Introduction to Computer Networks

Acknowledgement

- These slides are taken (as it is or with some modifications) from various resources including -
  - Comuter Networking: A top-down approach (Book and Slides) by Jim Kurose and Keith Ross
  - Data Communication and Networking with TCP/IP Protocol Suite (Book and Slides) by Behrouz A. Forouzan

Motivation

- When someone talks about computer networks, the first things that comes to our mind is the **internet.**
- But what internet actually is? and how it works?
- The actual definition of the internet can vary depending upon whom you are asking.
- In this course, we will talk about the internal workings of internet.

https://www.youtube.com/watch?v=Dxcc6ycZ73M&ab_channel=Code.org

The Internet: a "nuts and bolts" view

Billions of connected computing *devices*:

- *hosts = end systems*
- running *network apps* at Internet's "edge"

Packet switches: forward packets (chunks of data)

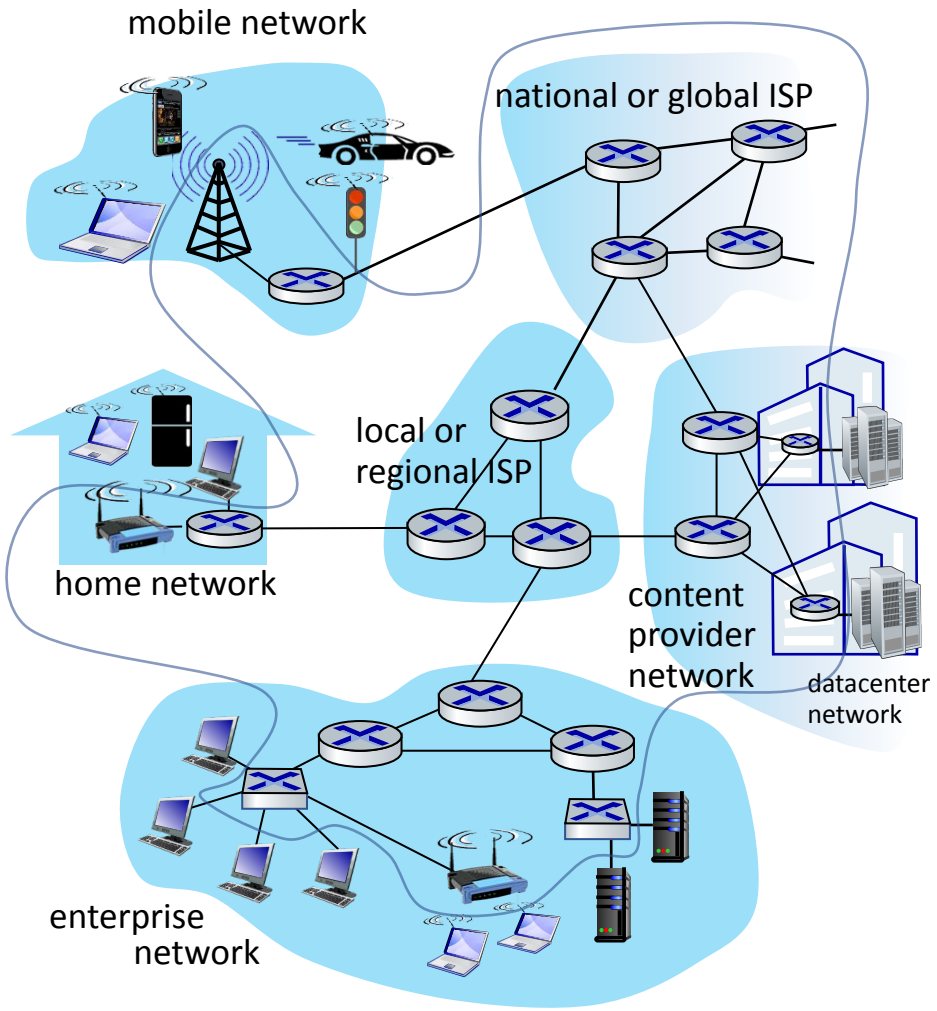
- *routers*, *switches*

Communication links

- fiber, copper, radio, satellite
- transmission rate: *bandwidth*

Networks

- collection of devices, routers, links: managed by an organization



“Fun” Internet-connected devices

Amazon Echo

Internet refrigerator

IP picture frame

Pacemaker & Monitor

Tweet-a-watt: monitor energy use

bikes

cars

scooters

Security Camera

Slingbox: remote control cable TV

Web-enabled toaster + weather forecaster

AR devices

Fitbit

Internet phones

Gaming devices

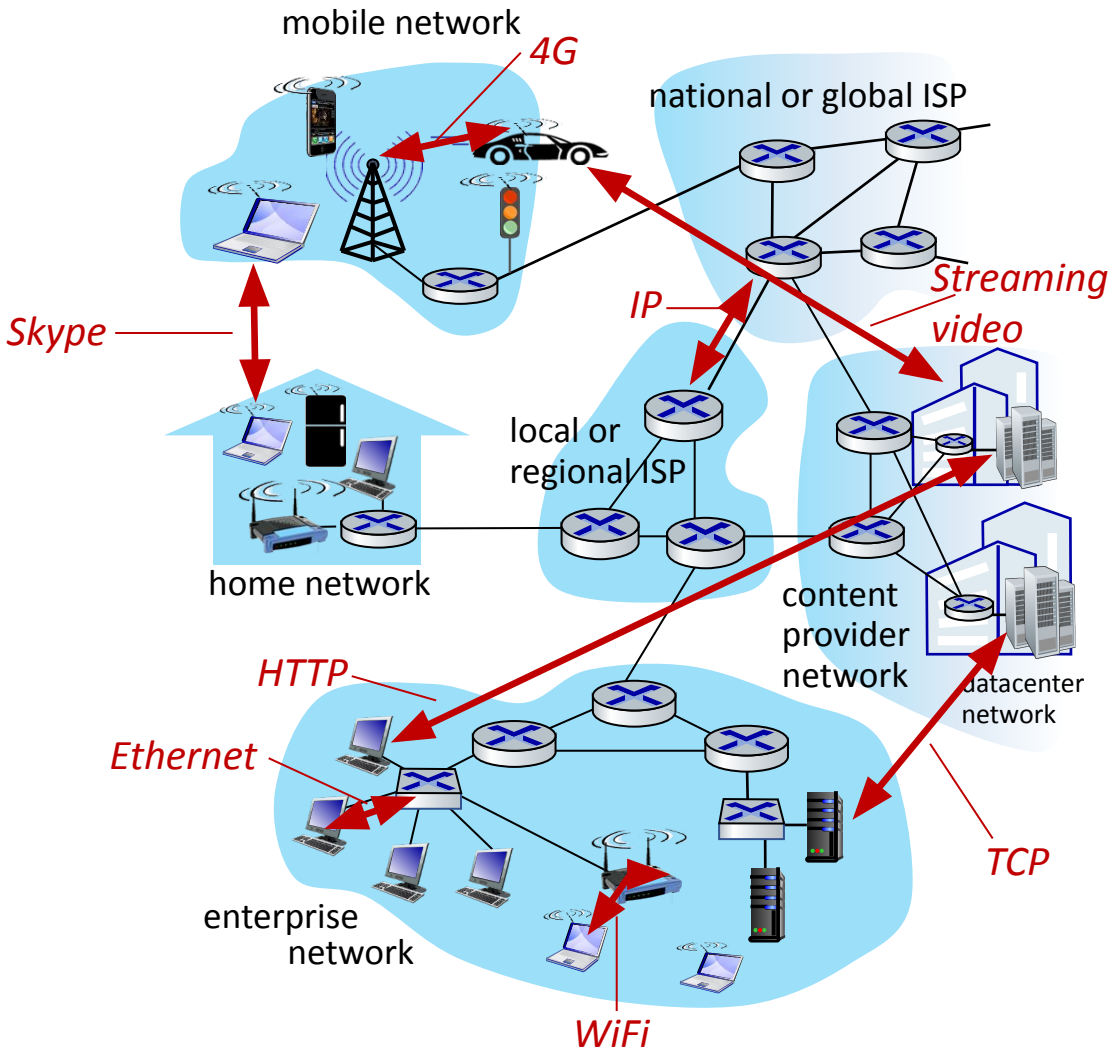
sensorized, bed mattress

diapers

Others?

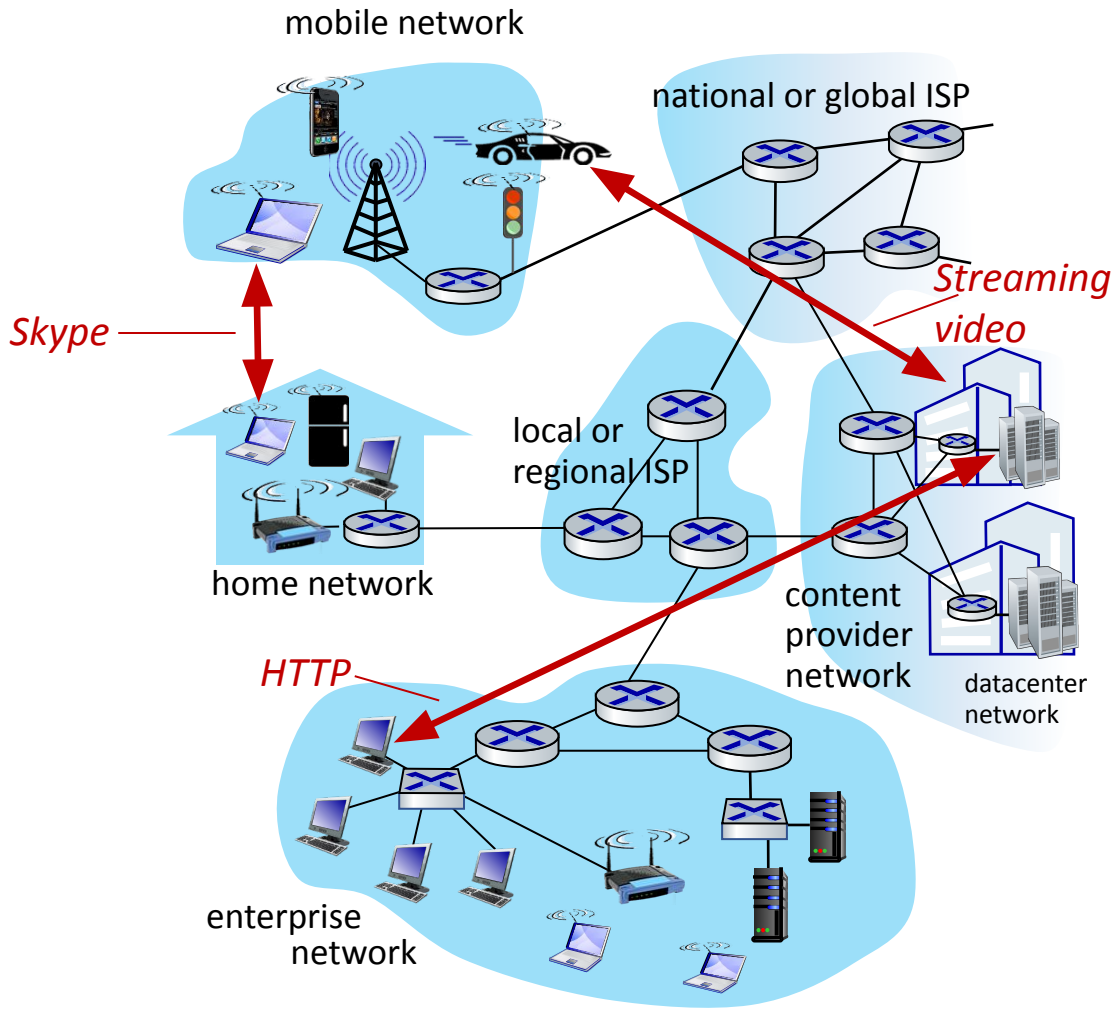
The Internet: a "nuts and bolts" view

- *Internet:* "network of networks"
  - Interconnected ISPs
- *protocols* are *everywhere*
  - control sending, receiving of messages
  - e.g., HTTP (Web), streaming video, Skype, TCP, IP, WiFi, 4/5G, Ethernet



The Internet: a "services" view

- *Infrastructure* that provides services to applications:
  - Web, streaming video, multimedia teleconferencing, email, games, e-commerce, social media, inter-connected appliances, …



What is a Computer Network?

- A computer network is the interconnection of a set of devices capable of communication.
- These devices can be either a **host** (also called **end system** sometimes) such as a large computer, desktop, laptop, workstation, cellular phone, TV, etc. Or a **connecting device** such as a router which connects the network to other network, a switch which connects devices together, a modem (modulator-demodulator) that changes the form of data, and so on.
- These devices in a network are connected using wired or wireless transmission media such as ethernet, WiFi, etc.

Connected computing *devices*:

- *hosts = end systems*
- running *networking apps* at the "edge" of the network

Packet switches: forward packets (chunks of data)

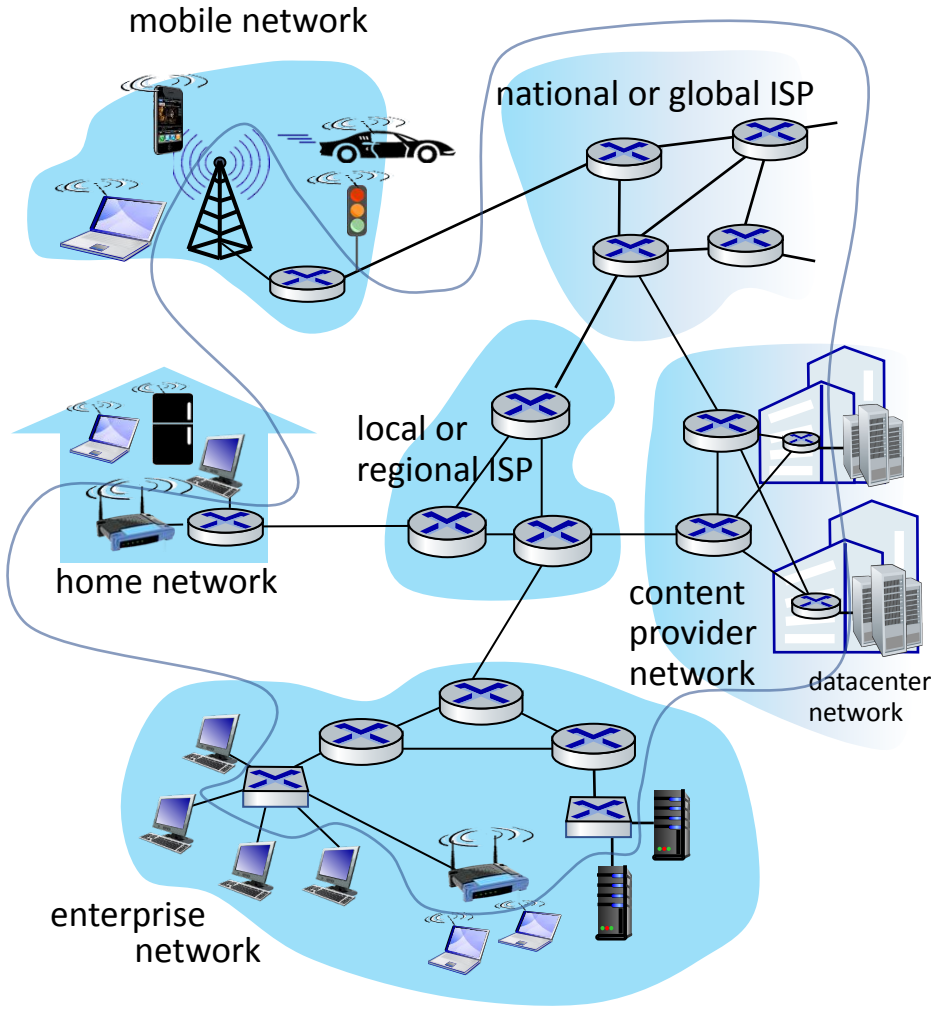
- *routers, switches*

Communication links

- fiber, copper, radio, satellite
- transmission rate: *bandwidth*

Networks

- collection of devices, routers, links: managed by an organization

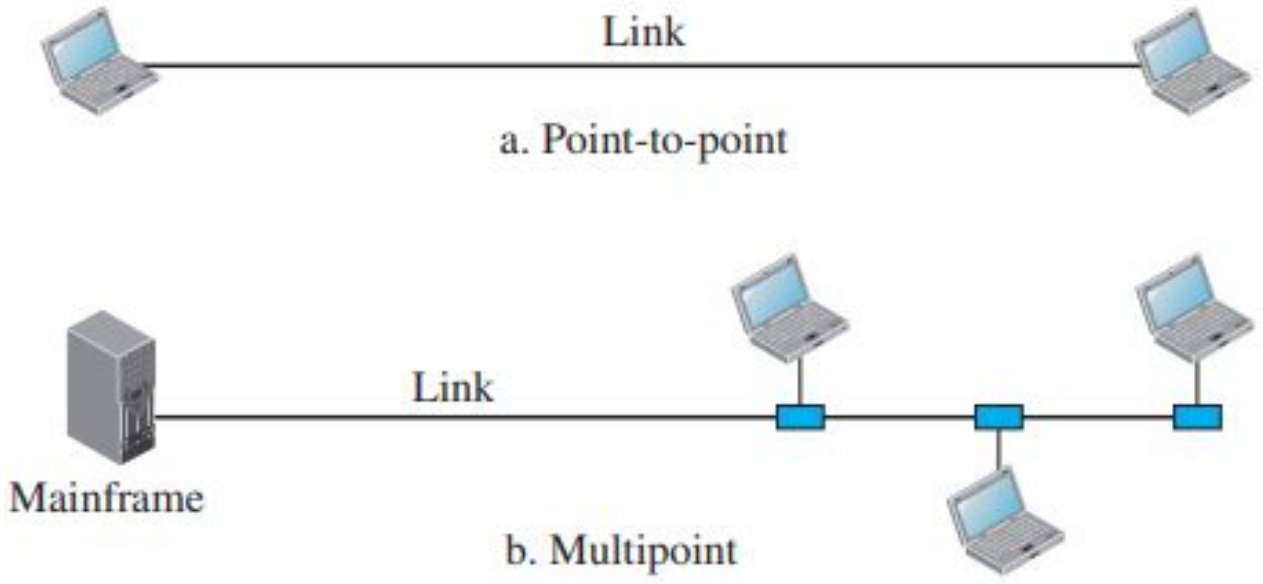


Network Criteria

- A network must be able to meet a certain number of criteria. Some of which are -
- **Performance -** A network should provide efficient communication. Performance of a network is measured using two things -
  - **Transit time** - Time spend by a message in the network before reaching to the destination.
  - **Response time** - After receiving a query, time taken by a computer to generate the response.
- **Reliability -** The Reliability of a network is measured by the frequency of failures it is undergoing and the time it takes to recover from the failures. Overall, the Robustness of the Network at times of catastrophic events is measured to check how reliable the Network is.
- **Security -** Network security issues include protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from breaches and data losses.

Physical Structures of Networks

- A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another.
- For communication to occur, two devices must be connected in some way to the same link at the same time.



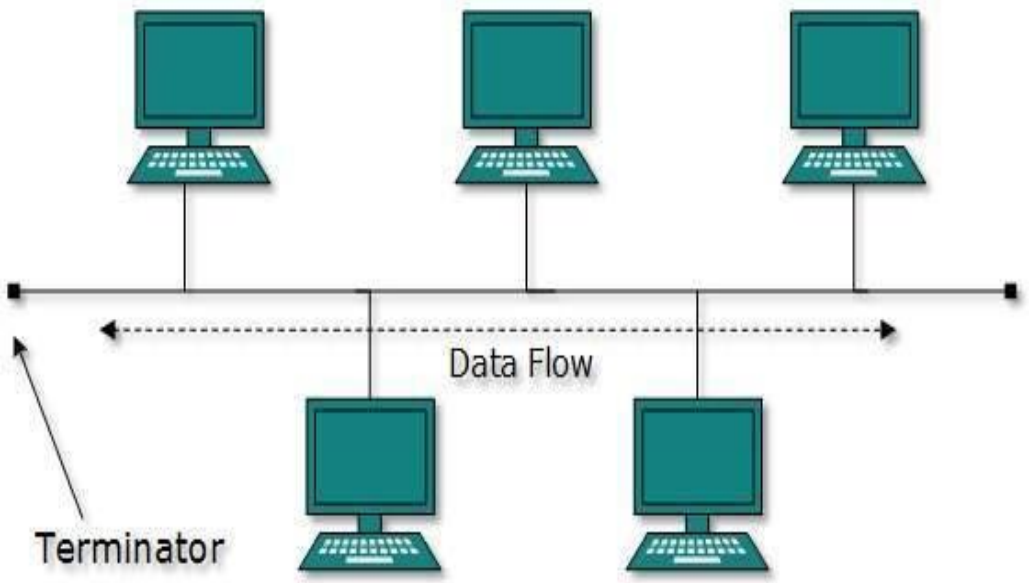
Network Topologies

- Network topology refers to the arrangement of different elements like nodes, links, and devices in a computer network.
- It defines how these components are connected and interact with each other.
- Understanding various types of network topologies helps in designing efficient and robust networks.
- Common types include *bus*, *star*, *ring*, *mesh*, and *tree* topologies, each with its own advantages and disadvantages.

Types of Network Topology

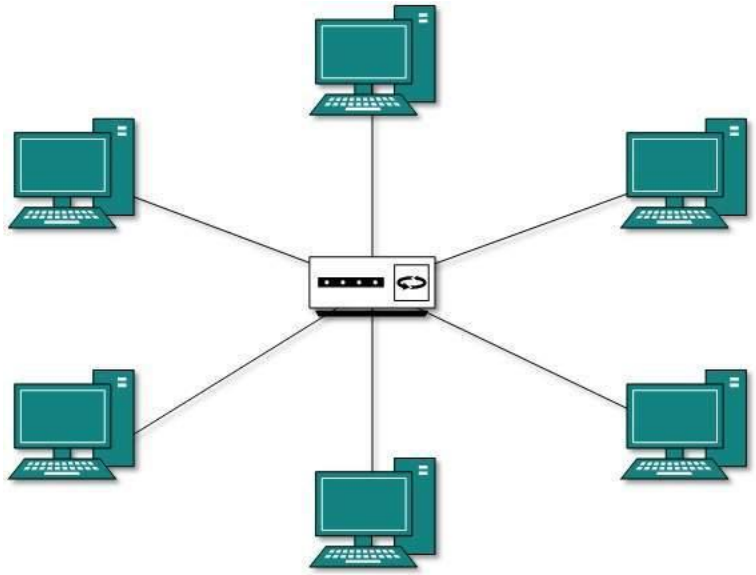
Bus Topology

- In case of Bus topology, all devices share single communication line or cable.
- It is one of the simple forms of networking where a failure of a device does not affect the other devices. But failure of the shared communication line can make all other devices stop functioning.
- Both ends of the shared channel have line terminator. The data is sent in both directions and as soon as it reaches the extreme ends, the terminator removes the data from the line.



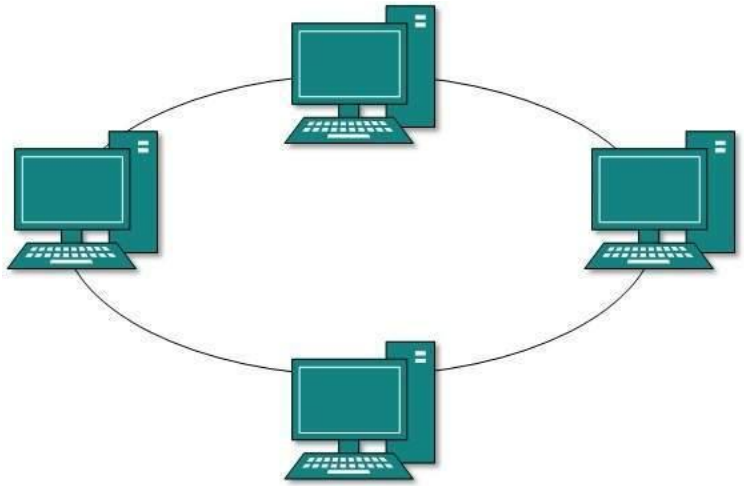
Types of Network Topology

- **Star Topology -**
  - In Star Topology, all the devices are connected to a single hub through a cable.
  - This hub is the central node and all other nodes are connected to the central node.
  - If hub fails, connectivity of all hosts to all other hosts fails. Every communication between hosts, takes place through only the hub.
  - Star topology is not expensive as to connect one more host, only one cable is required and configuration is simple.



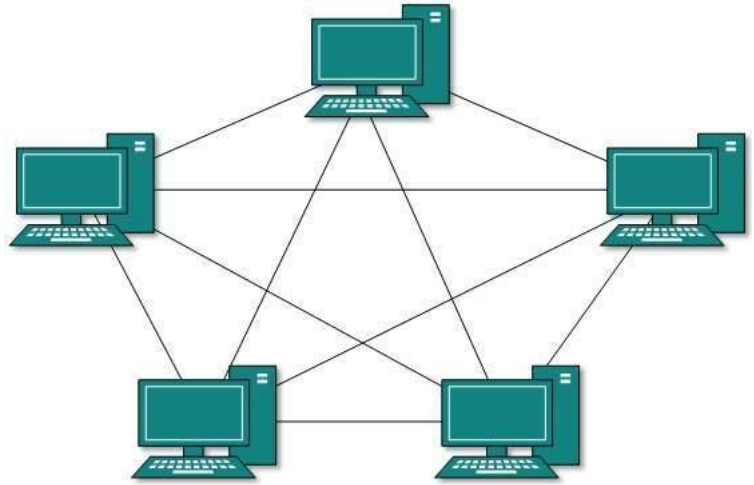
Types of Network Topology

- **Ring Topology -**
  - In ring topology, each host machine connects to exactly two other machines, creating a circular network structure.
  - When one host tries to communicate or send message to a host which is not adjacent to it, the data travels through all intermediate hosts.
  - To connect one more host in the existing structure, the administrator may need only one more extra cable.
  - Failure of any host results in failure of the whole ring as the data travels in only one direction in ring network structure. Thus, every connection in the ring is a point of failure.

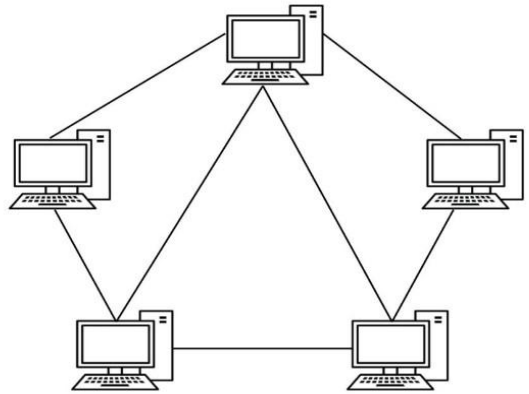


Types of Network Topology

- **Mesh Topology -**
  - In this type of topology, a host is connected to one or multiple hosts.
  - Hosts in Mesh topology also work as relay for other hosts which do not have direct point-to-point links.
  - **Full Mesh** - All hosts have a point-to-point connection to every other host in the network. It provides the most reliable network structure among all network topologies.
  - **Partially Mesh** - Not all hosts have point-to-point connection to every other host. Hosts connect to each other in some arbitrarily fashion. This topology exists where we need to provide reliability to some hosts out of all.



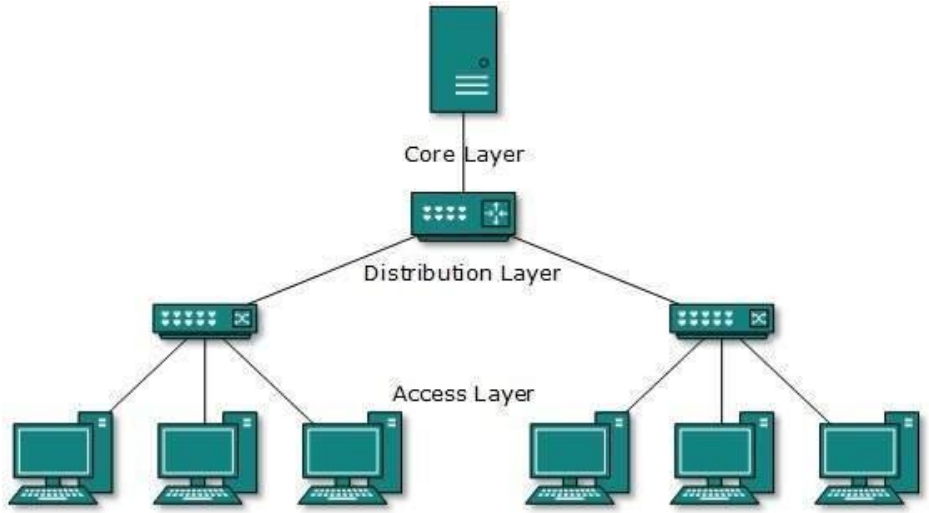
Partial Mesh Topology



Types of Network Topology

Tree Topology or Hierarchical Topology

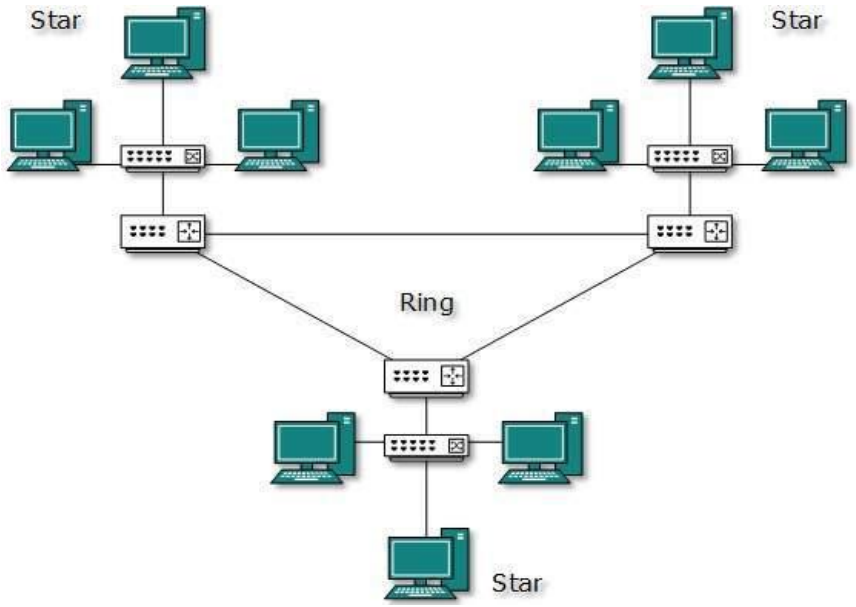
- This topology divides the network in to multiple levels/layers of network.
- The lowermost is access-layer where computers are attached.
- The middle layer is known as distribution layer, which works as mediator between upper layer and lower layer.
- The highest layer is known as core layer, and is central point of the network, i.e. root of the tree from which all nodes fork.
- Similar to the Bus topology, if the root goes down, then the entire network suffers even though it is not the single point of failure.
- Every connection serves as point of failure, failing of which divides the network into unreachable segment.



Types of Network Topology

Hybrid Topology

- A network structure whose design contains more than one topology is said to be hybrid topology. Hybrid topology inherits merits and demerits of all the incorporating topologies.
- Internet is the best example of largest Hybrid topology.



Comparison of Network Topologies

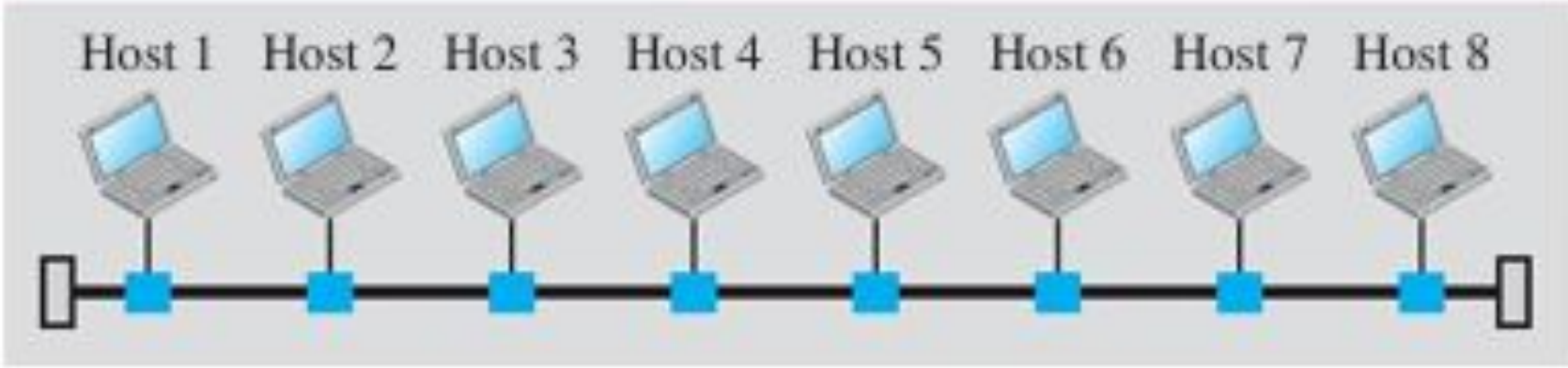
Topology	Advantages	Disadvantages
Bus	Simple, cost-effective, easy to extend	Difficult to troubleshoot, limited length and nodes, single point of failure (backbone)
Star	Easy to install/manage, failure isolation	Central point of failure (hub/switch), more cabling
Ring	High speed, no collisions	Failure of one device affects all, difficult to troubleshoot
Mesh	Highly reliable, redundant paths	Expensive, complex configuration
Tree	Scalable, easy management	More cabling, backbone failure affects entire network
Hybrid	Flexible, robust, scalable	omplex design/maintenance, costly

Types of Computer Networks

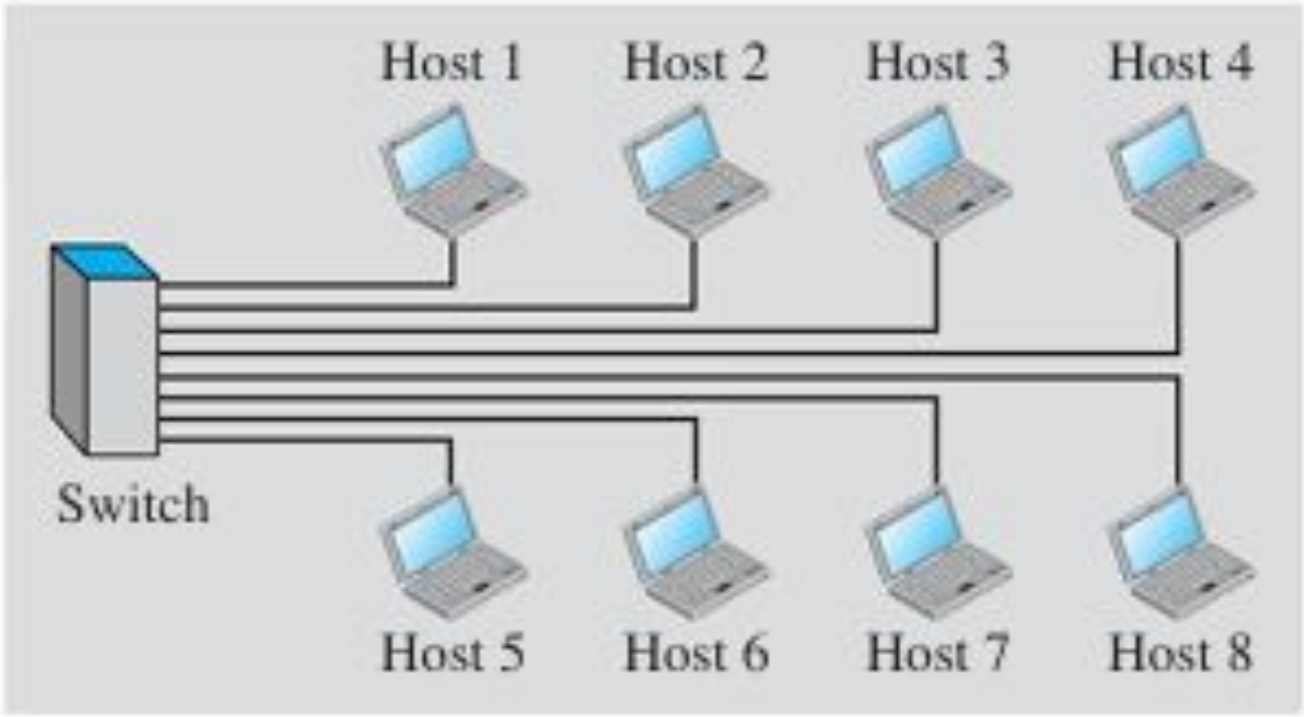
- Depending upon size and connection range, networks can be divided into several categories -
  - Personal Area Network (PAN)
  - Local Area Network (LAN)
  - Campus Area Network (CAN)
  - Metropolitan Area Network (MAN)
  - Wide Area Network (WAN)
    - Point-to-Point WAN
    - Switched WAN

Types of Computer Networks

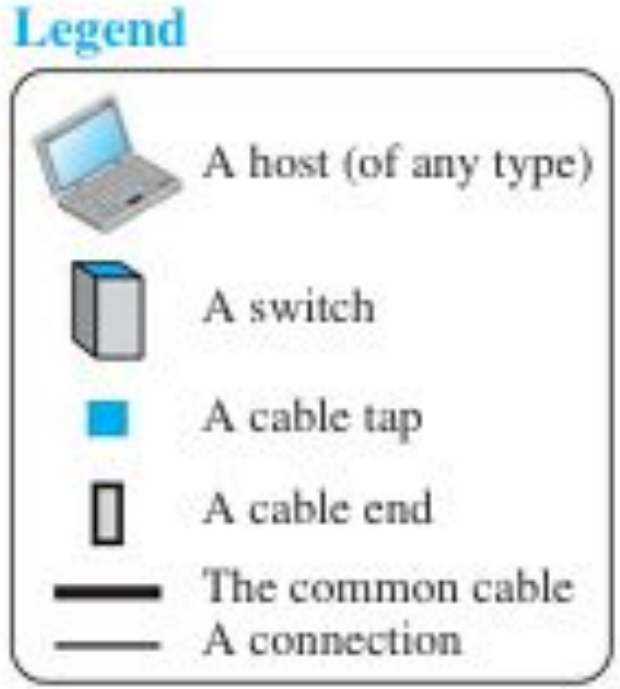
- **Personal Area Network (PAN)**
  - A PAN is the smallest type of network, covering a very limited area, typically within the range of an individual person.
  - Range - Up to a few meters
- **Local Area Network (LAN)**
  - A LAN covers a small geographic area such as a single building, office, or home. It is used to connect computers and devices within close proximity.
  - Range - Up to a few kilometers.



a. LAN with a common cable (past)



b. LAN with a switch (today)

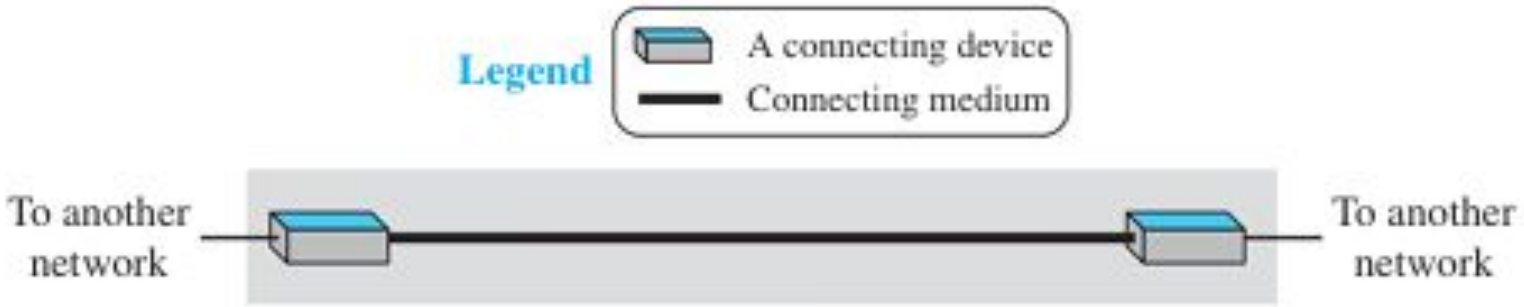


Types of Computer Networks

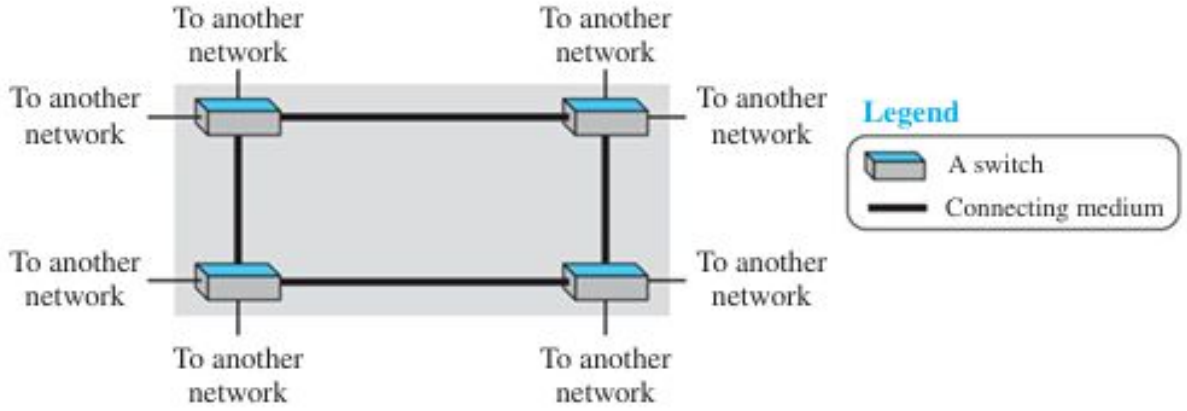
- **Metropolitan Area Network (MAN)**
  - A MAN covers a large geographic area than a LAN, typically spanning a city or a large campus. It connects multiple LANs within a specific region.
  - Range - Up to 50 kilometers.
- **Wide Area Network (WAN)**
  - A WAN covers a broad area, often a country or continent. It connects multiple LANs and MANs.
  - Range - Thousands of kilometers.

Wide Area Network

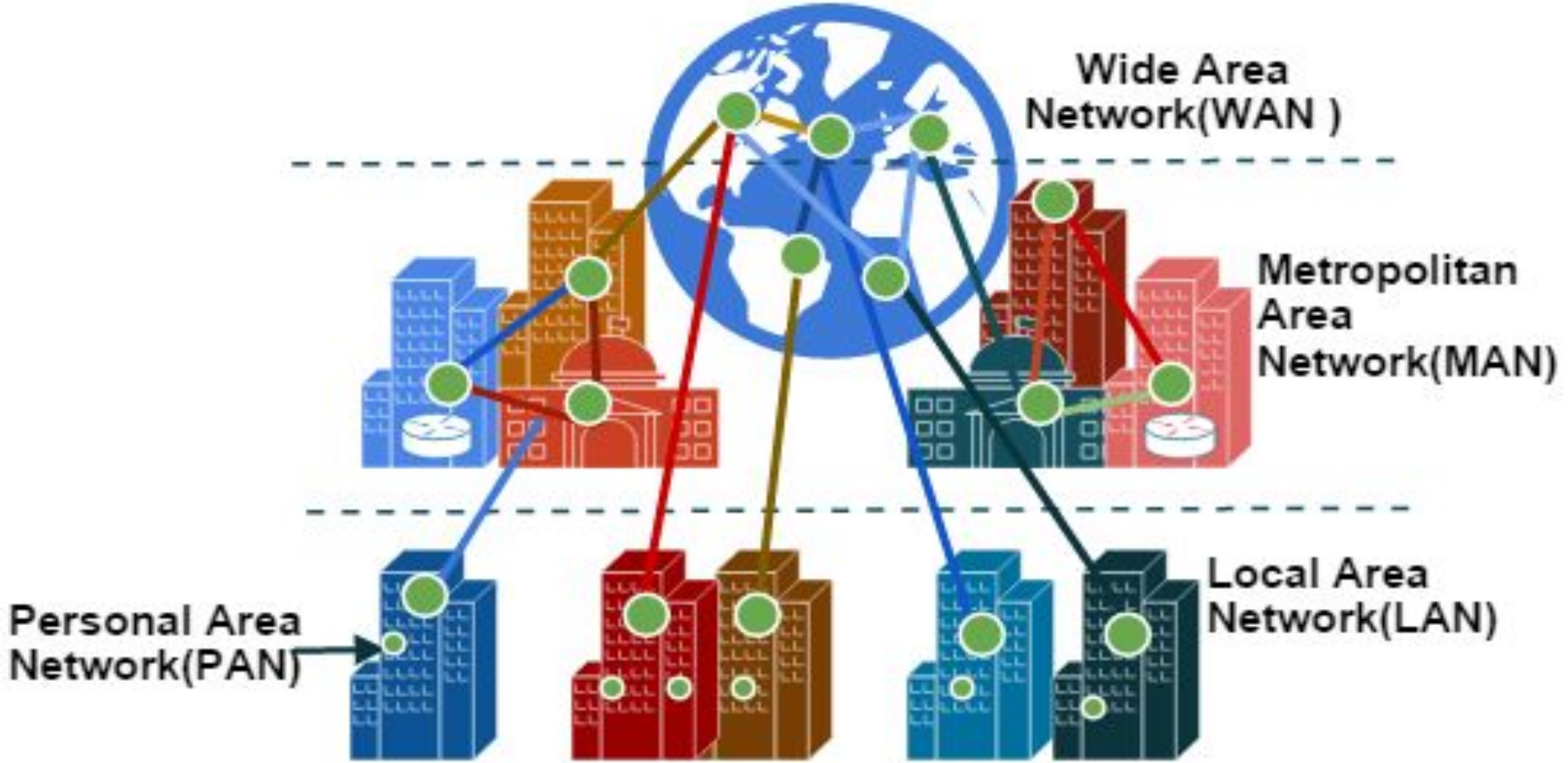
- Point-to-Point WAN:



- Switched WAN:

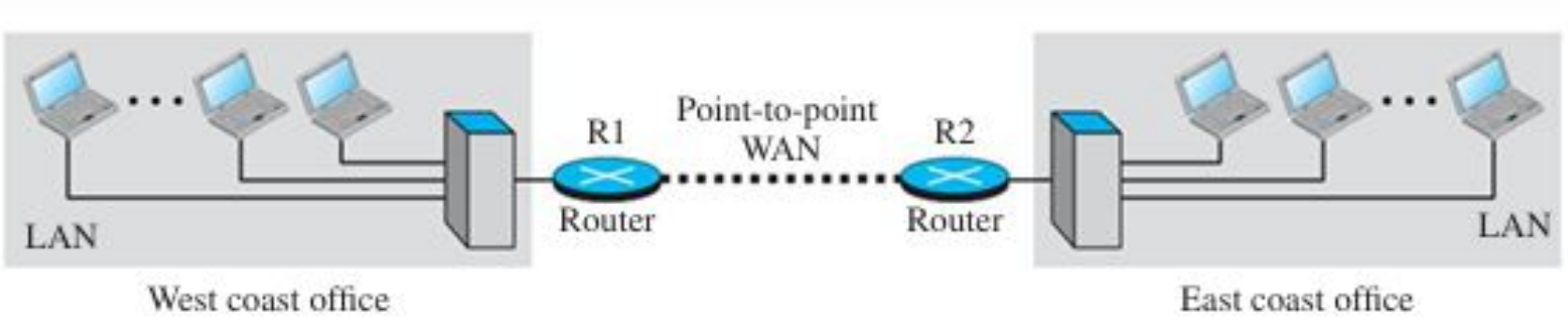


Types of Computer Networks

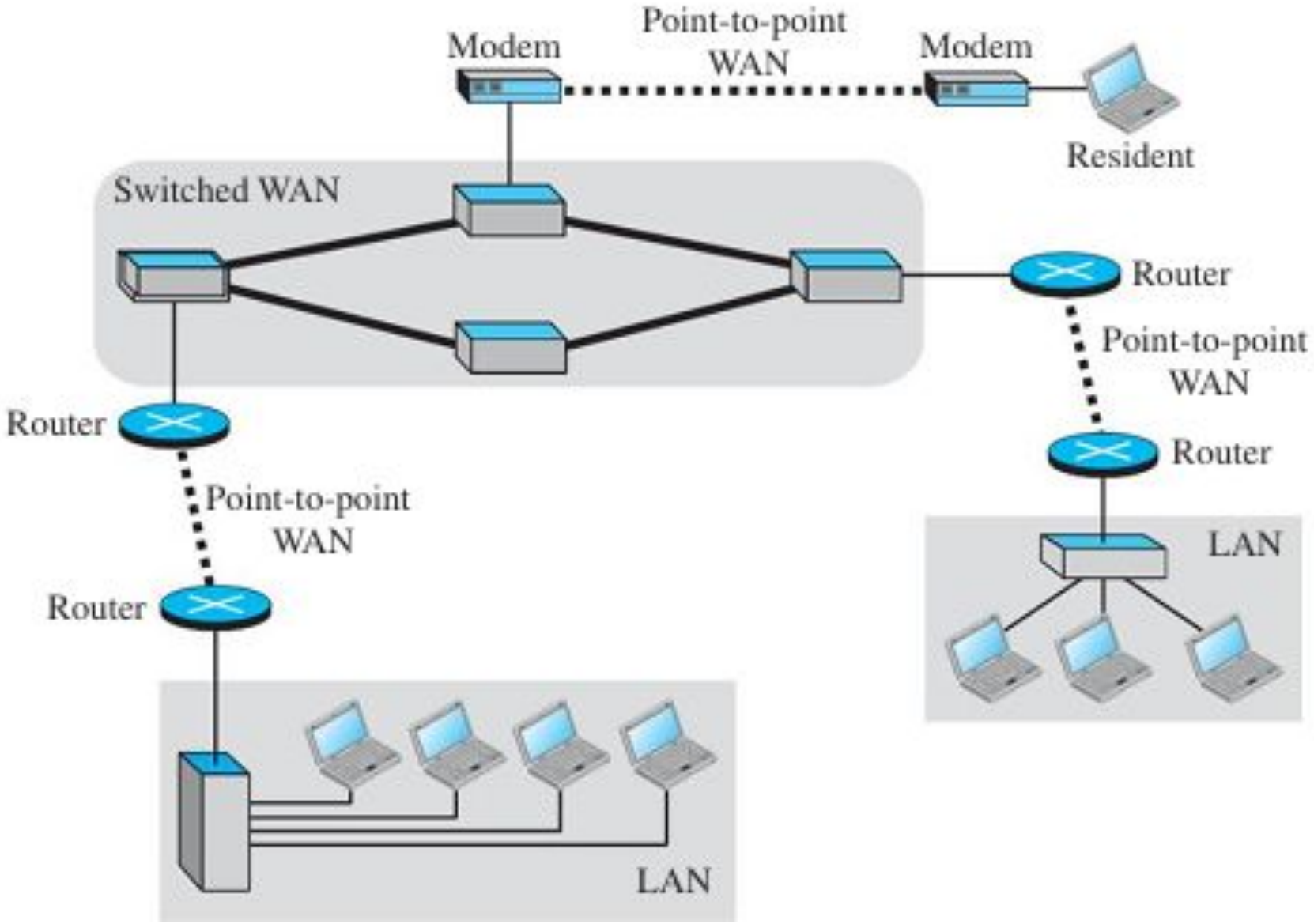


Inter-Network

- Today, it is very rare to see a LAN or a WAN in isolation; they are connected to one another. When two or more networks are connected, they make an *internetwork,* or an **internet.**
- As an example, assume that an organization has two offices, one on the east coast and the other on the west coast. Each office has a LAN that allows all employees in the office to communicate with each other. To make the communication between employees at different offices possible, the management leases a point-to-point dedicated WAN from a service provider, such as a telephone company, and connects the two LANs. Now the company has an internetwork, or a private internet (with lowercase i).



Inter-Network



The Internet: a "nuts and bolts" view

Billions of connected computing *devices*:

- *hosts = end systems*
- running *network apps* at Internet's "edge"

Packet switches: forward packets (chunks of data)

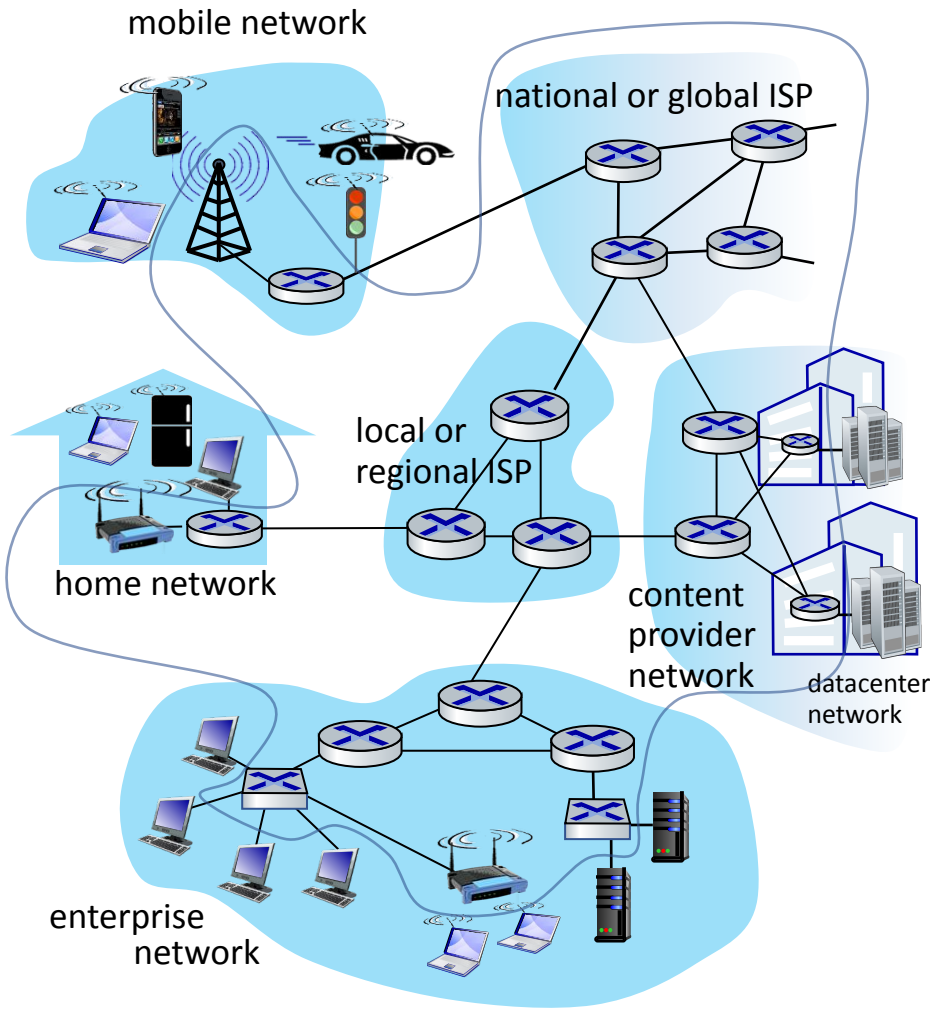
- *routers*, *switches*

Communication links

- fiber, copper, radio, satellite
- transmission rate: *bandwidth*

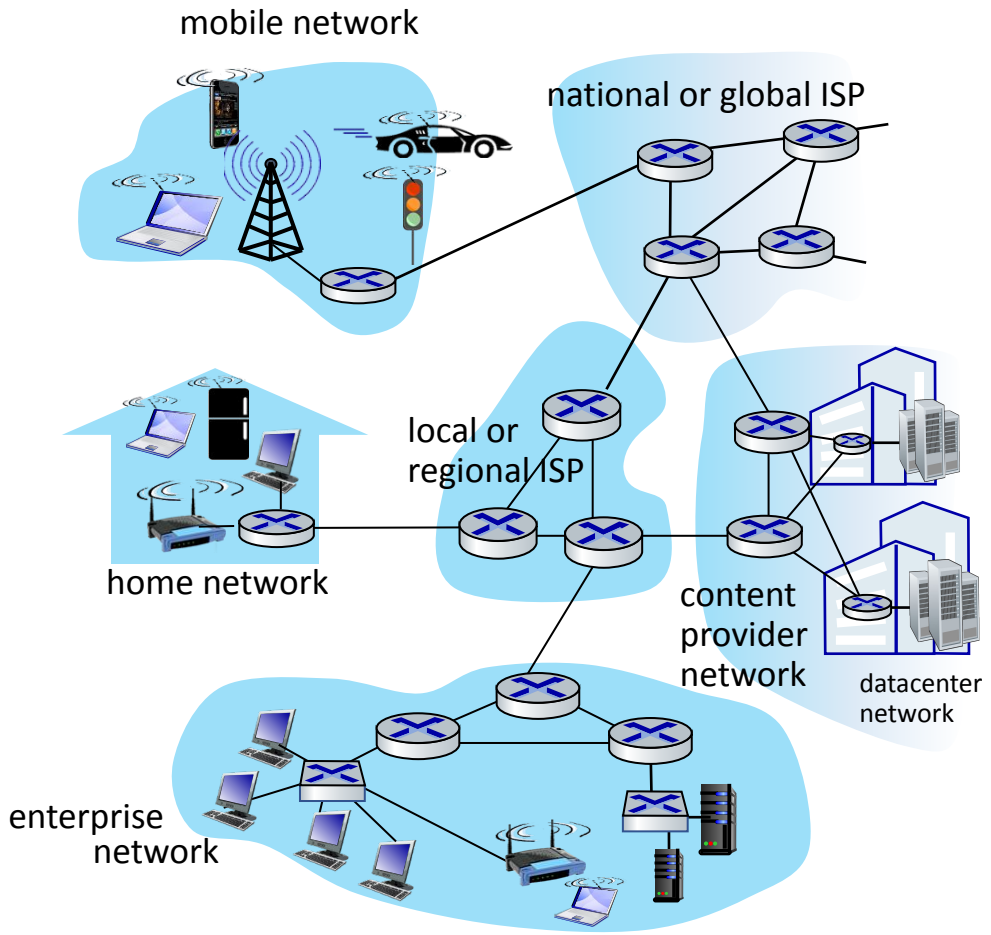
Networks

- collection of devices, routers, links: managed by an organization



Internet structure: a "network of networks"

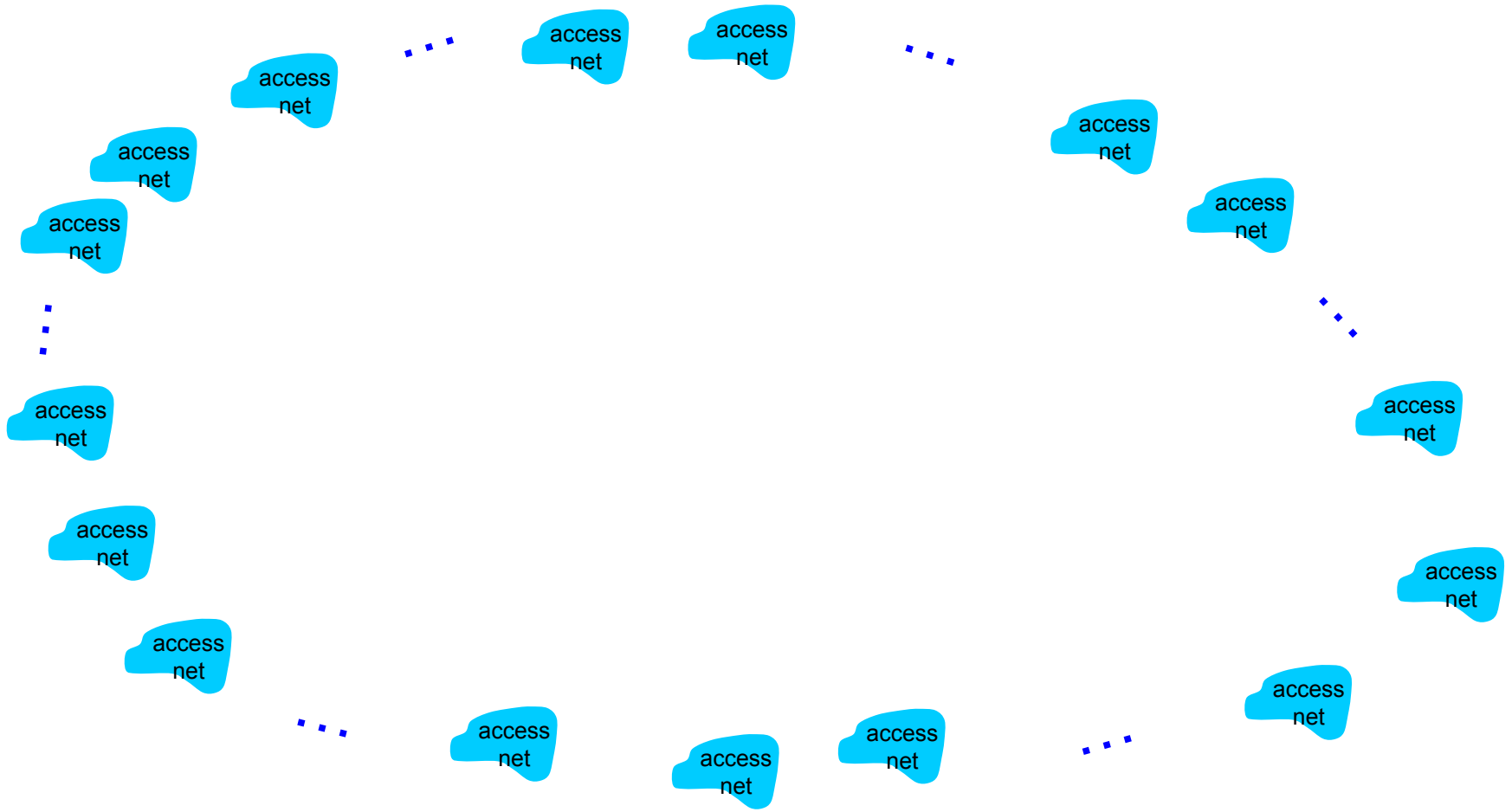
- hosts connect to Internet via access Internet Service Providers (ISPs)
- access ISPs in turn must be interconnected
  - so that *any* two hosts *(anywhere!)* can send packets to each other
- resulting network of networks is very complex
  - evolution driven by economics, national policies



Let's take a stepwise approach to describe current Internet structure

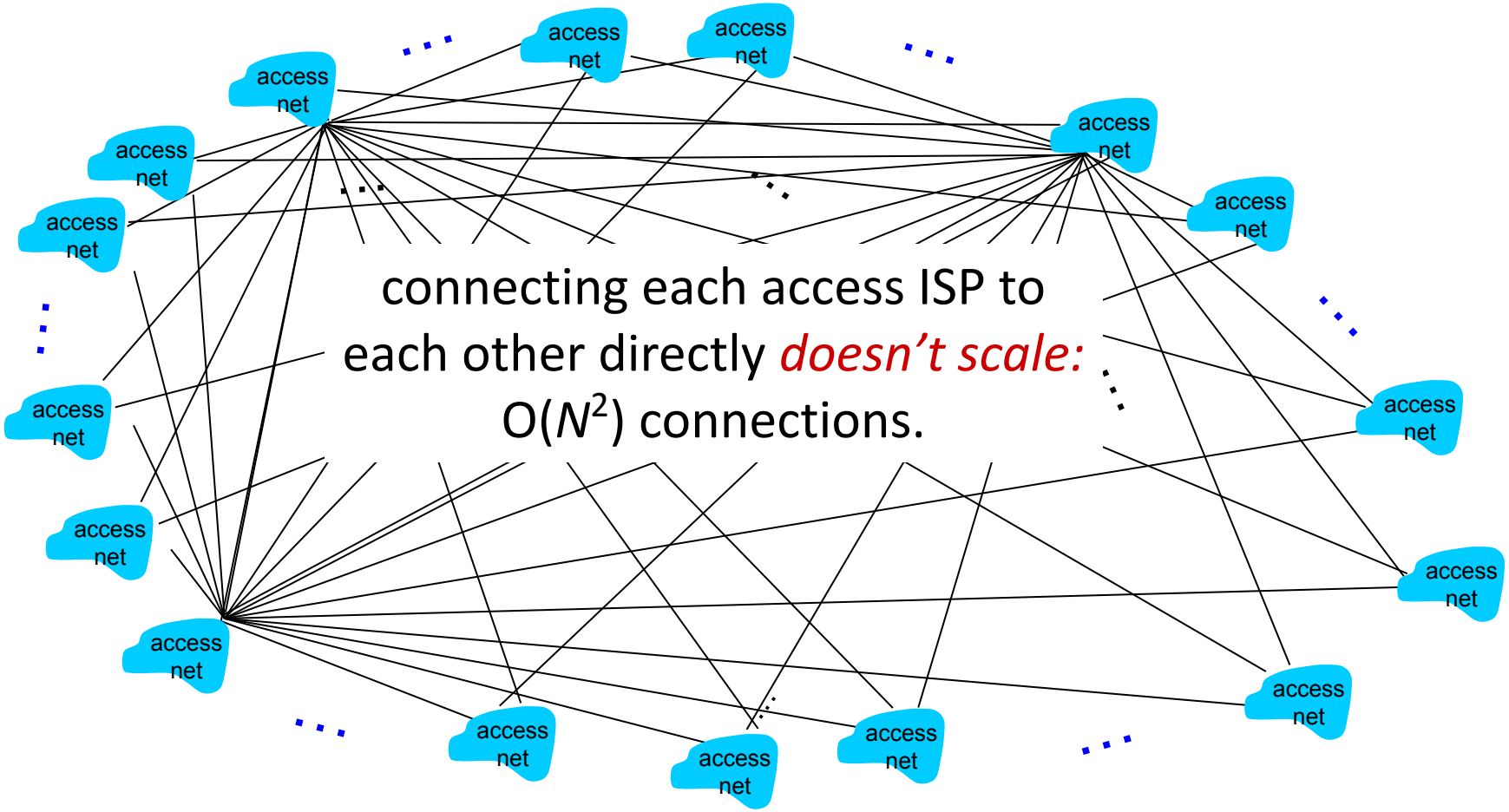
Internet structure: a “network of networks”

Question: given *millions* of access ISPs, how to connect them together?



Internet structure: a "network of networks"

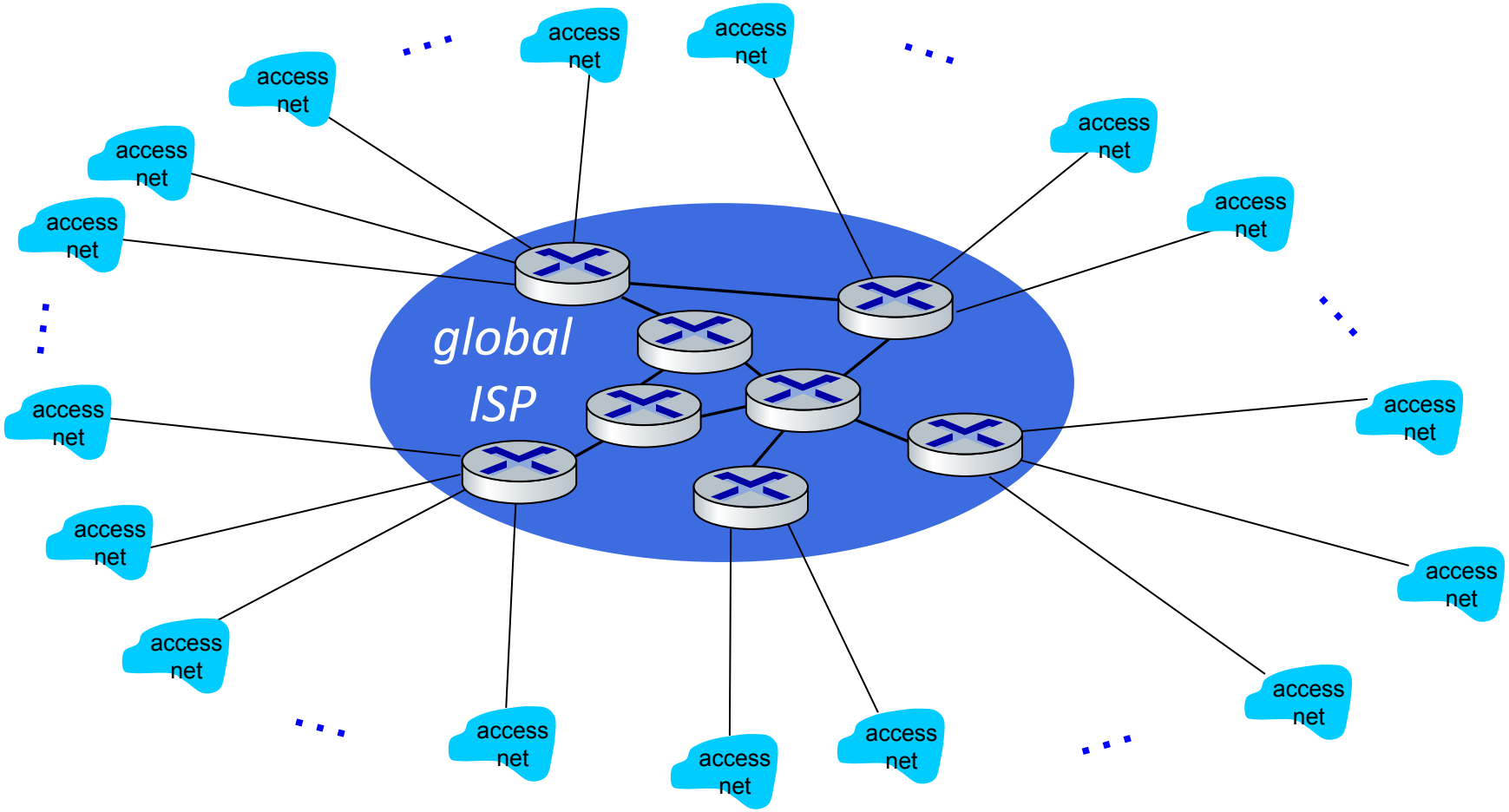
Question: given *millions* of access ISPs, how to connect them together?



Internet structure: a “network of networks”

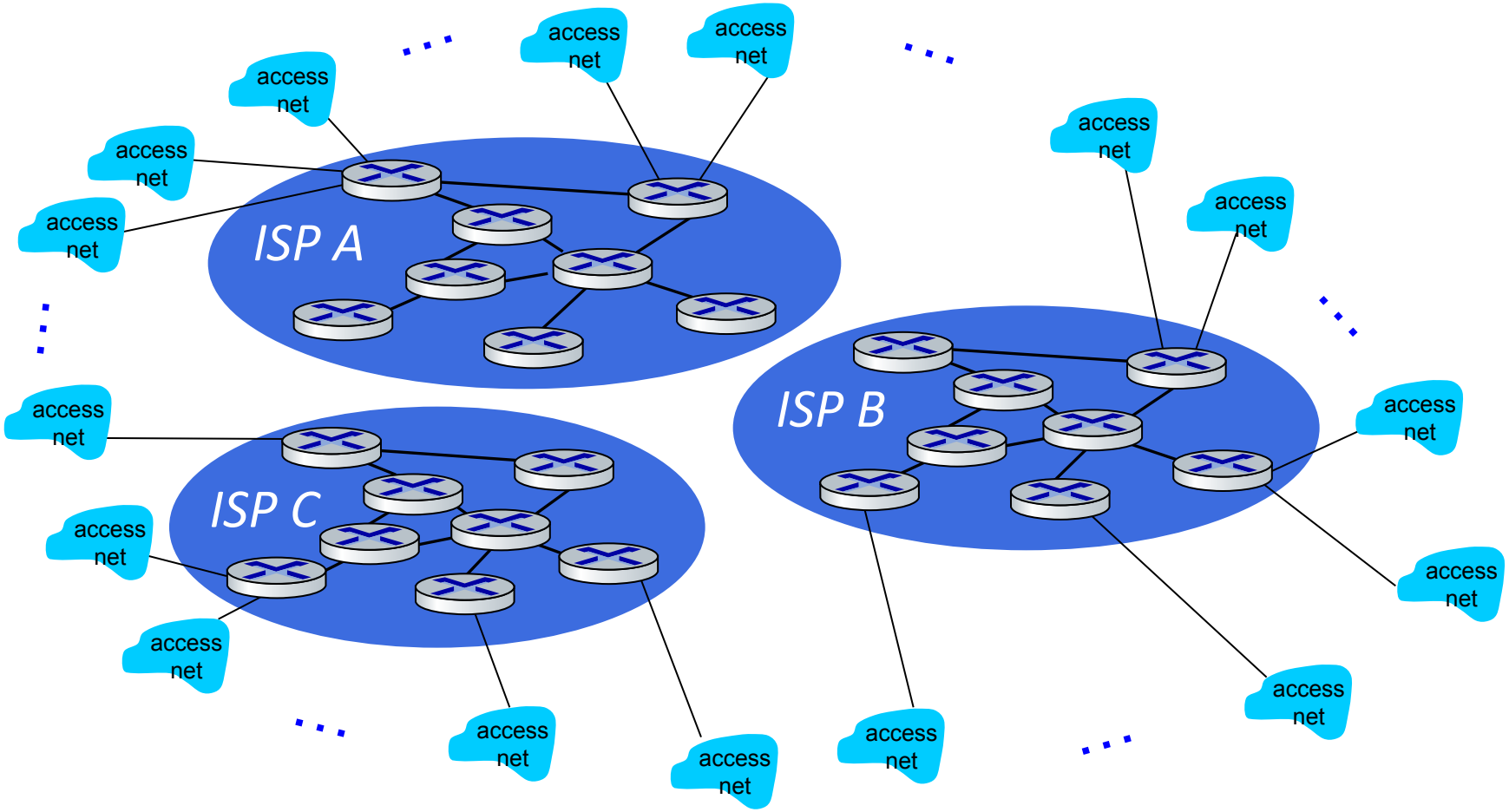
Option: connect each access ISP to one global transit ISP?

Customer and provider ISPs have economic agreement.



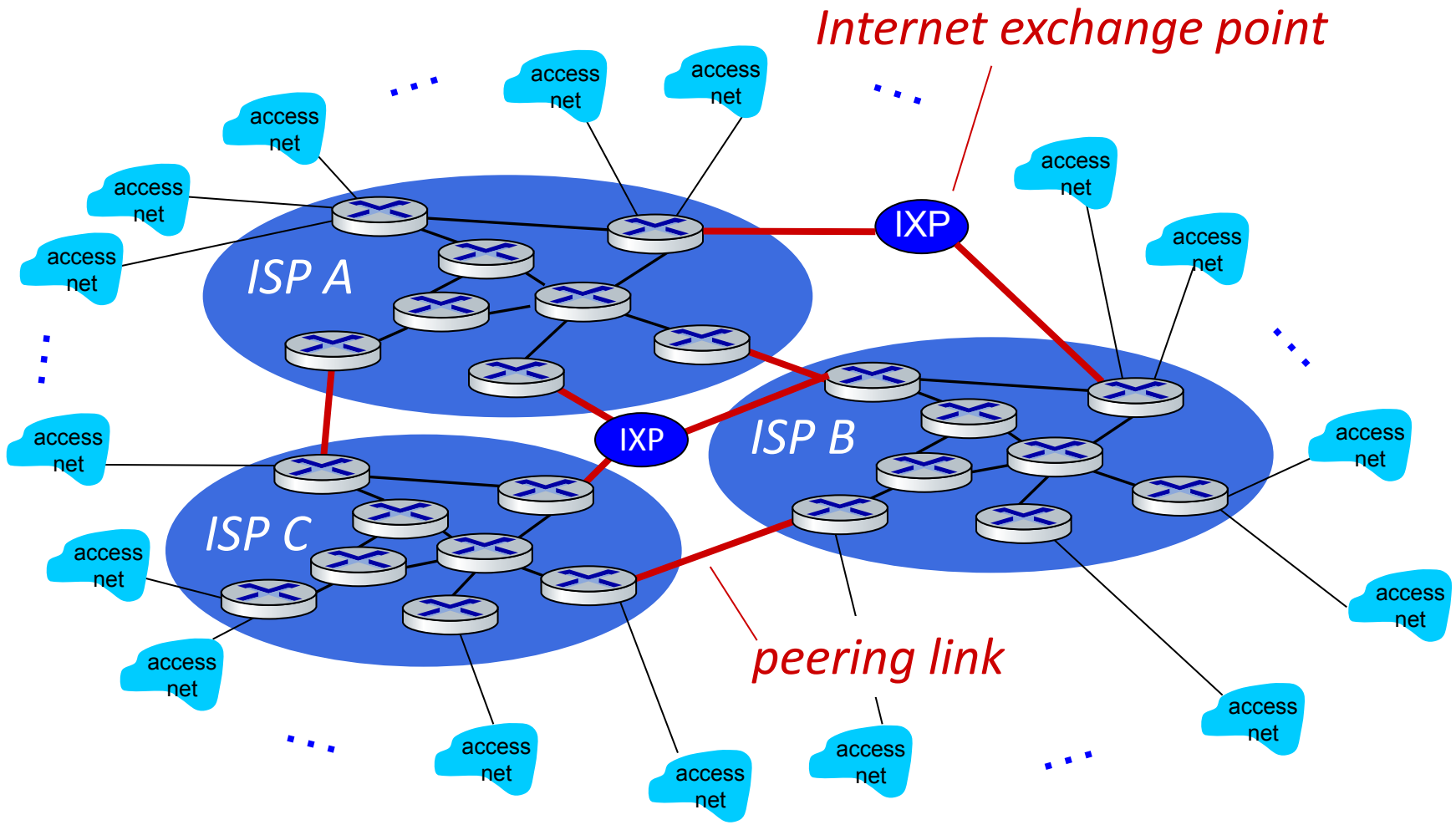
Internet structure: a “network of networks”

But if one global ISP is viable business, there will be competitors ….



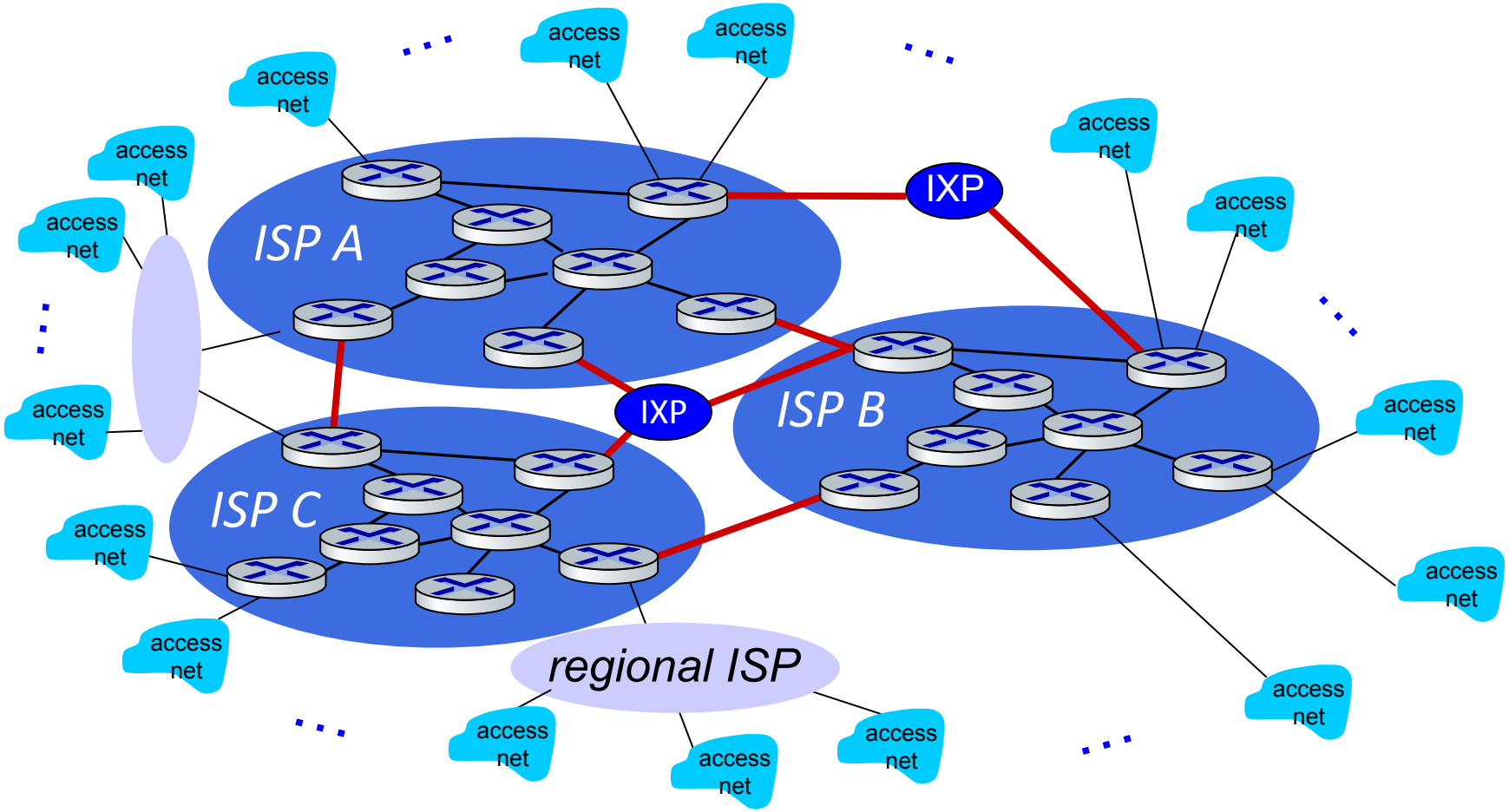
Internet structure: a "network of networks"

But if one global ISP is viable business, there will be competitors …. who will want to be connected



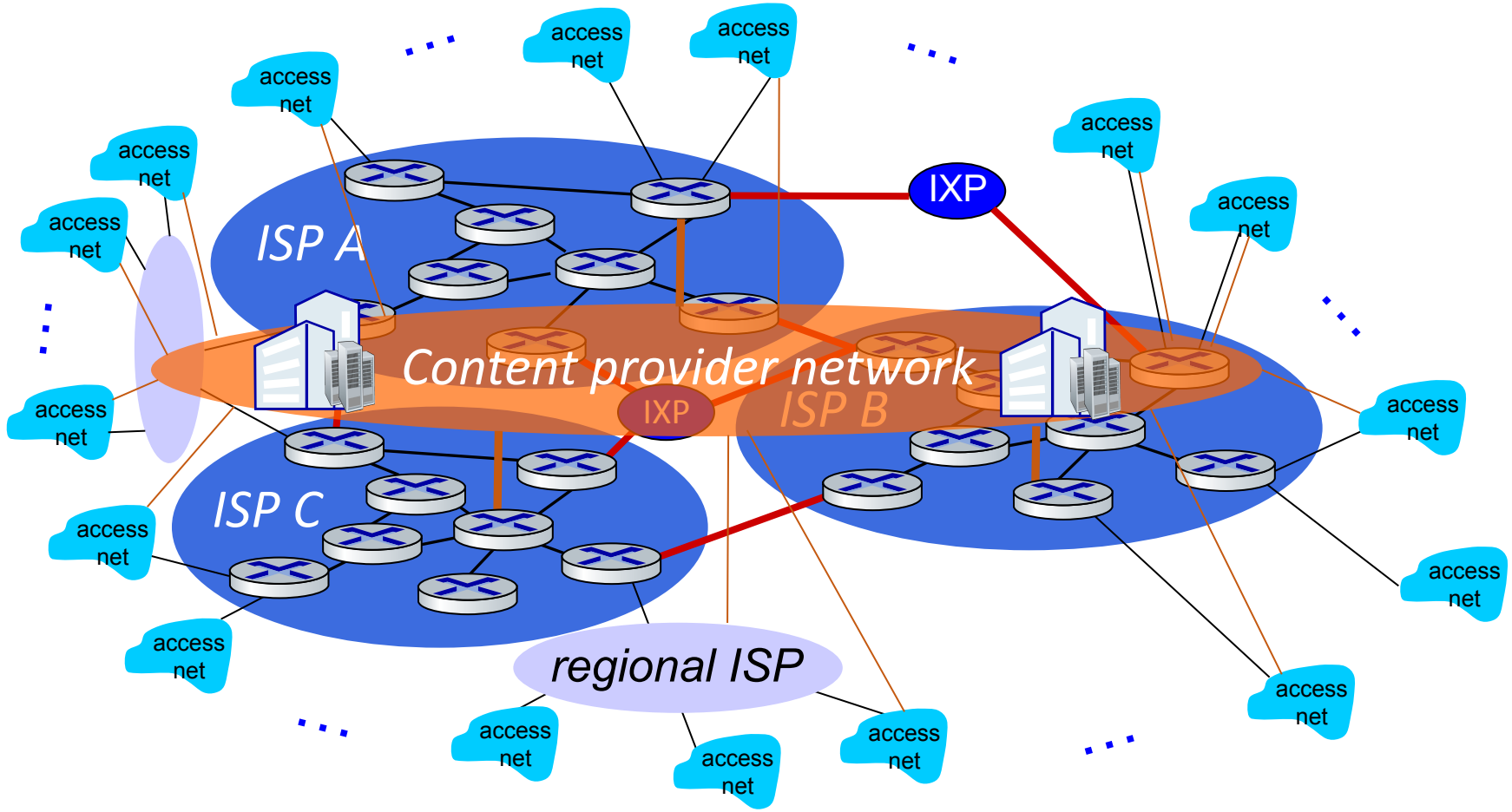
Internet structure: a “network of networks”

… and regional networks may arise to connect access nets to ISPs

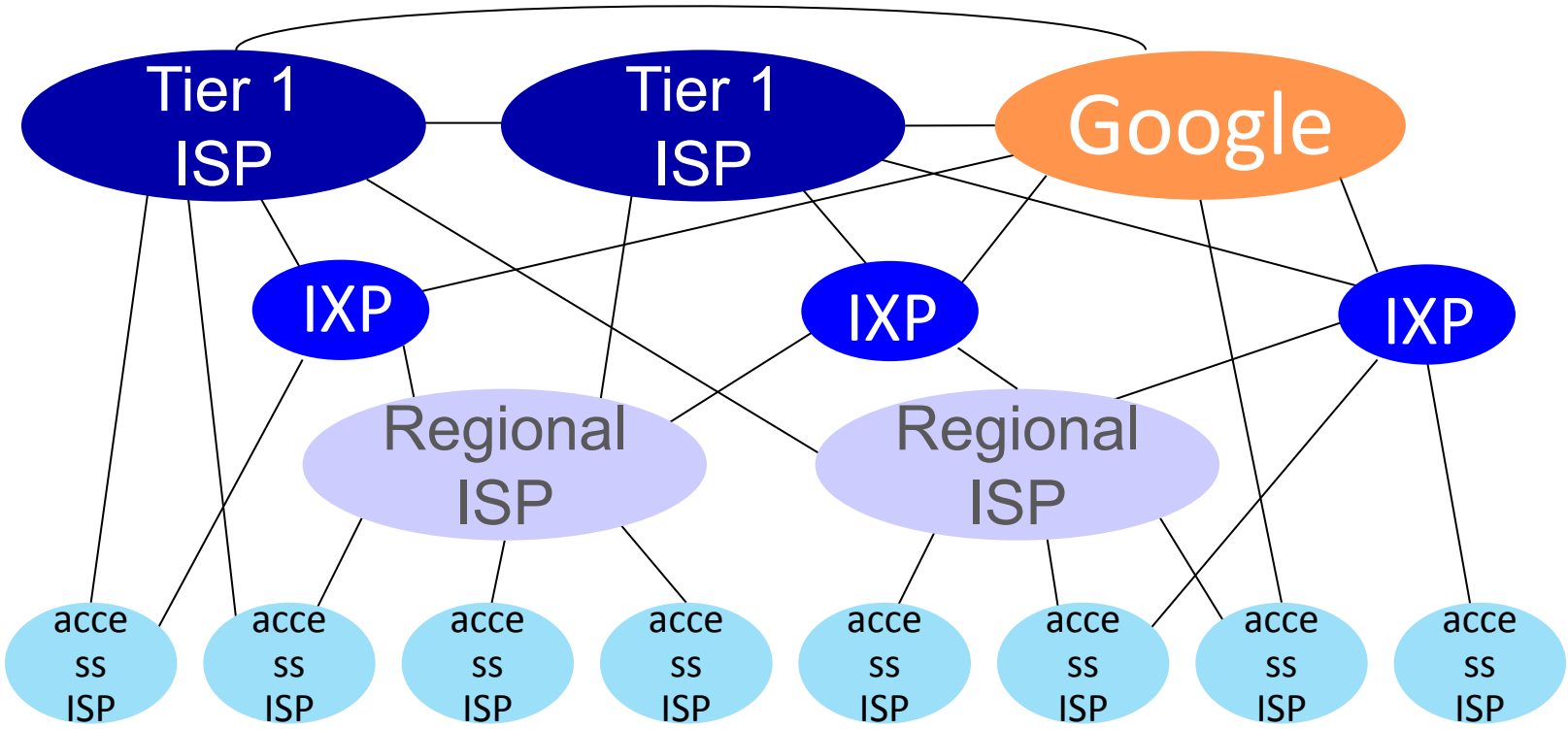


Internet structure: a “network of networks”

… and content provider networks (e.g., Google, Facebook) may run their own network, to bring services, content close to end users. ISP:- (National internet exchange of india), Peering network :- National knowledge network



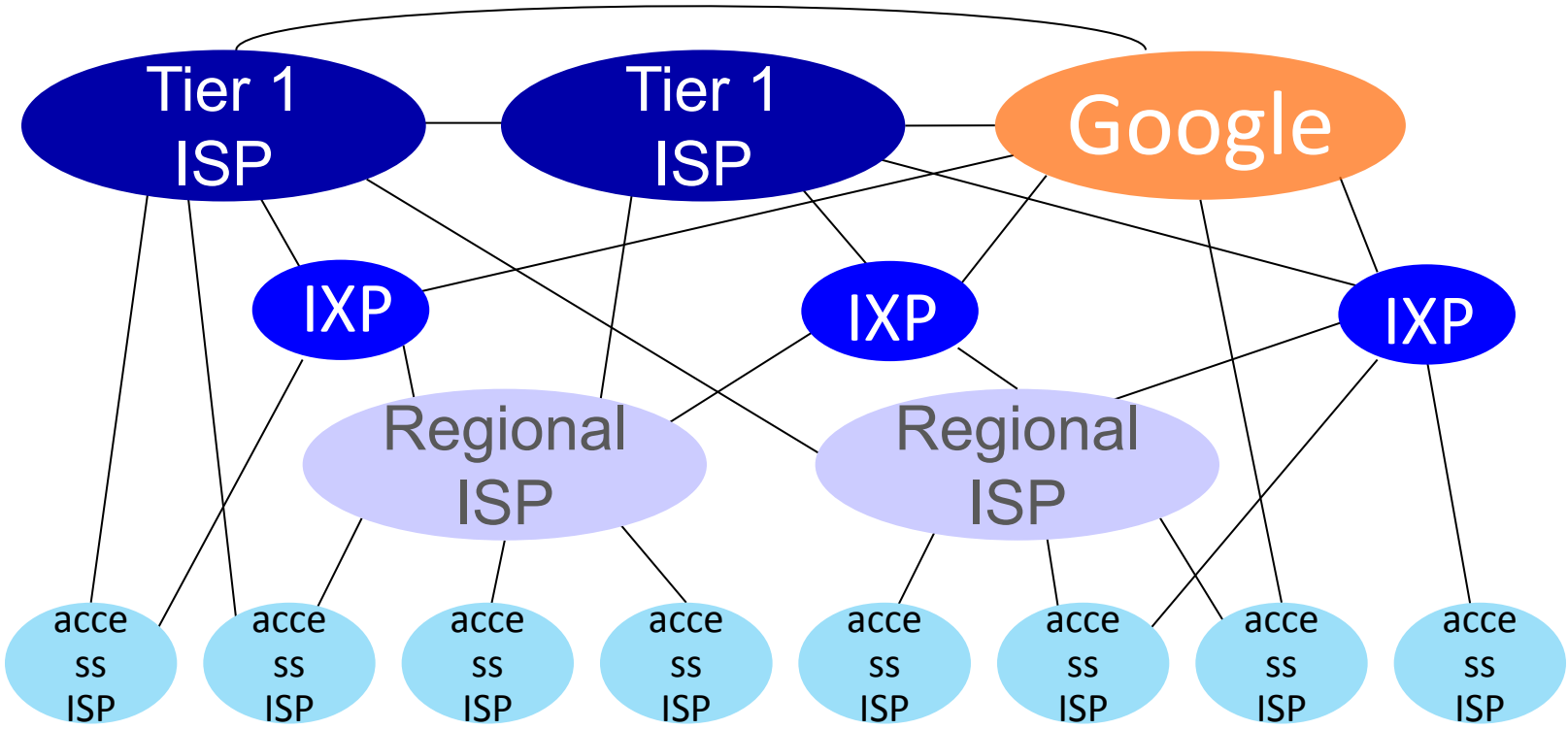
Internet structure: a "network of networks"



At "center": small # of well-connected large networks

- "tier-1" commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
- content provider networks (e.g., Google, Facebook): private network that connects its data centers to Internet, often bypassing tier-1, regional ISPs

Internet structure: a “network of networks”

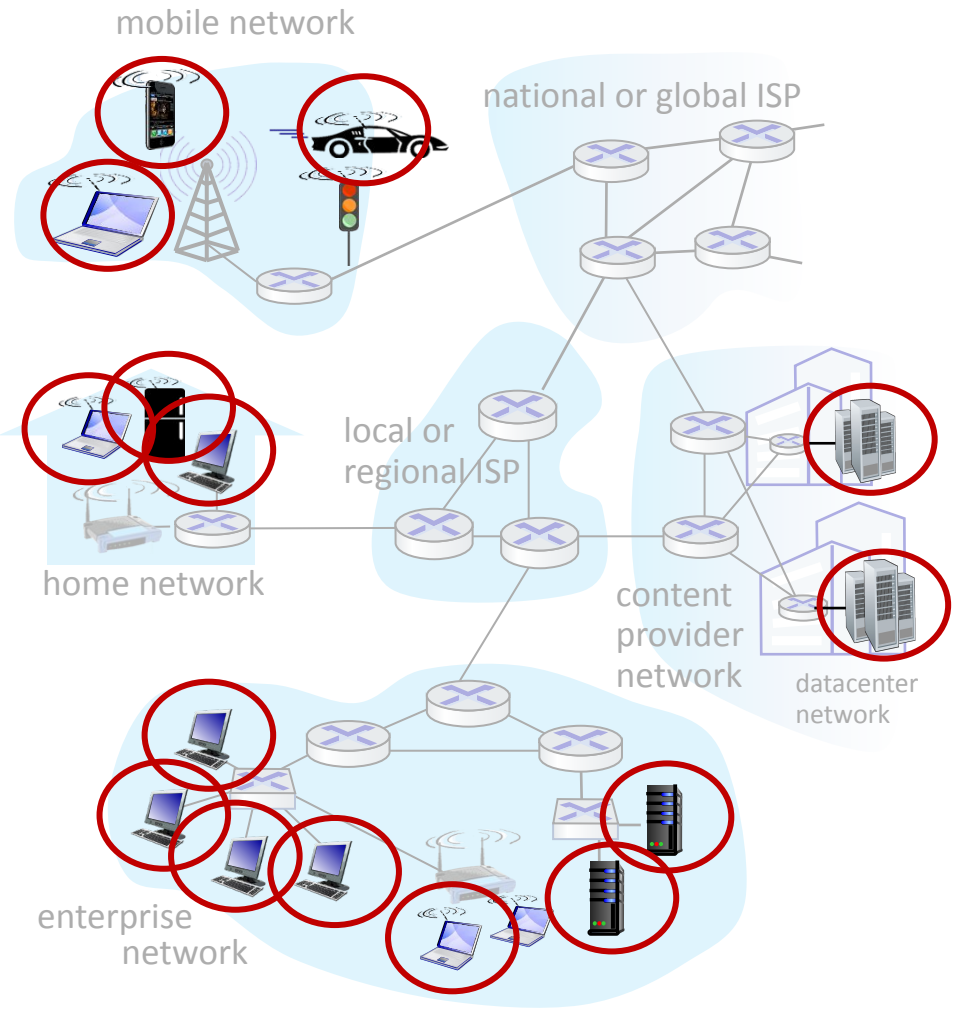


- Example Google Global Cache : Google installings caching servers with an ISP’s data center or network, allowing users to access youtube, google search, android updates etc... directly from nearby servers, instead of google’s global data center.

A closer look at Internet structure

Network edge:

- hosts: clients and servers
- servers often in data centers



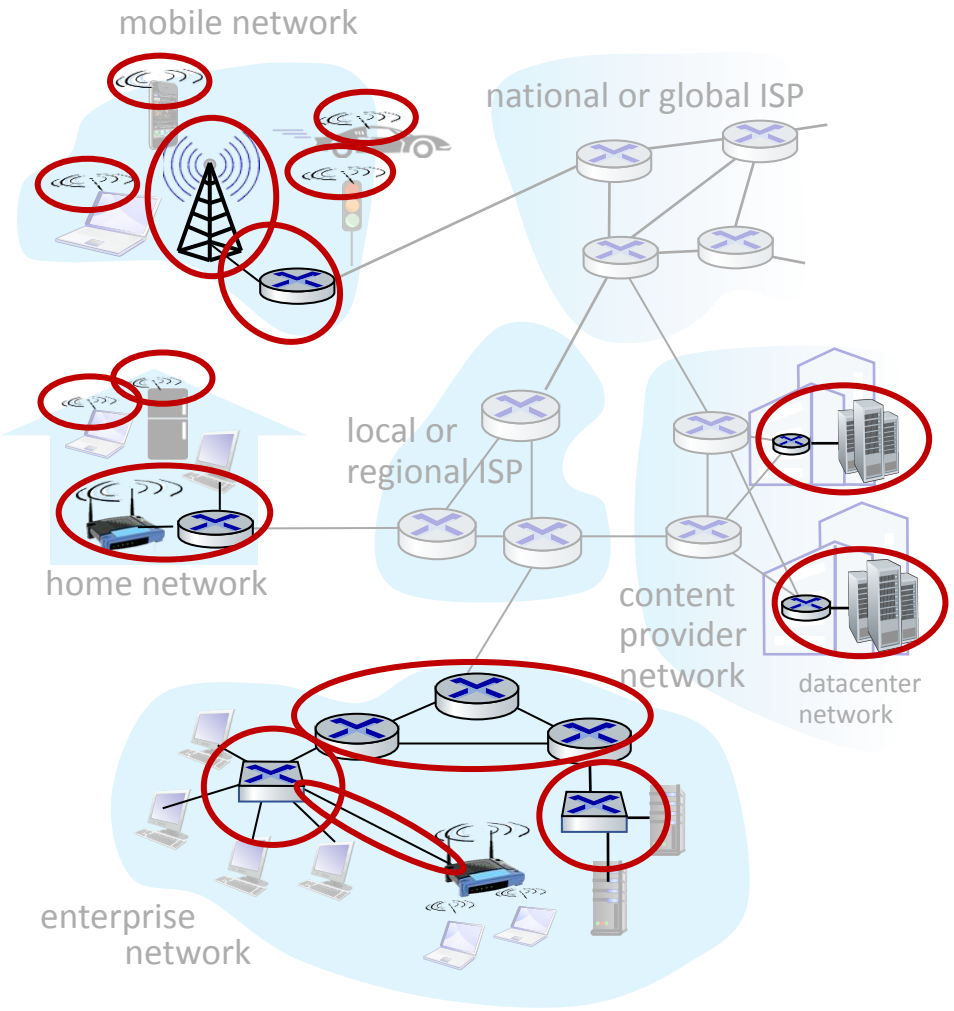
A closer look at Internet structure

Network edge:

- hosts: clients and servers
- servers often in data centers

Access networks, physical media:

- wired, wireless communication links



A closer look at Internet structure

Network edge:

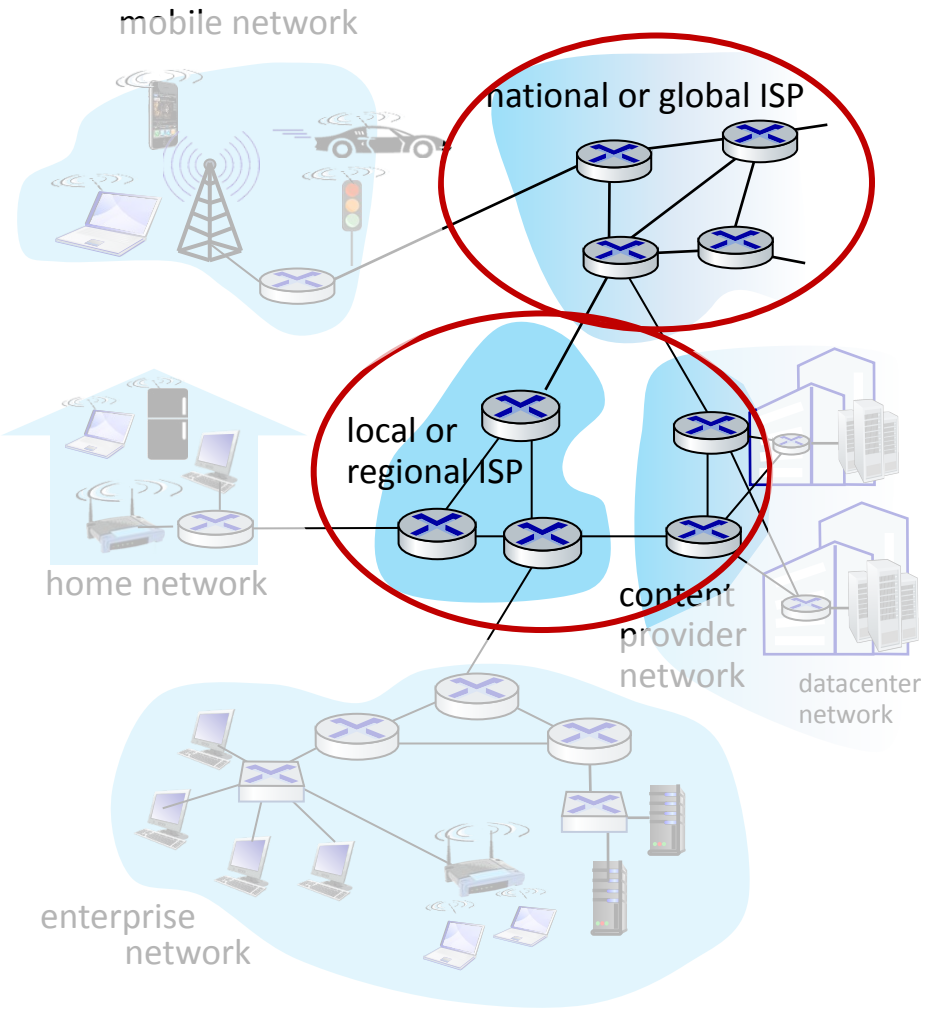
- hosts: clients and servers
- servers often in data centers

Access networks, physical media:

- wired, wireless communication links

Network core:

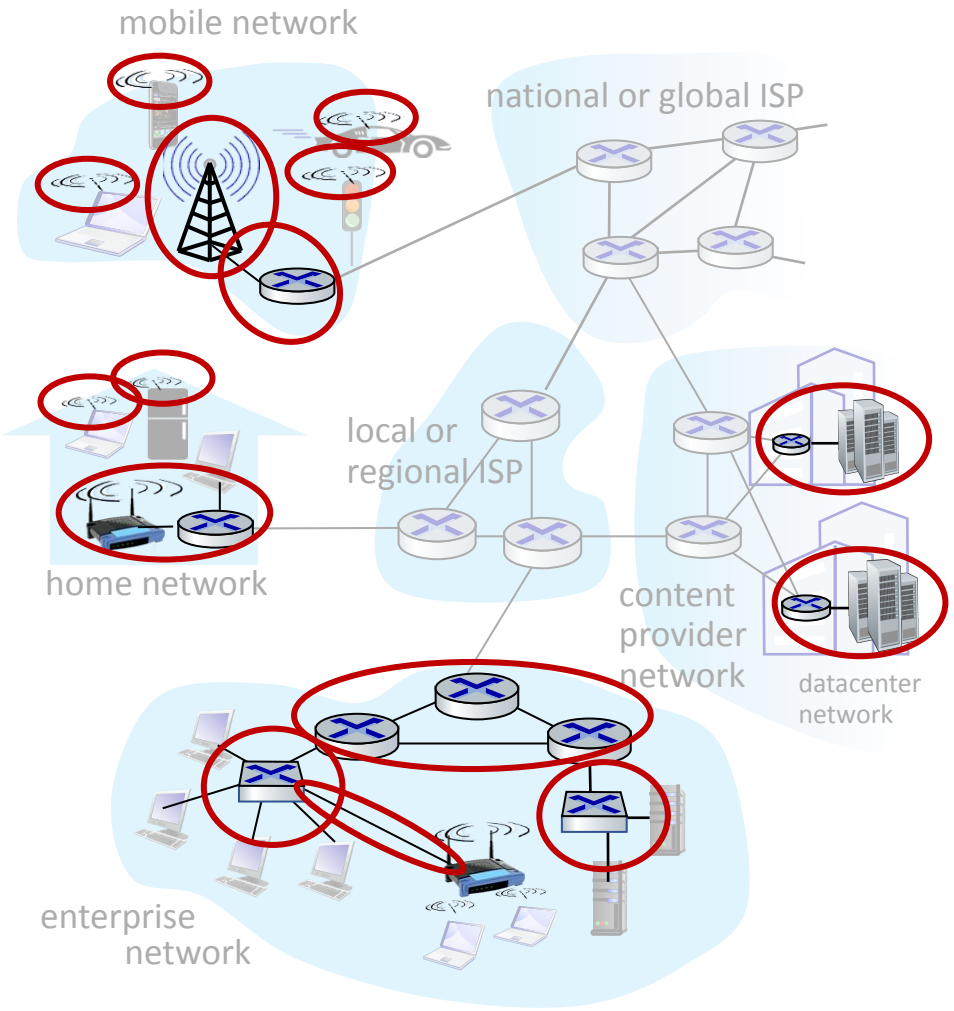
- interconnected routers
- network of networks



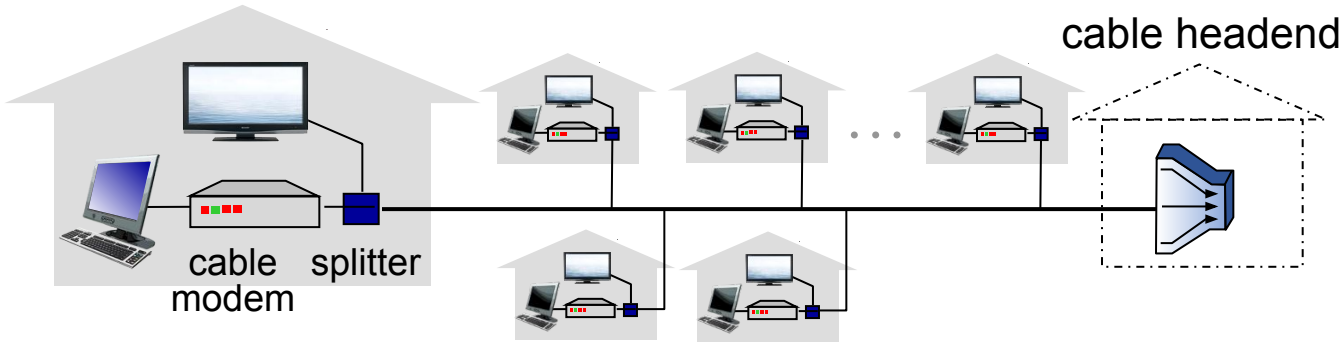
Access networks and physical media

Q: How to connect end systems to edge router?

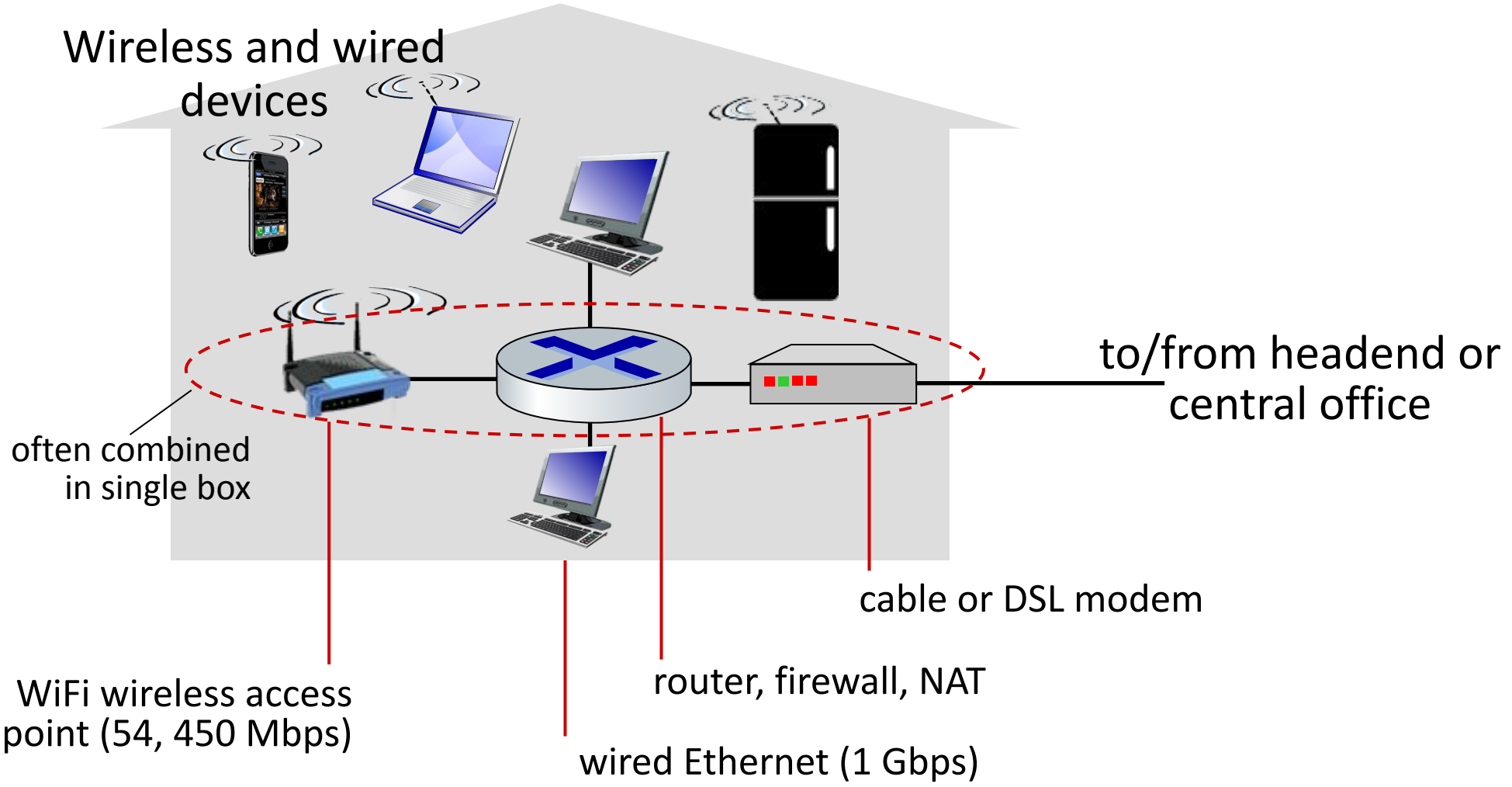
- residential access nets
- institutional access networks (school, company)
- mobile access networks (WiFi, 4G/5G)



Access networks: cable-based access



Access networks: home networks



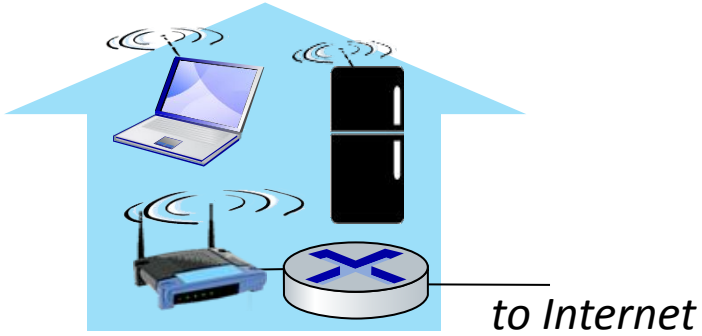
Access Networks: Wireless access networks

Shared *wireless* access network connects end system to router

- via base station aka “access point”

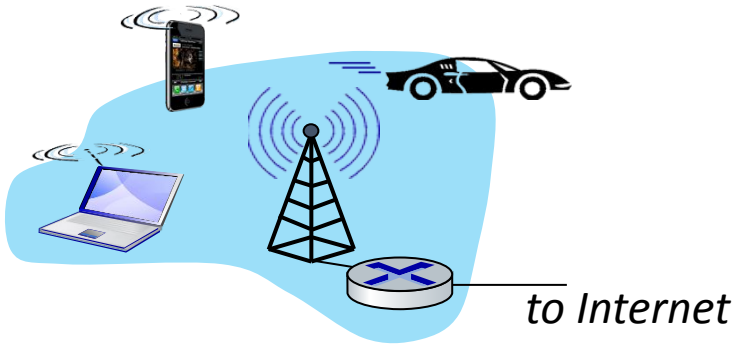
Wireless local area networks (WLANs)

- typically within or around building (~100 ft)
- 802.11b/g/n (WiFi): 11, 54, 450 Mbps transmission rate

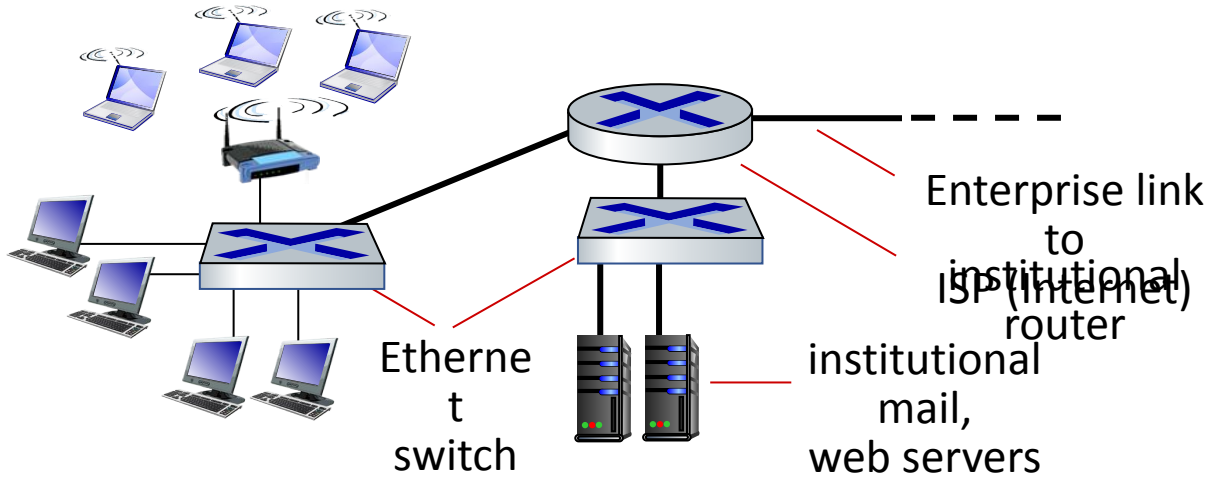


Wide-area cellular access networks

- provided by mobile, cellular network operator (10’s km)
- 10’s Mbps
- 4G/5G cellular networks



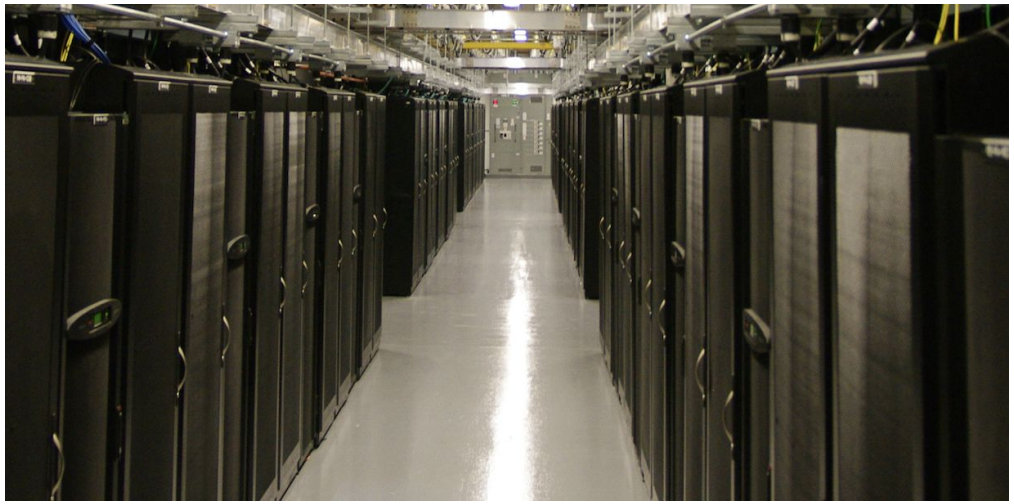
Access networks: enterprise networks



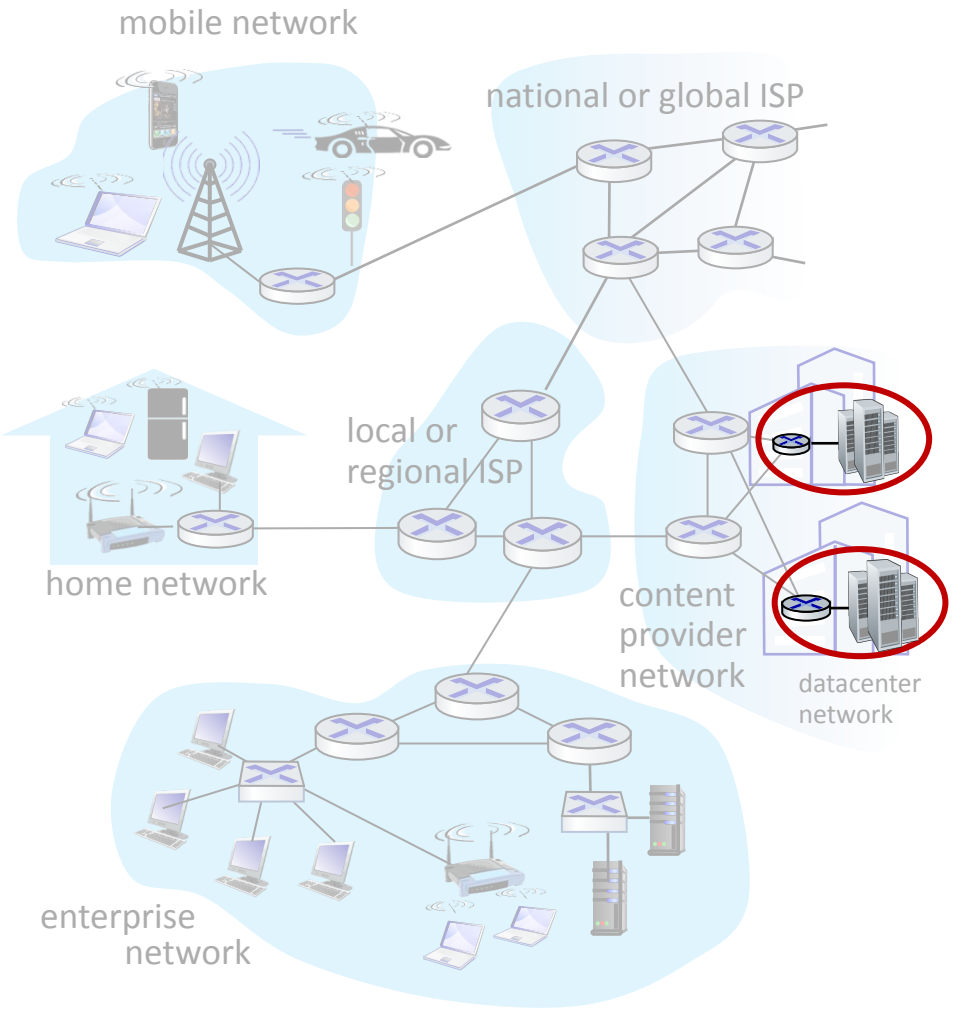
- companies, universities, etc.
- mix of wired, wireless link technologies, connecting a mix of switches and routers (we'll cover differences shortly)
  - Ethernet: wired access at 100Mbps, 1Gbps, 10Gbps
  - WiFi: wireless access points at 11, 54, 450 Mbps

Access networks: data center networks

- high-bandwidth links (10s to 100s Gbps) connect hundreds to thousands of servers together, and to Internet



Courtesy: Massachusetts Green High Performance Computing Center (mghpcc.org)



Switching

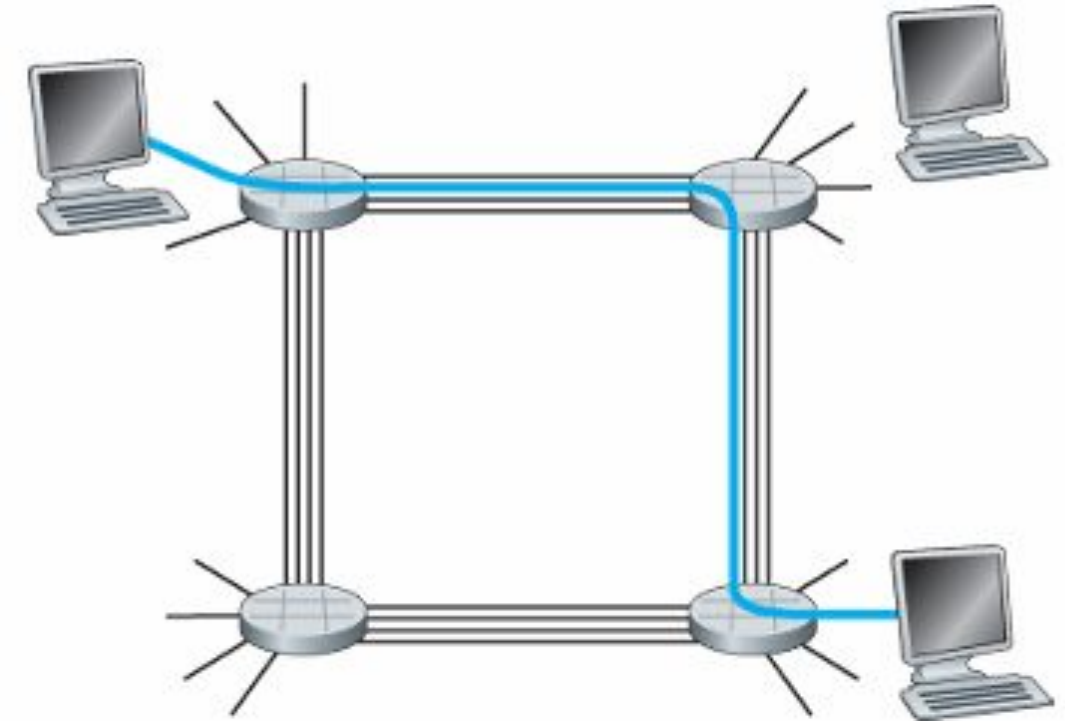
- We have discussed that internet is a network of networks in which two or more networks are connected to each other via switches and routers.
- Switching is the process of transferring data packets from one device to another in a network, or from one network to another via switches.
- There are two types of switching is popular in computer networks -
  - Circuit Switching
  - Packet Switching

Circuit Switching

- In a circuit-switched network, a dedicated connection, called a **circuit**, is always available between two end systems; the switch can only make it active or inactive.
- 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.
- Traditional telephone networks are examples of circuit-switched networks.

Circuit Switching

- In the example, a dedicated connection (in blue) is created between two end systems. This connection or circuit will be reserved for the entire duration of communication.
- Because each link has four circuits, for each link used by the end-to-end connection, the connection gets one fourth of the link's total transmission capacity for the duration of the connection. Thus, for example, if each link between adjacent switches has a transmission rate of 1 Mbps, then each end-to-end circuit-switch connection gets 250 kbps of dedicated transmission rate.



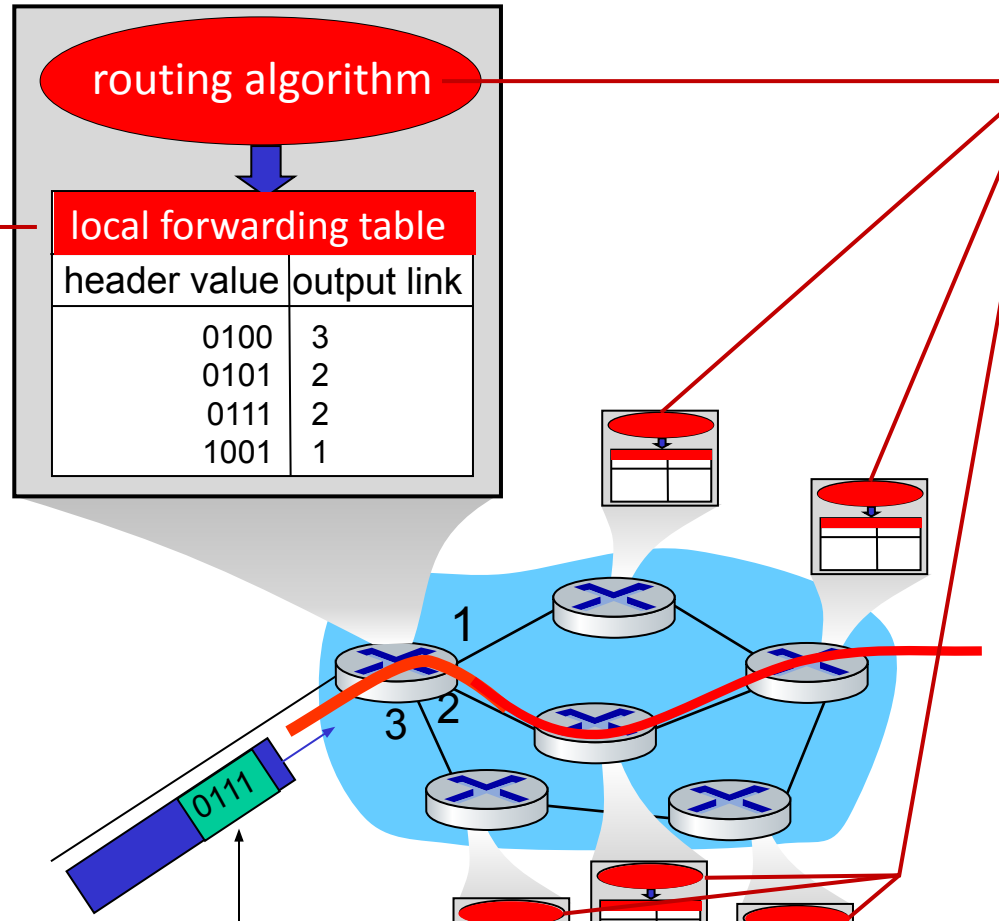
Packet Switching

- In a network application, end-systems exchanges messages with each other. These messages can be anything like a connection request or maybe some data transfer message like an email, file transfer, etc.
- 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* (routers, link-layer switches, etc.).
- Packets are transmitted over each communication link at 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 / second*** then the time to transmit the packet is ***L / R seconds***.

Two key network-core functions

Forwarding:

- aka "switching"
- *local* action: move arriving packets from router's input link to appropriate router output link



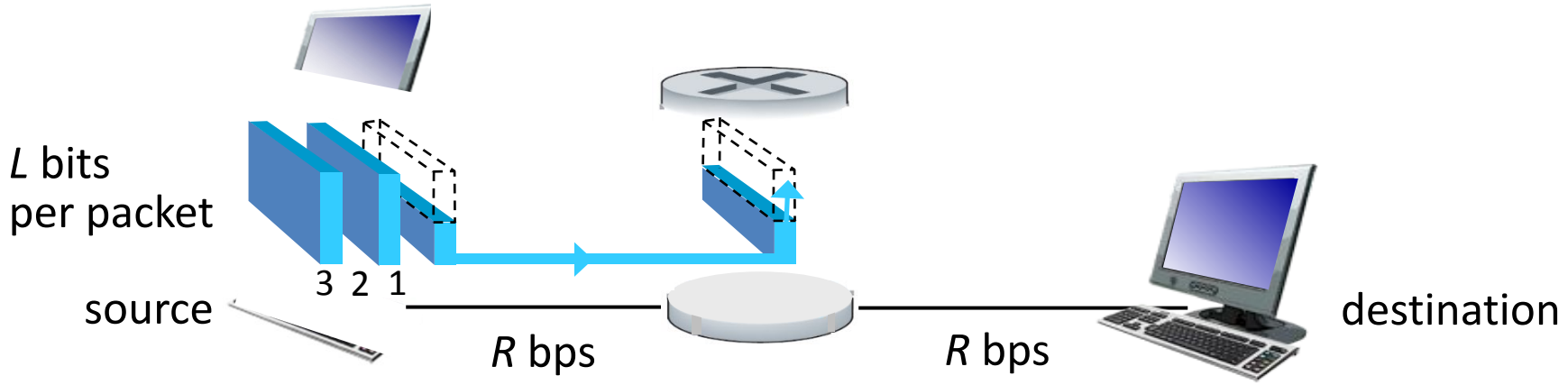
Routing:

- *global* action: determine source-destination paths taken by packets
- routing algorithms


routing
45 h
3,053 miles
48 h
3,309 miles
Victoria
Seattle
WASHINGTON
Portland
OREGON
IDAHO
MONTANA
NORTH DAKOTA
SOUTH DAKOTA
WYOMING
MINNESOTA
Minneapolis
WISCONSIN
MICHIGAN
Toronto
Quebec City
Montreal
Ottawa
MAINE
VERMONT
NEW HAMPSHIRE
Northampton
CT
RI
Detroit
Chicago
IOWA
NEVADA
UTAH
COLORADO
United States
Kansas City
ILLINOIS
INDIANA
Indianapolis
OHIO
PENNSYLVANIA
New York
Philadelphia
NJ
MARYLAND
DE
WEST VIRGINIA
VIRGINIA
KENTUCKY
Nashville
TENNESSEE
NORTH CAROLINA
Charlotte
SOUTH CAROLINA
Atlanta
GEORGIA
ALABAMA
MISSISSIPPI
ARKANSAS
OKLAHOMA
Dallas
TEXAS
Austin
Houston
San Antonio
LOUISIANA
New Orleans
Jacksonville
Orlando
Tampa
FLORIDA
Sacramento
San Francisco
San Jose
CALIFORNIA
Las Vegas
Los Angeles
San Diego
ARIZONA
Phoenix
Tucson
El Paso
NEW MEXICO
BAJA CALIFORNIA
SONORA
CHIHUAHUA
COAHUILA


forwarding
forwarding
45 h
3,053 miles
48 h
3,309 miles
United States
Victoria
Seattle
WASHINGTON
Portland
NORTH DAKOTA
SOUTH DAKOTA
MINNESOTA
Minneapolis
WISCONSIN
MICHIGAN
Toronto
Quebec City
Montreal
Ottawa
MAINE
VERMONT
NEW HAMPSHIRE
Northampton
WYOMING
IOWA
Chicago
Detroit
ILLINOIS
INDIANA
OHIO
Indianapolis
Kansas City
COLORADO
KANSAS
MISSOURI
KENTUCKY
Nashville
TENNESSEE
San Francisco
San Jose
CALIFORNIA
Las Vegas
OKLAHOMA
ARKANSAS
Atlanta
Los Angeles
ARIZONA
NEW MEXICO
Dallas
MISSISSIPPI
ALABAMA
San Diego
Phoenix
Tucson
El Paso
TEXAS
BAJA CALIFORNIA
SONORA
CHIHUAHUA
Austin
Houston
LOUISIANA
New Orleans
San Antonio
COAHUILA
Tampa
FLORIDA
Cleveland
DOWNTOWN
CENTRAL
OHIO CITY

Packet-switching: store-and-forward

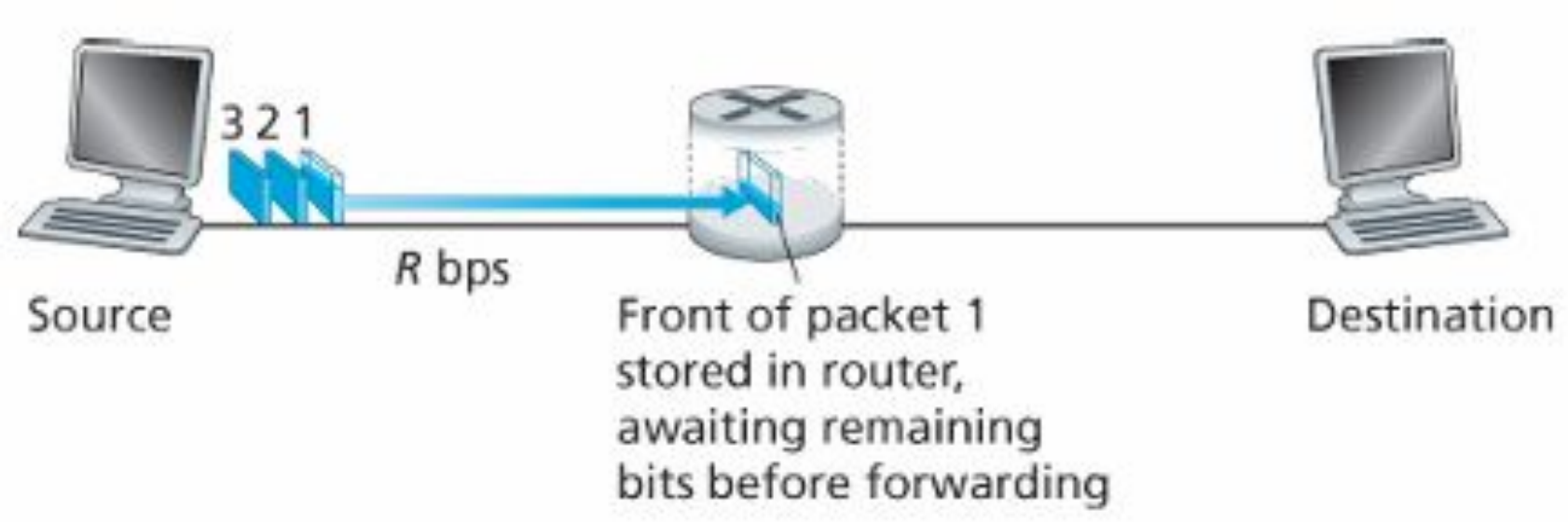


- packet transmission delay: takes L/R seconds to transmit (push out) L-bit packet into link at R bps
- *store and forward: entire* packet must arrive at router before it can be transmitted on next link

One-hop numerical example:

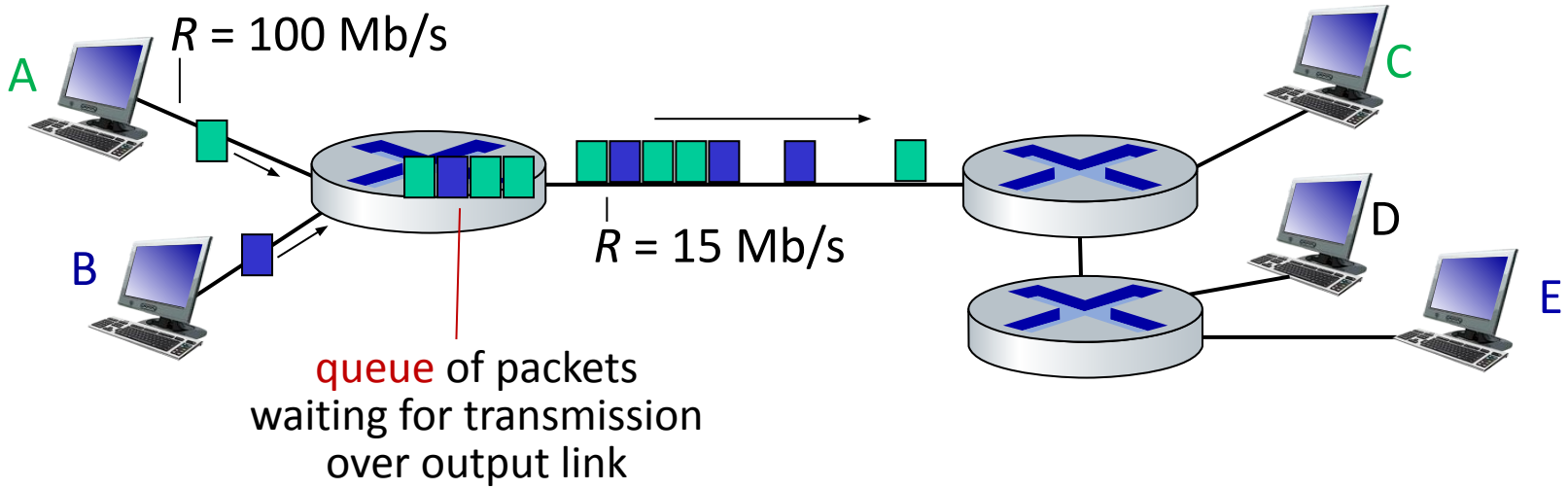
- L = 10 Kbits
- R = 100 Mbps
- one-hop transmission delay = 0.1 msec

Exercise



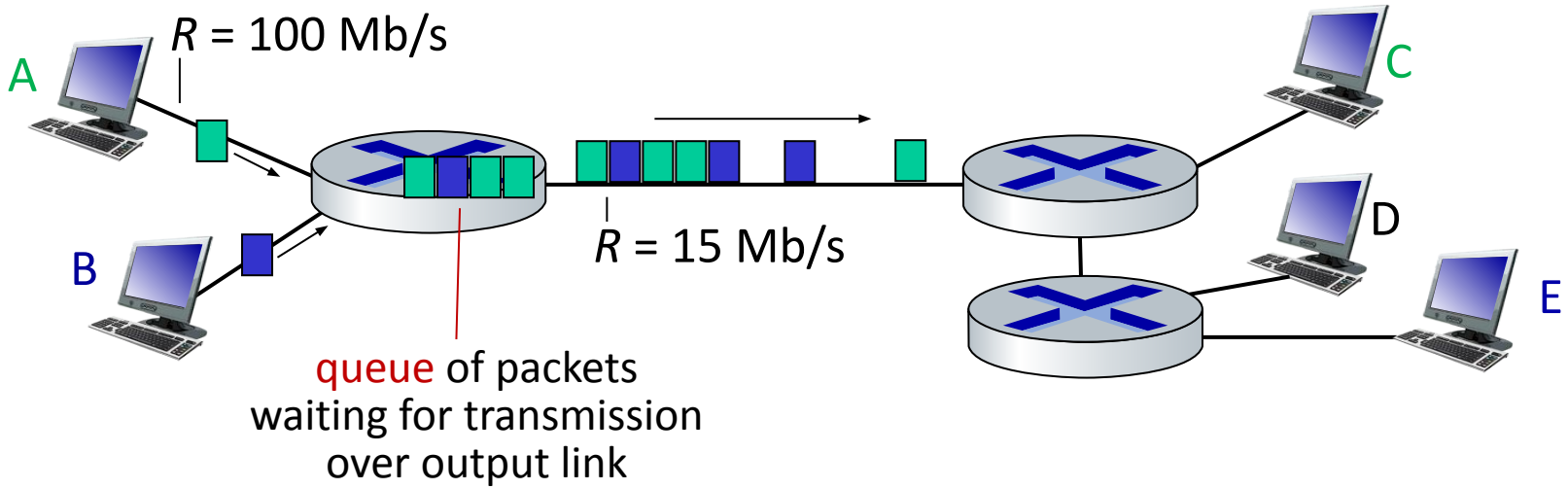
- In given example, how much time will it take for 1st packet to reach the destination? 0-L/R , L/R -> 2L/R,2L/R ->3L/R , 3L/R->4L/R
- Total time taken in transmitting all three packets from source to destination?

Packet-switching: queueing delays and packet loss



Queueing occurs when work arrives faster than it can be serviced:

Packet-switching: queueing delays and packet loss



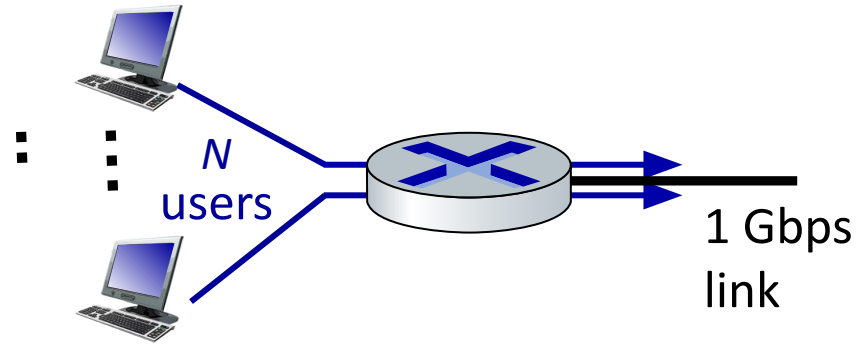
Packet queuing and loss: if arrival rate (in bps) to link exceeds transmission rate (bps) of link for some period of time:

- packets will queue, waiting to be transmitted on output link
- packets can be dropped (lost) if memory (buffer) in router fills up

Packet switching versus circuit switching

example:

- 1 Gb/s link
- each user:
  - 100 Mb/s when "active"
  - active 10% of time



Q: how many users can use this network under circuit-switching and packet switching?

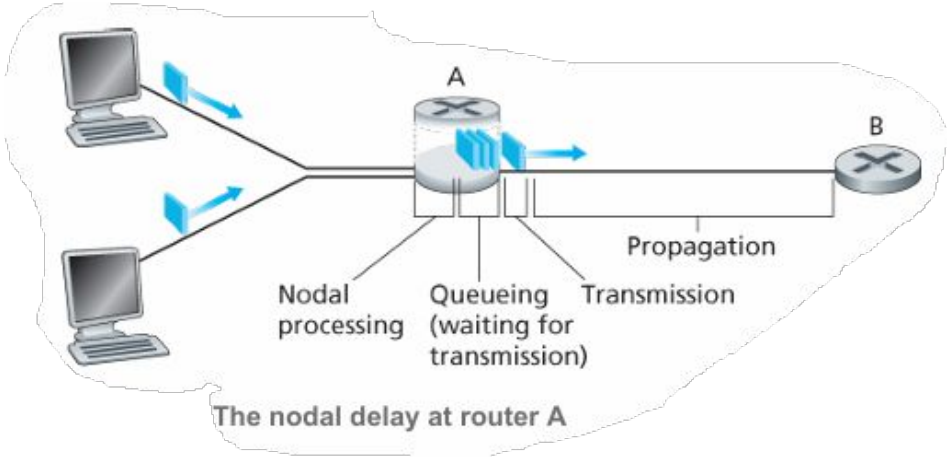
- *circuit-switching:* 10 users
- *packet switching:* with 35 users, probability > 10 active at same time is less than .0004 *

Q: how did we get value 0.0004?

A: HW problem

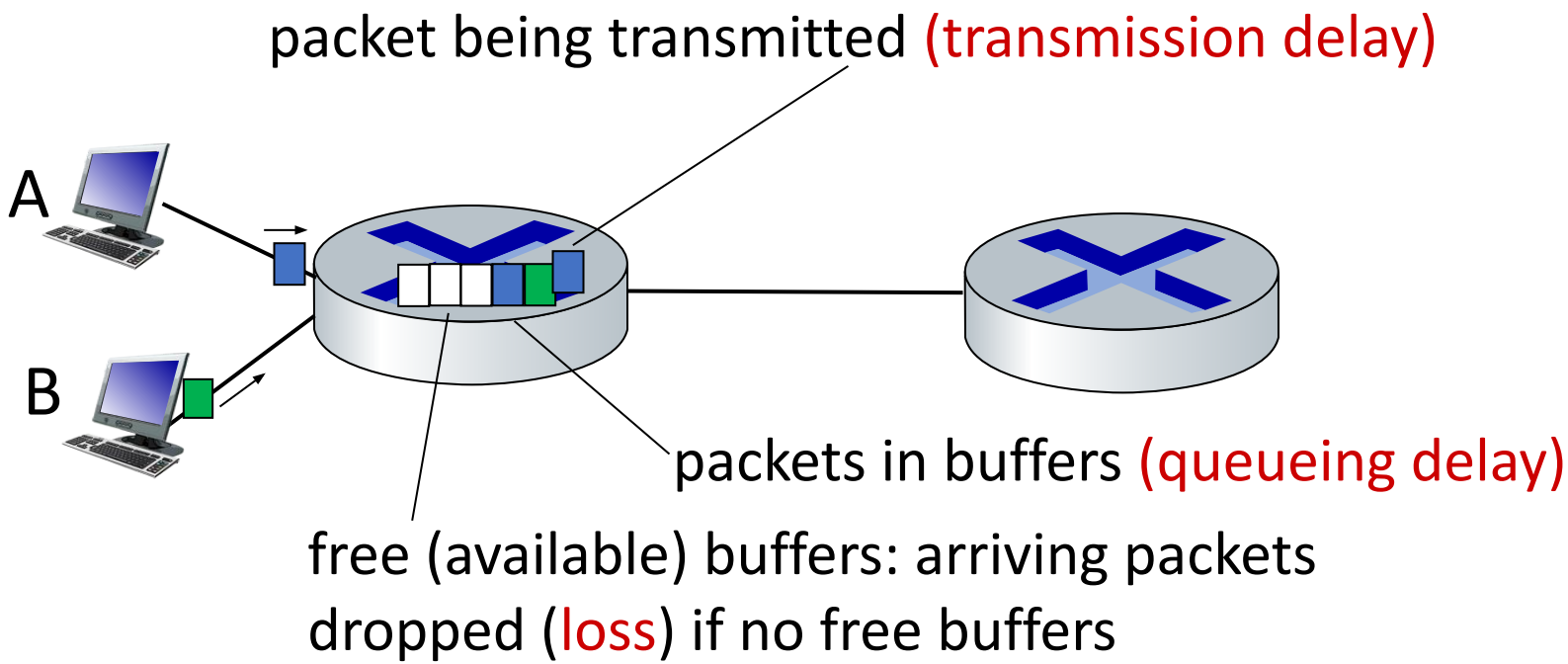
Delay in Packet-Switched Networks

- As a packet travels from one node (host or router) to the subsequent node (host or router) while going from source to destination, the packet suffers from several types of delays at each node along the path.
- The most important of these delays are the ***nodal processing delay***, ***queuing delay, transmission delay***, and ***propagation delay***; together, these delays accumulate to give a ***total nodal delay***.

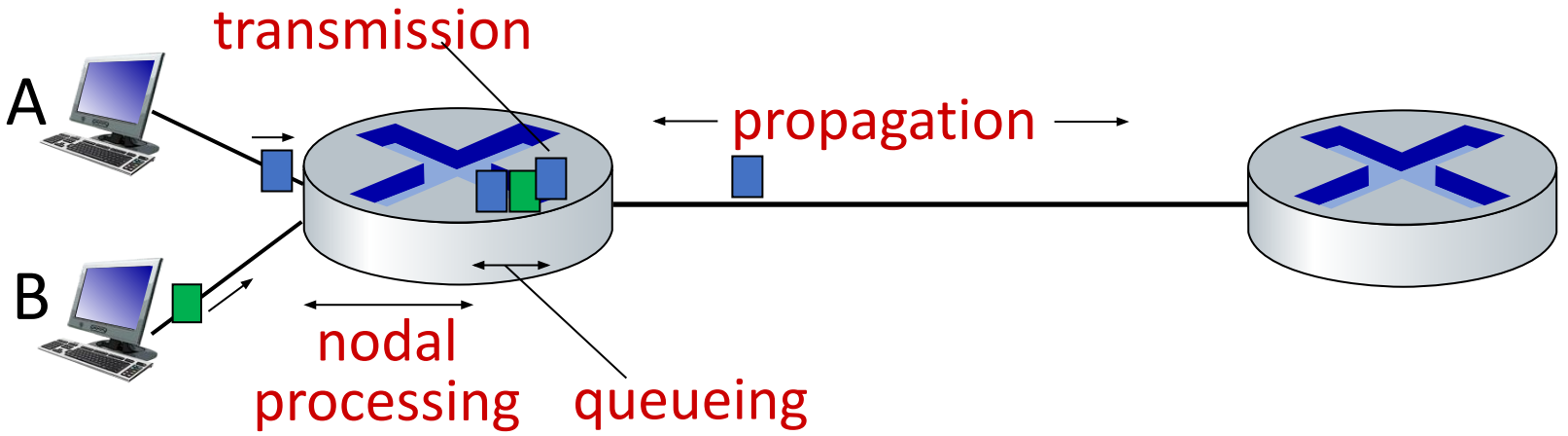


Delay in Packet-Switched Networks

- packets *queue* in router buffers, waiting for turn for transmission
  - queue length grows when arrival rate to link (temporarily) exceeds output link capacity
- packet *loss* occurs when memory to hold queued packets fills up



Packet delay: four sources



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

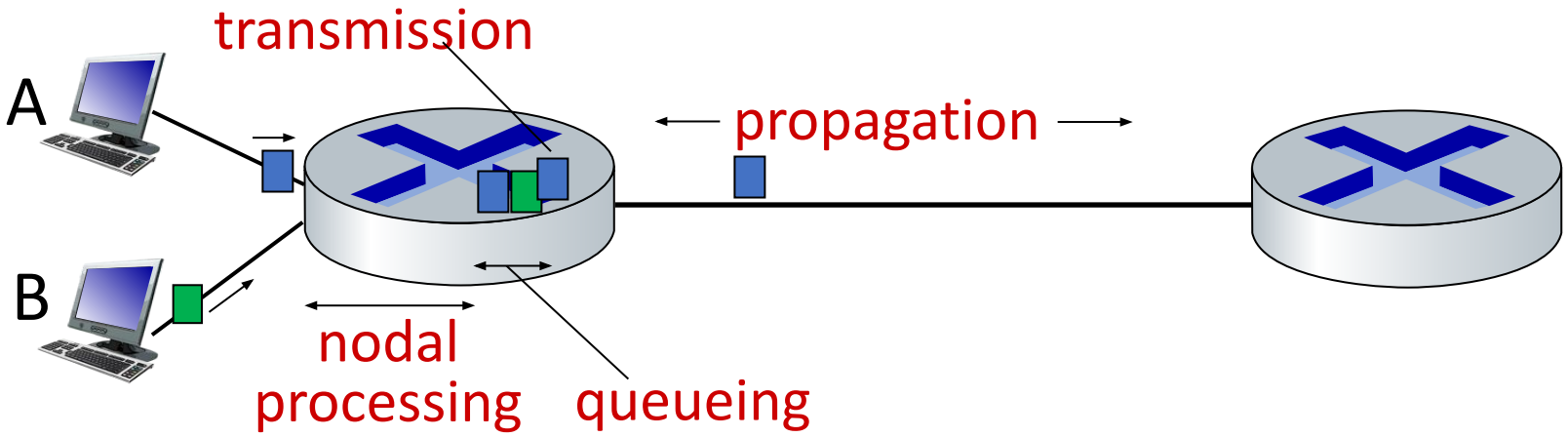
d_{proc}: nodal processing

- check bit errors
- determine output link
- typically < microsecs

d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on congestion level of router

Packet delay: four sources



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

d_{trans}: transmission delay:

- L: packet length (bits)
- R: link *transmission rate (bps)*
- $d_{trans} = L/R$

d_{prop}: propagation delay:

- d: length of physical link
- s: propagation speed (~3×10^8 m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop} *very* different

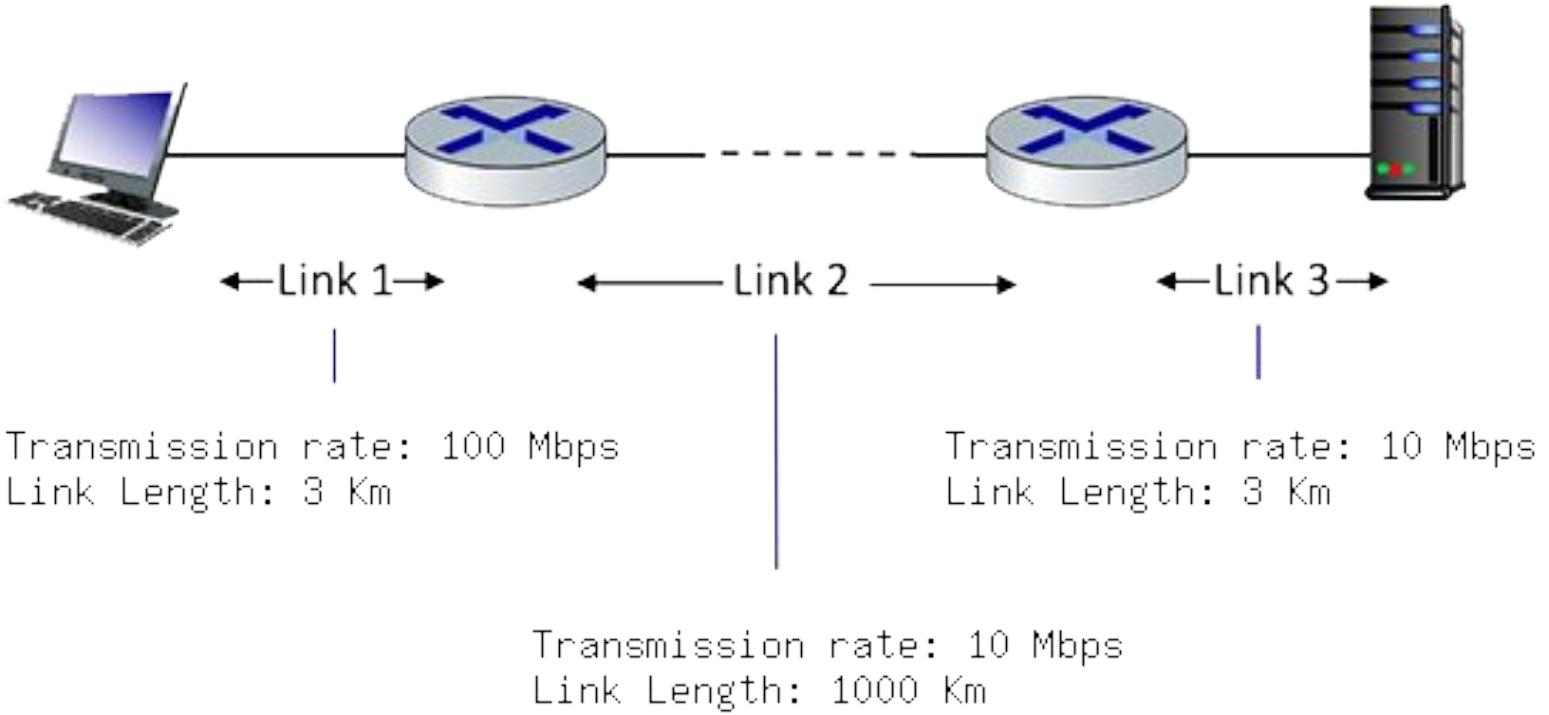
End-to-End Delay

- So far we have discussed delay at a single router, but what about total delay in the journey of packet from source to destination?
- Let's suppose there are **N routers** between the source host and the destination host.
- With the assumption that equal delays for the sender, routers, and receiver, the total end-to-end delay (source-to-destination delay) a packet encounters can be calculated as following -

$$\mathbf{d_{end\text{-}to\text{-}end} = (n + 1)\,(d_{processing} + d_{transmission} + d_{propagation}) + n\,d_{queuing}}$$

- Delay can have a significant impact on real-time applications such as voice and video chats, where low delay is essential to maintain smooth communication.

Exercise



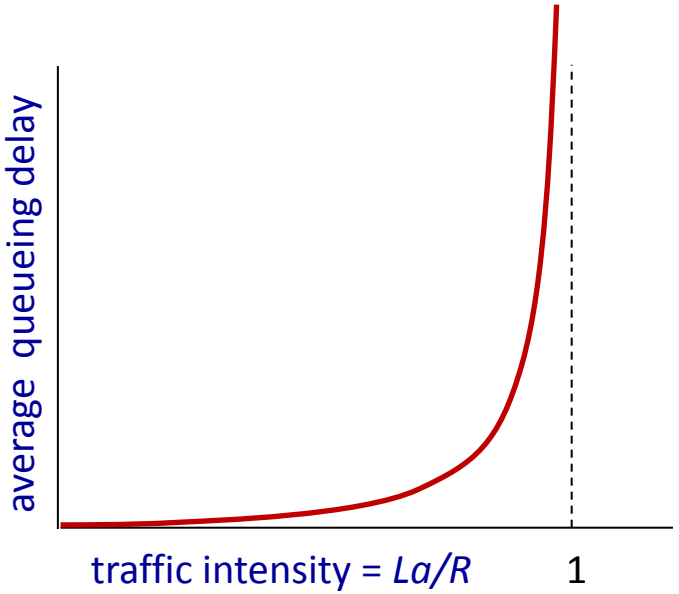
- Assume the length of a packet is 4000 bits. The propagation speed of each link is 3 x 10^8 m / sec. Assuming processing delay and queuing delay is negligible, answer following questions.
- What is the transmission delay of each link?
- What is the propogation delay of each link?
- What is the total end-to-end delay for one packet?
- Now suppose, source starts sending 3 packets at time t, at what time server/destination will receive all packets? Can you generalize the expression for n packets?

Packet queueing delay (revisited)

- *a:* average packet arrival rate
- *L:* packet length (bits)
- *R:* link bandwidth (bit transmission rate)

$$\frac{L \cdot a}{R} : \frac{\text{arrival rate of bits}}{\text{service rate of bits}}$$

"traffic intensity"



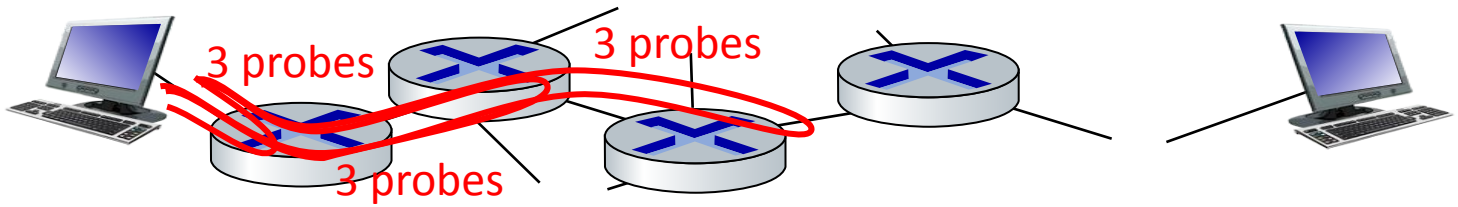
- *La/R* ~ 0: avg. queueing delay small
- *La/R* -> 1: avg. queueing delay large
- *La/R* > 1: more "work" arriving is more than can be serviced - average delay infinite!

La/R ~ 0

La/R -> 1

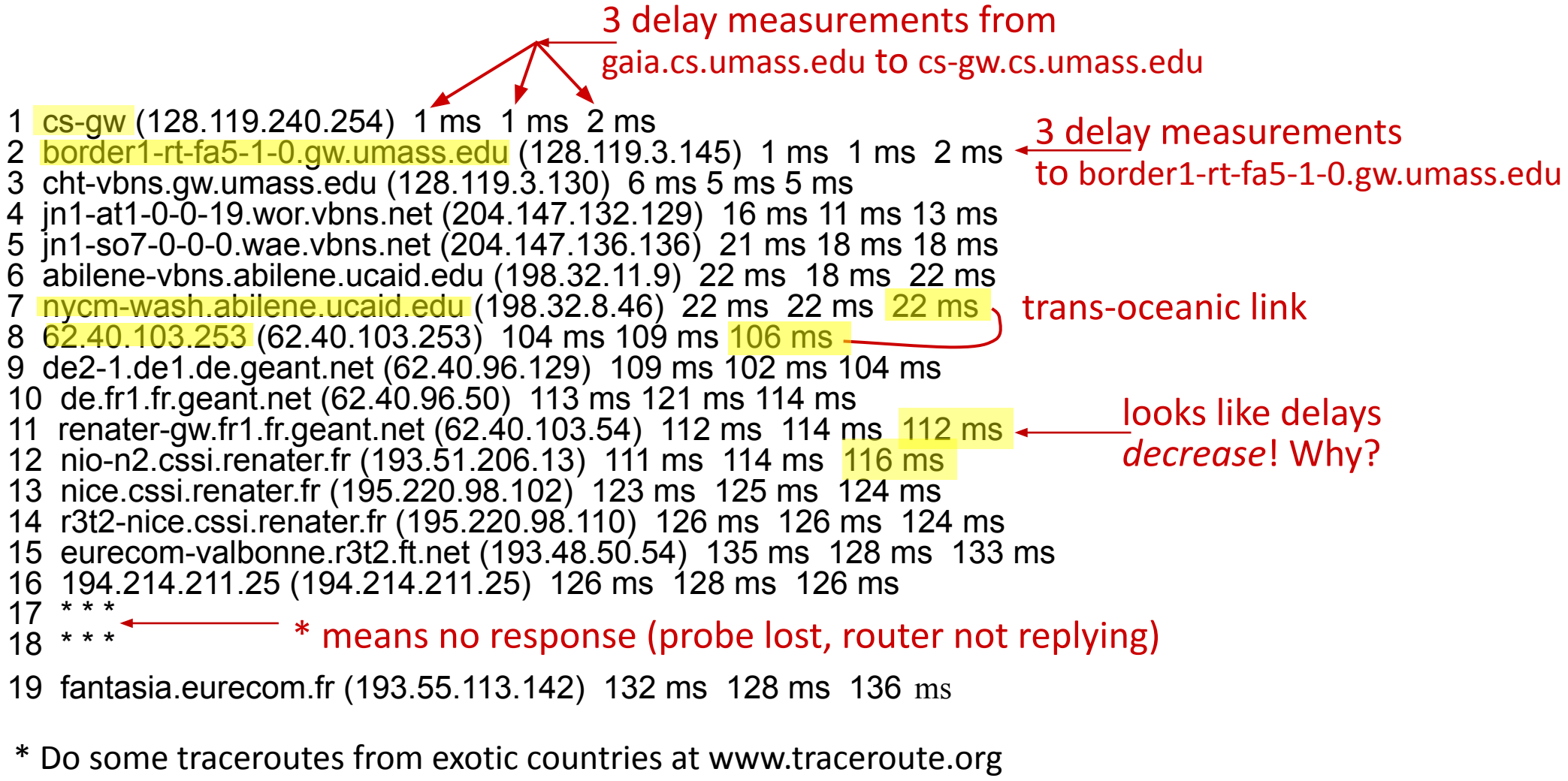
“Real” Internet delays and routes

- what do “real” Internet delay & loss look like?
- **traceroute** program: provides delay measurement from source to router along end-end Internet path towards destination. For all *i:*
  - sends three packets that will reach router *i* on path towards destination (with time-to-live field value of *i*)
  - router *i* will return packets to sender
  - sender measures time interval between transmission and reply



Real Internet delays and routes

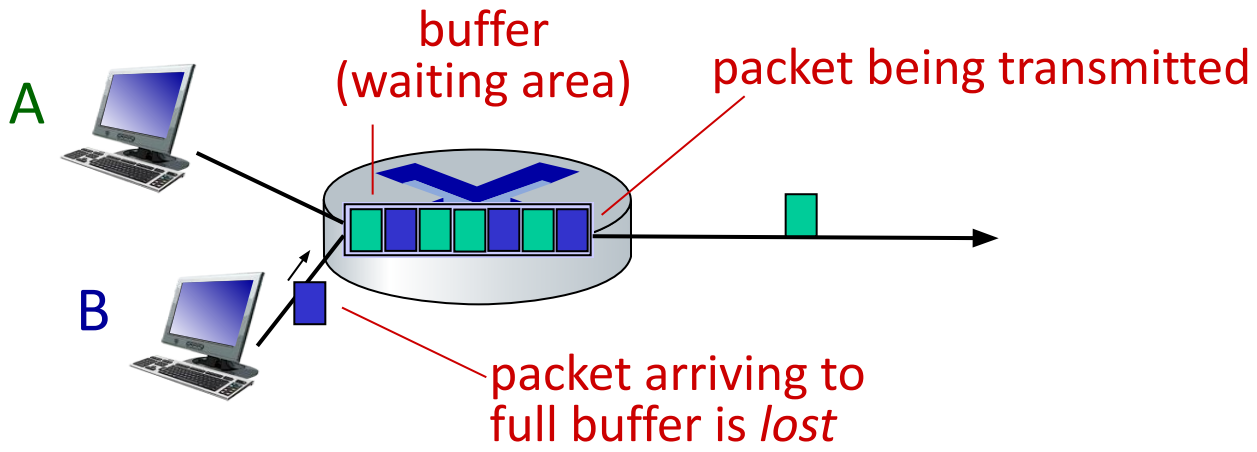
traceroute: gaia.cs.umass.edu to www.eurecom.fr



* Do some traceroutes from exotic countries at www.traceroute.org

Packet loss

- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



Throughput

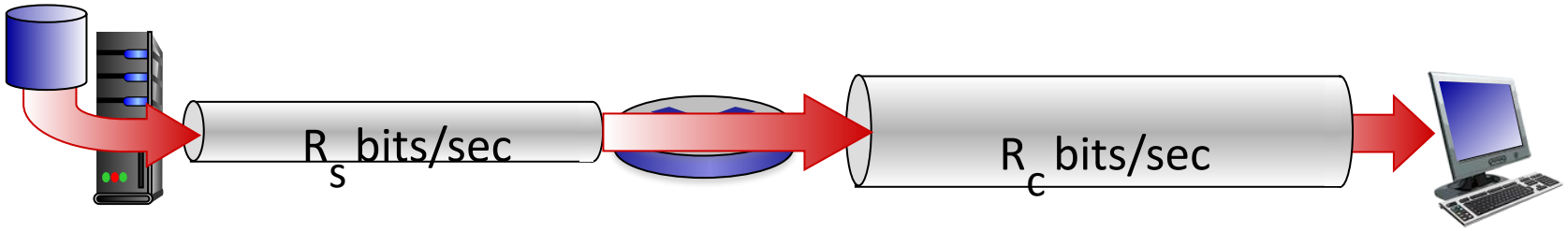
- Throughput in a packet-switched network refers to the rate at which data can be transmitted over the network link. It is usually measured in *bits per second (bps)* or *packets per second (pps)*. Throughput depends upon various factors -
- **Link Capacity** - The maximum data rate that the physical link can support. It is determined by the link's transmission rate and bandwidth.
- **Network Congestion** - High levels of network congestion can reduce throughput as packets experience more queueing delay and compete for available bandwidth.

Throughput

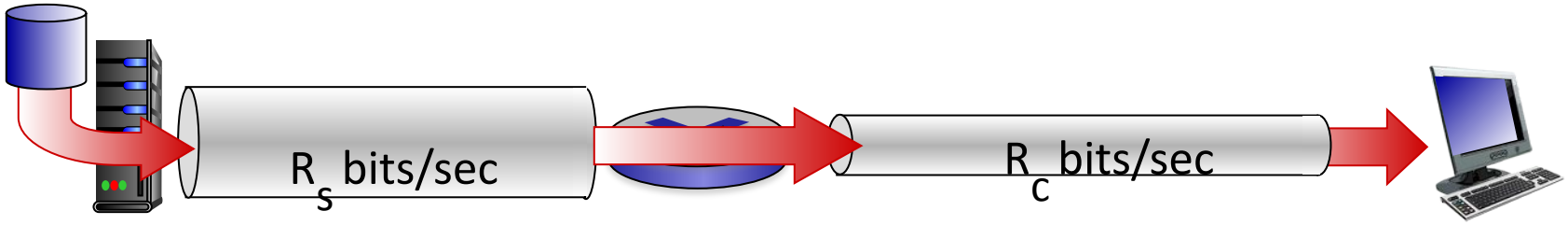
- **Protocols Overhead** - Packet-switched networks use various protocols to manage and control data transmission. These protocols add some overhead, which can slightly reduce the actual throughput.
- **Routing Efficiency** - The efficiency of the routing algorithms and protocols used in the network can impact how quickly packets reach their destination.
- Throughput is crucial for applications that require high data transfer rates, such as file downloads, streaming, and data-intensive processes.

Throughput

$R_s < R_c$ What is average end-end throughput?



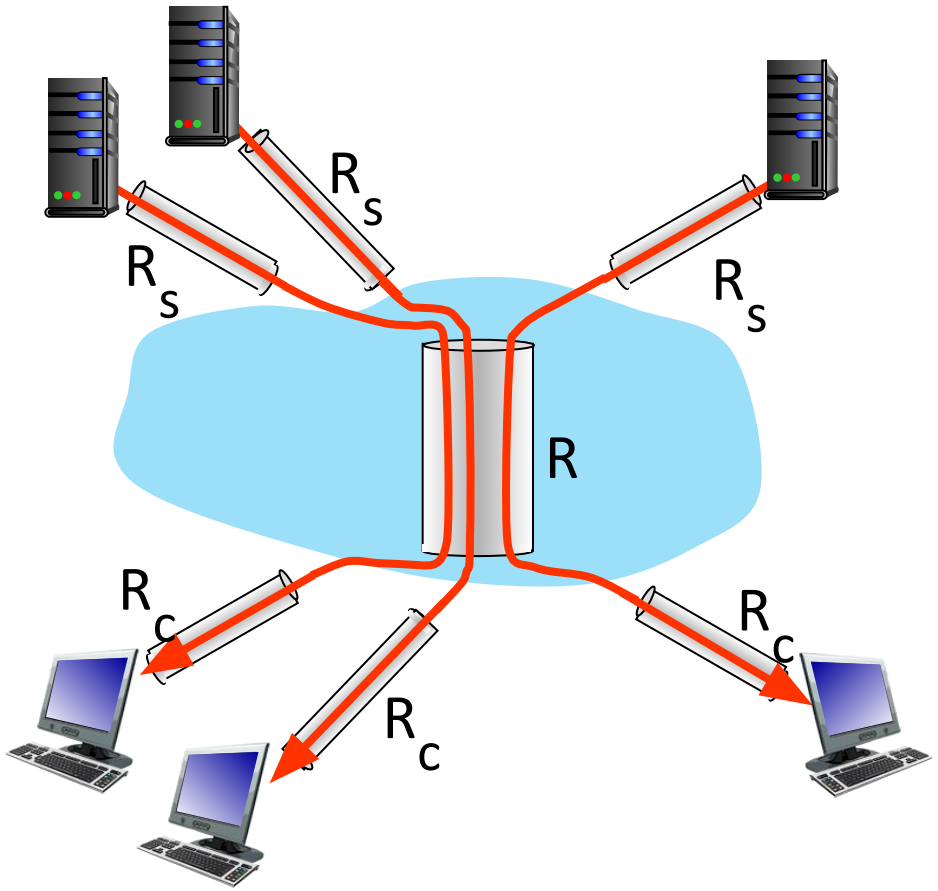
$R_s > R_c$ What is average end-end throughput?



bottleneck link

link on end-end path that constrains end-end throughput

Throughput: network scenario



10 connections (fairly) share backbone bottleneck link *R* bits/sec

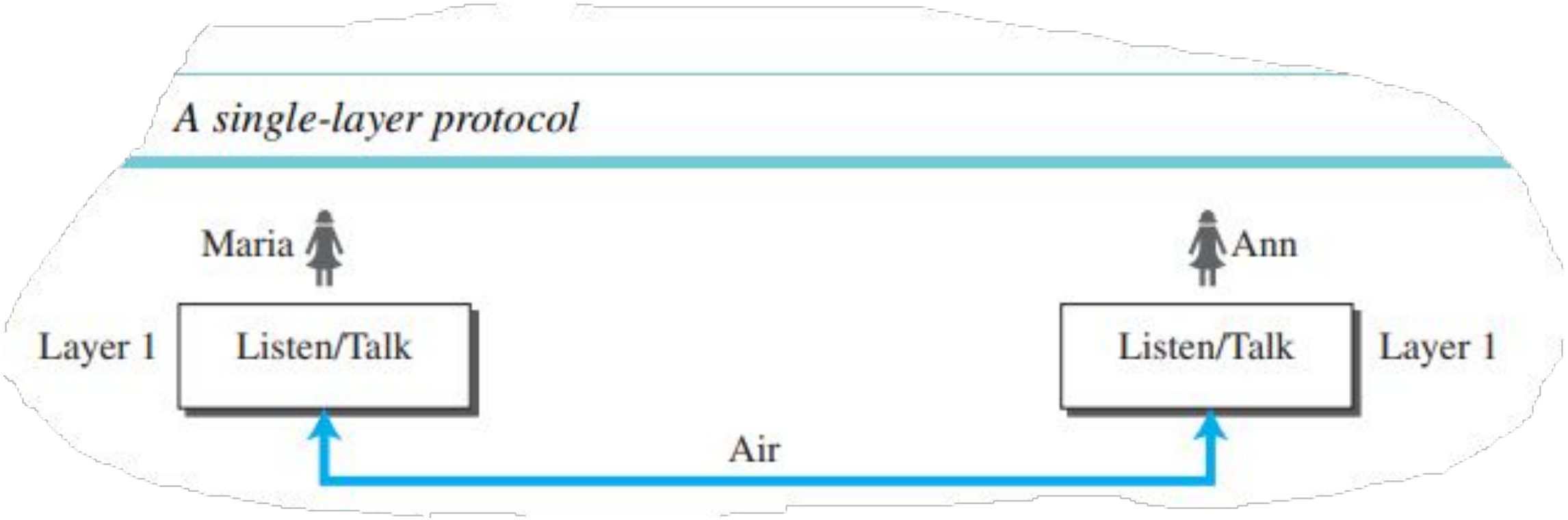
- per-connection end-end throughput: *min(*R_c*,*R_s*,R/10)*
- in practice: R_c or R_s is often bottleneck

PROTOCOLS & LAYERED ARCHITECTURE OF INTERNET

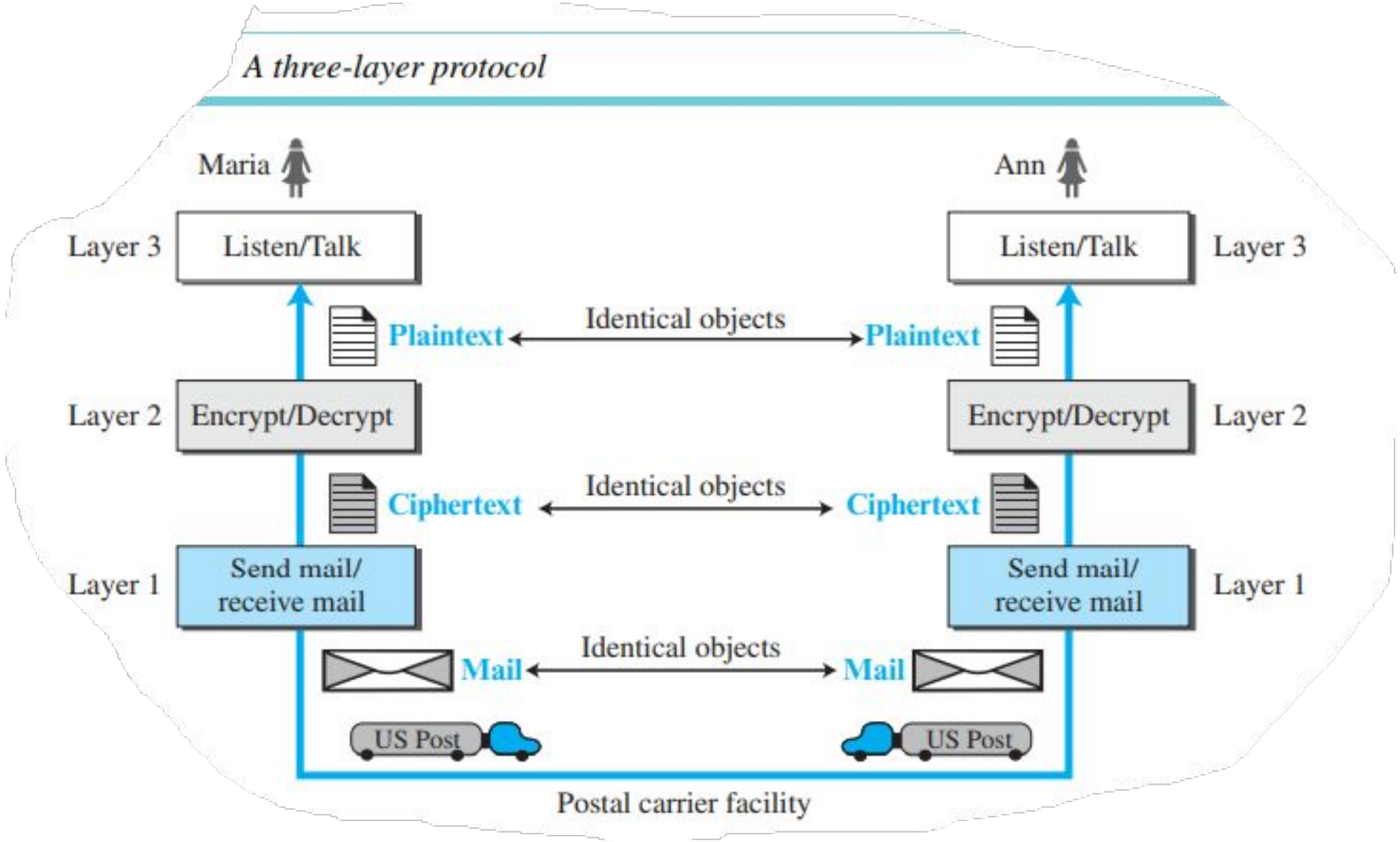
Protocols

- As we have discussed that internet is a complex network containing many hosts, routers, transmission links, etc. So how is this complex structure is managed or operates?
- The answer is Protocols. A protocol defines the rules that both sender and receiver and all intermediate devices need to follow to be able to communicate effectively.
- To reduce the complexity, the internet architecture is divided into protocol layers, where each layer serve a different purpose in the communication between two end-systems in the internet.

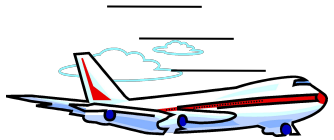
Protocols


A single-layer protocol
Maria
Ann
Layer 1
Listen/Talk
Listen/Talk
Layer 1
Air

Protocols


A three-layer protocol
Maria
Ann
Layer 3
Listen/Talk
Listen/Talk
Layer 3
Plaintext
Identical objects
Plaintext
Layer 2
Encrypt/Decrypt
Encrypt/Decrypt
Layer 2
Ciphertext
Identical objects
Ciphertext
Layer 1
Send mail/
receive mail
Send mail/
receive mail
Layer 1
Mail
Identical objects
Mail
US Post
US Post
Postal carrier facility

Example: organization of air travel



end-to-end transfer of person plus baggage

ticket (purchase)

baggage (check)

gates (load)

runway takeoff

airplane routing

ticket (complain)

baggage (claim)

gates (unload)

runway landing

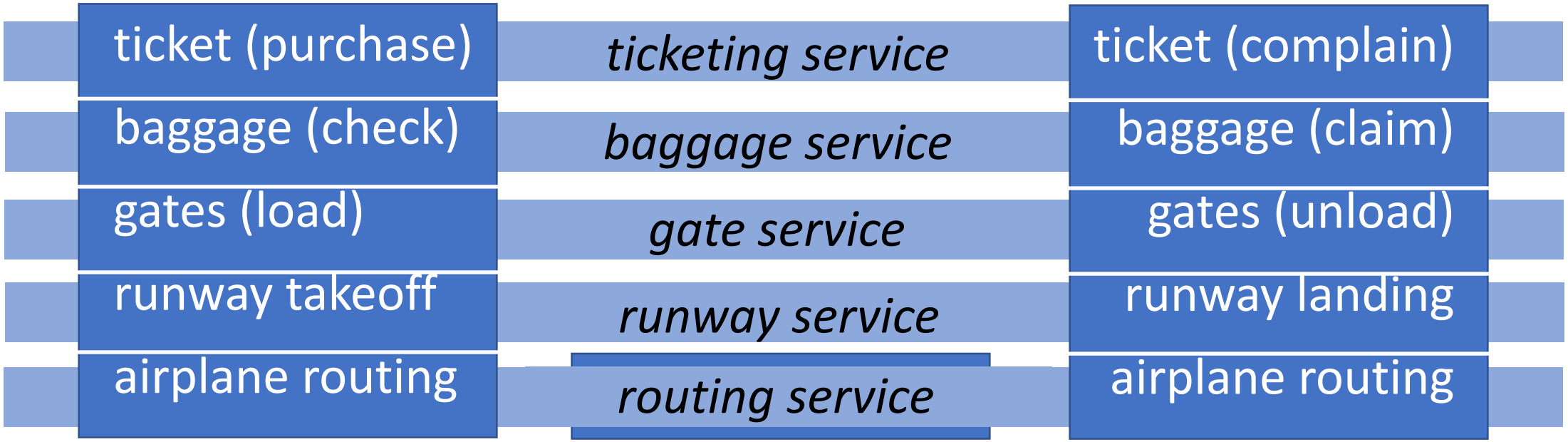
airplane routing

airplane routing

How would you *define/discuss* the *system* of airline travel?

- a series of steps, involving many services

Example: organization of air travel



layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

Why layering?

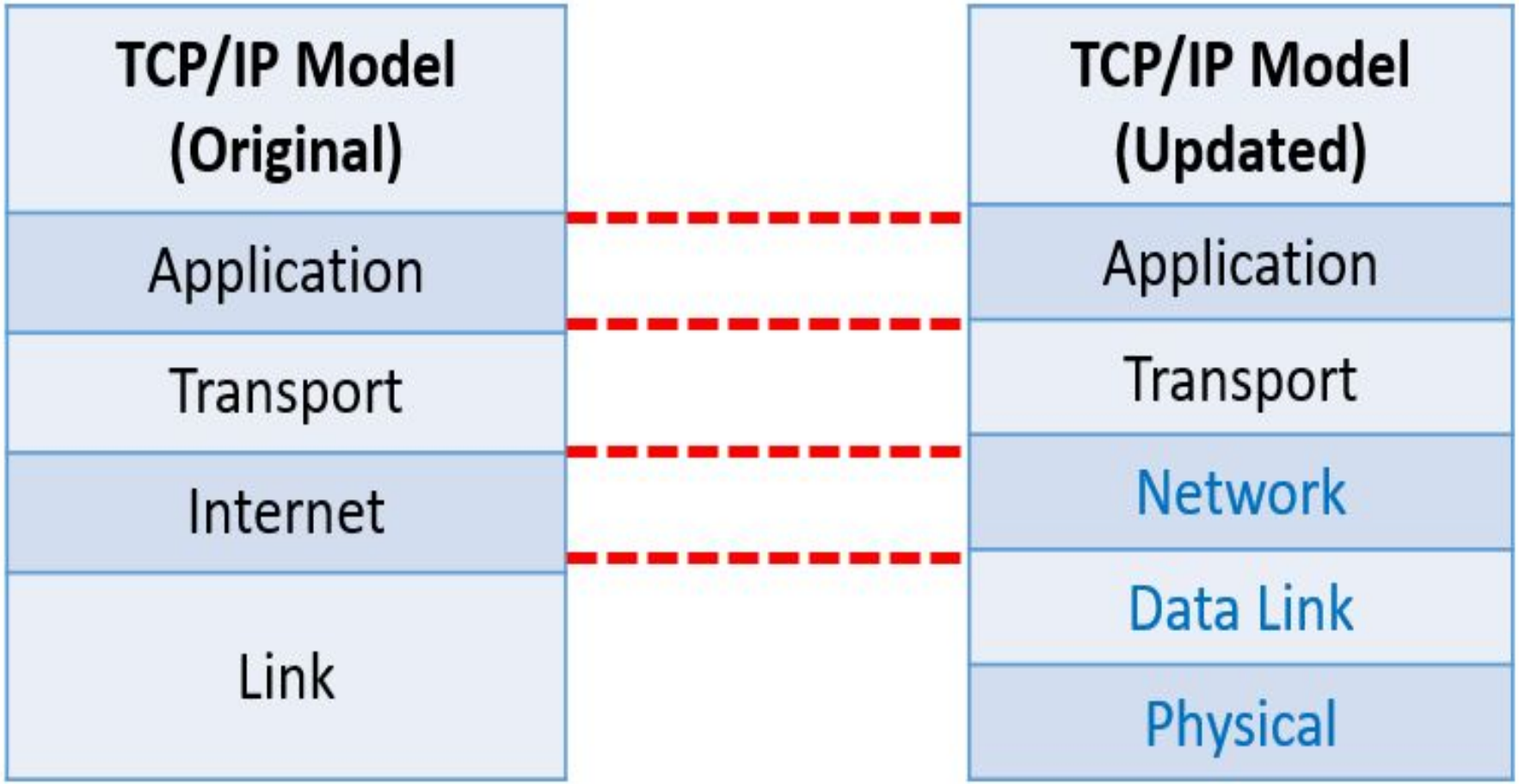
Approach to designing/discussing complex systems:

- explicit structure allows identification, relationship of system's pieces
  - layered *reference model* for discussion
- modularization eases maintenance, updating of system
  - change in layer's service *implementation*: transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system

TCP/IP Protocol Suite

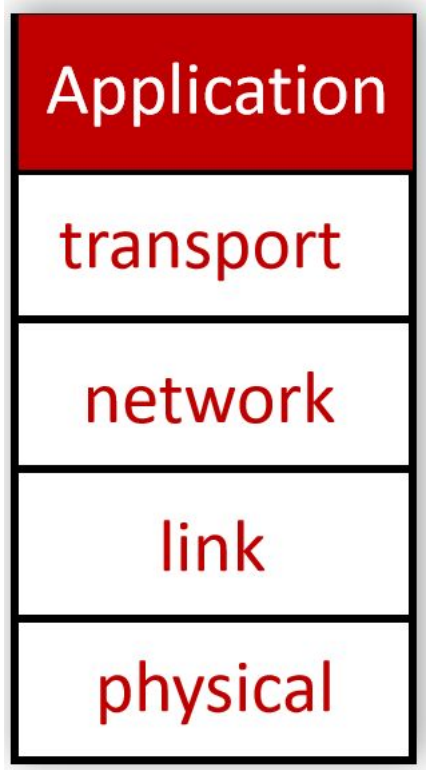
- **TCP/IP** (Transmission Control Protocol / Internet Protocol) is the protocol suite (a set of protocols organized in different layers) which is used in the internet today.
- TCP/IP is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality.
- The term hierarchical means that each upper level protocol is supported by the services provided by one or more lower level protocols.
- The original TCP/IP protocol suite was defined as four software layers built upon the hardware. Today, however, TCP/IP is thought of as a five-layer model.

LAYERS IN TCP/IP PROTOCOL SUITE



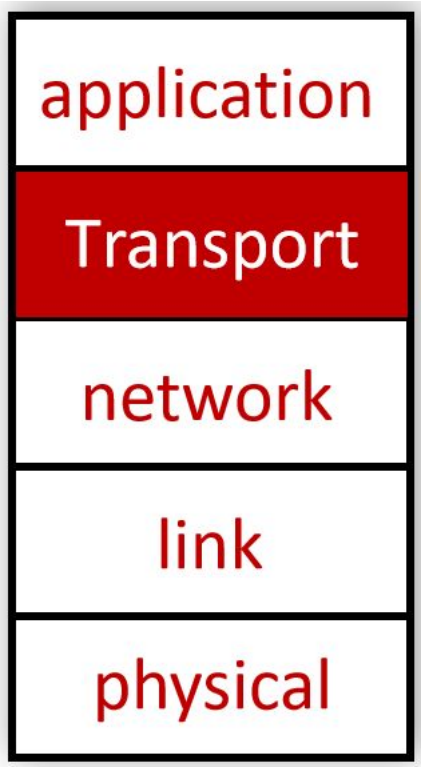
TCP/IP Protocol Suite: Application Layer

- Application layer provides services directly to user applications.
- It handles high-level protocols, data representation, etc.
- Some protocols which reside in application layer are -
  - **HTTP / HTTPS** (HyperText Transfer Protocol) **-** Used in web browsing.
  - **SMTP** (Simple Mail Transfer Protocol) **-** Used in email services.
  - **FTP (**File Transfer Protocol) - Used in transfer of files between two end systems.
  - **DNS (**Domain Name System) - Used for checking IP address of any websites/end systems.



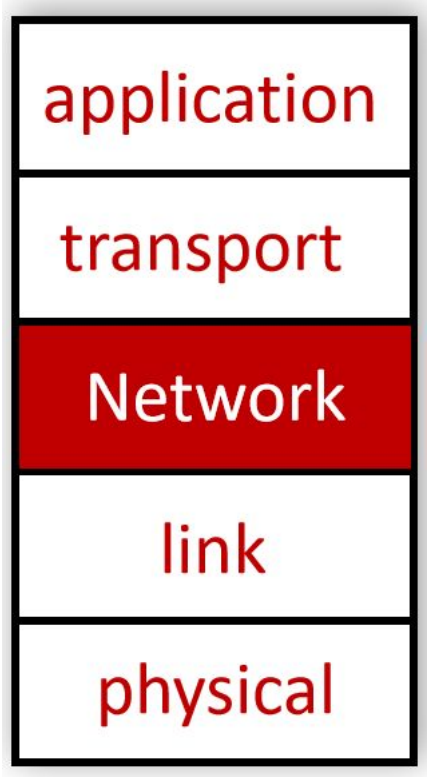
TCP/IP Protocol Suite: Transport Layer

- Transport layer is responsible for the delivery of a message (**segments**) from one process to another.
- There are two transport layer protocols - TCP (Transmission Control Protocol) and UDP (User Datagram Protocol)
- 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 congested.
- 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.



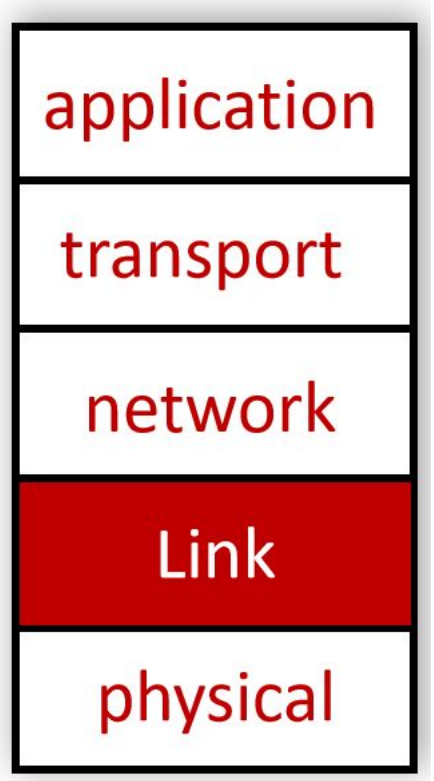
TCP/IP Protocol Suite: Network Layer

- The network layer is responsible for moving network-layer packets known as ***datagrams*** from one host to another.
- Transport layer in at source passes a transport-layer segment and a destination address to the network layer. The network layer then provides the service of delivering the segment to the transport layer of destination.
- The 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 network layer also contains many routing protocols that determine the routes that datagrams take between sources and destinations.



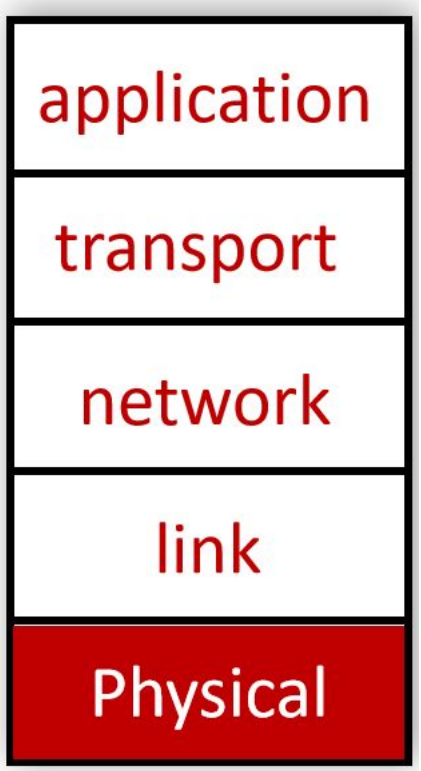
TCP/IP Protocol Suite: Link Layer

- 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.
- The link layer protocols include Ethernet, WiFi, etc.
- Link layer packets are known as **frames**.



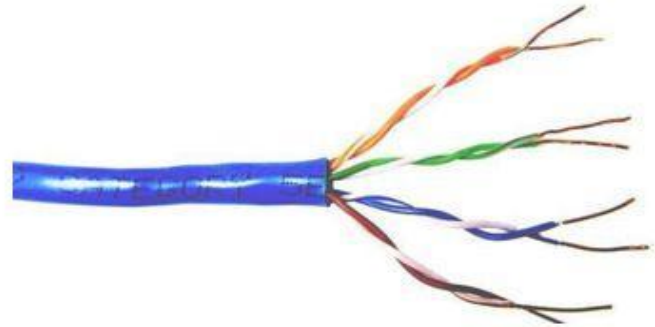
TCP/IP Protocol Suite: Physical Layer

- 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 physical layer depends on the actual tranmission medium of the link (for example, twisted-pair copper wire, single-mode fiber optics).
- For example, Ethernet has many physical-layer protocols - one for twisted-pair copper wire, another for coaxial cable, another for optical fiber, and so on. In each case, a bit is moved across the link in a different way.



Common network cable types

- Unshielded twisted pair (UTP)



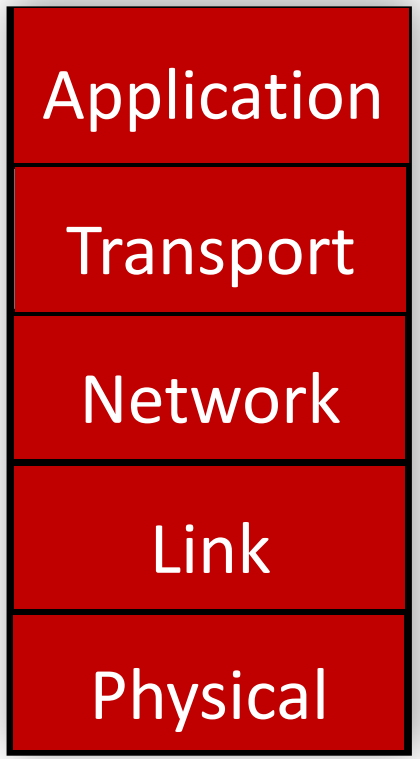
- Shielded twisted pair (STP)

- Coaxial cable

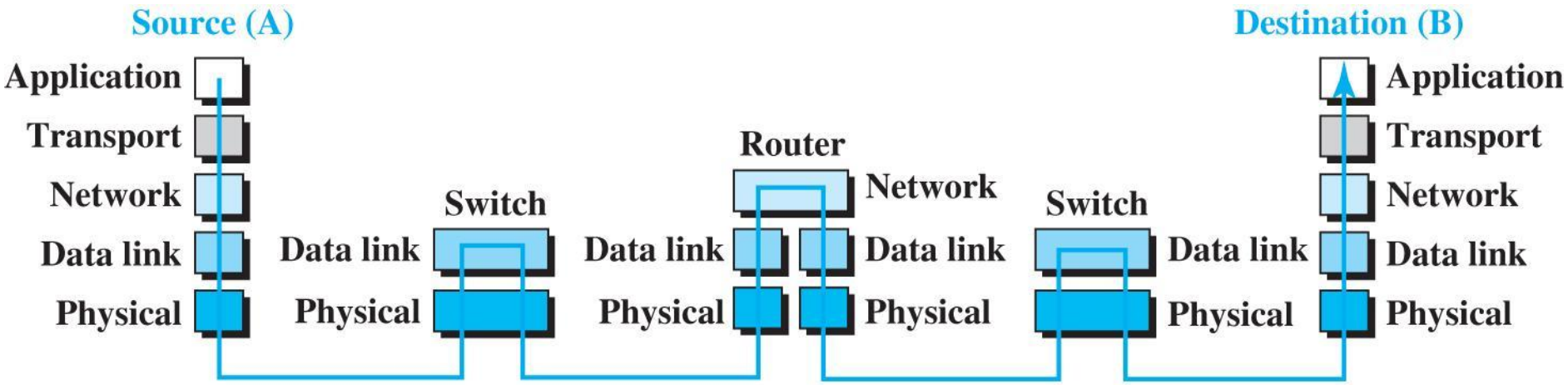
- Fiber optic

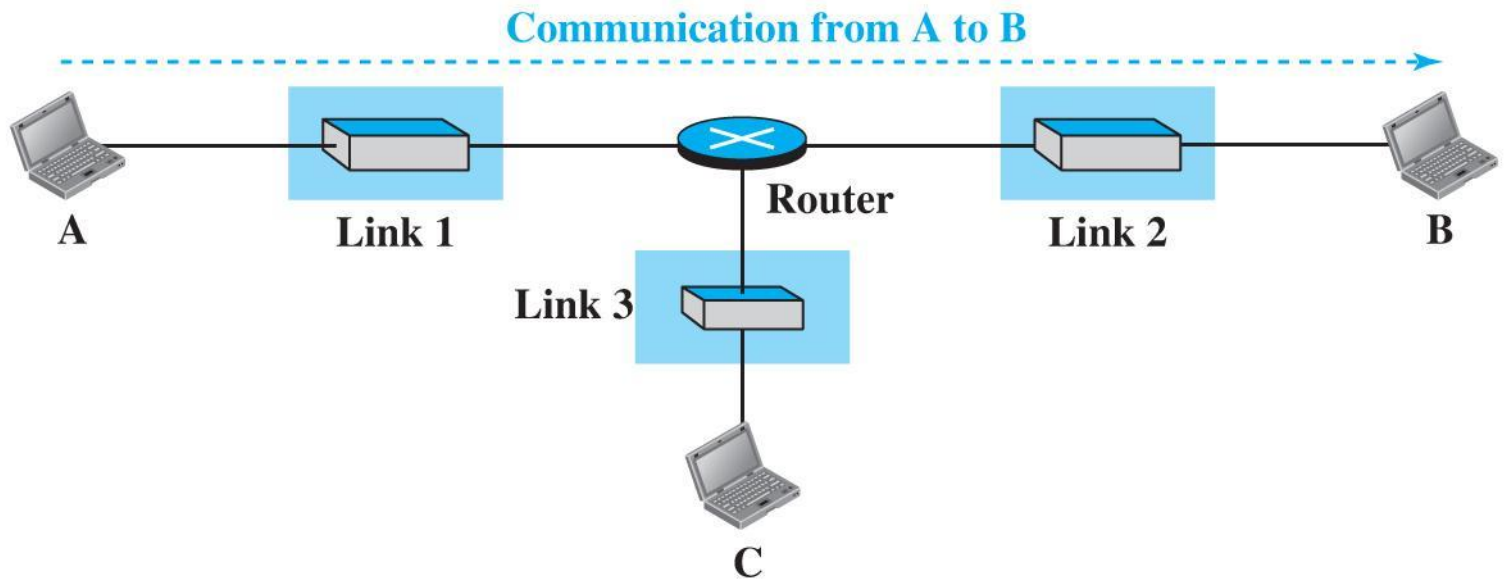
Layers in TCP/IP Protocol Suite

- *Application:* supporting network applications
  - HTTP, FTP, SMTP, DNS
- *Transport:* ensures process to process data transfer
  - TCP, UDP
- *Network:* handles path selection or routing of datagrams from source to destination
  - IP, routing protocols
- *Link:* data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi), PPP
- *Physical:* bits “on the wire”

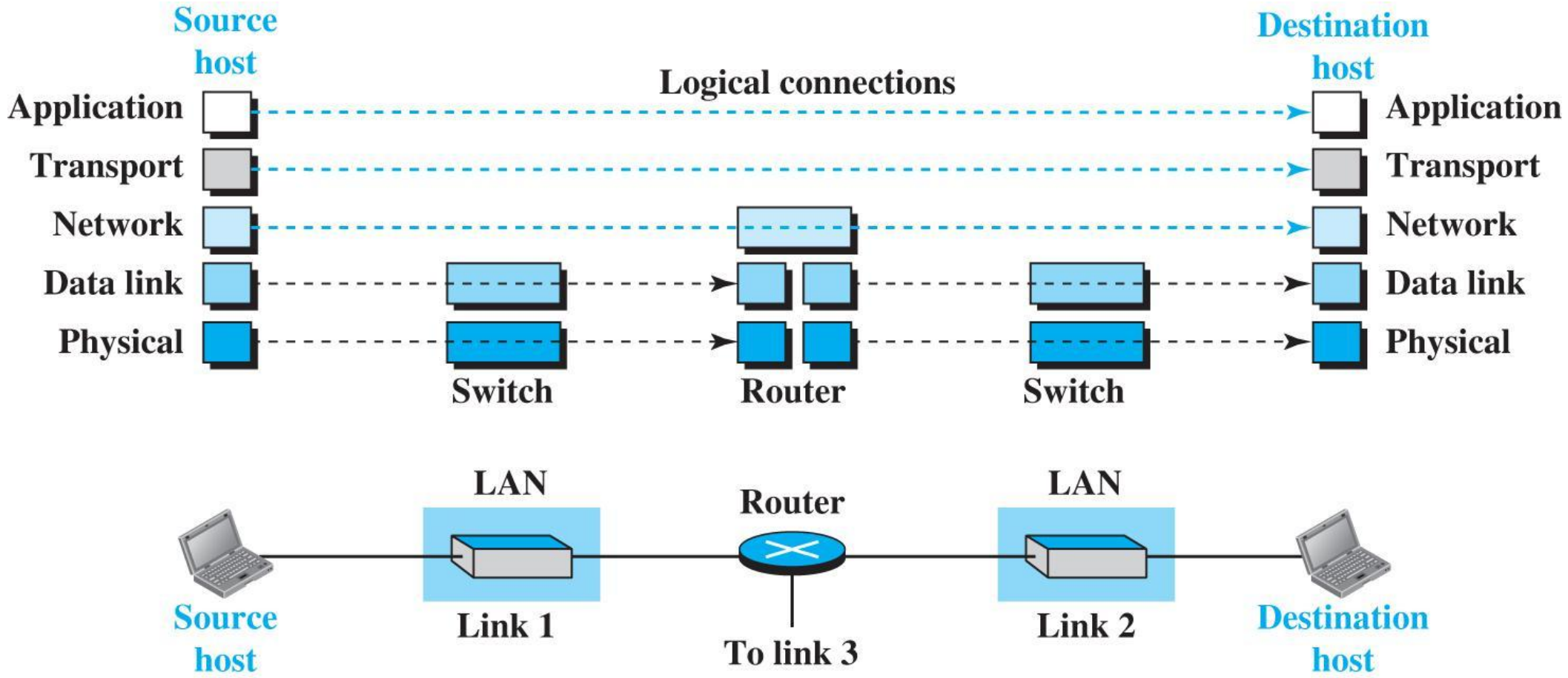


Packet Transmission in TCP/IP Stack

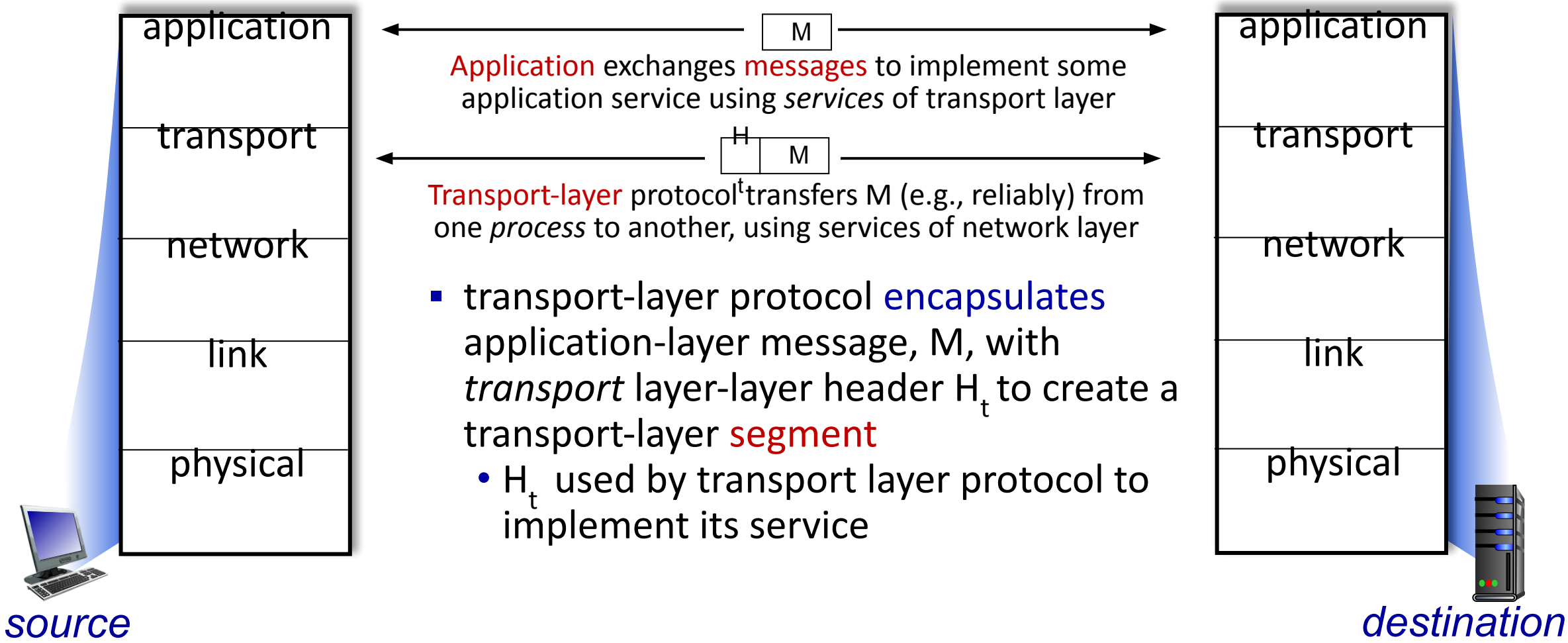




Logical Connections between layers in TCP/IP

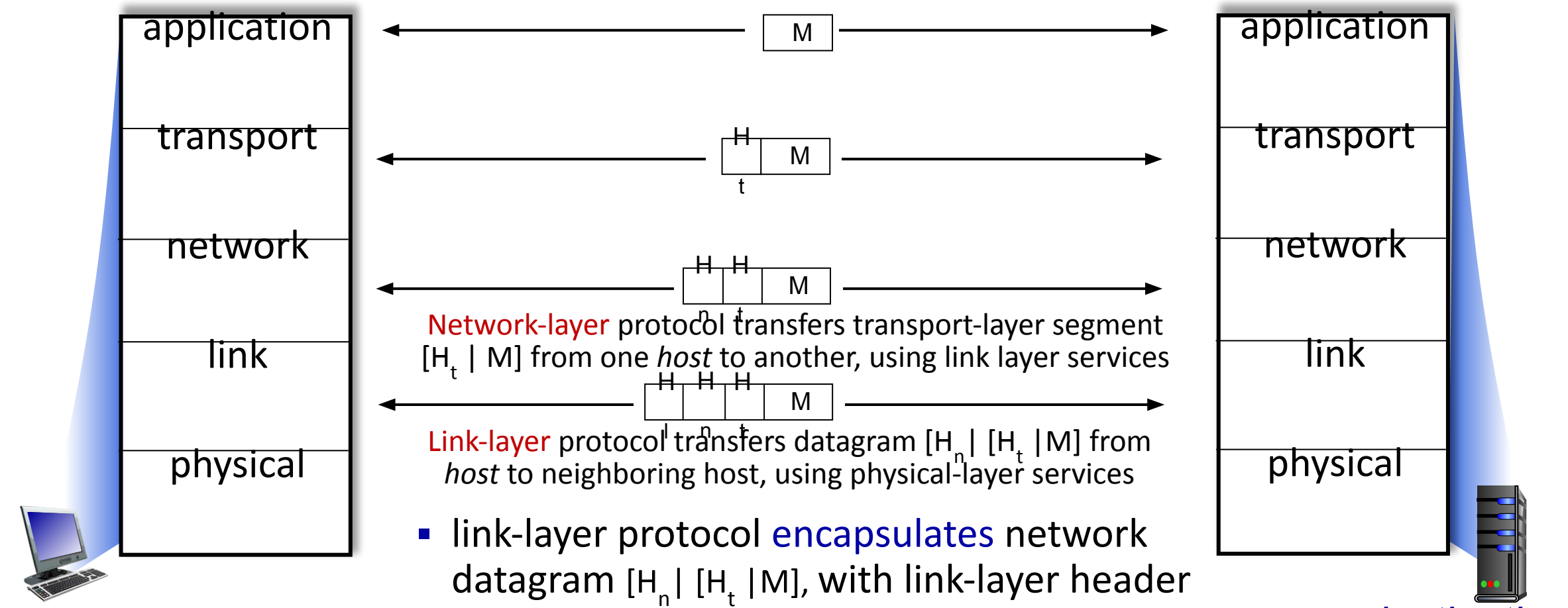


Services, Layering and Encapsulation



- transport-layer protocol encapsulates application-layer message, M, with *transport* layer-layer header H_t to create a transport-layer segment
  - H_t used by transport layer protocol to implement its service

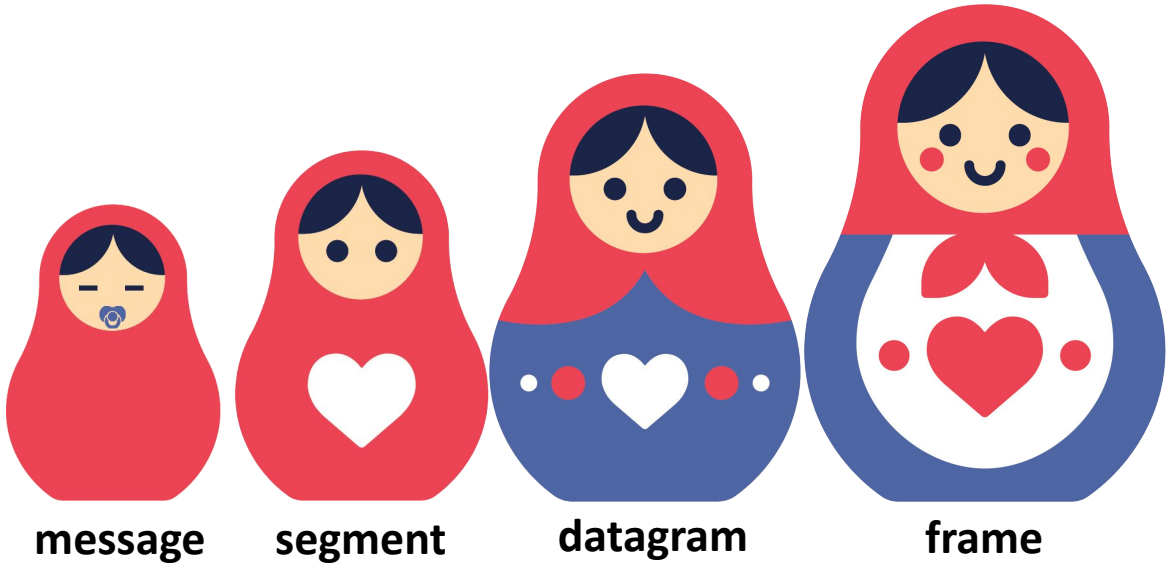
Services, Layering and Encapsulation



- link-layer protocol encapsulates network datagram $[H_n| [H_t |M]$, with link-layer header H_l to create a link-layer frame

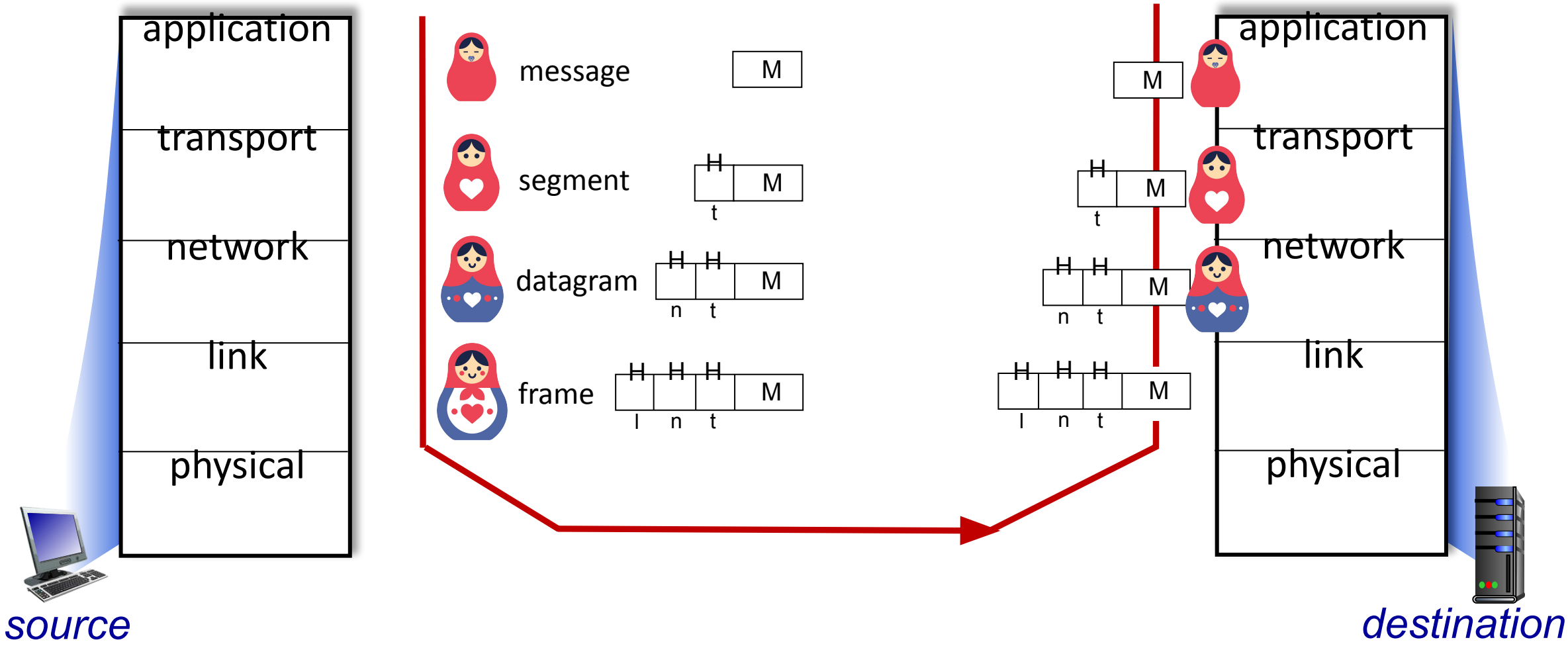
Encapsulation

Matryoshka dolls (stacking dolls)

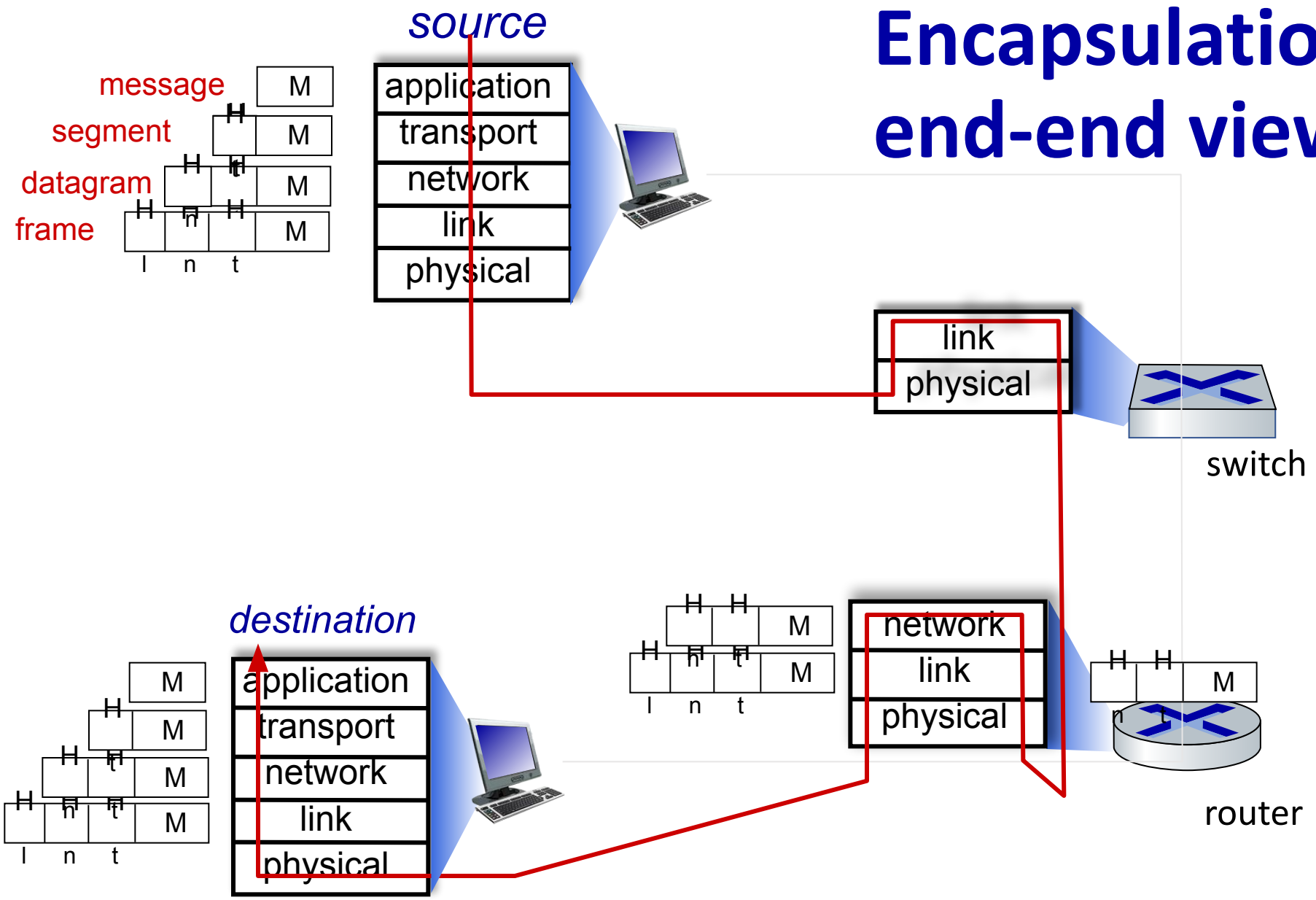


Credit: https://dribbble.com/shots/7182188-Babushka-Boi

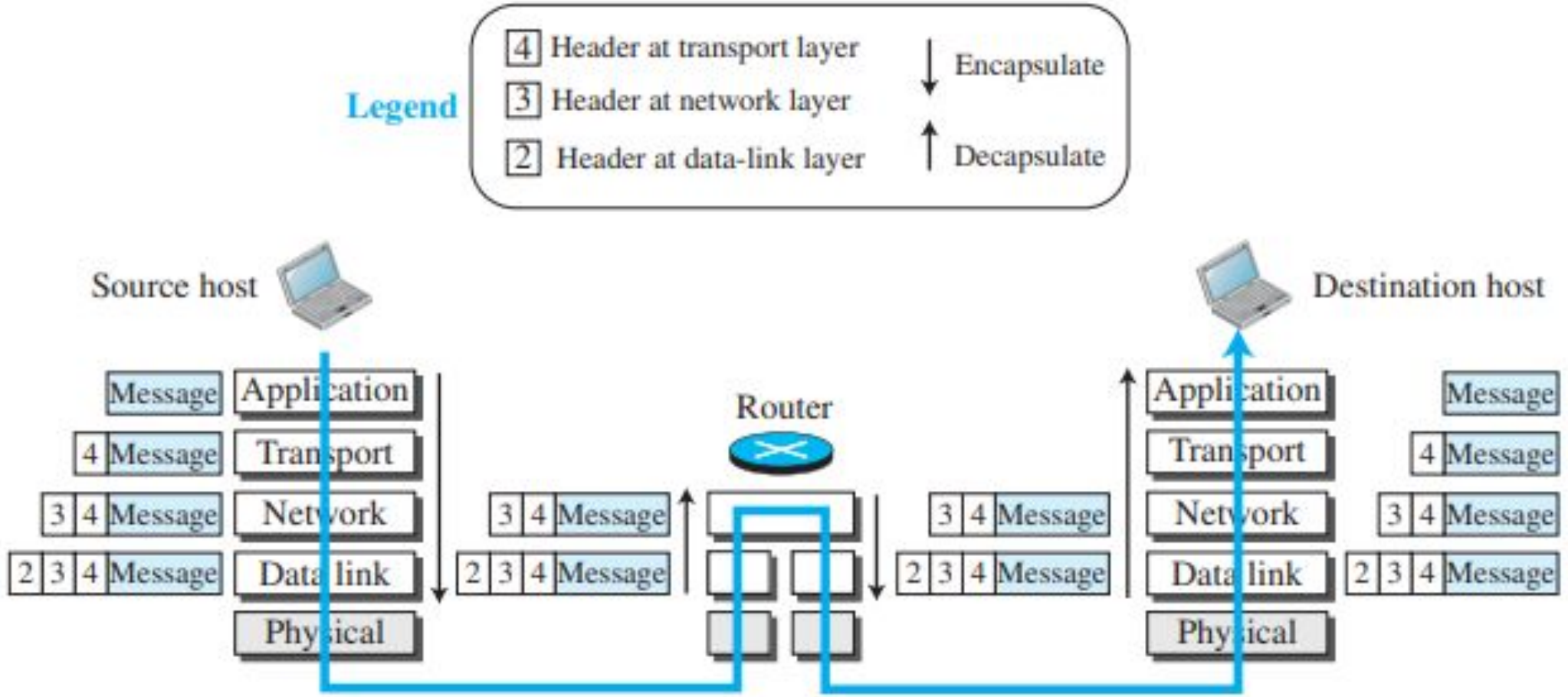
Services, Layering and Encapsulation



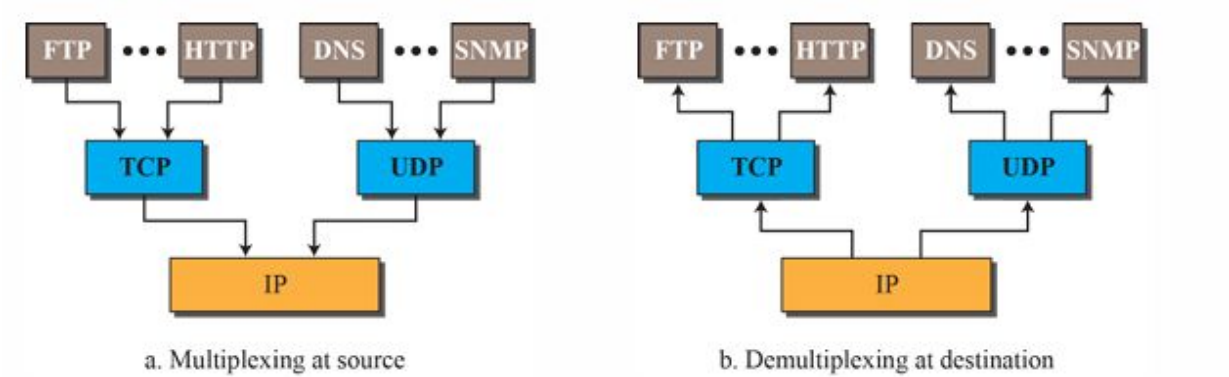
Encapsulation: an end-end view



Encapsulation and Decapsulation



Multiplexing and Demultiplexing

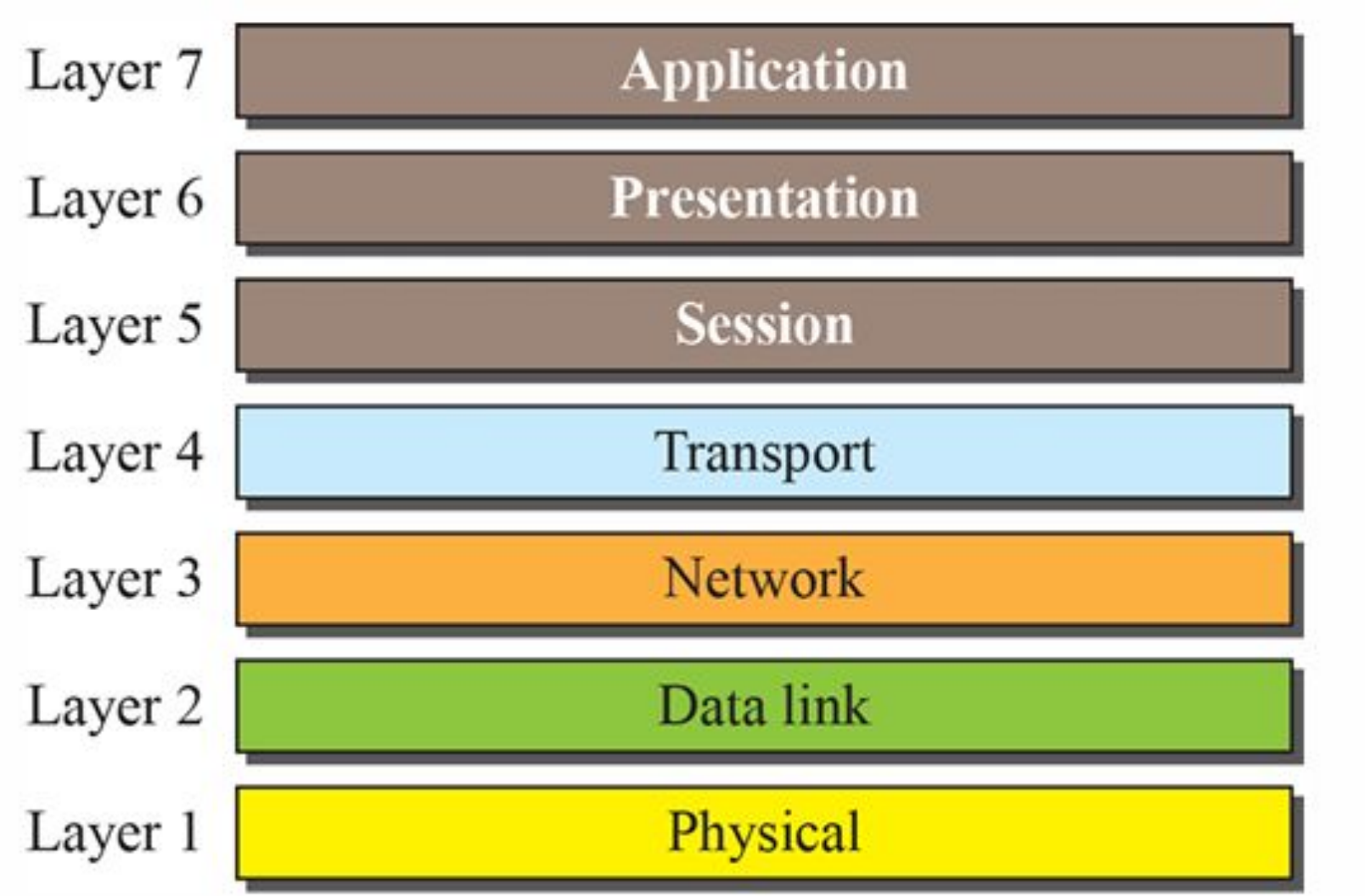


- **Multiplexing -** At the source, a protocol at a layer can take data packets from several next-higher layer protocols one at a time and encapsulate them together. This means that data from different applications can be combined into a single packet.
- **Demultiplexing -** At the destination, the receiving protocol can identify and extract the encapsulated packets from the received data packet and deliver them to their respective next-higher layer protocols one at a time. This ensures that the data is correctly routed to the appropriate application or service.

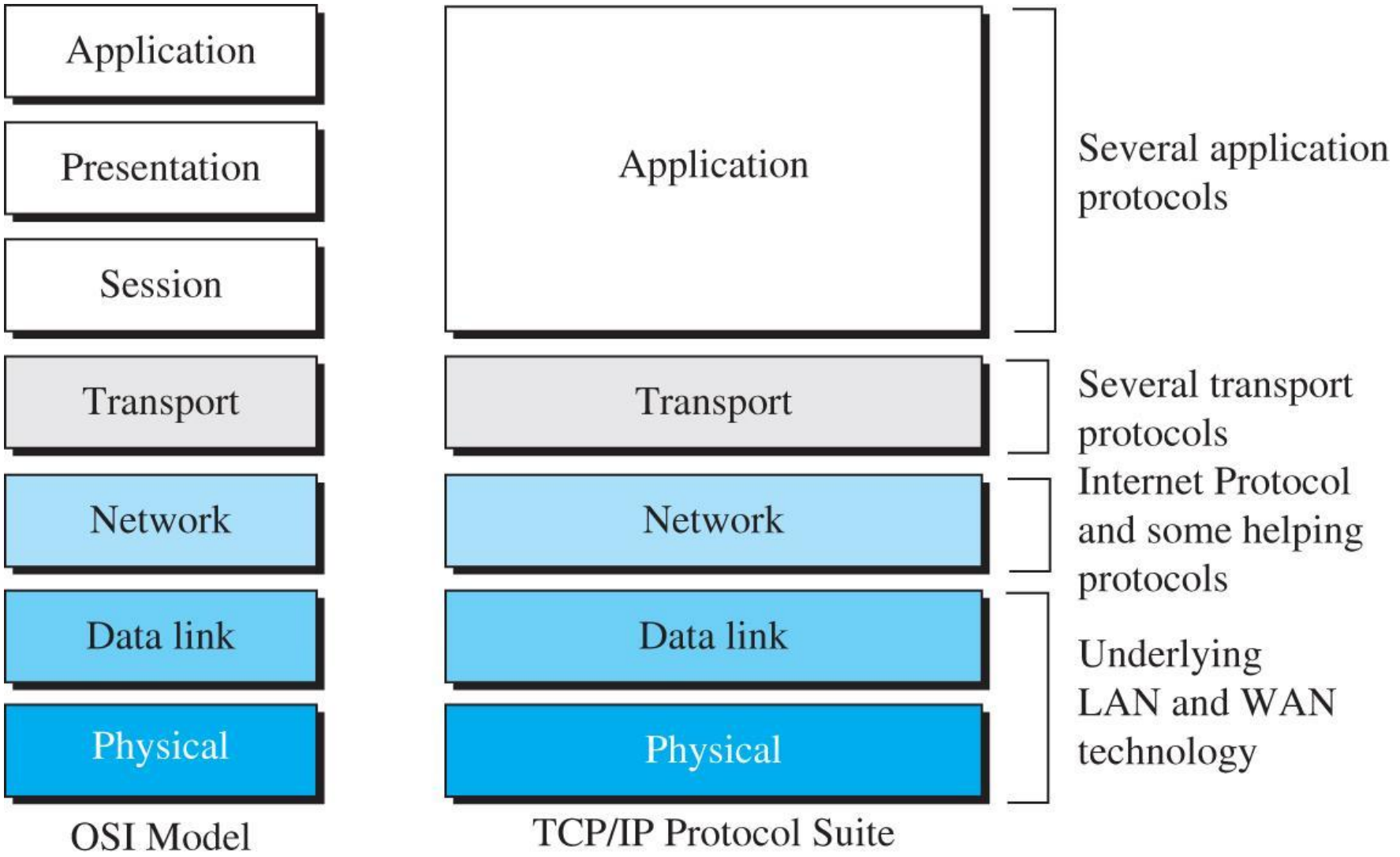
The OSI Model

- Established in 1947, the **International Organization for Standardization (ISO)** is a multinational body dedicated to worldwide agreement on international standards.
- An ISO standard that covers all aspects of network communications is the **Open Systems Interconnection (OSI) model**. It was first introduced in the late 1970s.
- An open system is like a universal translator that enables different systems to talk to each other, regardless of their differences. The OSI model is a blueprint that shows how to make this communication happen without changing the underlying hardware or software.
- The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network.

The OSI Model



OSI Vs TCP/IP


Application
Presentation
Session
Transport
Network
Data link
Physical
OSI Model
Application
Transport
Network
Data link
Physical
TCP/IP Protocol Suite
Several application protocols
Several transport protocols
Internet Protocol and some helping protocols
Underlying LAN and WAN technology

OSI vs TCP/IP

- **Multiple Transport-layer Protocols**: TCP/IP has more than one transport-layer protocol, and some of them provide functionalities that were originally associated with the session layer in the OSI model.
- **Application Layer Variability:** The application layer in TCP/IP is not limited to a single piece of software. Instead, multiple applications can be developed at this layer. If a specific application requires functionalities similar to those in the session and presentation layers, those functionalities can be included in the application's development.
- The absense of session and presentation layers in TCP/IP was a deliberate decision, influenced by the presence of multiple transport layer protocols and the flexibility of the application layer to accommodate various application specific functionalities.

Lack of OSI Model's Success

- **Timing and Cost:** By the time the OSI model was completed, TCP/IP was already widely implemented and used. Switching to the OSI model would have required a significant amount of time, effort, and money to replace the existing TCP/IP infrastructure, which was not practical.
- **Incomplete Definitions:** Some layers in the OSI model, like the presentation and session layers, were not fully defined with concrete protocols and software. Although the services provided by these layers were listed, the actual implementation details were lacking, making it difficult for organizations to adopt them.
- **Performance Issues:** When organizations attempted to implement the OSI model in different applications, it didn't demonstrate significantly better performance than the established TCP/IP protocol suite. As a result, there was no compelling reason for the Internet authority to switch from TCP/IP to OSI.