort** to deliver packets in a timely manner, but it does not make any guarantees.

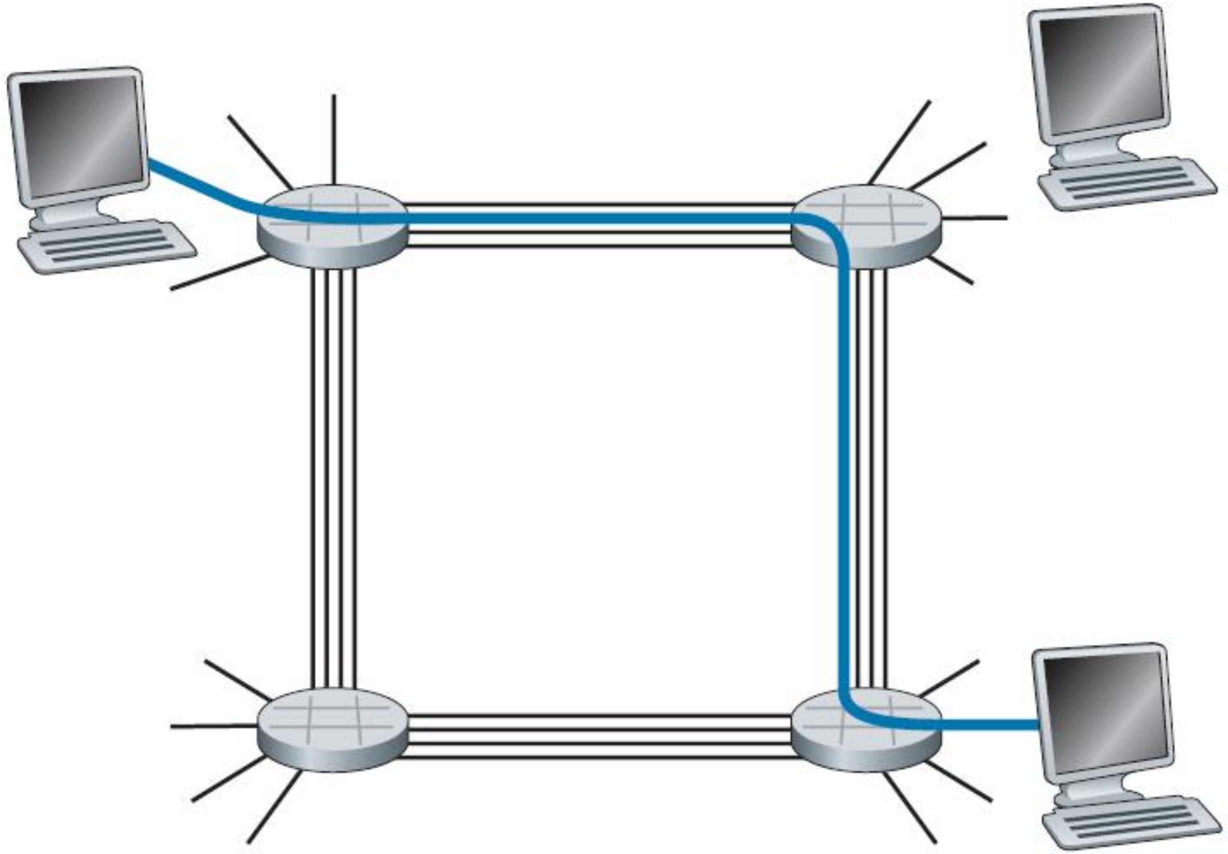

**Figure 1.13** ♦ A simple circuit-switched network consisting of four switches and four links

# Multiplexing in Circuit-Switched Networks

- A circuit in a link is implemented with either **frequency-division multiplexing (FDM)** or **time-division multiplexing (TDM)**.
- e.g.
  - **FM radio stations** 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**.
- Circuit switching is wasteful because the dedicated circuits are idle during **silent periods**.
- Establishing end-to-end circuits is complicated and requires **complex signaling software** to coordinate the operation of the switches along the end-to-end path

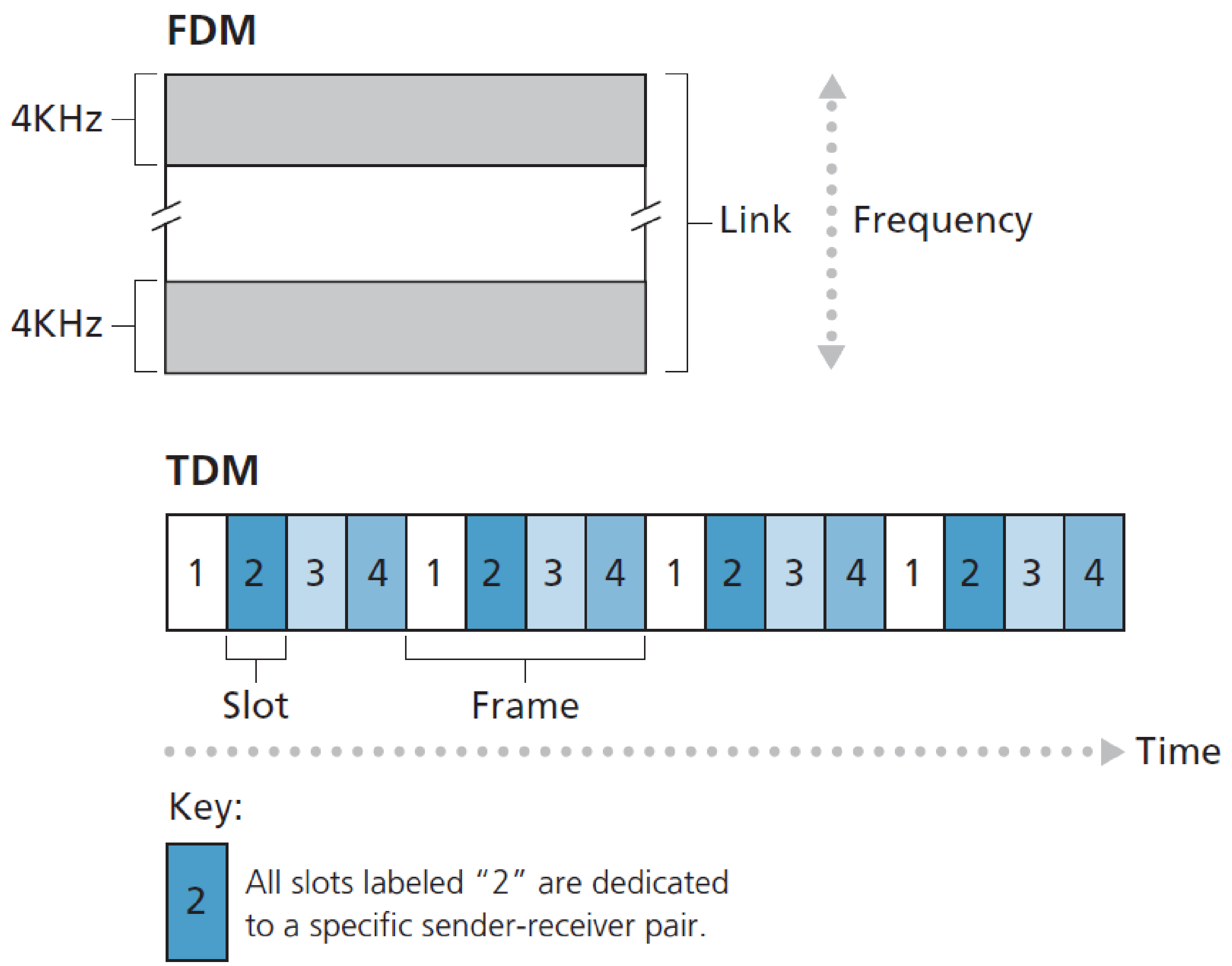


**Figure 1.14** ♦ With FDM, each circuit continuously gets a fraction of the bandwidth. With TDM, each circuit gets all of the bandwidth periodically during brief intervals of time (that is, during slots)

# Example #1

- How long it takes to send a file of **640,000 bits** from Host A to Host B over a circuit-switched network.
- All links use TDM with **24 slots** and have a bit rate of **1.536 Mbps**.
- It takes **500 msec** to establish an end-to-end circuit.
- Answer
  - Each circuit has a transmission rate of **(1.536 Mbps)/24 = 64 kbps**.
  - It takes **(640,000 bits)/(64 kbps) = 10 seconds** to transmit the file
  - Adding the circuit establishment time, giving **10.5 seconds** to send the file.

# Packet Switching Versus Circuit Switching

- Packet switching is **not** suitable for **real-time services** (telephone calls and video conference calls).
- Packet switching offers better **sharing** of transmission capacity and is **simpler**, more **efficient**, and **less costly** to implement.
- Circuit switching **pre-allocates** use of the transmission link regardless of demand, with allocated but unneeded link time going unused.
- Packet switching on the other hand allocates link use ***on demand***.

# Example #2 (1/2)

- Suppose users share a **1 Mbps** link.
- A user is active only **10 percent** of the time
- Each user alternates between periods of activity, when a user generates data at a constant rate of **100 kbps.**
- With **circuit switching**, **100 kbps** must be *reserved* for *each* user at all times.
- Circuit-switched link can support only **10 (= 1 Mbps/100 kbps)** simultaneous users.

# Example #2 (2/2)

- If there are **35 users**, the probability that there are **11 or more** simultaneously active users is approximately **0.0004**.
- When there are **10 or fewer** active users, users' packets flow through the link without delay.
- Because the probability of having **more than 10** simultaneously active users is minuscule in this example, packet switching provides essentially the **same performance** as circuit switching, *but does so while allowing for* ***more than three times the number of users.***

# Example #3 (1/2)

- There are **10 users** and that one user suddenly generates one **thousand 1,000-bit packets**, while other users remain quiescent and do not generate packets.
- Under **TDM circuit switching** with **10 slots** per frame and each slot consisting of **1,000 bits**.
  - The active user can only use its one time slot per frame to transmit data, while the remaining nine time slots in each frame remain idle.
- It will be **10 seconds** before all of the active user's one million bits of data has been transmitted.

# Example #3 (2/2)

- In the case of **packet switching**, the active user can continuously send its packets at the full link rate of **1 Mbps**.
- All of the active user’s data will be transmitted within **1 second**.

# A Network of Networks (1/4)

- Over the years, the network of networks that forms the Internet has evolved into a very **complex structure**.
- Much of this evolution is driven by **economics and national policy**, rather than by performance considerations.
- One **naive approach** would be to have each **access ISP** *directly* connect with every other **access ISP**. (hundreds of thousands access ISPs all over the world)
- **Network Structure 1**: interconnects all of the access ISPs with a *single global transit ISP.* (costly global ISP)
- **Network Structure 2 (**two-tier hierarchy**)**: hundreds of thousands of access ISPs and *multiple* global transit ISPs. (competing global transit providers as a function of their pricing and services)

# A Network of Networks (2/4)

- **Network Structure 3** (multi-tier hierarchy):
  - In any given region, there is a **regional ISP** to which the access ISPs in the region connect.
  - Each regional ISP then connects to **tier-1 ISPs** that do not have a presence in every city in the world. (a **dozen tier-1 ISPs**)
  - Each access ISP pays the regional ISP to which it connects, and each regional ISP pays the tier-1 ISP to which it connects.
  - There may be a larger regional ISP to which the smaller regional ISPs in that region connect.

# A Network of Networks (3/4)

- **Network Structure 4:** To build a network that more closely resembles today's Internet, we must add to the hierarchical Network Structure 3
  - **points of presence (PoPs):** a group of one or more routers in the provider's network where customer ISPs can connect into the provider ISP (not at the access level)
  - **multi-homing:** connect to two or more provider ISPs
  - **peering:** pair of nearby ISPs at the same level of the hierarchy can directly connect their networks together
  - **Internet exchange points (IXPs):** a third-party company can create an **IXP** that is a meeting point where multiple ISPs can peer together

# A Network of Networks (4/4)

- **Network Structure 5**: builds on top of Network Structure 4 by adding **content-provider networks**.
- e.g.: The **Google data centers** are all interconnected via Google's private TCP/IP network, which spans the entire globe but is nevertheless separate from the public Internet.

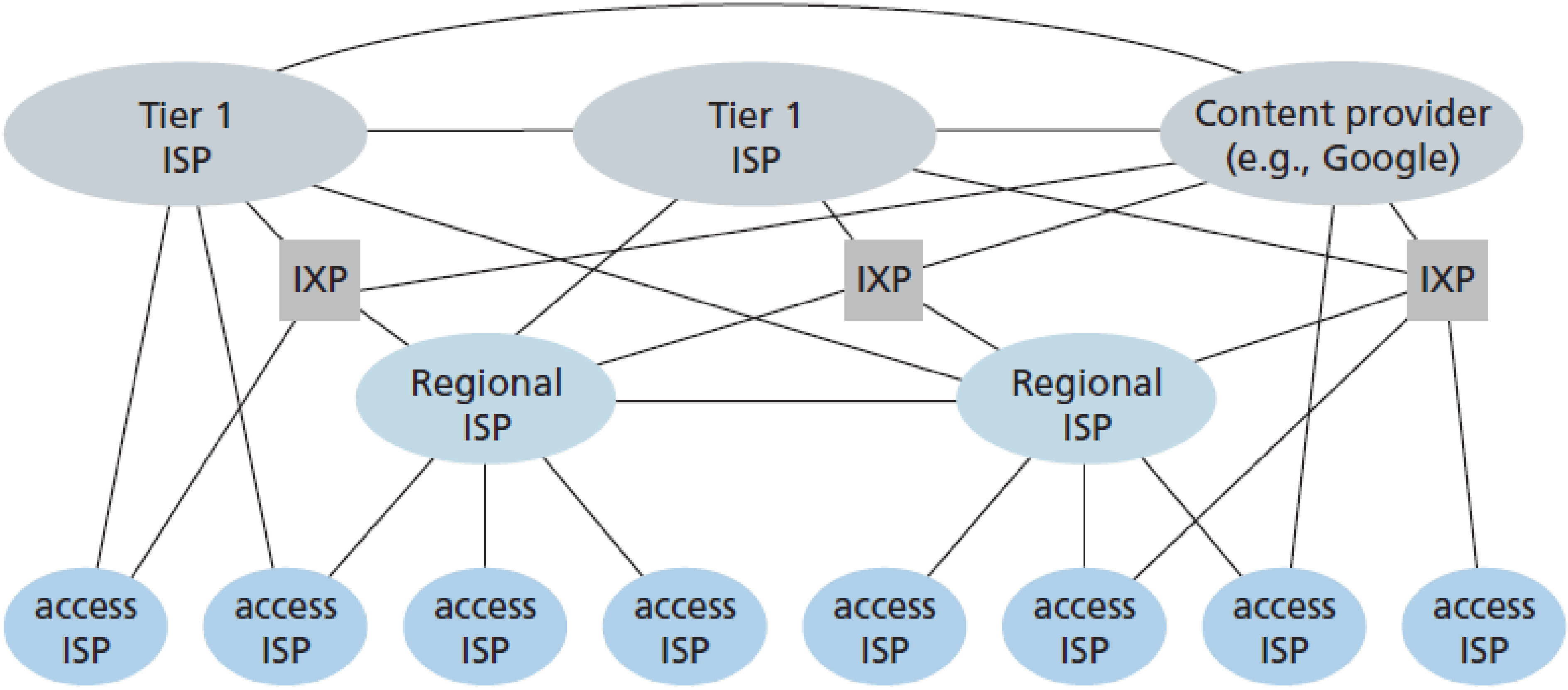


**Figure 1.15** ♦ Interconnection of ISPs

# Delay, Loss, and Throughput

# Overview

- the physical laws introduce **delay** and **loss** as well as constrain **throughput**
  - **throughput** is the amount of data per second that can be transferred between end systems
- The packet suffers from several **types of delays** at *each* node along the path
  - **processing delay** (microseconds or less)
  - **queuing delay** **(**microseconds to milliseconds**)** **(**depend on the number of earlier-arriving packets)
  - **transmission delay** is $\boldsymbol{L/R}$ (packet length $L$ bits; $R$ transmission rate in bps) (amount of time required to push all of the packet's bits into the link)
  - **propagation delay** ($\boldsymbol{d/s}$ distance between two routers divided by the propagation speed) (depends on the physical medium) (milliseconds)

    $2 \times 10^8$ meters/sec to $3 \times 10^8$ meters/sec

# Nodal Delay

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

The contribution of these delay components can vary significantly.

- e.g. LAN: $d_{prop}$ is negligible
- e.g. routers interconnected by a geostationary satellite link: $d_{prop}$ is hundreds of milliseconds (dominant)
- The processing delay, $d_{proc}$, is often negligible
  - however, it strongly influences a router's maximum throughput

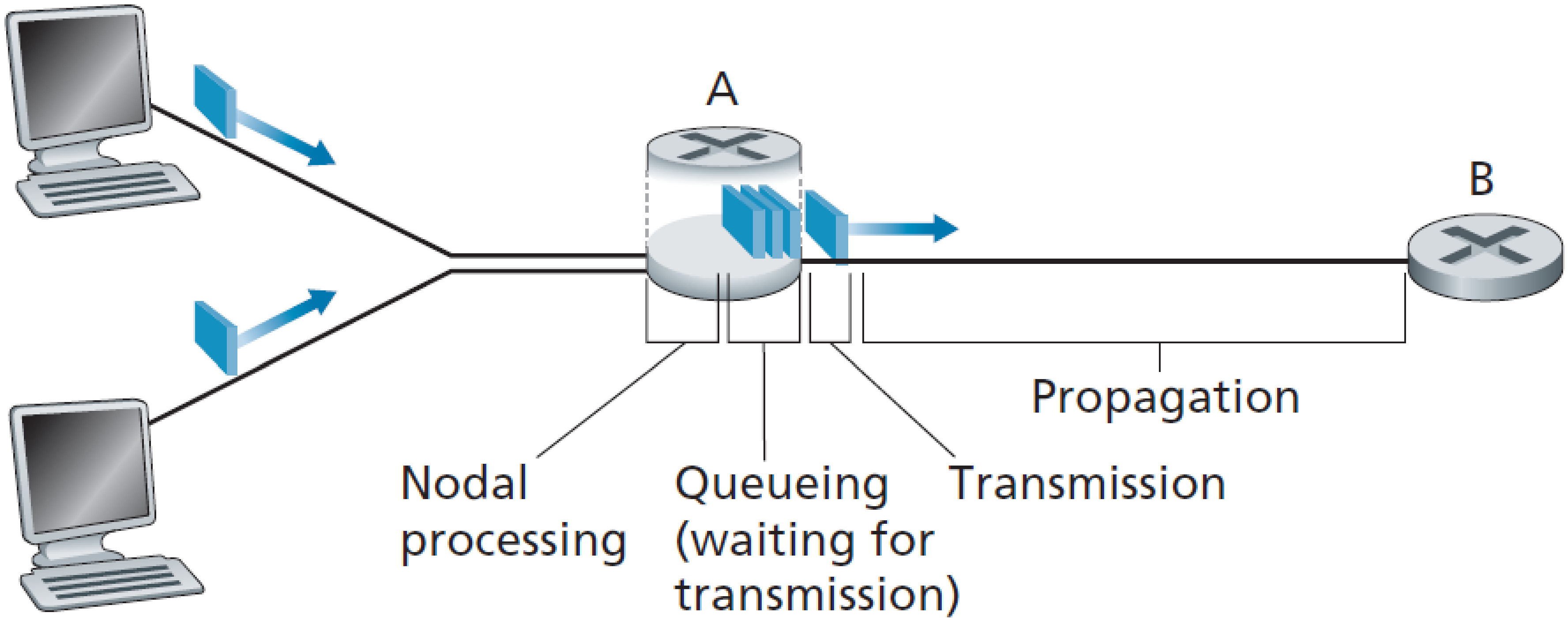


**Figure 1.16** ♦ The nodal delay at router A

# Queuing Delay and Packet Loss (1/2)

- The queuing delay can vary from packet to packet (uses **statistical measures** such as average, variance, probability)
- Queuing delay depends on
  - the rate at which traffic arrives ($\boldsymbol{a}$ packets/sec) (assume each is $\boldsymbol{L}$ bits),
  - the transmission rate of the link ($\boldsymbol{R}$ bps), and
  - the nature of the arriving traffic (periodically or in bursts; or random)
- **Traffic intensity = $\boldsymbol{La/R}$**
  - *If* $\boldsymbol{La/R > 1}$ the queue will tend to increase without bound and the queuing delay will approach infinity!

# Queuing Delay and Packet Loss (2/2)

- $La/R < 1$: the nature of the arriving traffic impacts the queuing delay
  - If packets arrive periodically, then every packet will arrive at an empty queue and there will be **no queuing delay**
  - If packets arrive in bursts but periodically, there can be a **significant** average queuing delay
  - *e.g.* $N$ packets arrive simultaneously every $(L/R)N$ seconds, $n$th packet transmitted has a queuing delay of $(n - 1)L/R$ seconds
- A small percentage increase in the intensity will result in a much larger percentage-wise increase in delay.
- Performance at a node is often measured not only in terms of **delay**, but also in terms of the probability of **packet loss**.

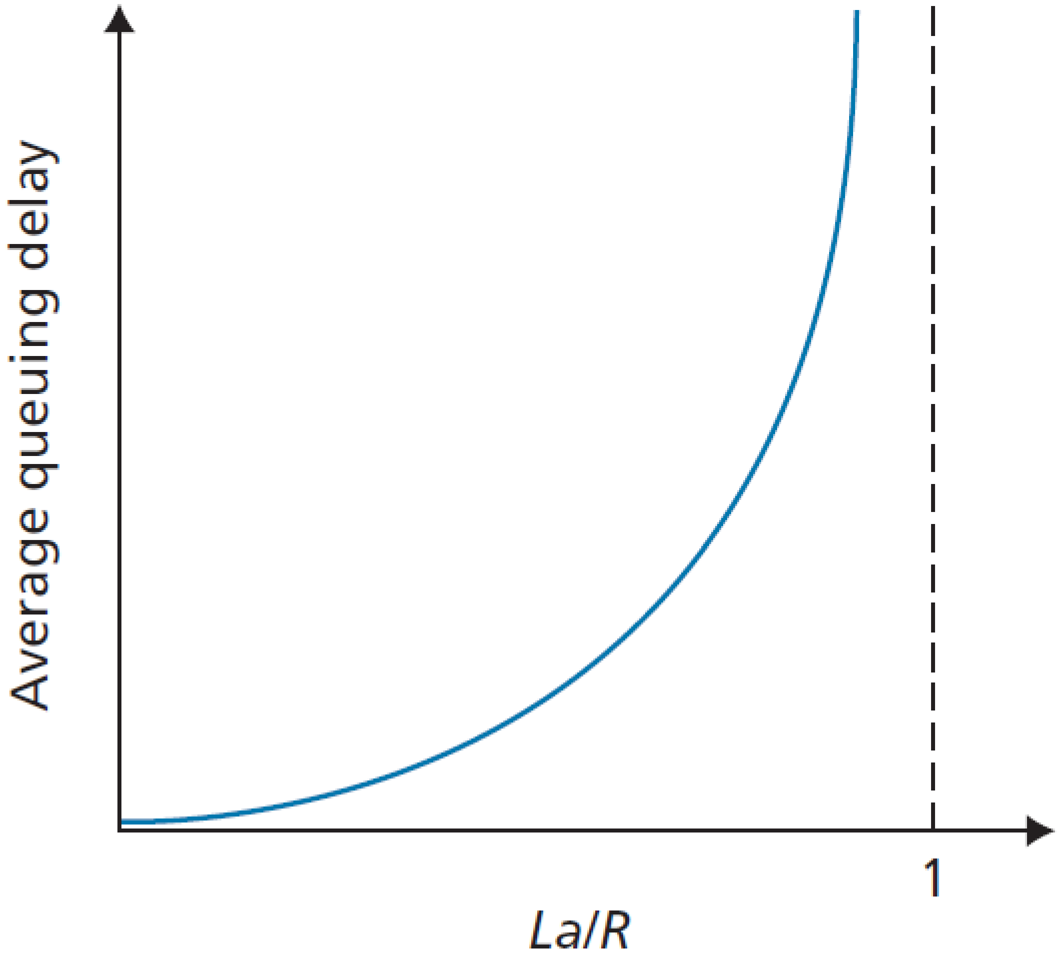


**Figure 1.18** ♦ Dependence of average queuing delay on traffic intensity

# End-to-End Delay

- Assume , $N - 1$ routers, no queuing delay

$$d_{end-to-end} = N\,(d_{proc} + d_{trans} + d_{prop})$$

- **Traceroute** is a simple program, when the user specifies a destination hostname, the program in the source host sends multiple, special packets toward that destination. (graphical interface **PingPlotter**)
  - The source sends $3 \times N$ packets to the destination.
  - As these packets work their way toward the destination, they pass through a series of routers.
  - When a router receives one of these special packets, it sends back to the source a short message that contains the **name and address of the router**.
  - The source can **reconstruct the route** taken by packets flowing from source to destination, and the source can determine the **round-trip delays** to all the intervening routers.

```
1  gw-vlan-2451.cs.umass.edu (128.119.245.1)  1.899 ms 3.266 ms  3.280 ms
2  j-cs-gw-int-10-240.cs.umass.edu (10.119.240.254) 1.296 ms 1.276 ms
   1.245 ms
3  n5-rt-1-1-xe-2-1-0.gw.umass.edu (128.119.3.33) 2.237 ms  2.217 ms
   2.187 ms
4  core1-rt-et-5-2-0.gw.umass.edu (128.119.0.9) 0.351 ms 0.392 ms 0.380 ms
5  border1-rt-et-5-0-0.gw.umass.edu (192.80.83.102) 0.345 ms 0.345 ms
   0.344 ms
6  nox300gw1-umass-re.nox.org (192.5.89.101) 3.260 ms  0.416 ms 3.127 ms
7  nox300gw1-umass-re.nox.org (192.5.89.101) 3.165 ms 7.326 ms  7.311 ms
8  198.71.45.237 (198.71.45.237) 77.826 ms 77.246 ms 77.744 ms
9  renater-lb1-gw.mx1.par.fr.geant.net (62.40.124.70) 79.357 ms 77.729
   79.152 ms
10 193.51.180.109 (193.51.180.109) 78.379 ms  79.936 80.042 ms
11 * 193.51.180.109 (193.51.180.109) 80.640 ms *
12 * 195.221.127.182 (195.221.127.182) 78.408 ms *
13 195.221.127.182 (195.221.127.182) 80.686 ms 80.796 ms 78.434 ms
14 r-upmc1.reseau.jussieu.fr (134.157.254.10) 78.399 ms * 81.353 ms
```

# Additional Delays

- Delay of the transmission as part of its protocol for sharing the medium with other end systems as in a **WiFi**.
- Packetization delay (to fill a packet), which is present in Voice-over-IP (**VoIP**) applications.

# Throughput (1/2)

- Use the **speedtest** application to measure the end-to-end delay and download throughput between a host and servers
- If a file consists of $\boldsymbol{F}$ bits and the transfer takes $\boldsymbol{T}$ seconds for Host B to receive all $\boldsymbol{F}$ bits, then the **average throughput** of the file transfer is $\boldsymbol{F/T}$ bits/sec.
- We may think of bits as ***fluid*** and communication links as ***pipes***.
- In a simple two-link network, the throughput is $\min\{\boldsymbol{R_c}, \boldsymbol{R_s}\}$, that is, it is the transmission rate of the **bottleneck link**.
- For a network with $\boldsymbol{N}$ links between the server and the client, with the transmission rates of the $\boldsymbol{N}$ links being $\boldsymbol{R_1}, \boldsymbol{R_2}, \dots, \boldsymbol{R_N}$. The throughput for a file transfer from server to client is $\min\{\boldsymbol{R_1}, \boldsymbol{R_2}, \dots, \boldsymbol{R_N}\}$.

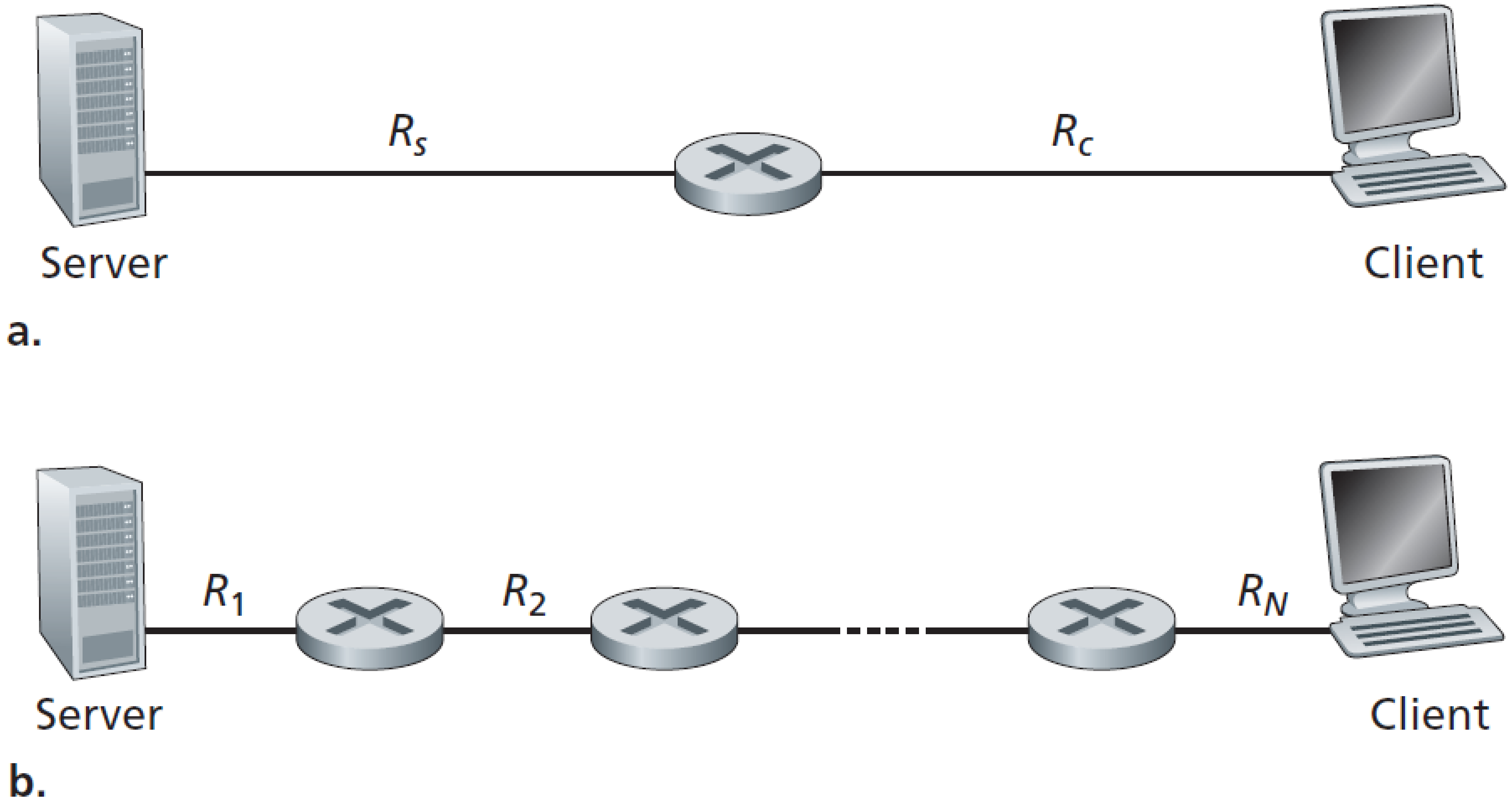


**Figure 1.19** ♦ Throughput for a file transfer from server to client

# Throughput (2/2)

- When there is **no other intervening traffic**, the throughput can simply be approximated as the minimum transmission rate along the path between source and destination.
- Links in the **core** of the communication network have very high transmission rates.
- The constraining factor for throughput in today's Internet is typically the **access network**.

# Example #4 (1/2)

- There are **10** simultaneous downloads: **10 servers** and **10 clients** connected to the core of the computer network.
- Server access links have the same rate $R_s$, all client access links have the same rate $R_c$.
- There is a link in the core that is traversed by **all 10** downloads with $R$ transmission rate.
- The transmission rates of all other links in the core are **much larger** than $R_s$, $R_c$, and $R$.
- If the rate of the common link, $R$, is **very large**, then the throughput for each download will be $\min\{R_s, R_c\}$.

# Example #4 (2/2)

- If the rate of the common link is of the same order as $\boldsymbol{R_s}$ and $\boldsymbol{R_c}$, **bottleneck is now the shared link in the core**.
- e.g. $R_s = 2\ Mbps, R_c = 1\ Mbps, R\ =\ 5\ Mbps$
- each download has **$500\ kbps$** of throughput.

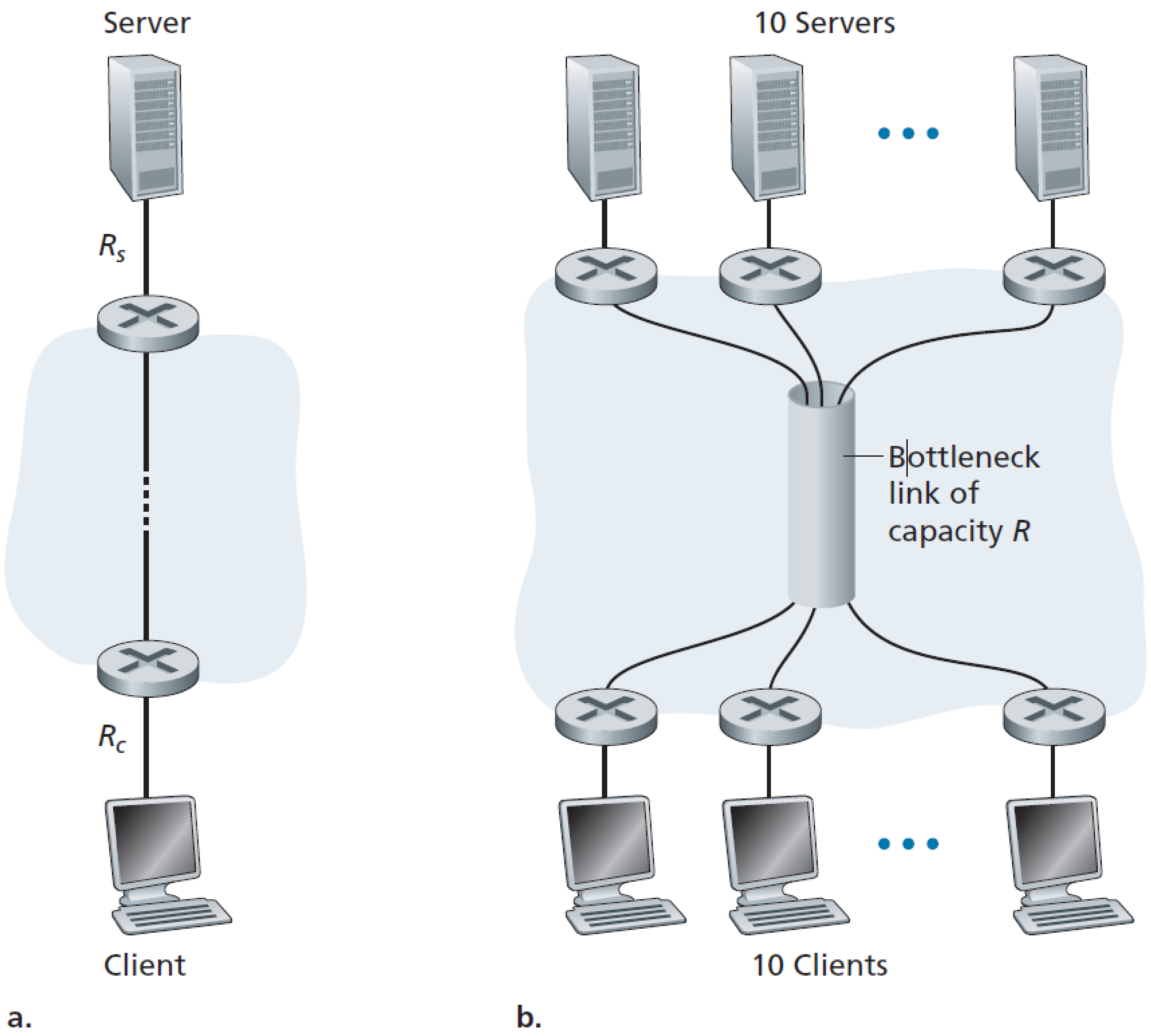


**Figure 1.20** ♦ End-to-end throughput: (a) Client downloads a file from server; (b) 10 clients downloading with 10 servers

# Protocol Layers

# Layered Architecture

- There are **many pieces** to the Internet: numerous applications and protocols, various types of end systems, packet switches, and various types of link-level media.
- Given this **enormous complexity**, is there any hope of organizing a network architecture, or at least our discussion of network architecture?
- A **layered architecture** allows us to discuss a well-defined, specific part of a large and complex system.
  - Each layer provides its service by performing certain actions and using the services of the layer directly below it.
  - Modularity makes it much easier to change the **implementation** of the service provided by a layer without affecting other components.

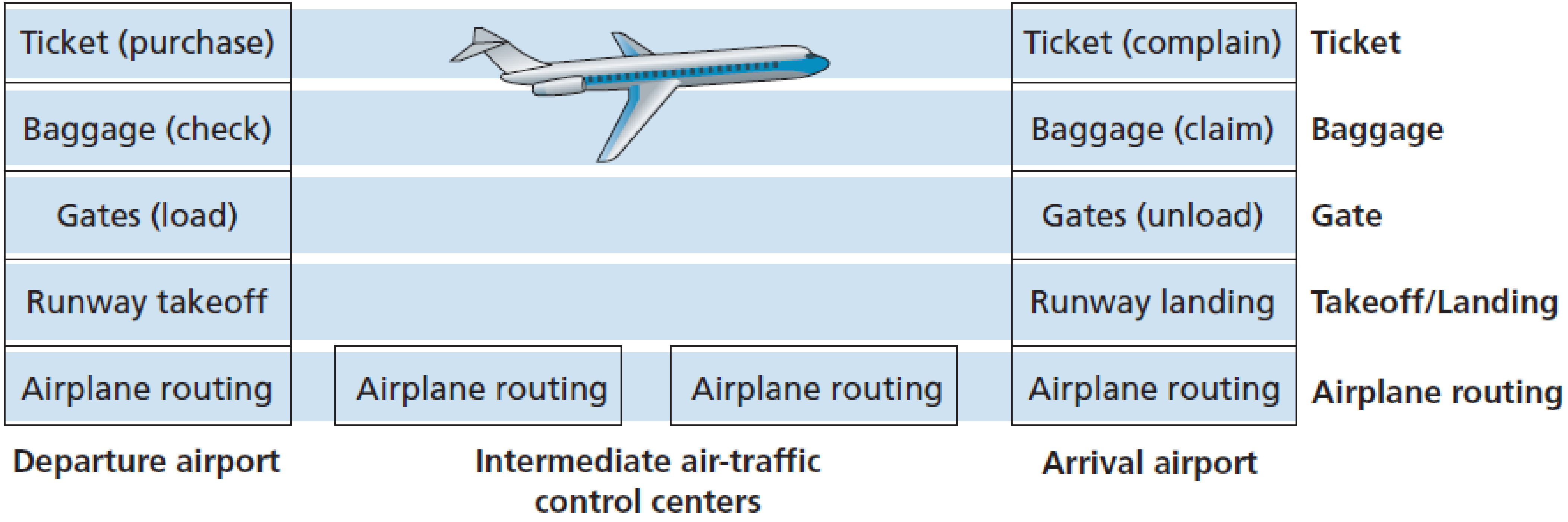


**Figure 1.22** ♦ Horizontal layering of airline functionality

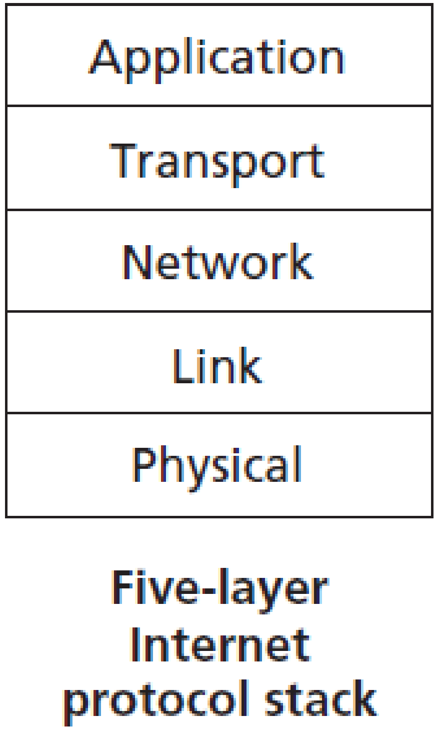


**Figure 1.23** ♦ The Internet protocol stack

# Protocol Layering (1/4)

- Network designers organize protocols in **layers**.
- A protocol layer can be implemented in software, in hardware, or in a combination of the two.
  - **Application-layer** protocols are almost always implemented in software and so are **transport-layer** protocols
  - The **physical layer** and **data link layers** are responsible for handling communication over a specific link, they are typically implemented in a network interface card
  - The **network layer** is often a mixed implementation of hardware and software.
- Potential **drawbacks** of layering is that one layer may duplicate lower-layer functionality and the functionality at one layer may need information that is present only in another layer.

# Protocol Layering (2/4)

- The **application layer** is where network applications and their application-layer protocols reside.
  - With the application in one end system using the protocol to exchange packets of information (called **messages**) with the application in another end system.
  - e.g. HTTP, SMTP, FTP, DNS
- The **transport layer** transports application-layer messages between application endpoints. (a transport-layer packet is refered as a **segment**)
  - The **UDP protocol** provides a connectionless service to its applications.
  - **TCP** provides a connection-oriented service to its applications: guaranteed delivery; flow control; congestion-control.

# Protocol Layering (3/4)

- The **network layer** (IP layer) is responsible for moving network-layer packets known as **datagrams** from one host to another.
  - IP protocol defines the fields in the datagram as well as how the end systems and routers act on these fields.
  - This layer contains **routing protocols** that determine the routes that datagrams take between sources and destinations.
- The **link layer** 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.
  - A datagram may be handled by **different** link-layer protocols at different links along its route.
  - The link-layer packets are refereed as **frames**.
  - e.g. Ethernet, WiFi

# Protocol Layering (4/4)

- The job of the **physical layer** is to move the **individual bits** within the frame from one node to the next.
  - Depends on the actual transmission medium of the link.
  - e.g. Ethernet has many physical-layer protocols: one for twisted-pair copper wire, another for coaxial cable, another for fiber, and so on

# Encapsulation

- The transport layer takes the **message** and appends additional information. The **transport-layer segment encapsulates** the application-layer message.
- The network layer adds network-layer header information 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**.

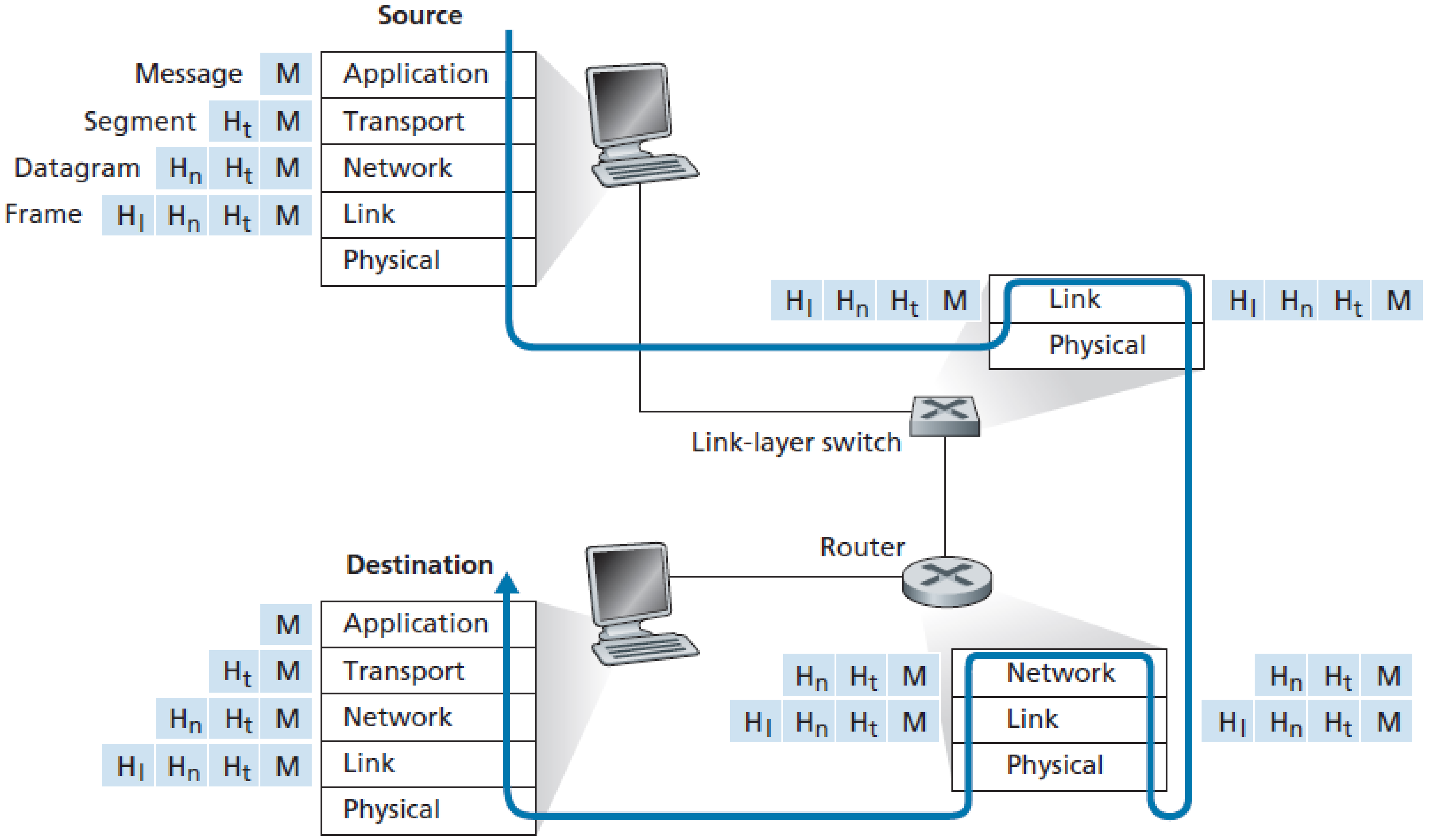


**Figure 1.24** ♦ Hosts, routers, and link-layer switches; each contains a different set of layers, reflecting their differences in functionality

# Summary

- What Is the Internet?
- The Network Edge
- The Network Core
- Delay, Loss, and Throughput in Packet-Switched Networks
- Protocol Layers and Their Service Models