Module 1
Introduction

What's a Distributed System?

A distributed system is a collection of independent computers that appear to the users of the system as a single computer

- Example:
  - a network of workstations allocated to users
  - a pool of processors in the machine room allocated dynamically
  - a single file system (all users access files with the same path name)
  - user command executed in the best place (user workstation, a workstation belonging to someone else, or on an unassigned processor in the machine room)

Why Distributed?

Economics	Microprocessors offer a better price/ performance than mainframes
Speed	A distributed system may have more total computing power than a mainframe
Inherent distribution	Some applications involve spatially separated machines
Reliability	If one machine crashes, the system as a whole can still survive
Incremental growth	Computing power can be added in small increments

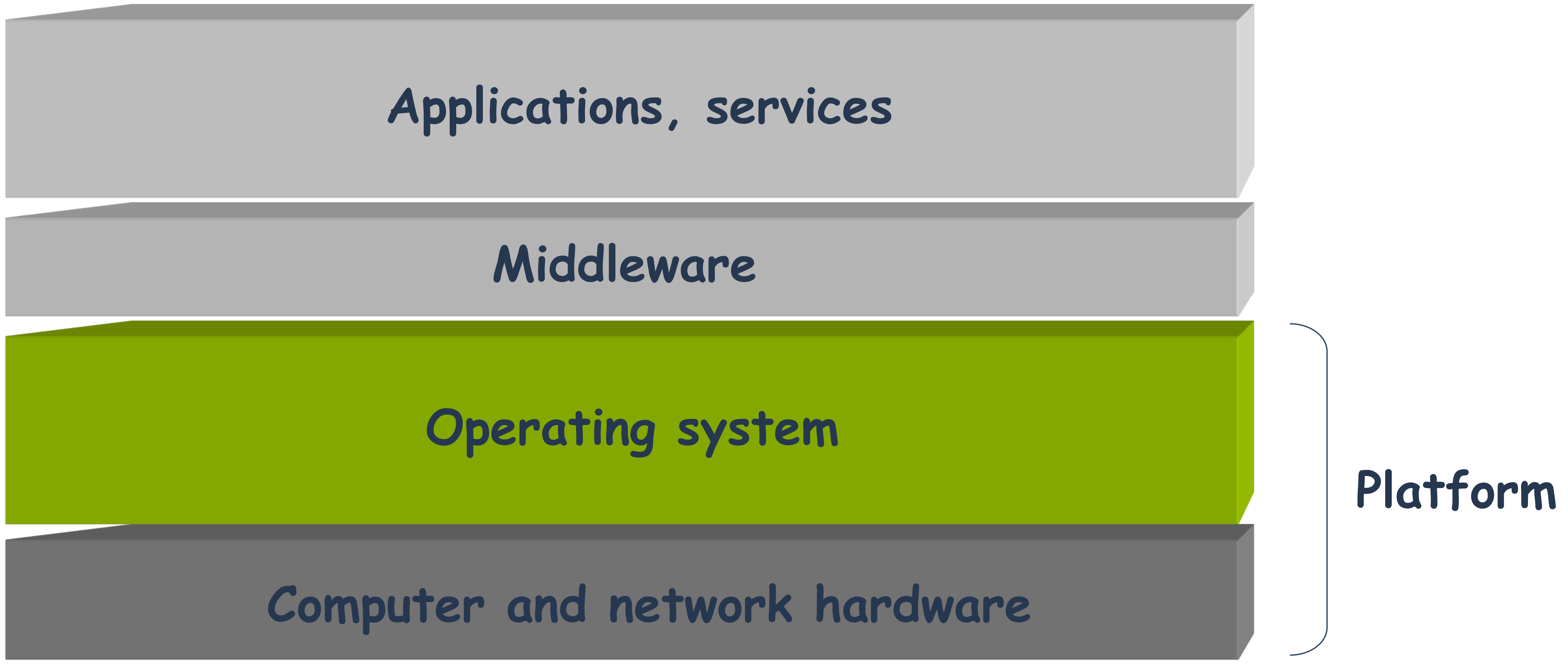
Primary Features

- Multiple computers
  - ➡ Concurrent execution
  - ➡ Independent operation and failures
- Communications
  - ➡ Ability to communicate
  - ➡ No tight synchronization (no global clock)
- "Virtual" Computer
  - ➡ Transparency

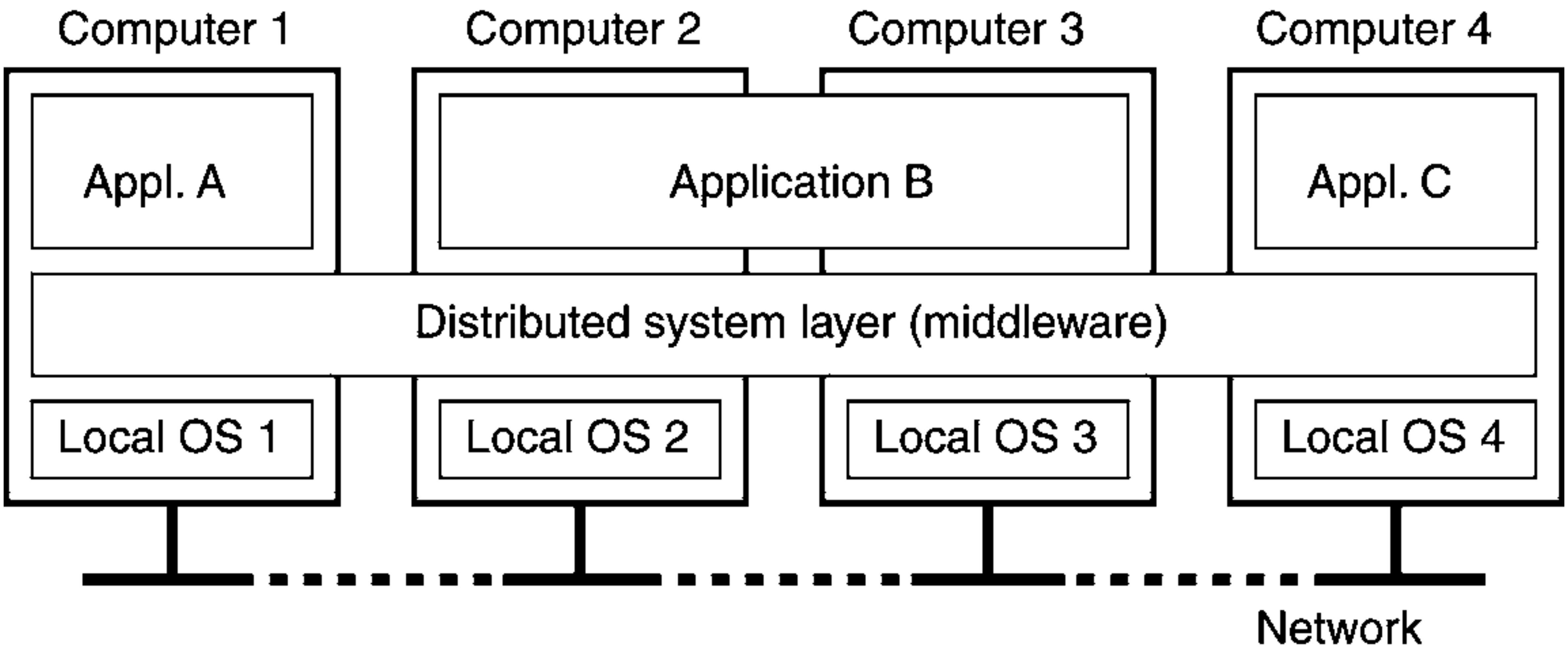
Types of Transparency

Access	local and remote resources are accessed using identical operations
Location	resources are accessed without knowledge of their location
Concurrency	several processes operate concurrently using shared resources without interference between them
Replication	multiple instances of resources appeare as a single instance
Failure	the concealment of faults from users
Mobility	the movement of resources and clients within a system (also called migration transparency)
Performance	the system can be reconfigured to improve performance
Scaling	the system and applications can expand in scale without change to the system structure or the application algorithms

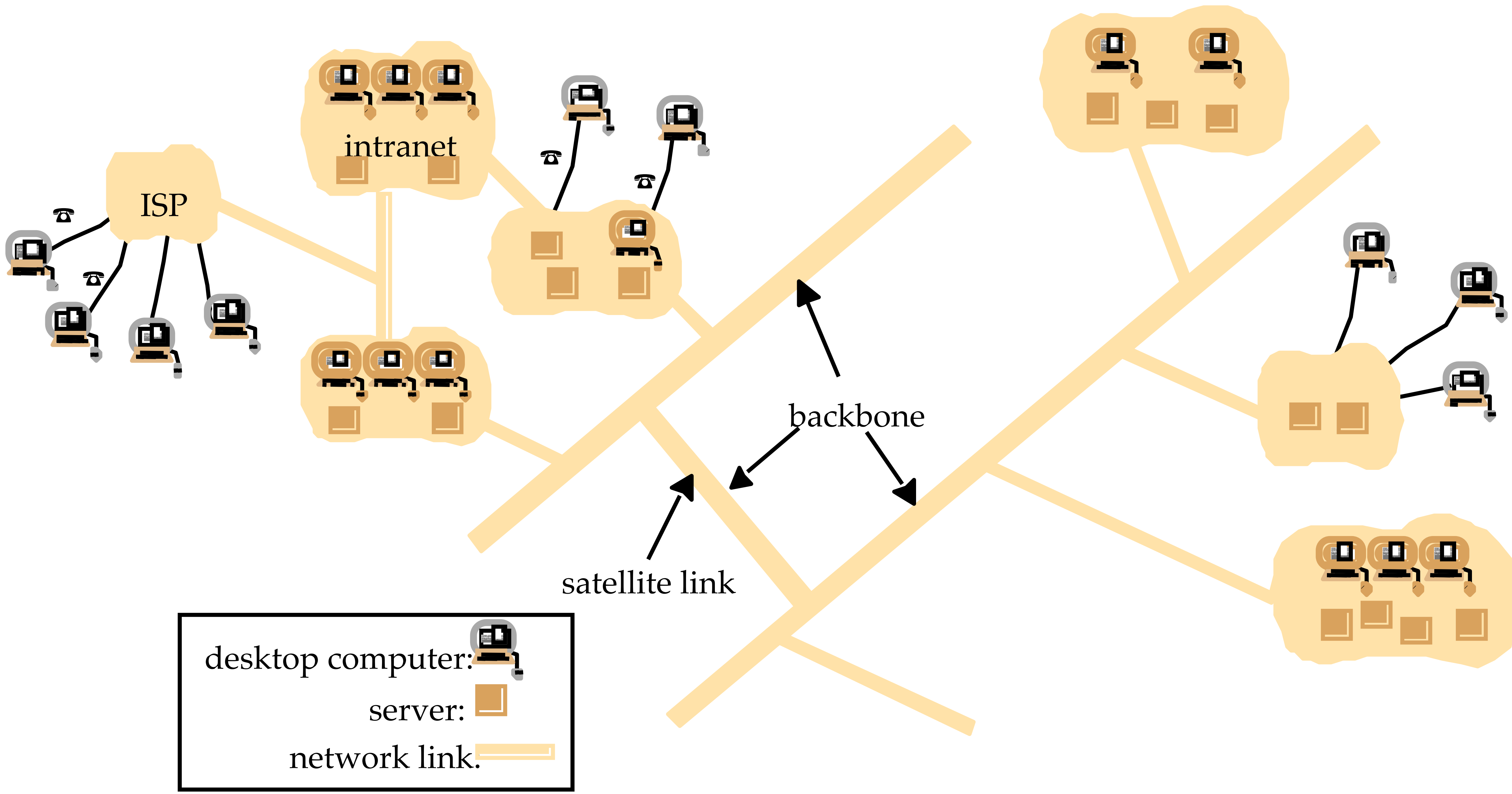
Typical Layering in DSs



Distributed System as Middleware

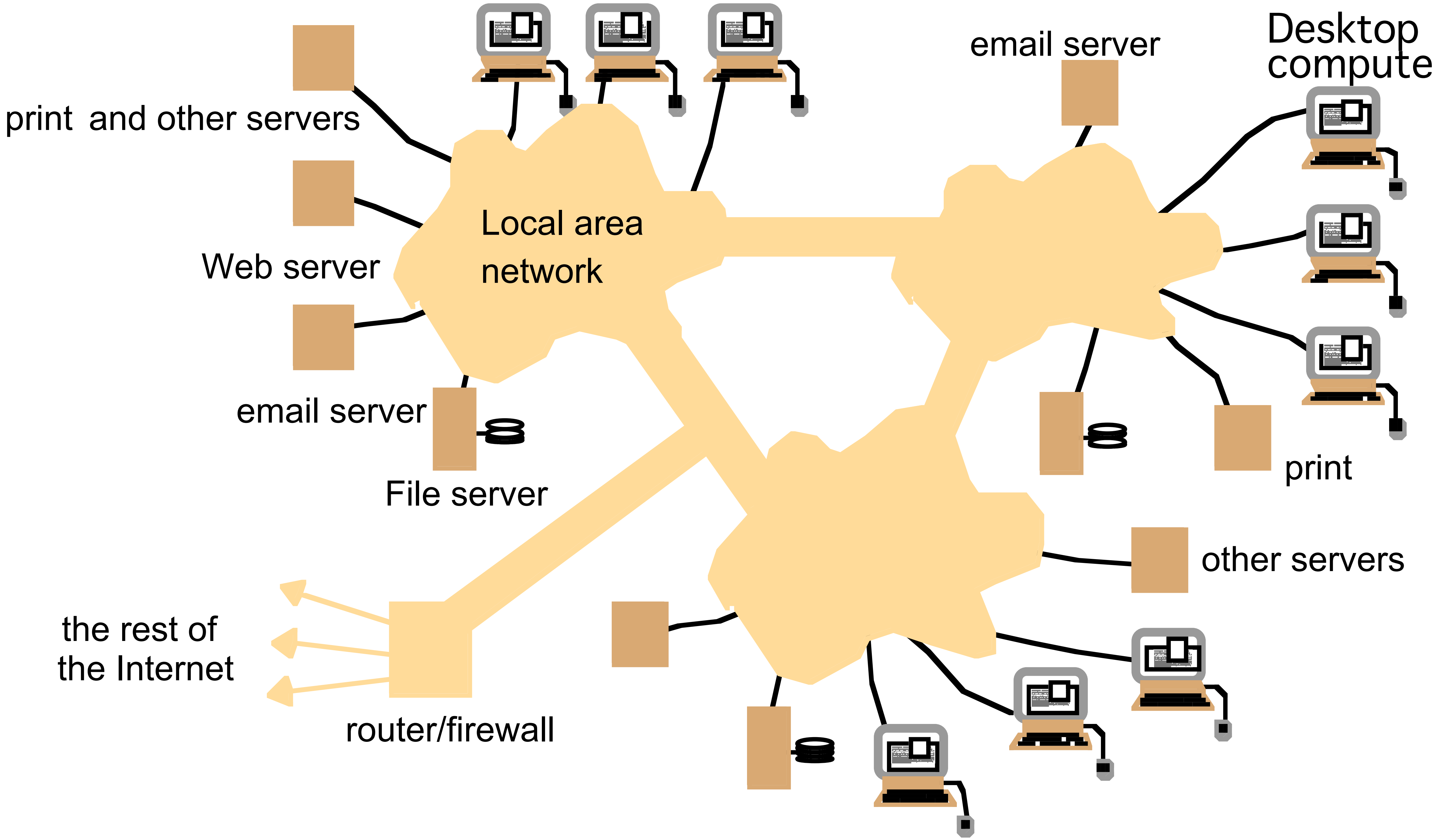


Example – Internet (Portion of it)



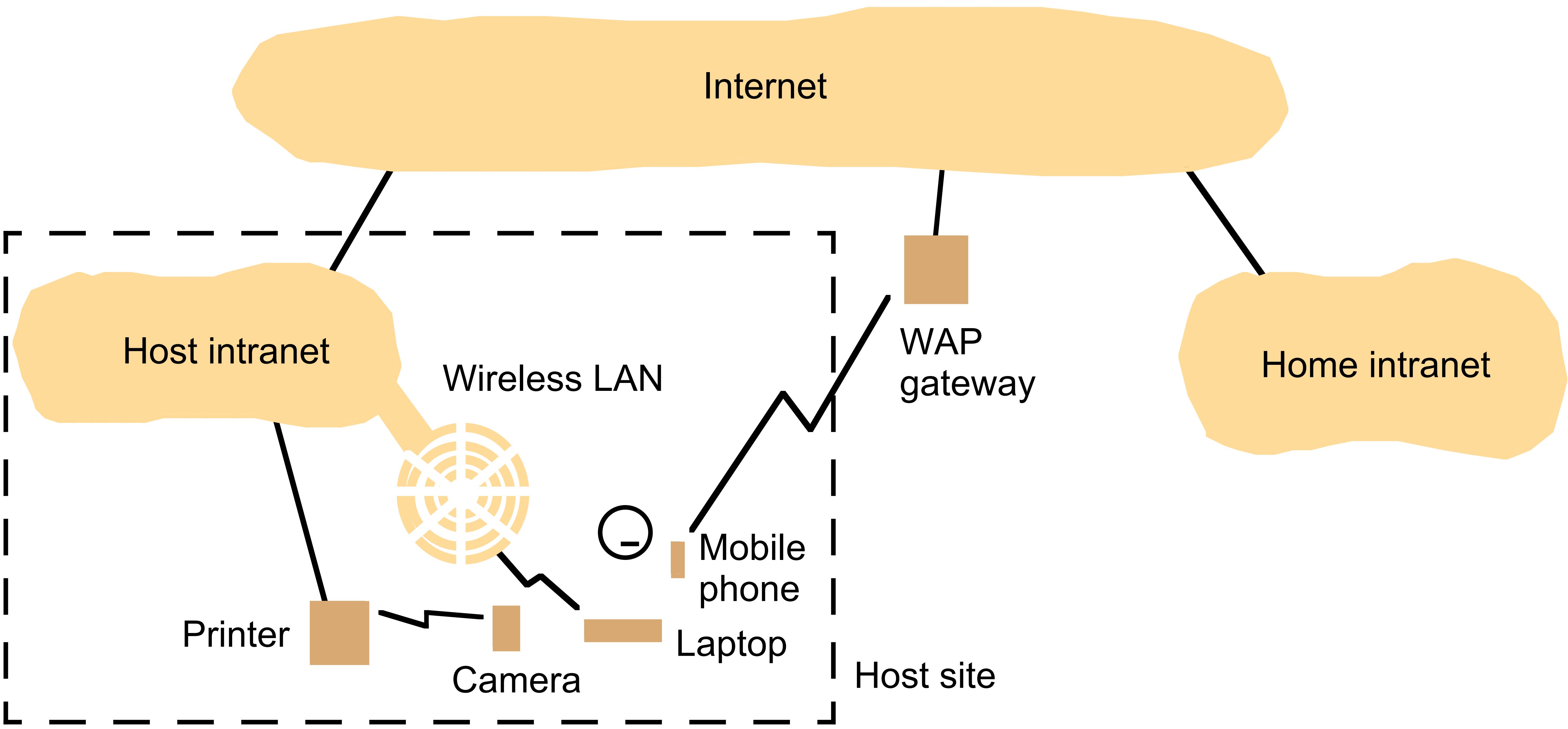
From Coulouris, Dollimore and Kindberg, Distributed Systems: Concepts and Design, 3rd ed.

Example – An Intranet



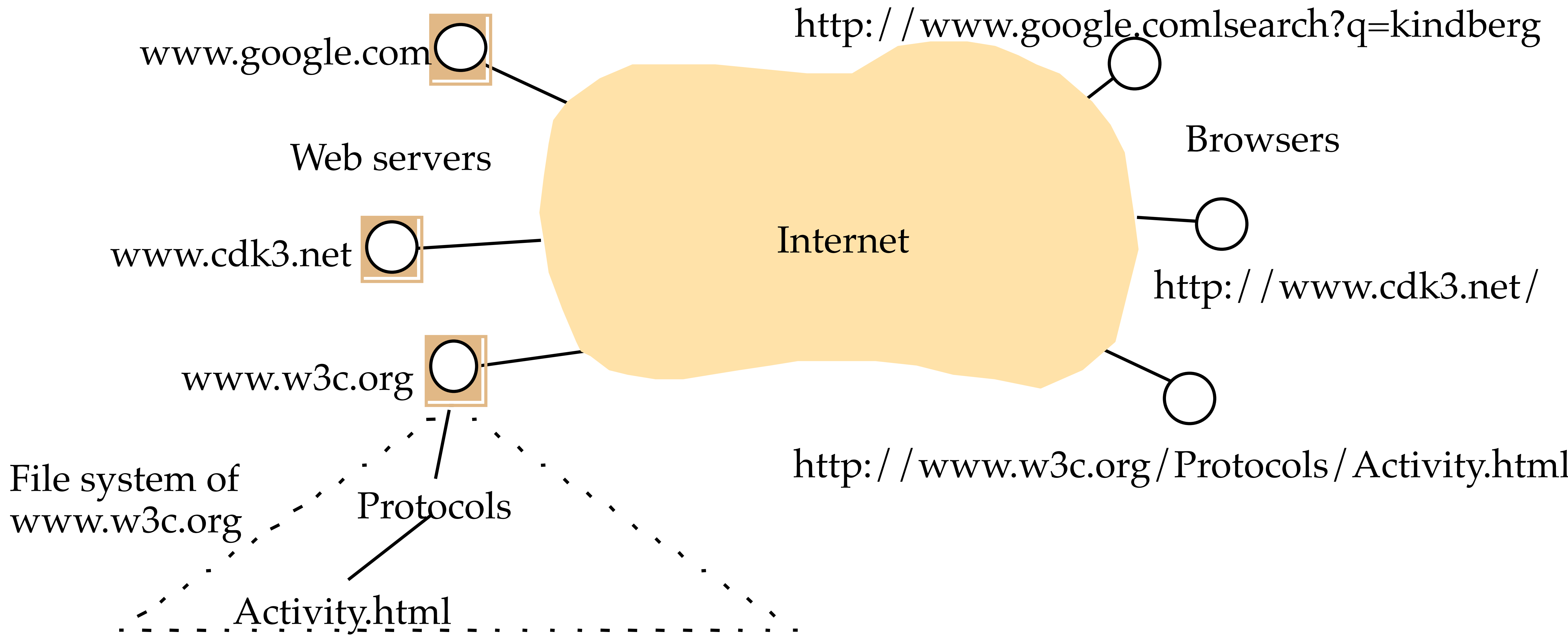
From Couloris, Dollimore and Kindberg, Distributed Systems: Concepts and Design, 3rd ed.

Example – Mobile Environment



From Coulouris, Dollimore and Kindberg, Distributed Systems: Concepts and Design, 3rd ed.

Example - Web



www.google.com
http://www.google.comlsearch?q=kindberg
Web servers
Browsers
www.cdk3.net
Internet
http://www.cdk3.net/
www.w3c.org
File system of
www.w3c.org
Protocols
http://www.w3c.org/Protocols/Activity.html
Activity.html

Challenges (I)

- Heterogeneity
  - ➡ Networks
  - ➡ Hardware
  - ➡ OS
  - ➡ Programming Languages
  - ➡ Implementations
- Openness
  - ➡ Can you extend and re-implement the system?
  - ➡ Public and published interfaces
  - ➡ Uniform communication mechanism

Challenges (II)

- Scalability
  - ➡ Can the system behave properly if the number of users and components increase?
    - ✦ Scale-up: increase the number of “users” while keeping the system performance unaffected
    - ✦ Speed-up: improvement in the system performance as system resources are increased
  - ➡ Impediments
    - ✦ Centralized data
      - ✓ A single file
    - ✦ Centralized services
      - ✓ A single server
    - ✦ Centralized algorithms
      - ✓ Algorithms that “know it all”

Challenges (III)

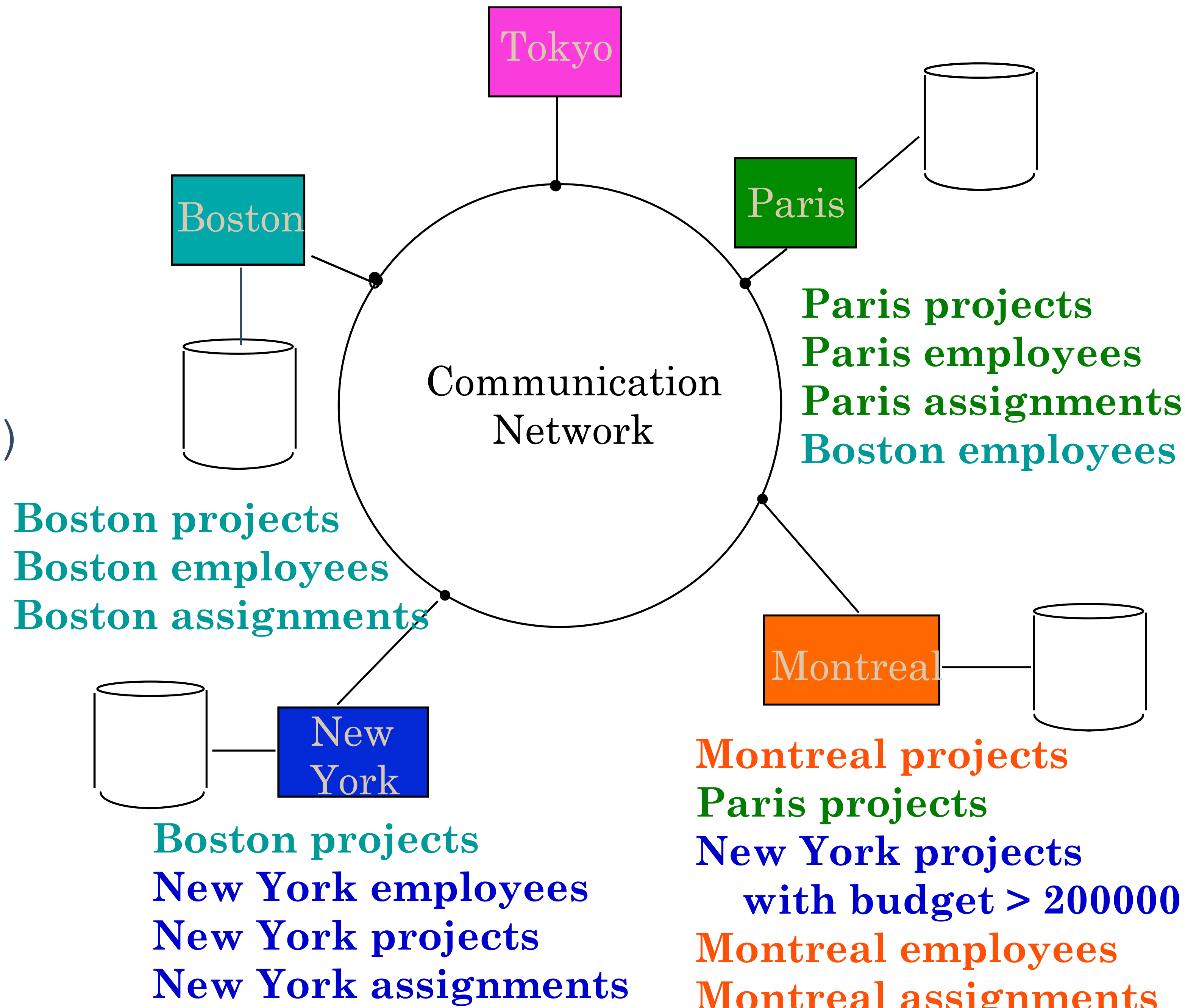
- Failure handling
  - Partial failures
    - Can non-failed components continue operation?
    - Can the failed components easily recover?
  - Failure detection
  - Failure masking
  - Failure tolerance
  - Recovery
  - An important technique is replication
    - Why does the space shuttle has 3 on-board computers?

Challenges (IV)

- Concurrency
  - ➡ Multiple "clients" sharing a resource ➔how to you maintain integrity of the resource and proper operation (without interference) of these clients?
  - ➡ Example: bank account
    - ✦ Assume your account has a balance of $100
    - ✦ You are depositing from an ATM a cheque for $50 into that account
    - ✦ Someone else is withdrawing from the same account $30 using another ATM but at the same time
    - ✦ What should the final balance of the account be?
      - ✓ $120
      - ✓ $70
      - ✓ $150

Challenges (V)

- Transparency
  - ➡ Not easy to maintain
  - ➡ Example:



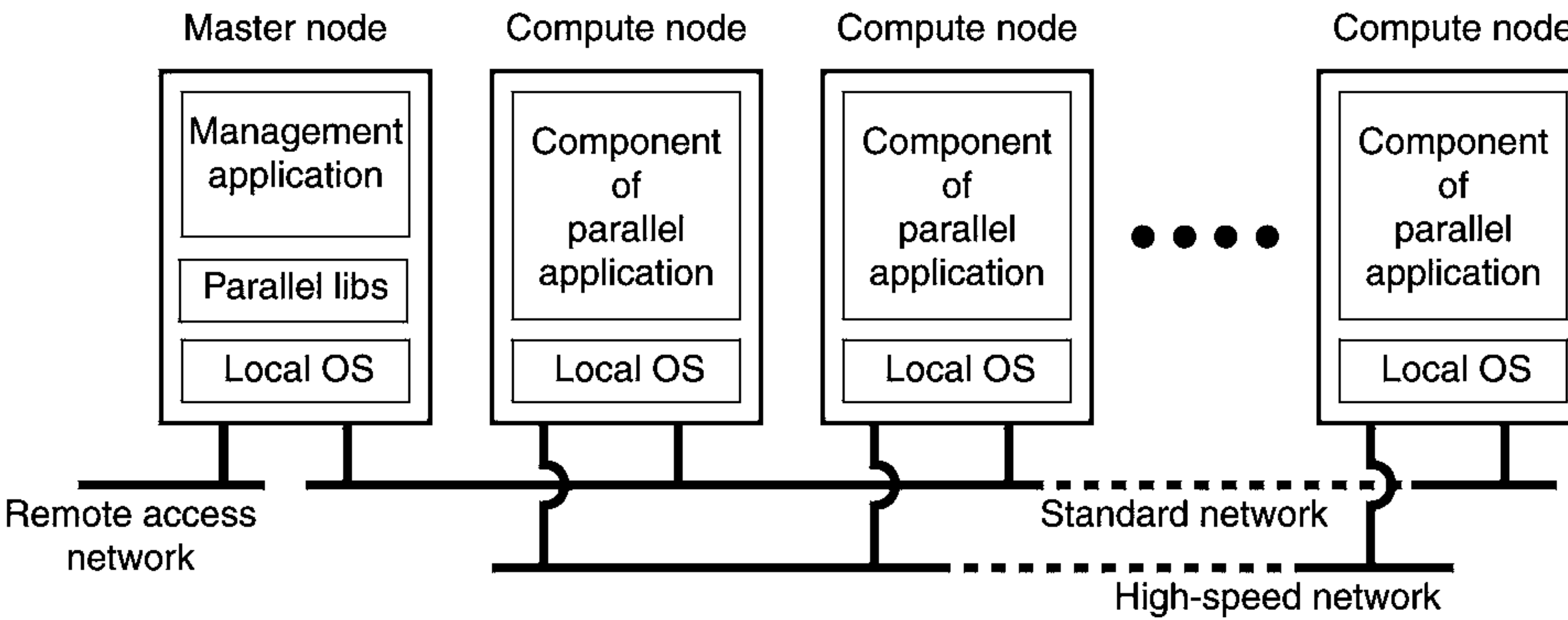

```
EMP(ENO,ENAME,TITLE,LOC)
PROJECT(PNO,PNAME,LOC)
PAY(TITLE,SAL)
ASG(ENO,PNO,DUR)

SELECT  ENAME,SAL
FROM    EMP,ASG,PAY
WHERE   DUR > 12
AND     EMP.ENO = ASG.ENO
AND     PAY.TITLE = EMP.TITLE
```

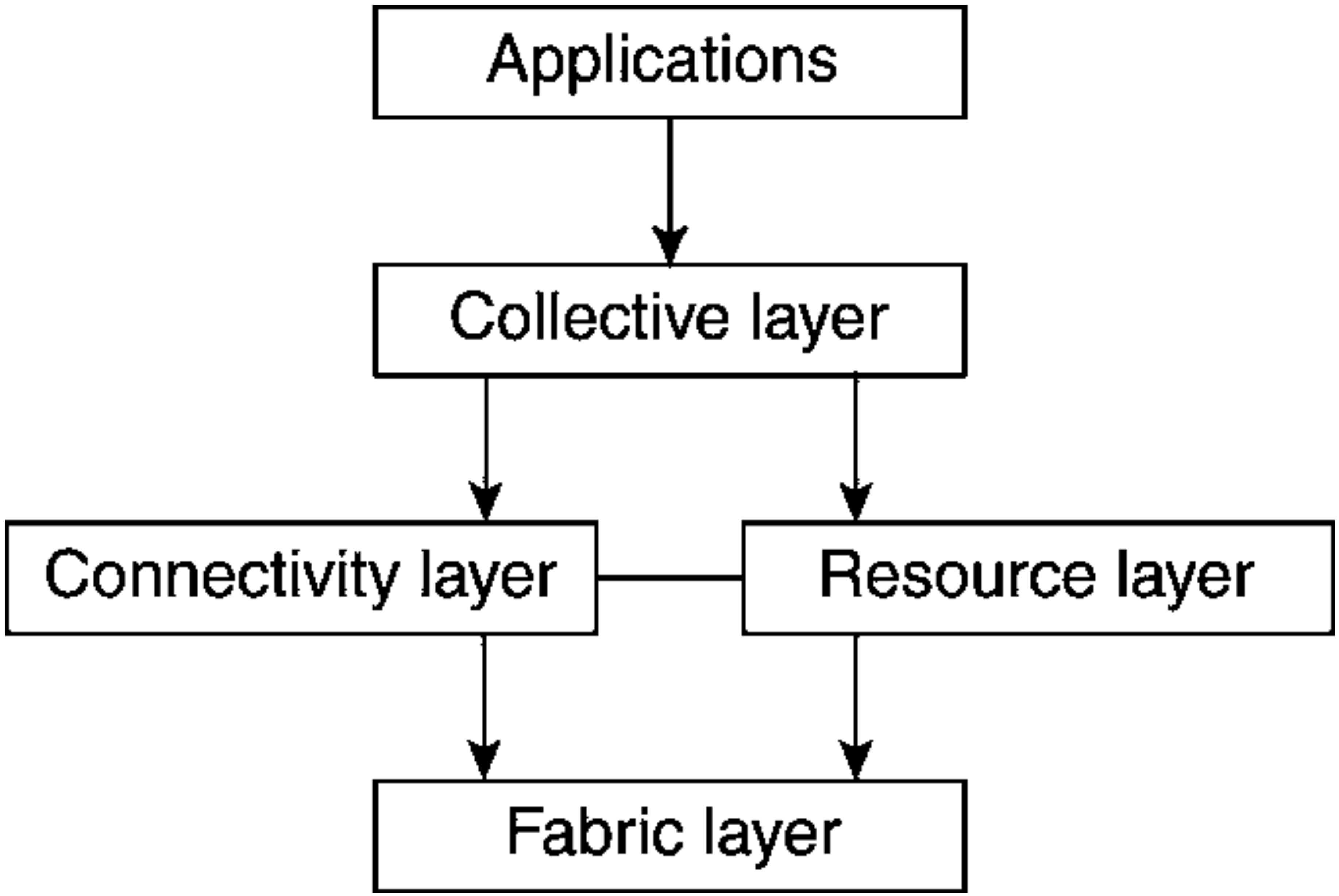
Pitfalls when Developing Distributed Systems

- False assumptions made by first time developer:
  - The network is reliable.
  - The network is secure.
  - The network is homogeneous.
  - The topology does not change.
  - Latency is zero.
  - Bandwidth is infinite.
  - Transport cost is zero.
  - There is one administrator.

Cluster Computing Systems



Grid Computing Systems

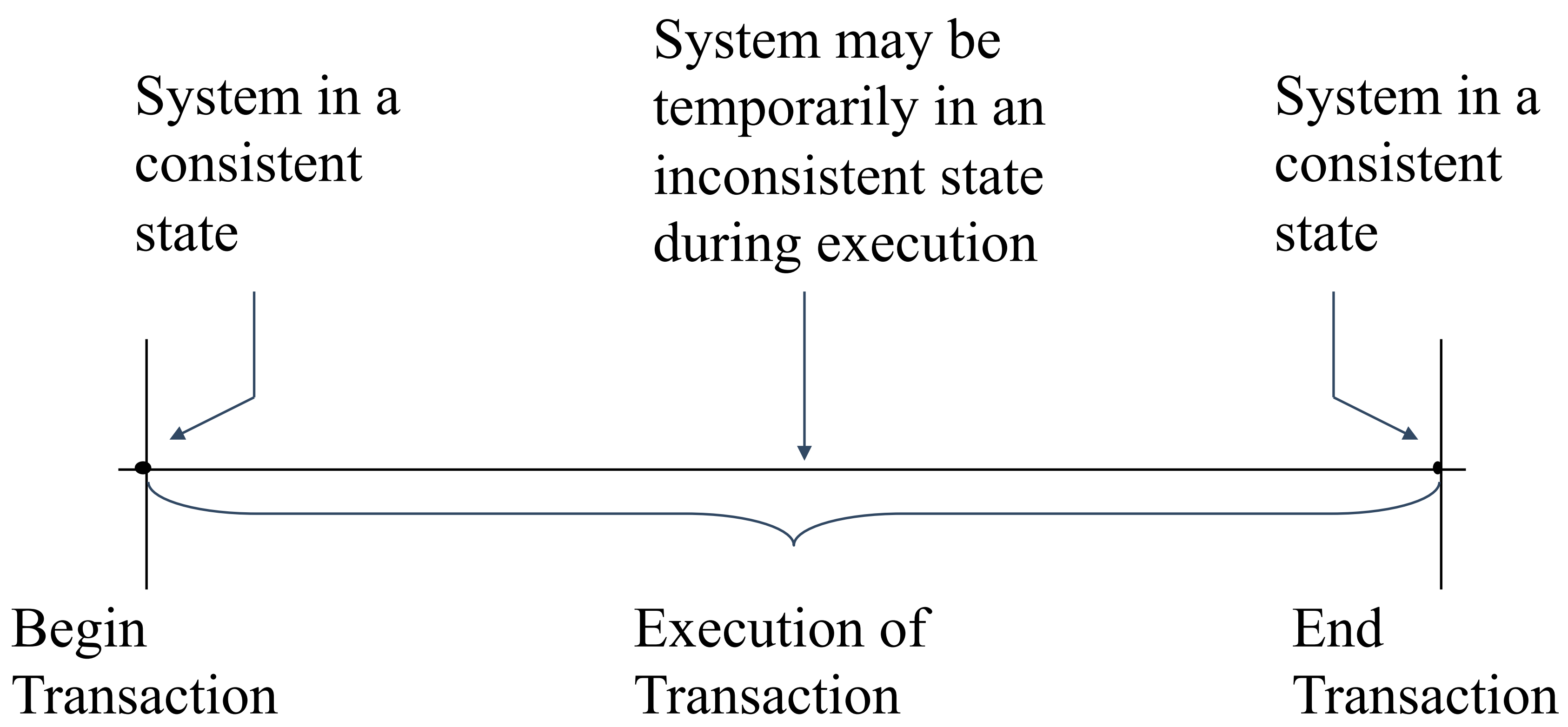


From Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e

Transaction Processing Systems (1)

A transaction is a collection of actions that make consistent transformations of system states while preserving system consistency.

- ➡ concurrency transparency
- ➡ failure transparency



Transaction Processing Systems (2)

Primitive	Description
BEGIN_TRANSACTION	Mark the start of a transaction
END_TRANSACTION	Terminate the transaction and try to commit
ABORT_TRANSACTION	Kill the transaction and restore the old values
READ	Read data from a file, a table, or otherwise
WRITE	Write data to a file, a table, or otherwise

From Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e

Transaction Processing Systems (3)

ATOMICITY

- All or nothing
- Multiple operations combined as an atomic transaction

CONSISTENCY

- No violation of integrity constraints
- Transactions are correct programs

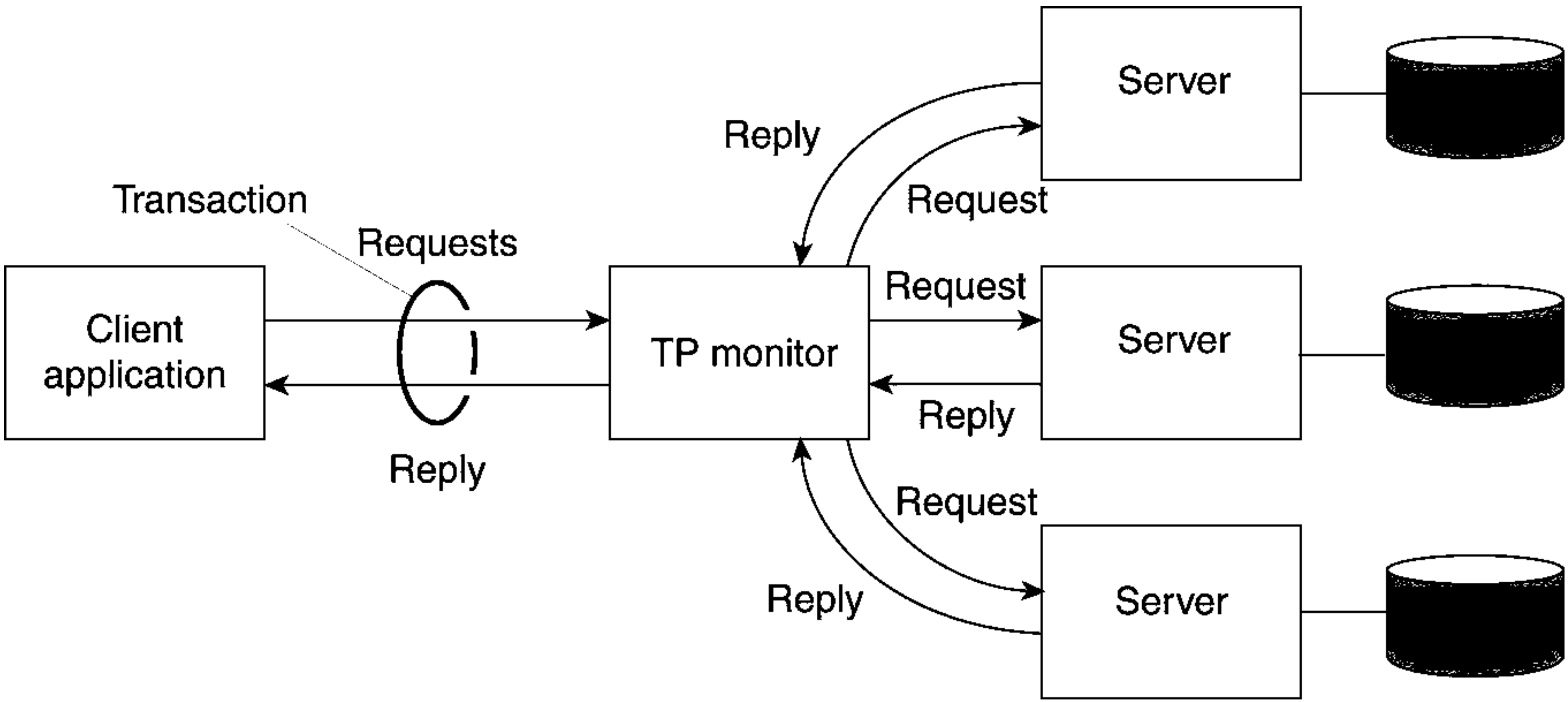
ISOLATION

- Concurrent changes invisible ➔serializable

DURABILITY

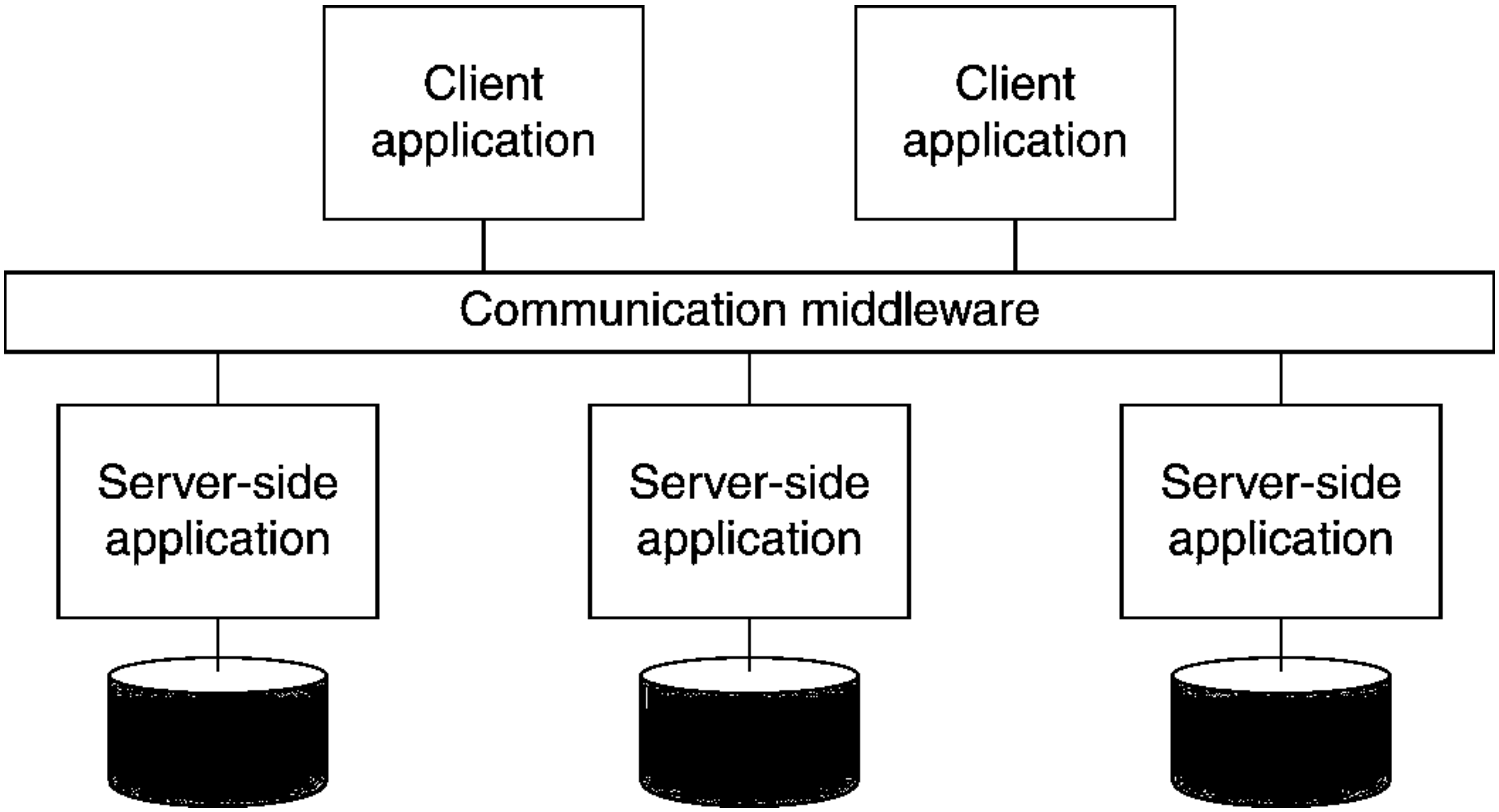
- Committed updates persist
- Database recovery

Transaction Processing Systems (4)



From Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e

Enterprise Application Integration



From Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e

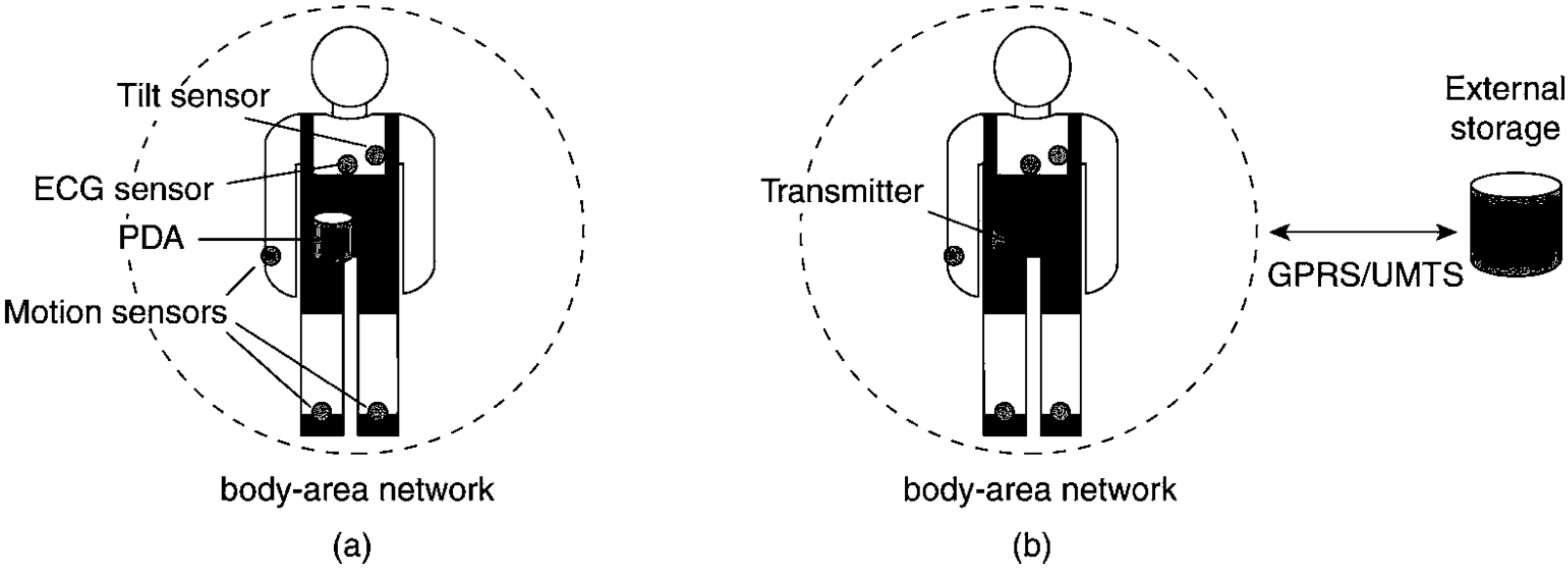
Distributed Pervasive Systems

- Requirements for pervasive systems
  - ➡ Embrace contextual changes.
  - ➡ Encourage ad hoc composition.
  - ➡ Recognize sharing as the default.

Electronic Health Care Systems (1)

- Questions to be addressed for health care systems:
  - ➡ Where and how should monitored data be stored?
  - ➡ How can we prevent loss of crucial data?
  - ➡ What infrastructure is needed to generate and propagate alerts?
  - ➡ How can physicians provide online feedback?
  - ➡ How can extreme robustness of the monitoring system be realized?
  - ➡ What are the security issues and how can the proper policies be enforced?

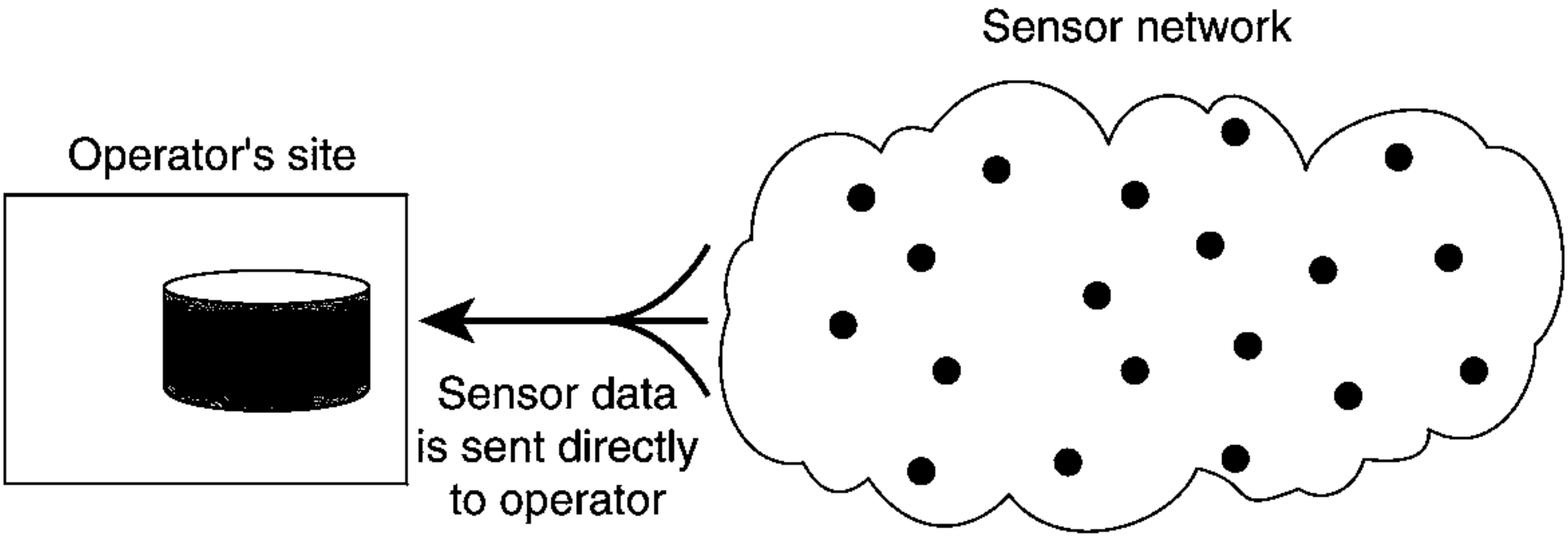
Electronic Health Care Systems (2)



Sensor Networks (1)

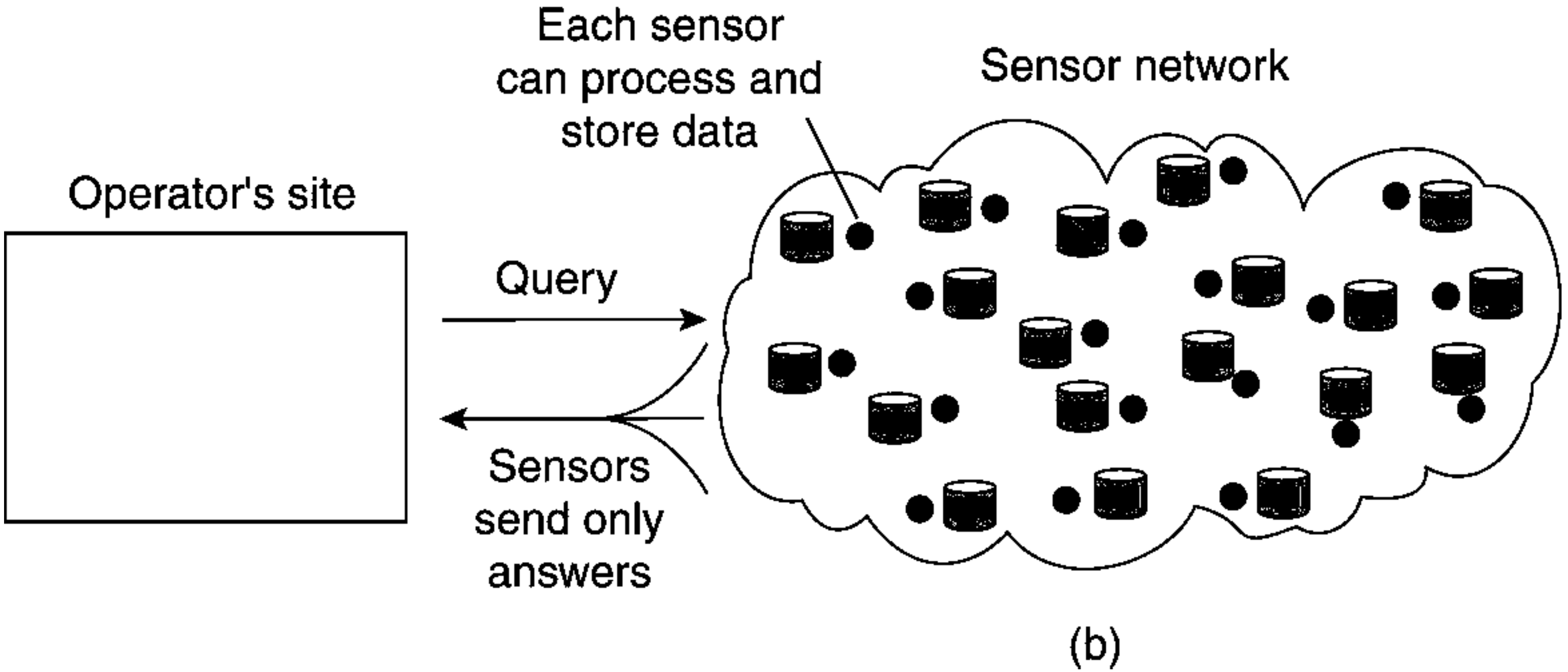
- Questions concerning sensor networks:
  - ➡ How do we (dynamically) set up an efficient tree in a sensor network?
  - ➡ How does aggregation of results take place? Can it be controlled?
  - ➡ What happens when network links fail?

Sensor Networks (2)

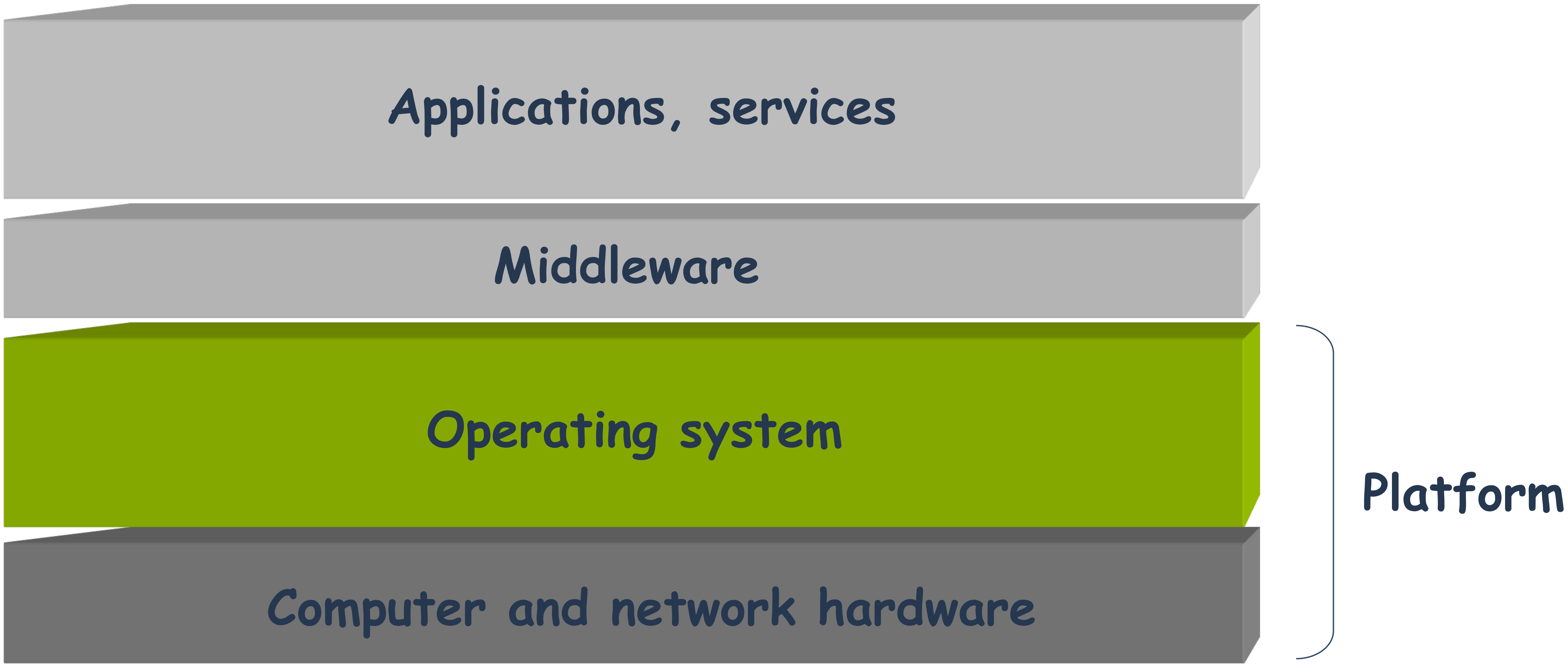


(a)

Sensor Networks (3)

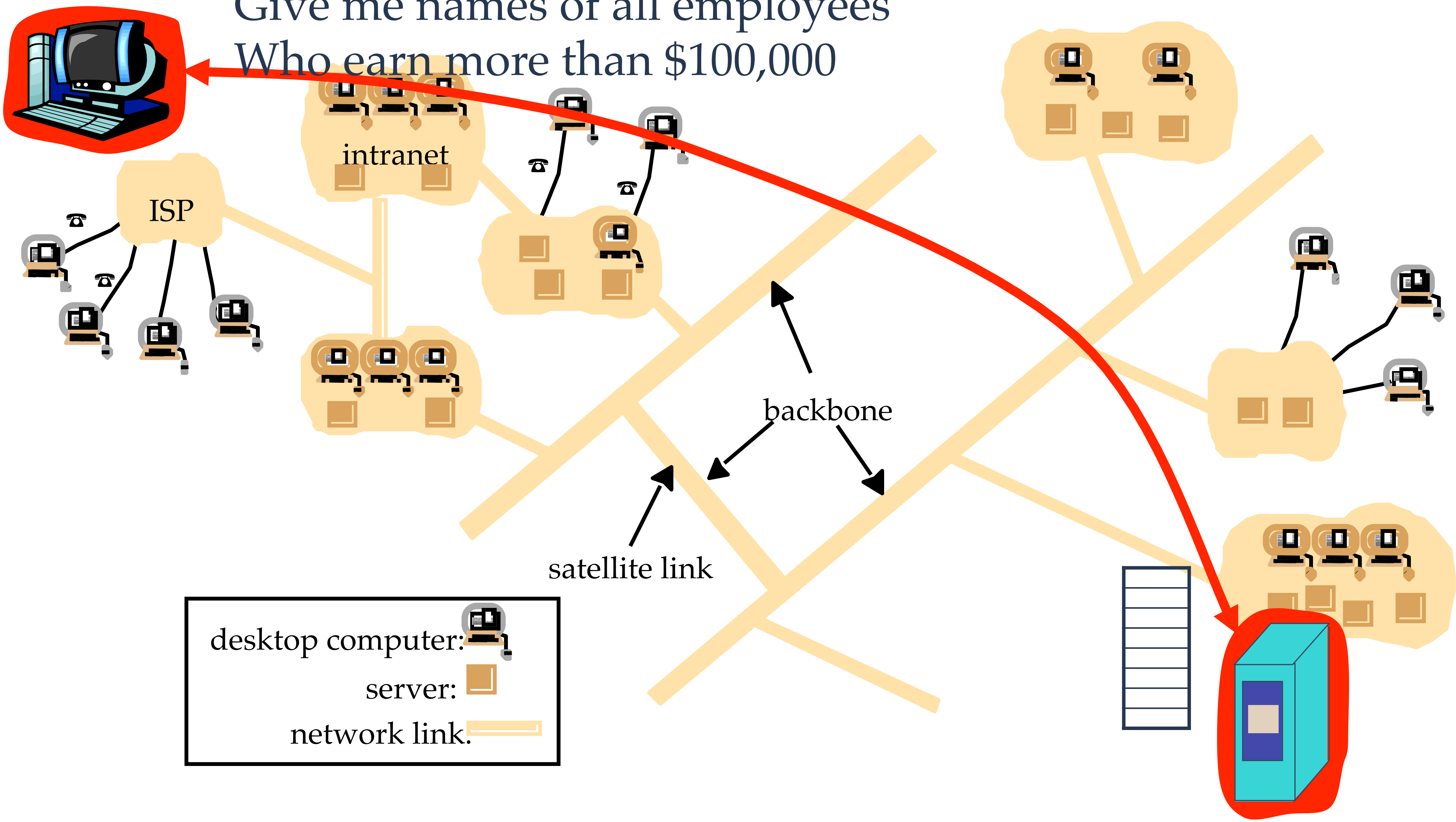


What is the Platform?



Networks and Communication

Give me names of all employees
Who earn more than $100,000

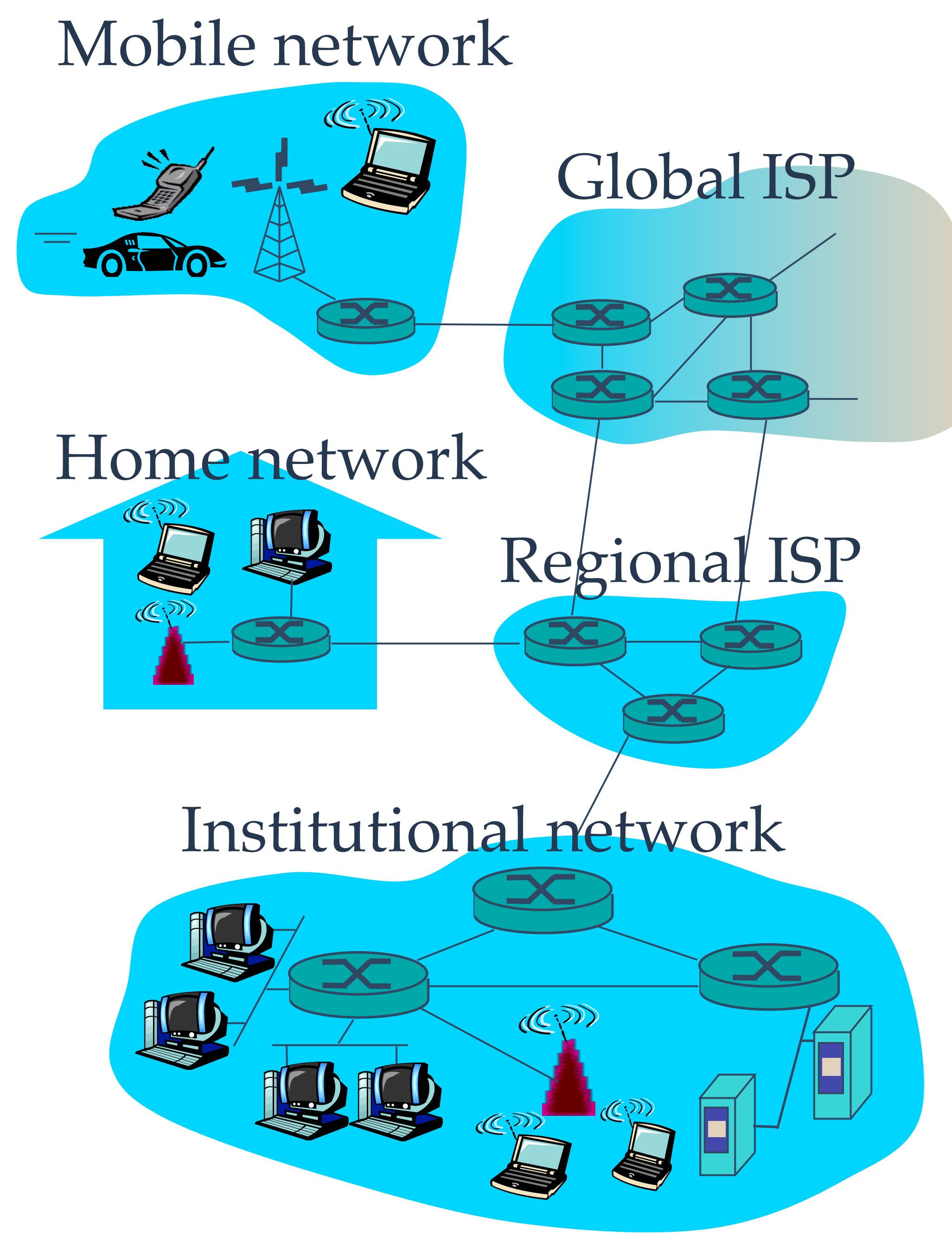


Issues

- How do the request and response get transmitted between the requestor and the server?
  - ➡ Protocols to facilitate communication
  - ➡ Moving the request and reply messages through the network
- What are the modes of communication between the requestor and the responder?
  - ➡ Connectionless service
  - ➡ Connection-oriented service

What's the Internet: "nuts and bolts" view

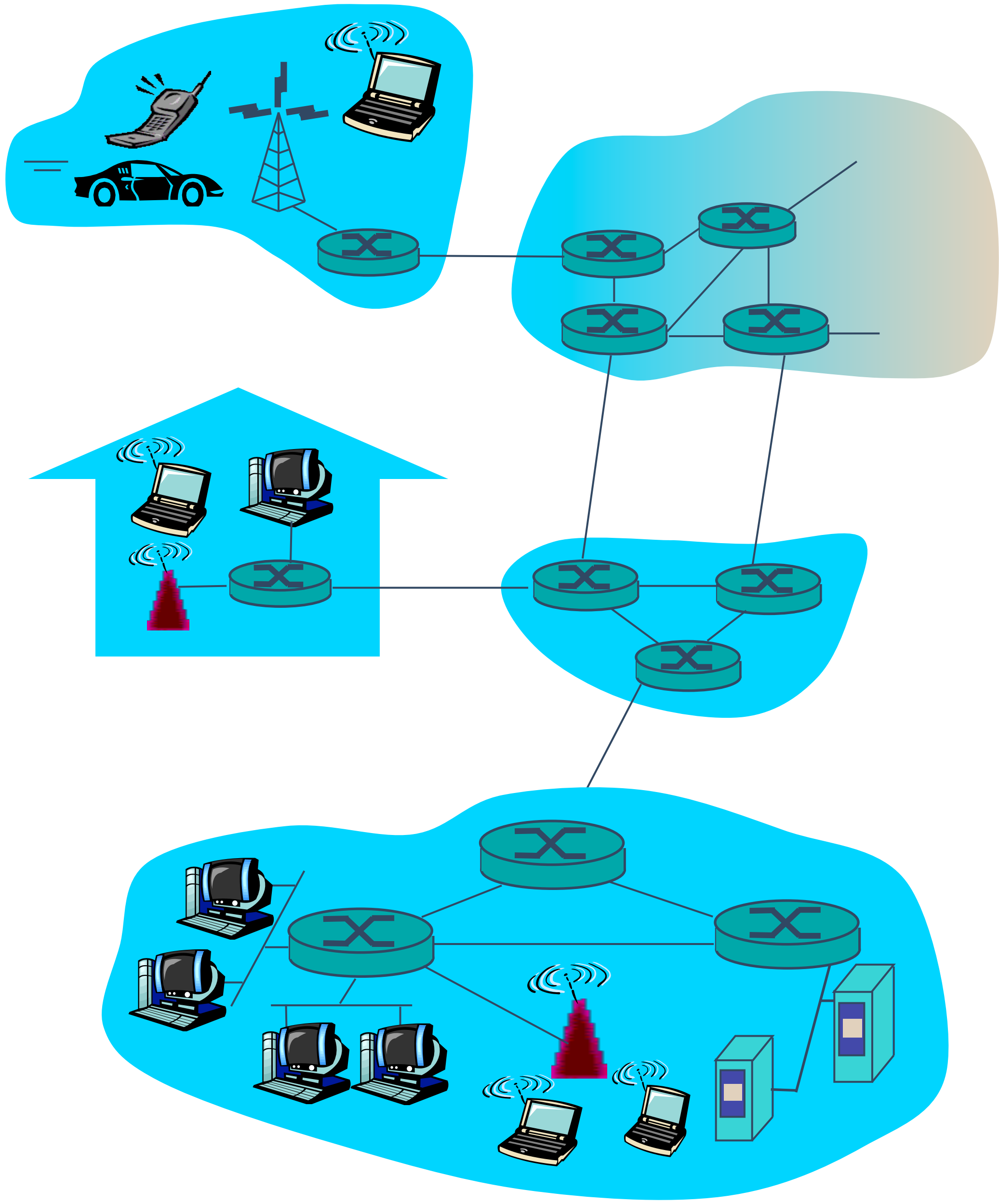
- millions of connected computing devices: *hosts = end systems*
  - running *network apps*
- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate = *bandwidth*
- *routers:* forward packets (chunks of data)



From Kurose & Ross, Computer Networking: A Top-Down Approach, 5e

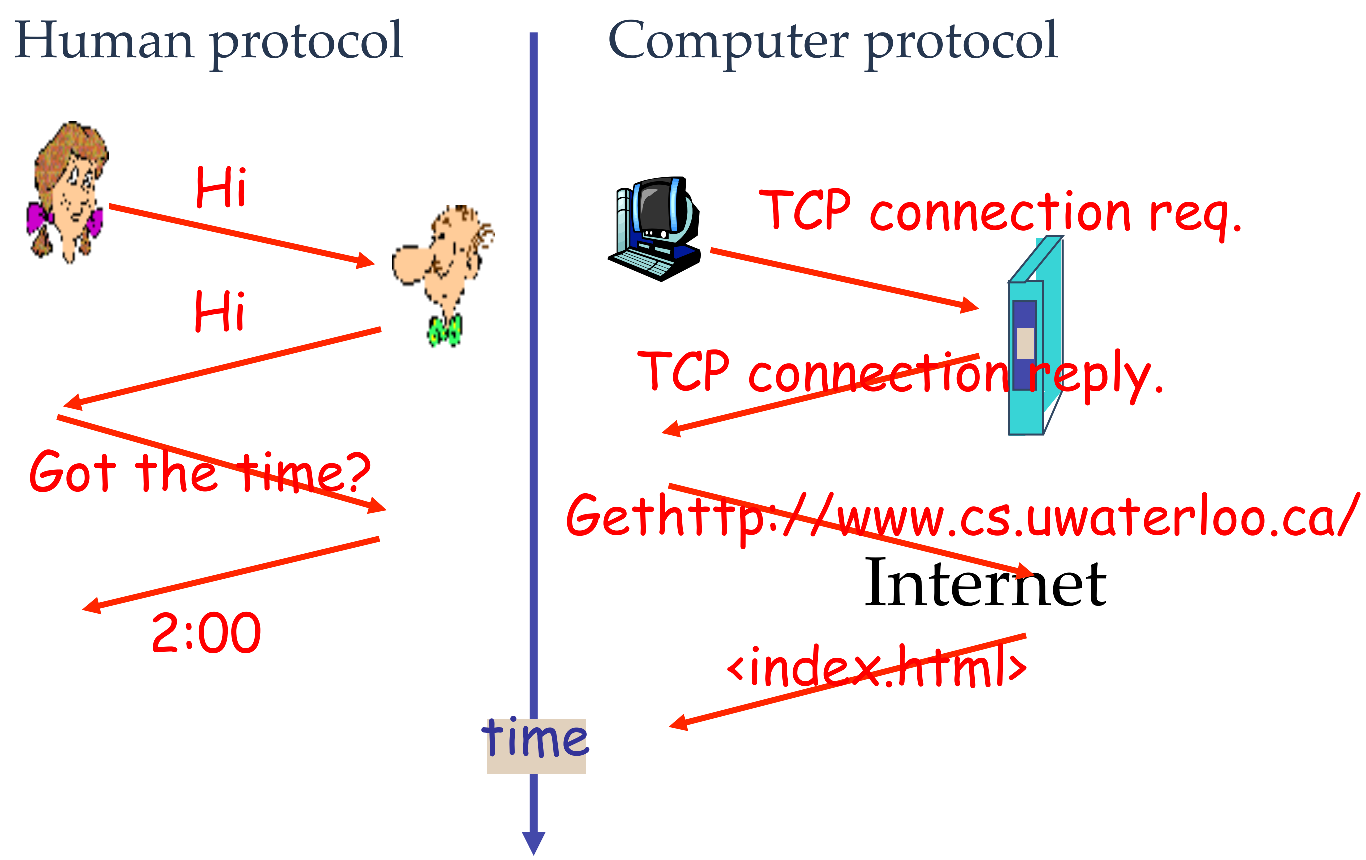
What’s the Internet: a service view

- communication *infrastructure* enables distributed applications:
  - Web, VoIP, email, games, e-commerce, file sharing
- communication services provided to apps:
  - reliable data delivery from source to destination
  - “best effort” (unreliable) data delivery



What is a protocol?

- A protocol defines the format and the order of messages sent and received among network entities, and the actions taken on message transmission and receipt
- Human protocols:
  - ➡ "What's the time?"
  - ➡ "I have a question"
  - ➡ introductions
- Network protocols:
  - ➡ machines rather than humans
  - ➡ all communication activity in governed by protocols



Protocol "Layers"

Networks are complex,
with many "pieces":

- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

Is there any hope of *organizing* structure of network?

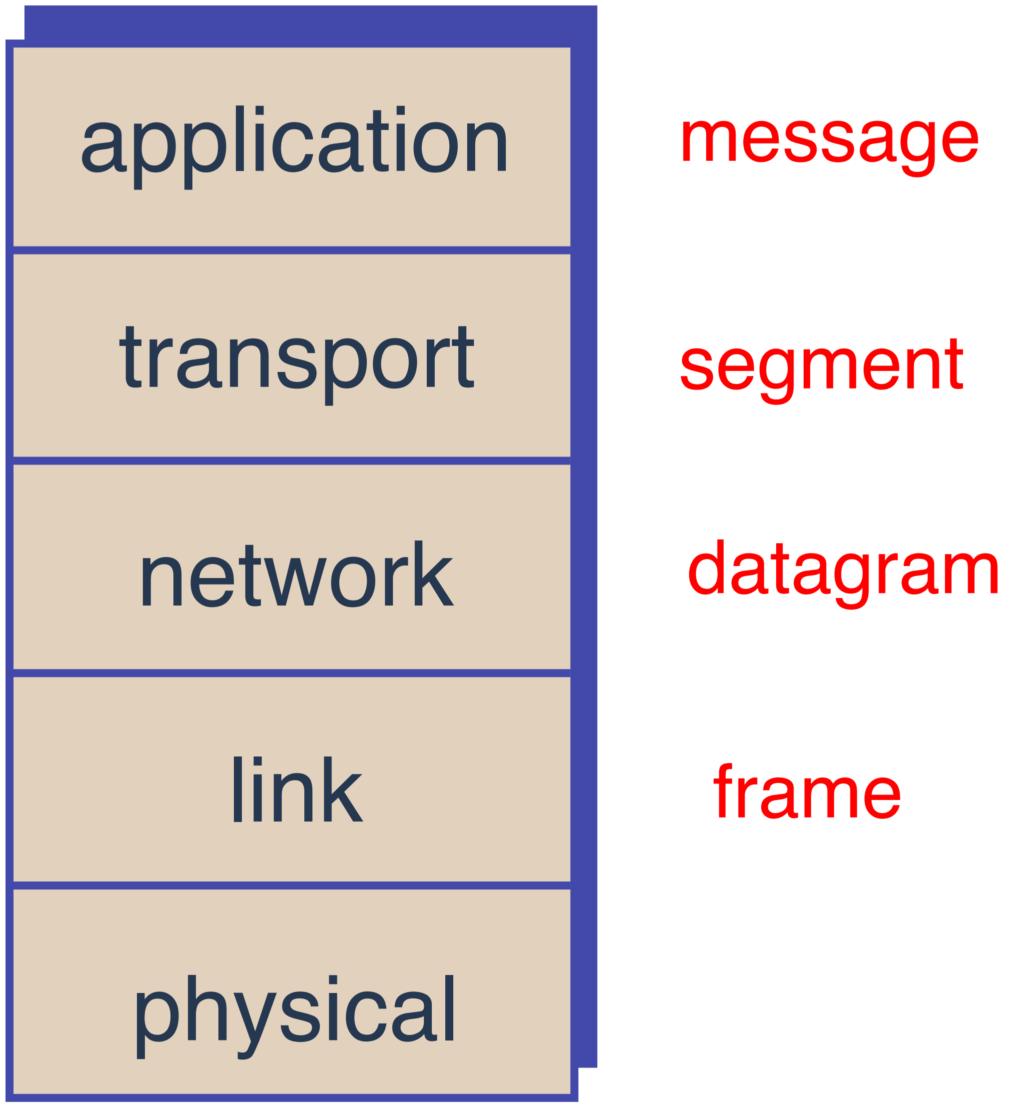
Or at least our discussion of networks?

Layered Architecture

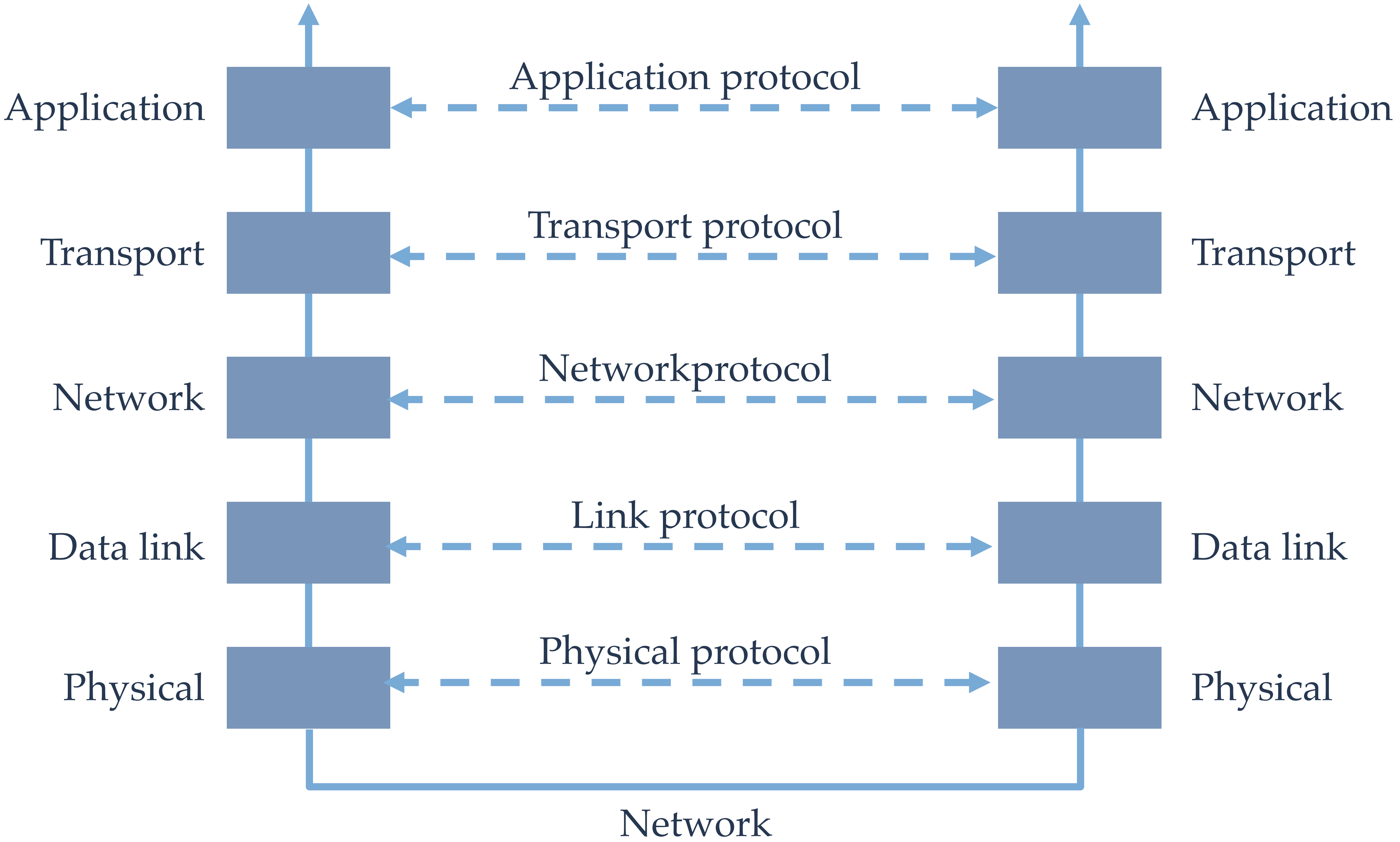
- Many requirements result in a large and complex system
- Introduce modularity by providing required functionality in layers
- Each layer provides a well defined set of services to next upper layer and exploits only services provided by next lower layer
- Can change implementation of a layer without affecting rest of the system
  - ➡ E.g., need to replace only a single layer when moving from wireless to wired network access

Internet Protocol Stack

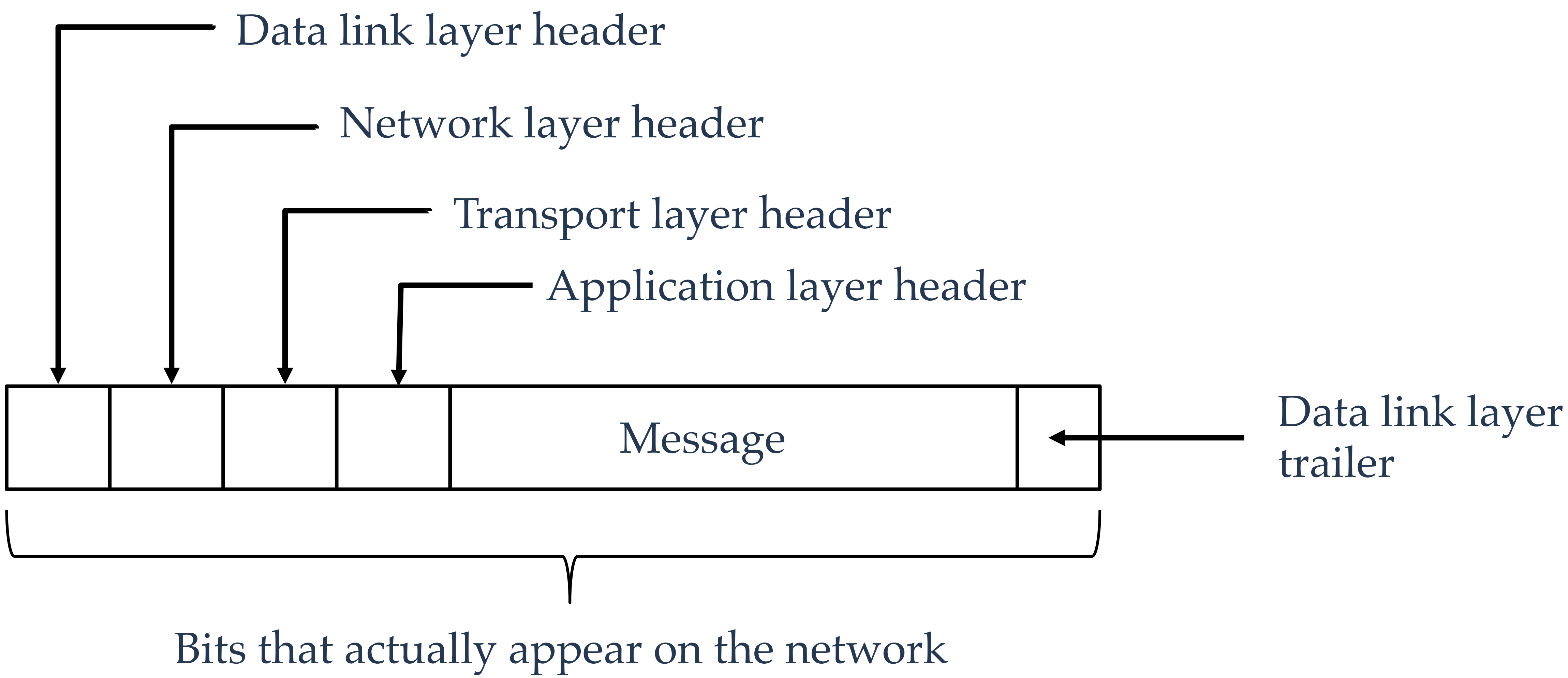
- application: supporting network applications
  - ftp, smtp, http
- transport: host-host data transfer
  - tcp, udp
- network: routing of datagrams from source to destination
  - ip, routing protocols
- link: data transfer between neighboring network elements
  - ppp, ethernet
- physical: bits “on the wire”



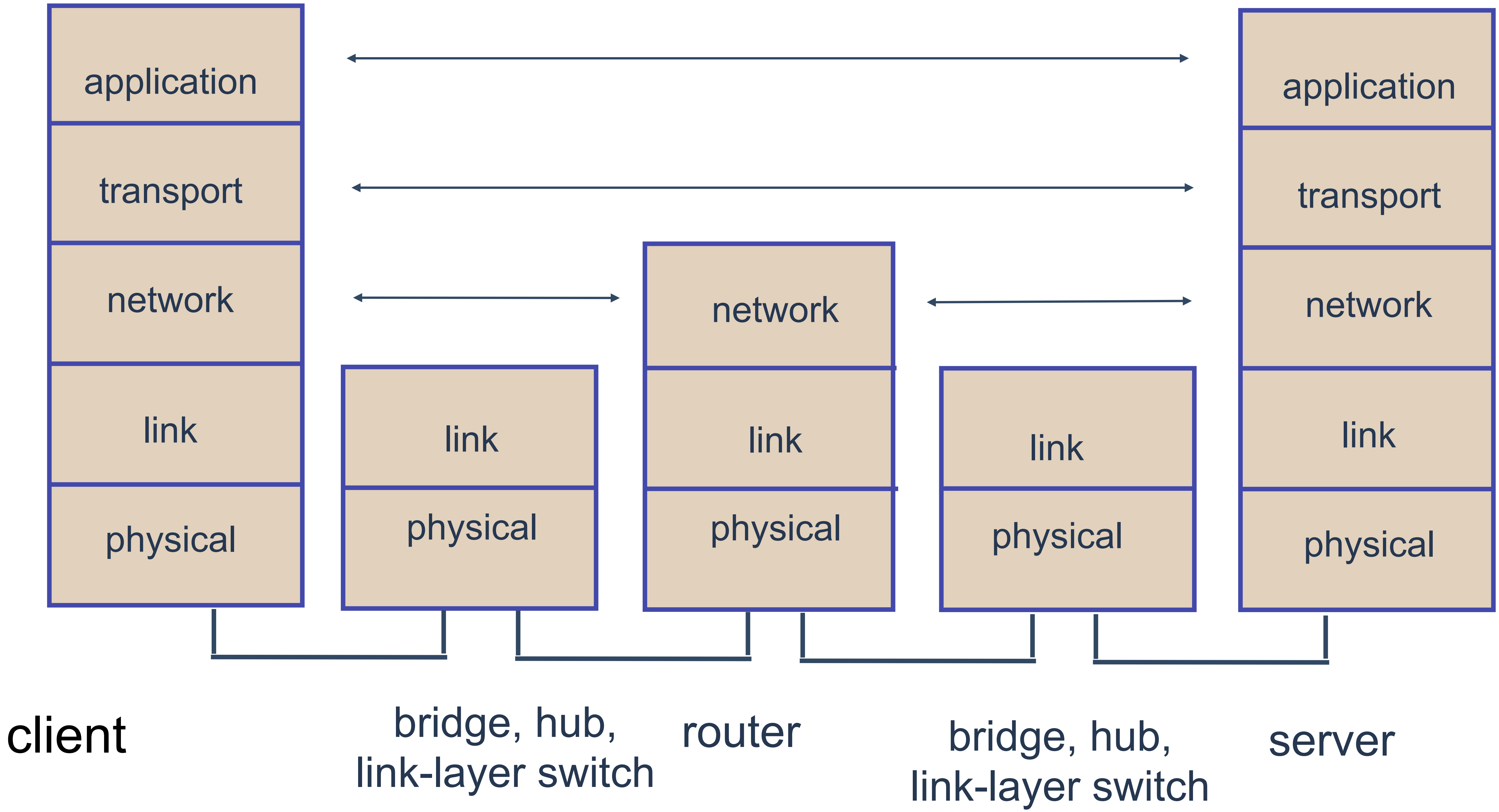
Layered Architecture



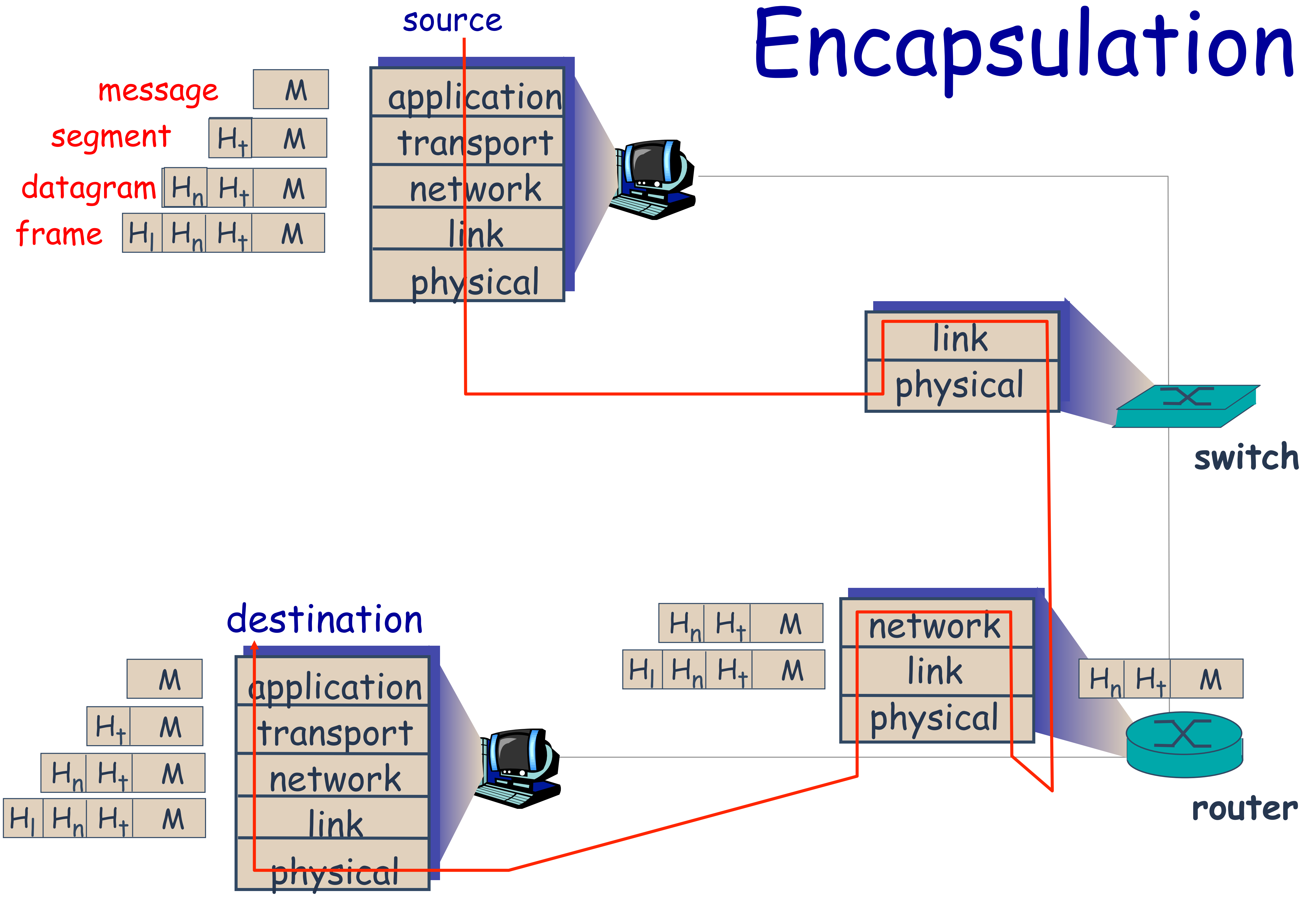
Message Format



Only Endnodes Implement all Layers



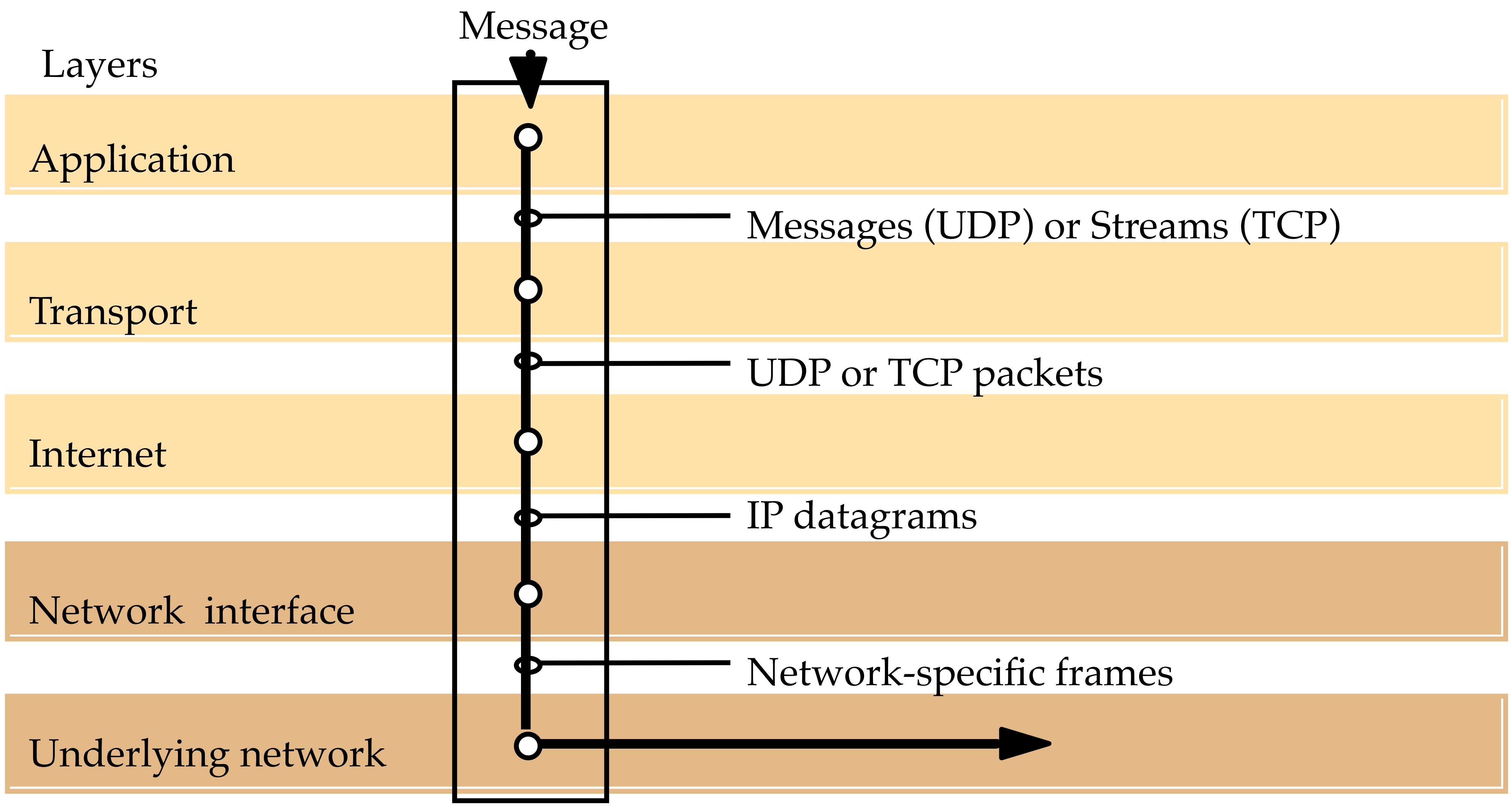
Encapsulation



Internet Protocols

Layer	Protocols
Application	FTP Telnet NFS SMTP HTTP …
Transport	TCP / UDP
Network	IP
Data Link Physical	X.25 / Ethernet / Packet Radio / ATM / FDDI / …

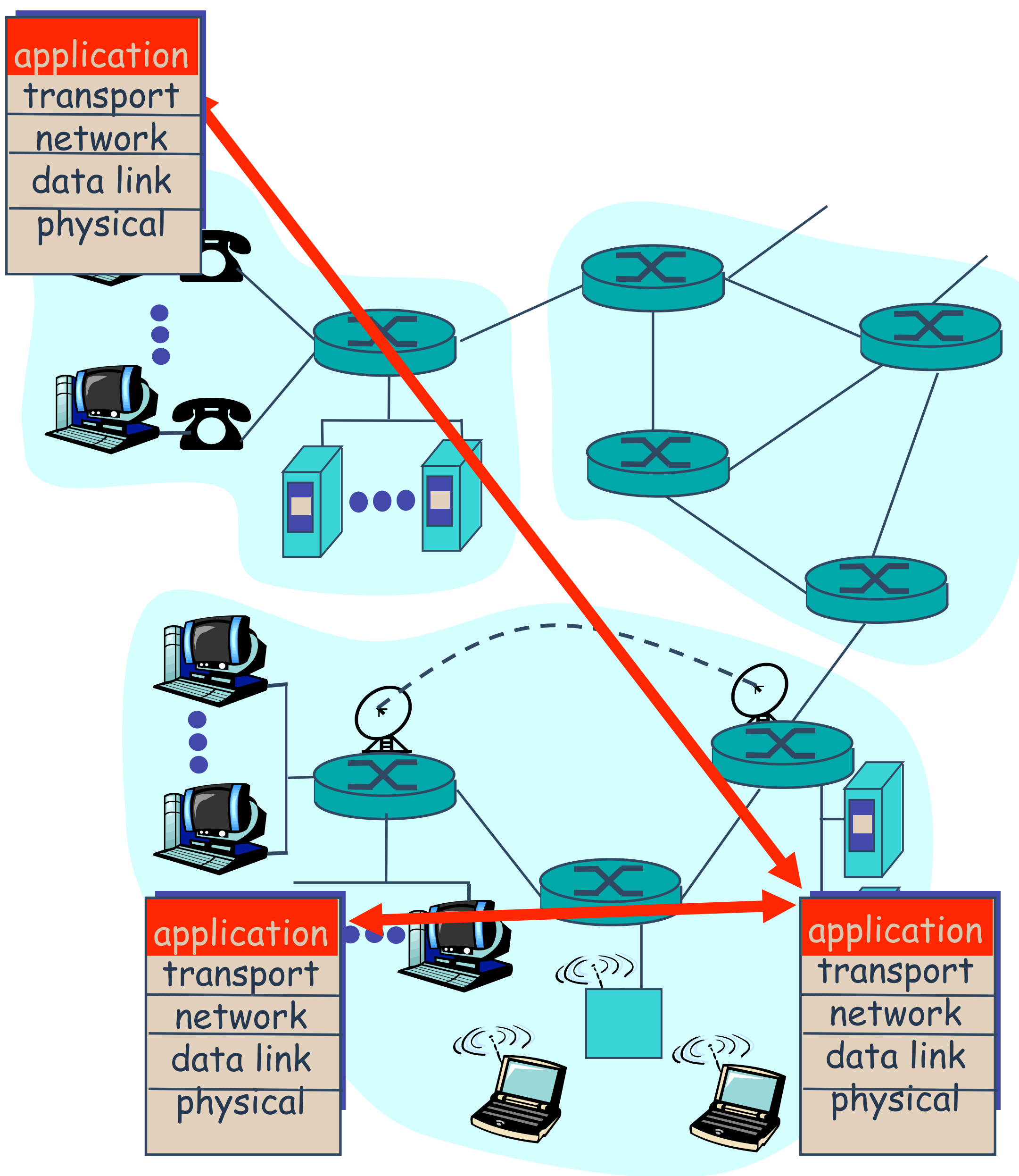
TCP/IP Layers



From Coulouris, Dollimore and Kindberg, Distributed Systems: Concepts and Design, 3rd ed.

Applications and Application-Layer Protocols

- Application: communicating, distributed processes
  - ➡ running in network hosts in "user space"
  - ➡ exchange messages to implement app
  - ➡ e.g., email, file transfer, the Web
- Application-layer protocols
  - ➡ one "piece" of an app
  - ➡ define messages exchanged by apps and actions taken
  - ➡ use services provided by lower layer protocols



Application-Layer Protocols (2)

- API: application programming interface
  - Defines interface between application and transport layer
  - socket: Internet API
    - two processes communicate by sending data into socket, reading data out of socket
- What transport services does an application need?
  - Data loss
    - some apps (e.g., audio) can tolerate some loss
    - other apps (e.g., file transfer) require 100% reliable data transfer
  - Bandwidth
    - some apps (e.g., multimedia) require a minimum amount of bandwidth to be "effective"
    - other apps ("elastic apps") make use of whatever bandwidth they get
  - Timing
    - some apps (e.g., Internet telephony, interactive games) require low delay to be "effective"

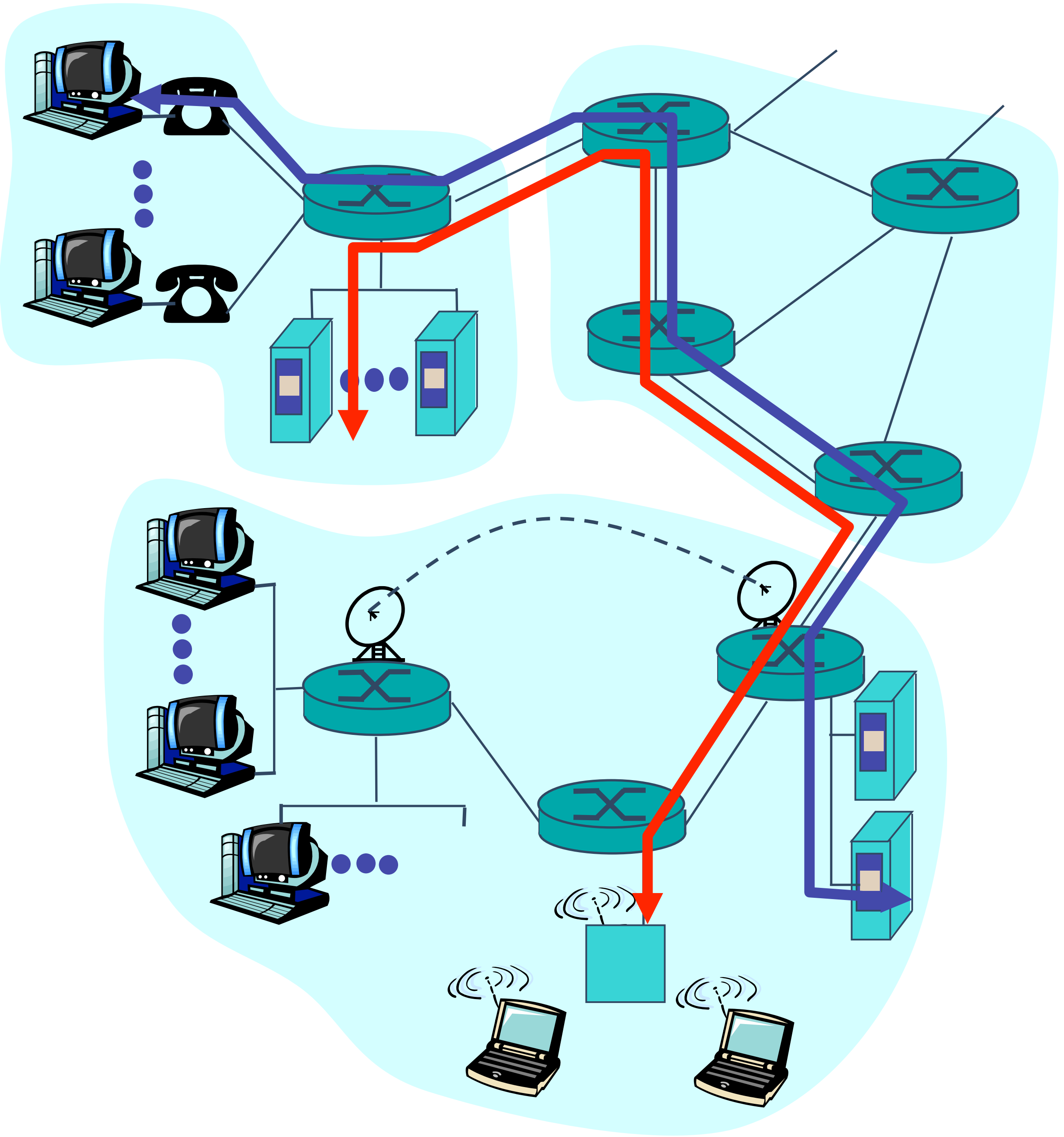
Network Core – How Data Move Through the Network

- Network is a mesh of interconnected routers
- Two ways of setting up a connection between two computers
  - ➡ circuit switching: dedicated circuit per call – e.g., telephone net
  - ➡ packet-switching: data sent thru the network in discrete "chunks"

Circuit Switching

End-end resources reserved for "call"

- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required

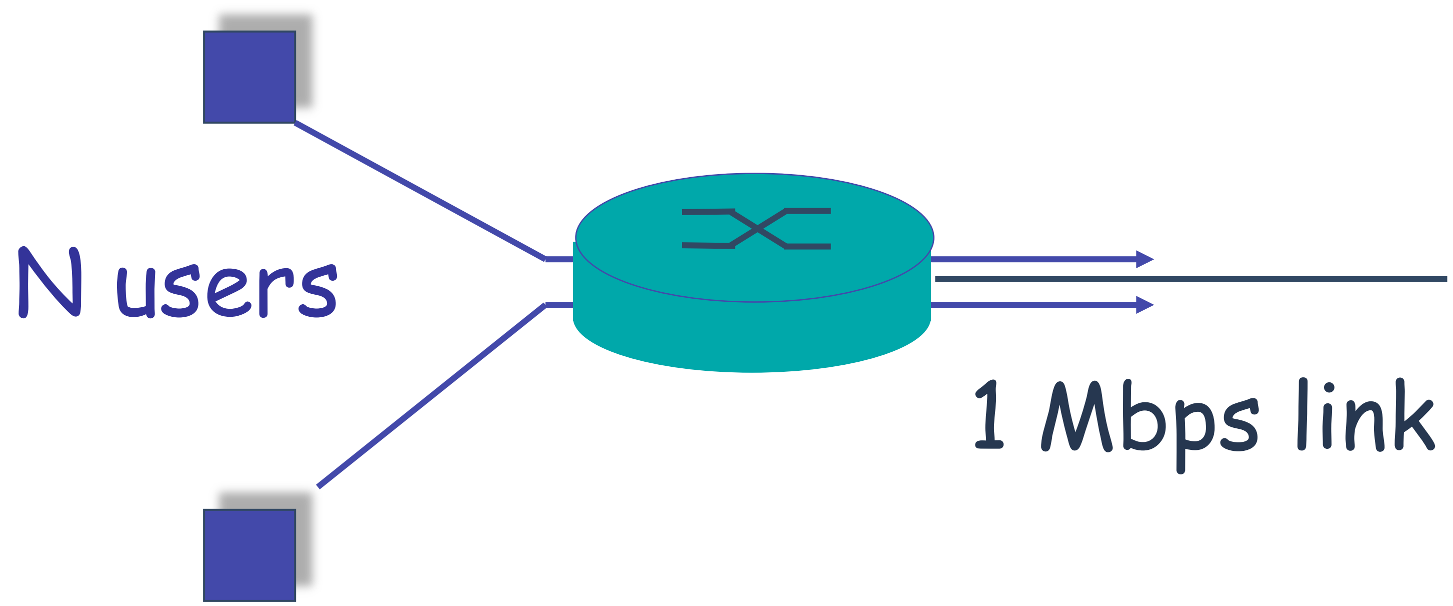


Packet Switching

- Each end-end data stream divided into packets
  - ➡ User A, B packets share network resources
  - ➡ Each packet uses full link bandwidth
  - ➡ Resources used as needed
- Resource contention:
  - ➡ Aggregate resource demand can exceed amount available
  - ➡ Congestion: packets queue, wait for link use
- Store and forward: packets move one hop-at-a-time (store-and-forward)
  - ➡ Transmit over link
  - ➡ Wait turn at next link

Packet Switching vs Circuit Switching

- Packet switching allows more users to use network!
- 1 Mbit link; each user:
  - 100Kbps when "active"
  - active 10% of time
- circuit-switching:
  - 10 users
- packet switching:
  - 35 users, probability > 10 active less than .0004



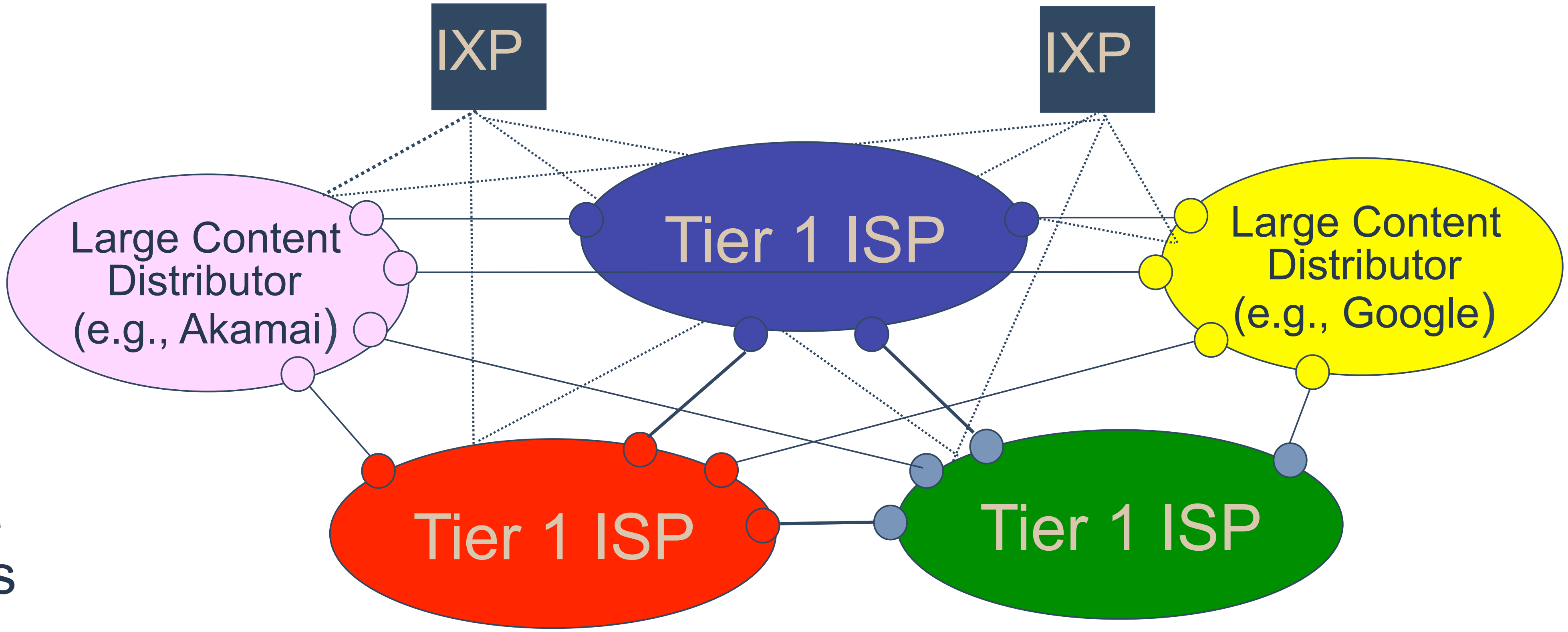
Packet Switching vs Circuit Switching (2)

- Packet switching is great for bursty data
  - ➡ resource sharing
  - ➡ no call setup
- It incurs excessive congestion: packet delay and loss
  - ➡ protocols needed for reliable data transfer, congestion control
- How to provide circuit-like behavior?
  - ➡ bandwidth guarantees needed for audio/video apps
  - ➡ still an unsolved problem, but solutions such as ATM have been developed

Internet structure: network of networks

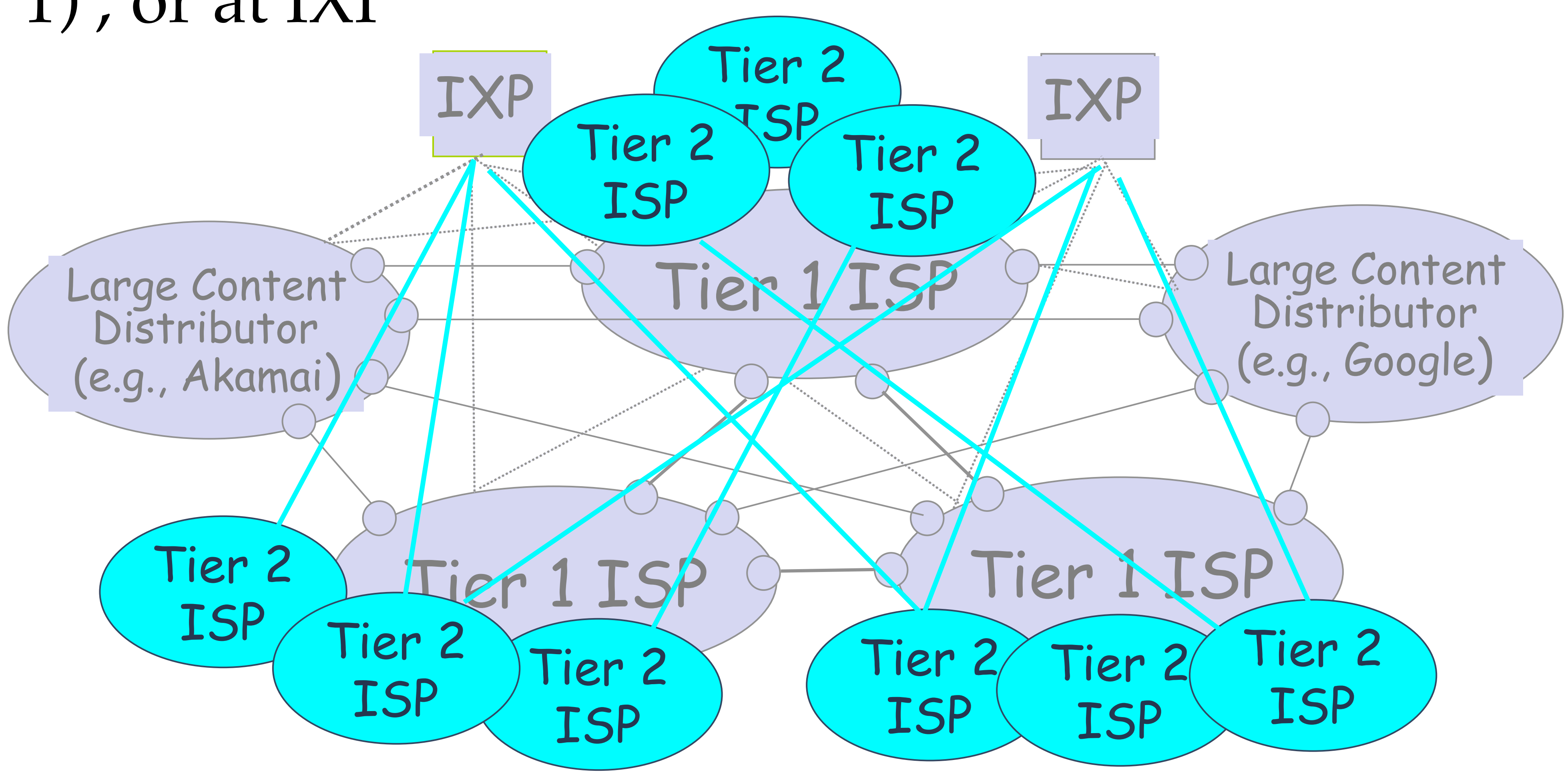
- Roughly hierarchical
- At center: small # of well-connected large networks
  - "tier-1" commercial ISPs (e.g., Rogers, Telus, Bell, Verizon, Sprint, AT&T, Qwest, Level3), national & international coverage
  - large content distributors (Google, Akamai, Microsoft)
  - treat each other as equals (no charges)

Tier-1 ISPs & Content Distributors, interconnect (peer) privately … or at Internet Exchange Points IXPs



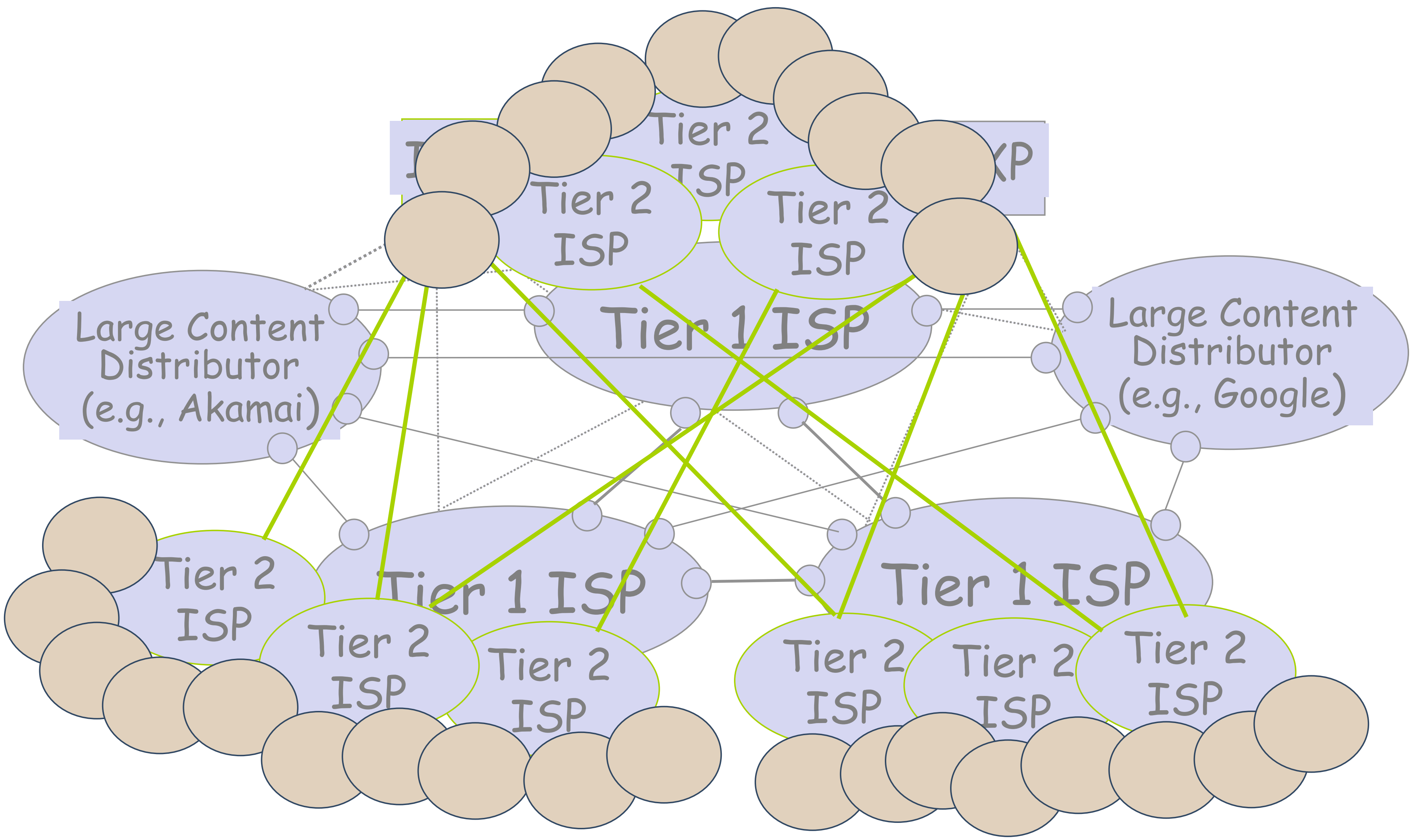
Internet structure: network of networks

- "tier-2" ISPs: smaller (often regional) ISPs
  - ➡ connect to one or more tier-1 *(provider)* ISPs
    - ✦ each tier-1 has many tier-2 *customer nets*
    - ✦ tier 2 pays tier 1 provider
- tier-2 nets sometimes peer directly with each other (bypassing tier 1) , or at IXP



Internet structure: network of networks

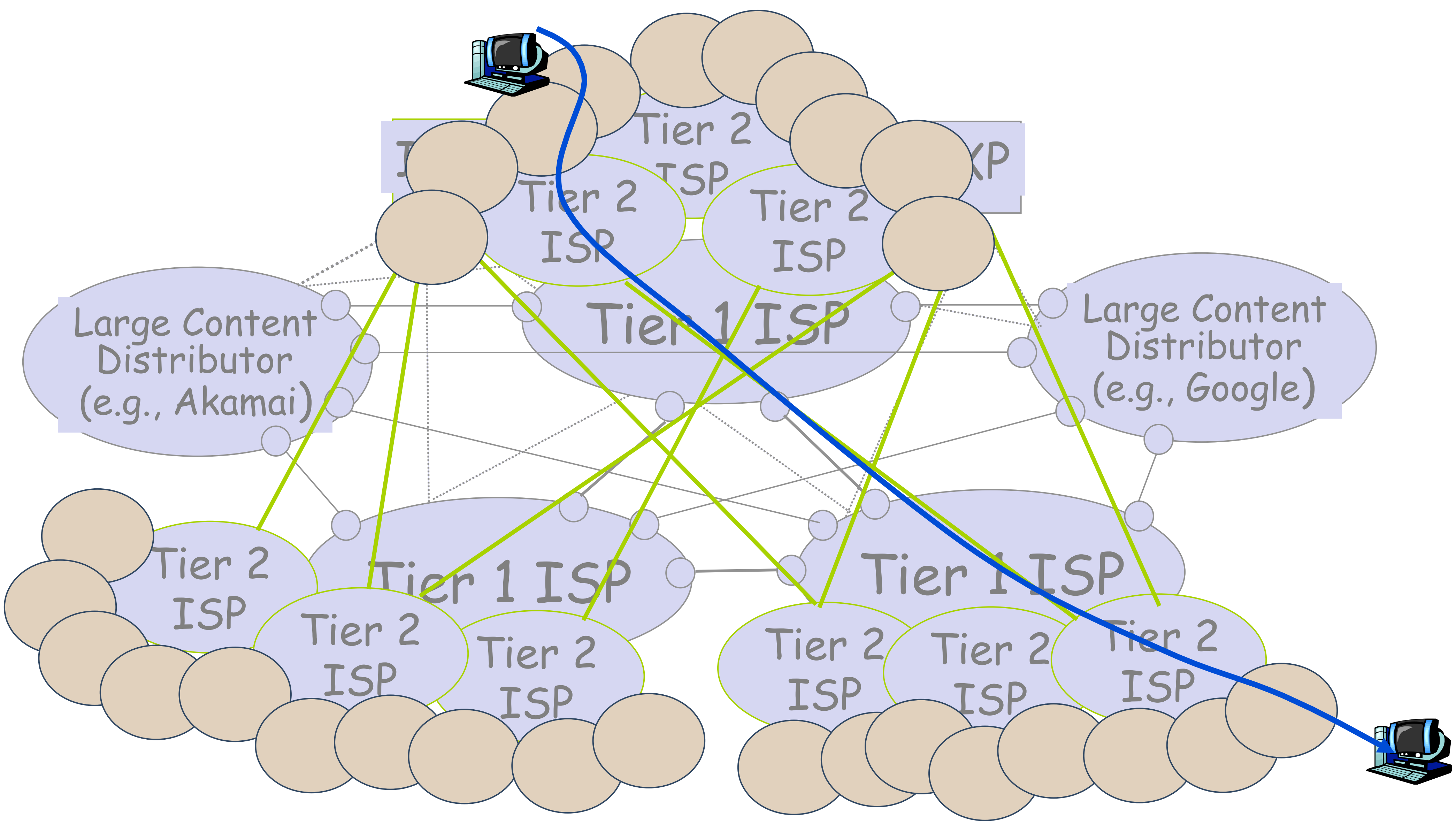
- "Tier-3" ISPs, local ISPs
- customer of tier 1 or tier 2 network
  - ➡ last hop ("access") network (closest to end systems)



From Kurose & Ross, Computer Networking: A Top-Down Approach, 5e

Internet structure: network of networks

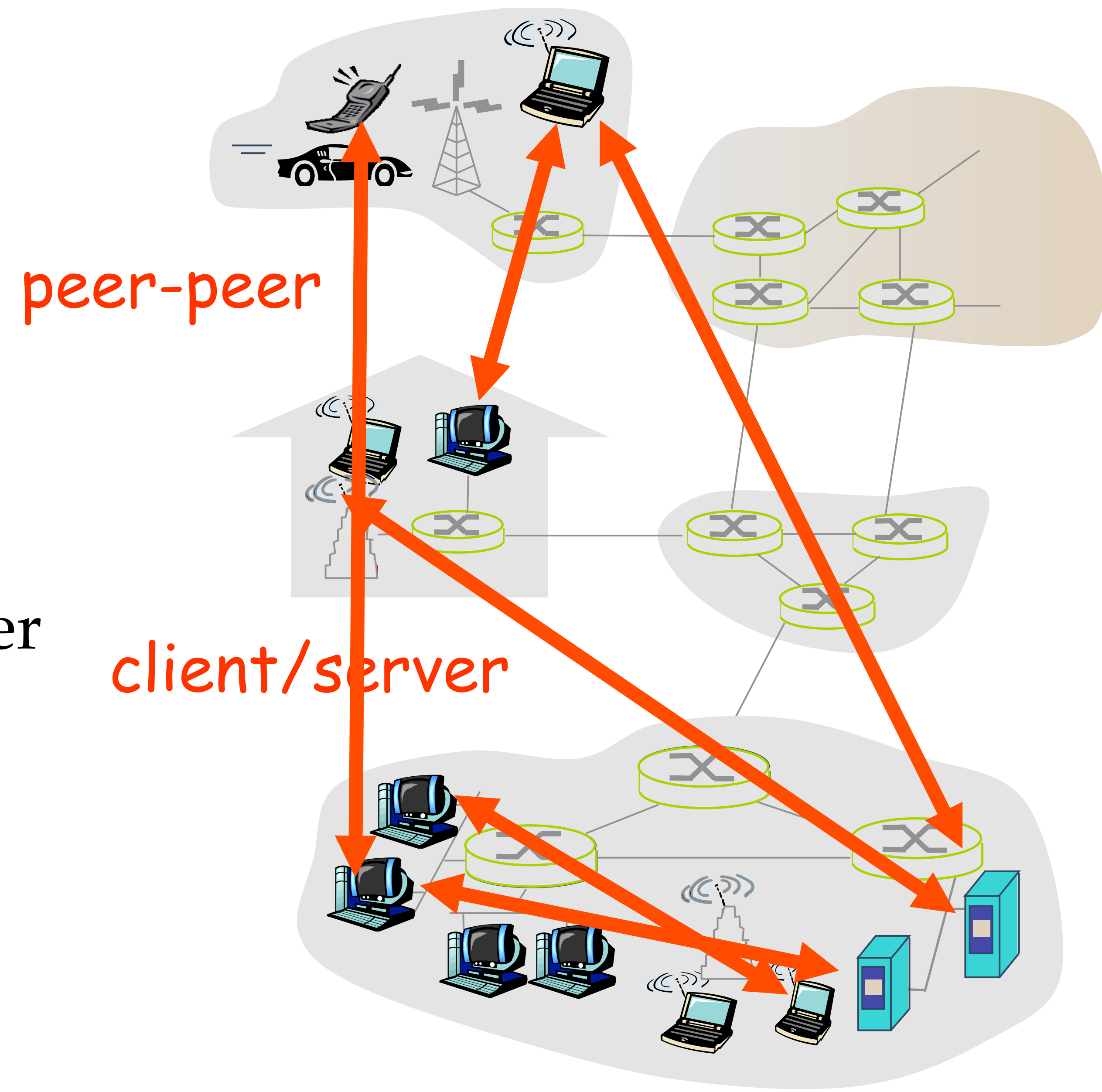
- a packet passes through *many* networks from source host to destination host



From Kurose & Ross, Computer Networking: A Top-Down Approach, 5e

The Network Edge: How to Connect to the Network

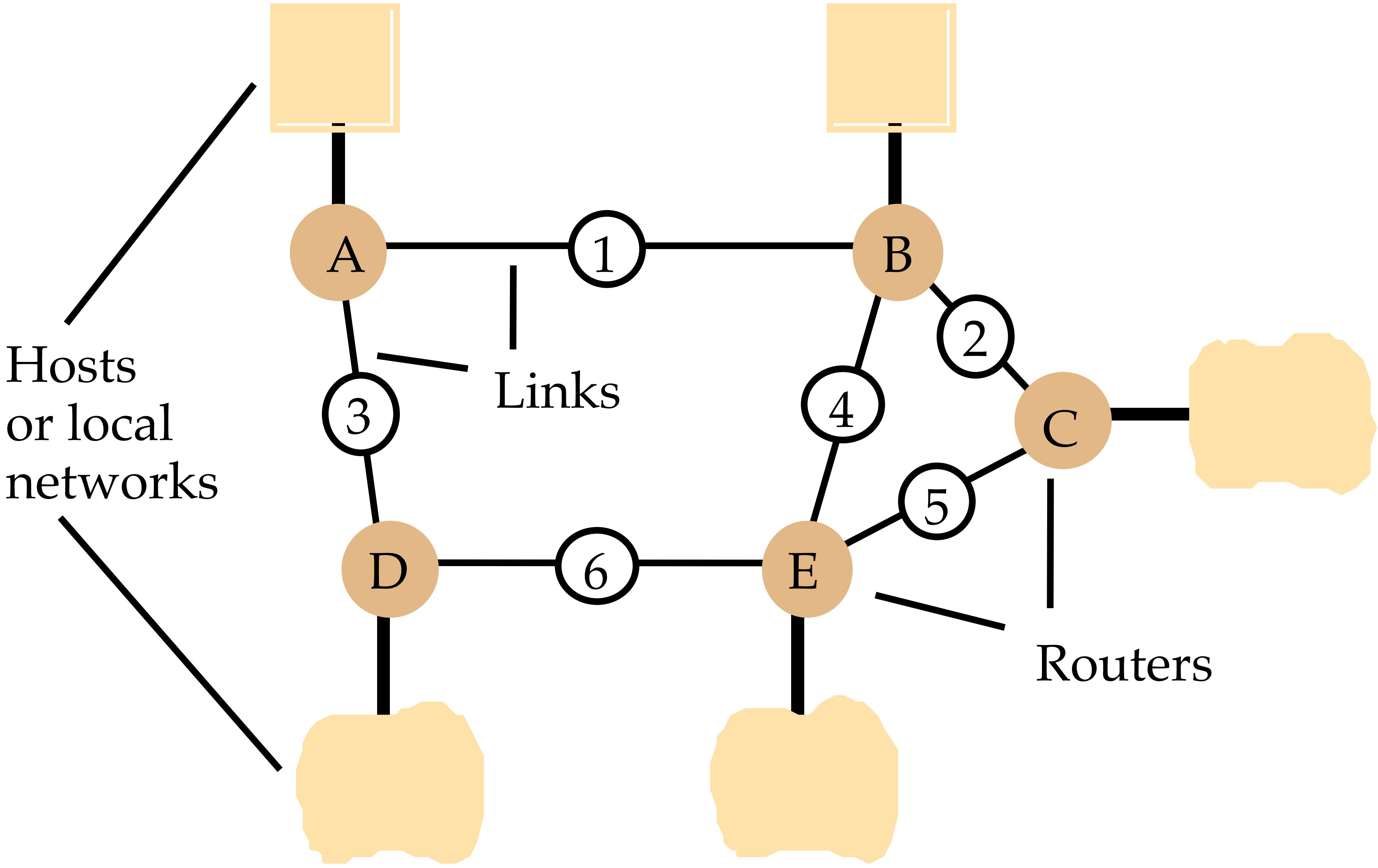
- end systems (hosts):
  - run application programs
  - e.g. Web, email
  - at "edge of network"
- client/server model
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server
- peer-peer model
  - minimal (or no) use of dedicated servers
  - e.g. Skype, BitTorrent



Routing

- Goal: move packets among routers from source to destination
- It is an issue in packet switched networks
- datagram network:
  - ➡ destination address determines next hop
  - ➡ routes may change during session
  - ➡ analogy: driving, asking directions
- virtual circuit network:
  - ➡ each packet carries tag (virtual circuit ID), tag determines next hop
  - ➡ fixed path determined at call setup time, remains fixed thru call
  - ➡ routers maintain per-call state

Routing in a Wide Area Network



From Coulouris, Dollimore and Kindberg, Distributed Systems: Concepts and Design, 3rd ed.

Routing Tables

Routings from A

To	*Link*	*Cost*
A	local	0
B	1	1
C	1	2
D	3	1
E	1	2

Routings from B

To	*Link*	*Cost*
A	1	1
B	local	0
C	2	1
D	1	2
E	4	1

Routings from C

To	*Link*	*Cost*
A	2	2
B	2	1
C	local	0
D	5	2
E	5	1

Routings from D

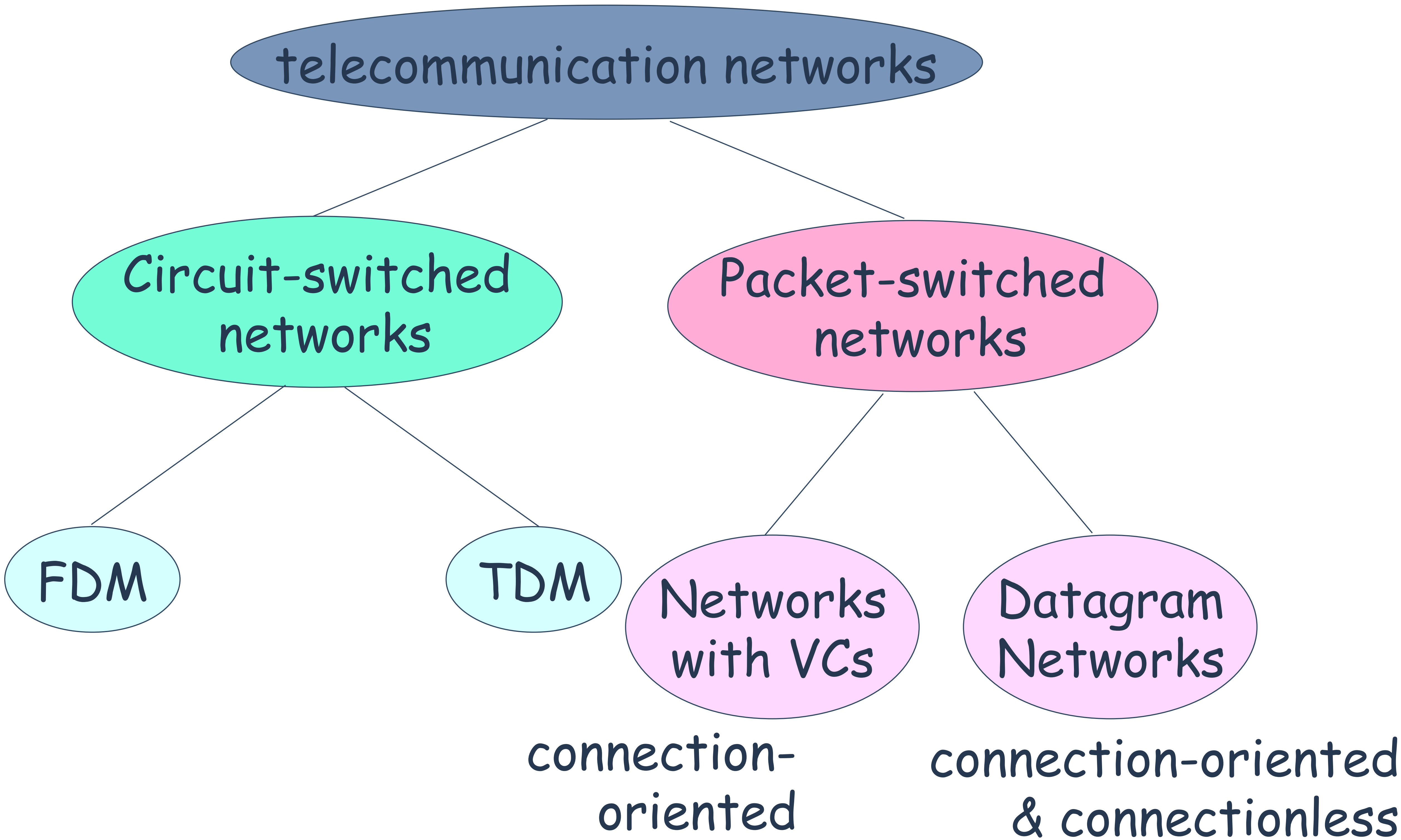
To	*Link*	*Cost*
A	3	1
B	3	2
C	6	2
D	local	0
E	6	1

Routings from E

To	*Link*	*Cost*
A	4	2
B	4	1
C	5	1
D	6	1
E	local	0

From Coulouris, Dollimore and Kindberg, Distributed Systems: Concepts and Design, 3rd ed.

Network Taxonomy



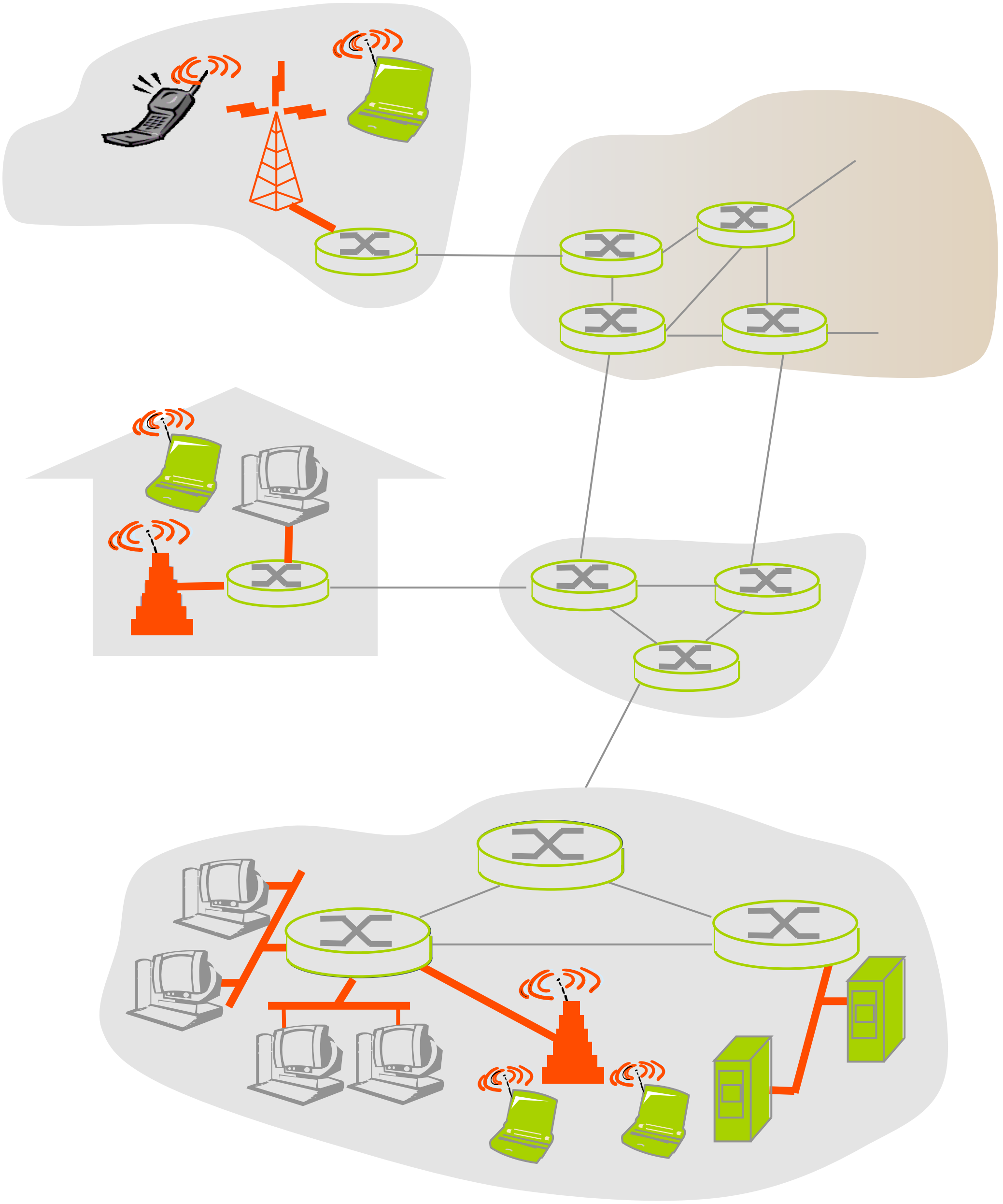
Access networks and physical media

How to connect end systems to edge router?

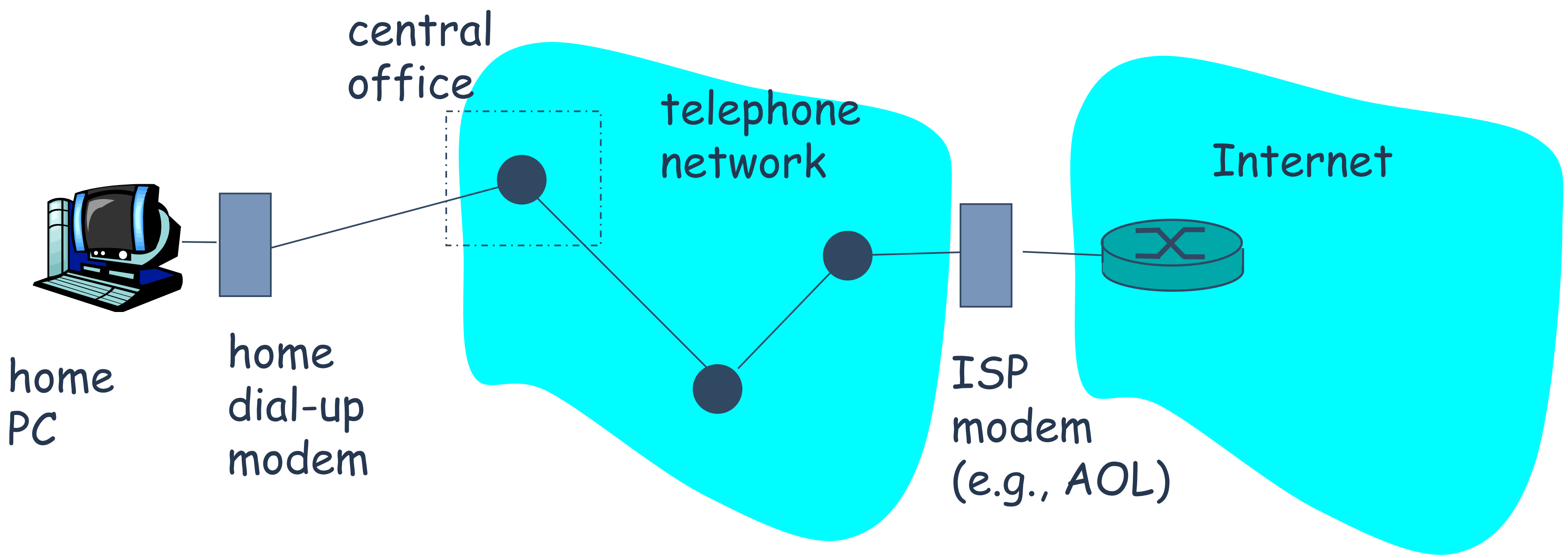
- Residential access nets
- Institutional access networks (school, company)
- Mobile access networks

Keep in mind:

- Bandwidth (bits per second) of access network
- Shared or dedicated?

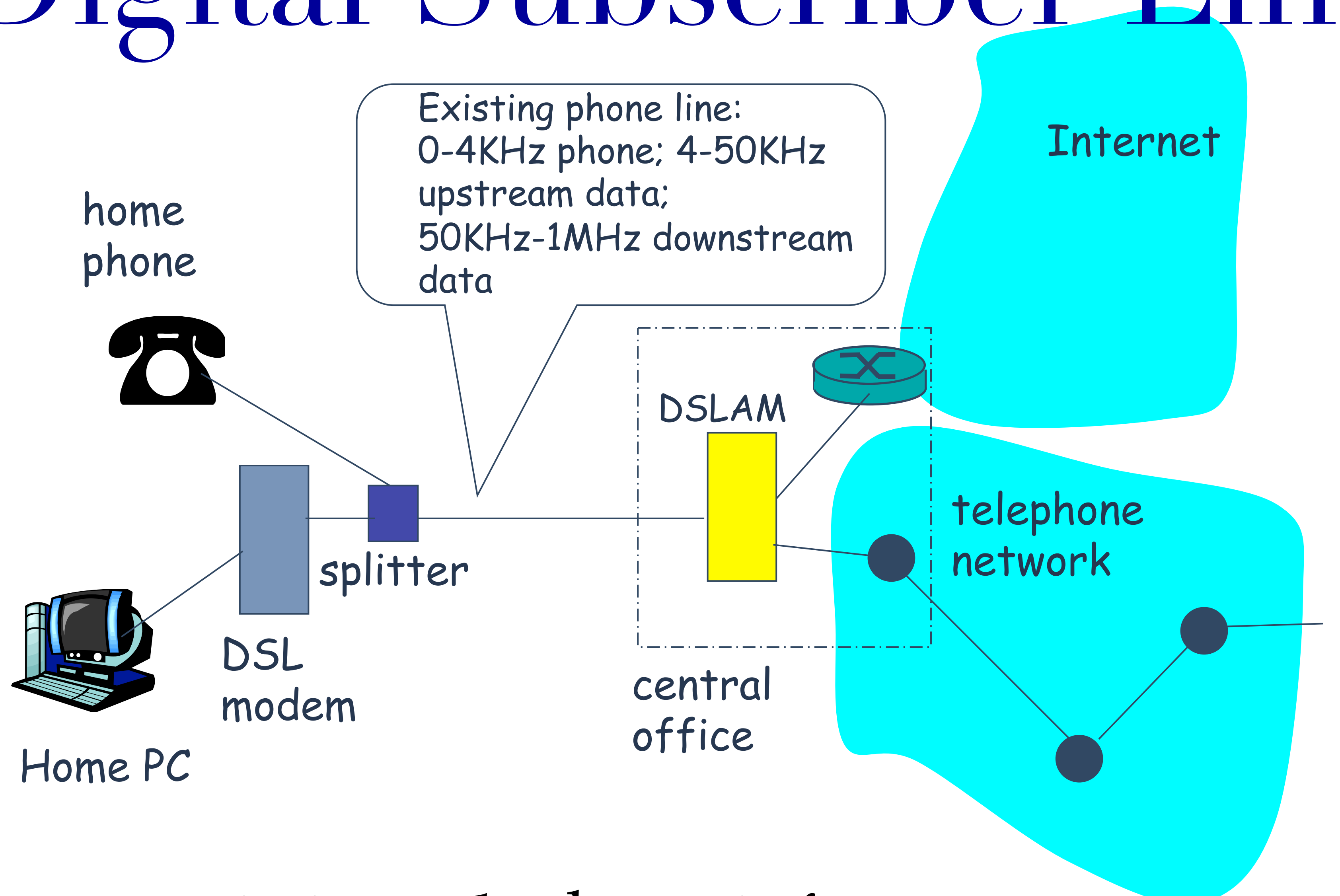


Dial-up Modem



- uses existing telephony infrastructure
  - ➡ home directly-connected to central office
- up to 56Kbps direct access to router (often less)
- can't surf, phone at same time: not "always on"

Digital Subscriber Line (DSL)



- uses existing telephone infrastructure
  - up to 1 Mbps upstream (today typically < 256 kbps)
  - up to 8 Mbps downstream (today typically < 1 Mbps)
  - dedicated physical line to telephone central office

Residential access: cable modems

- uses cable TV infrastructure, rather than telephone infrastructure
- HFC: hybrid fiber coax
  - asymmetric: up to 30Mbps downstream, 2 Mbps upstream
- network of cable, fiber attaches homes to ISP router
  - homes share access to router
  - unlike DSL, which has dedicated access

Residential access: cable modems

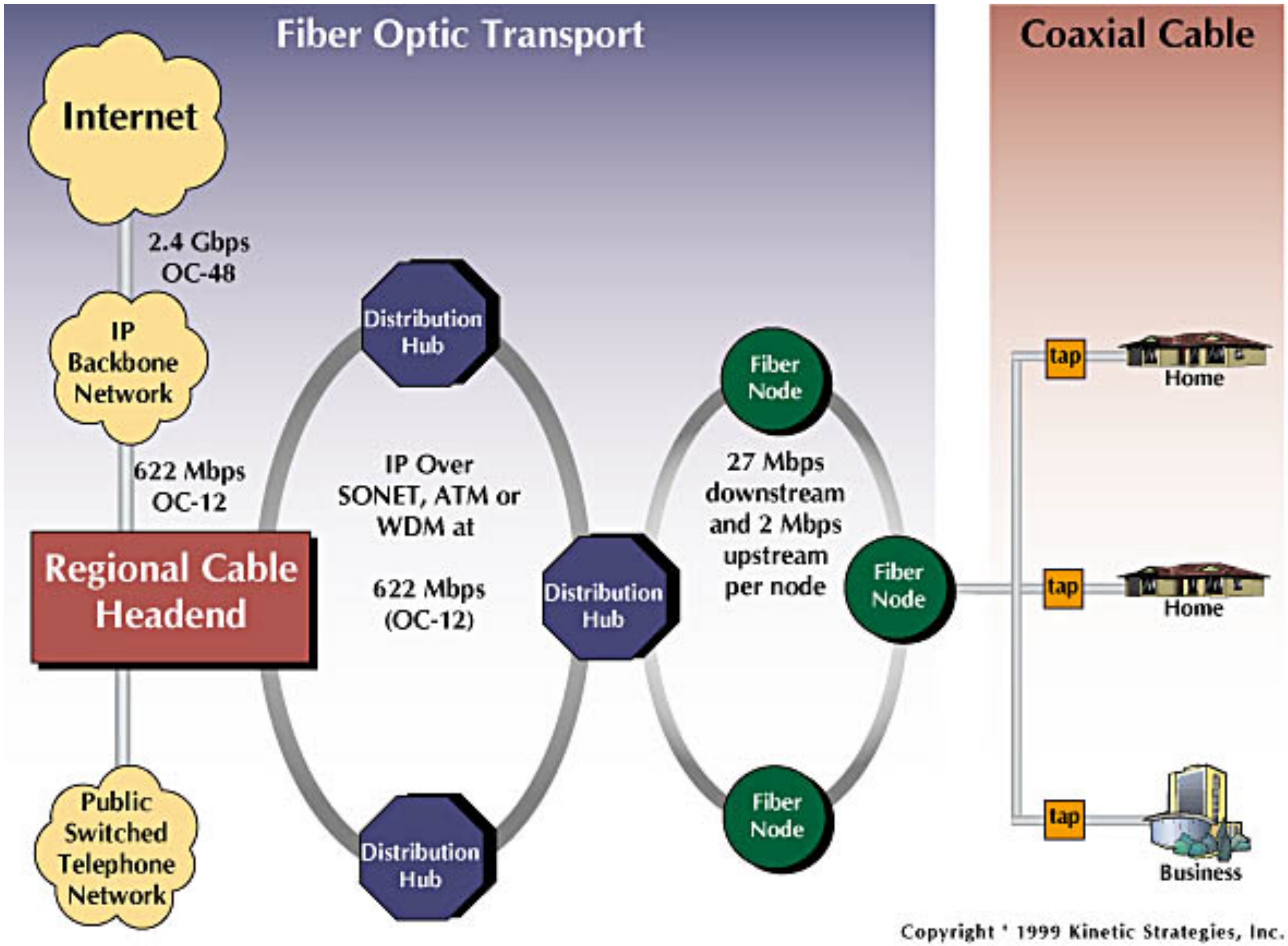
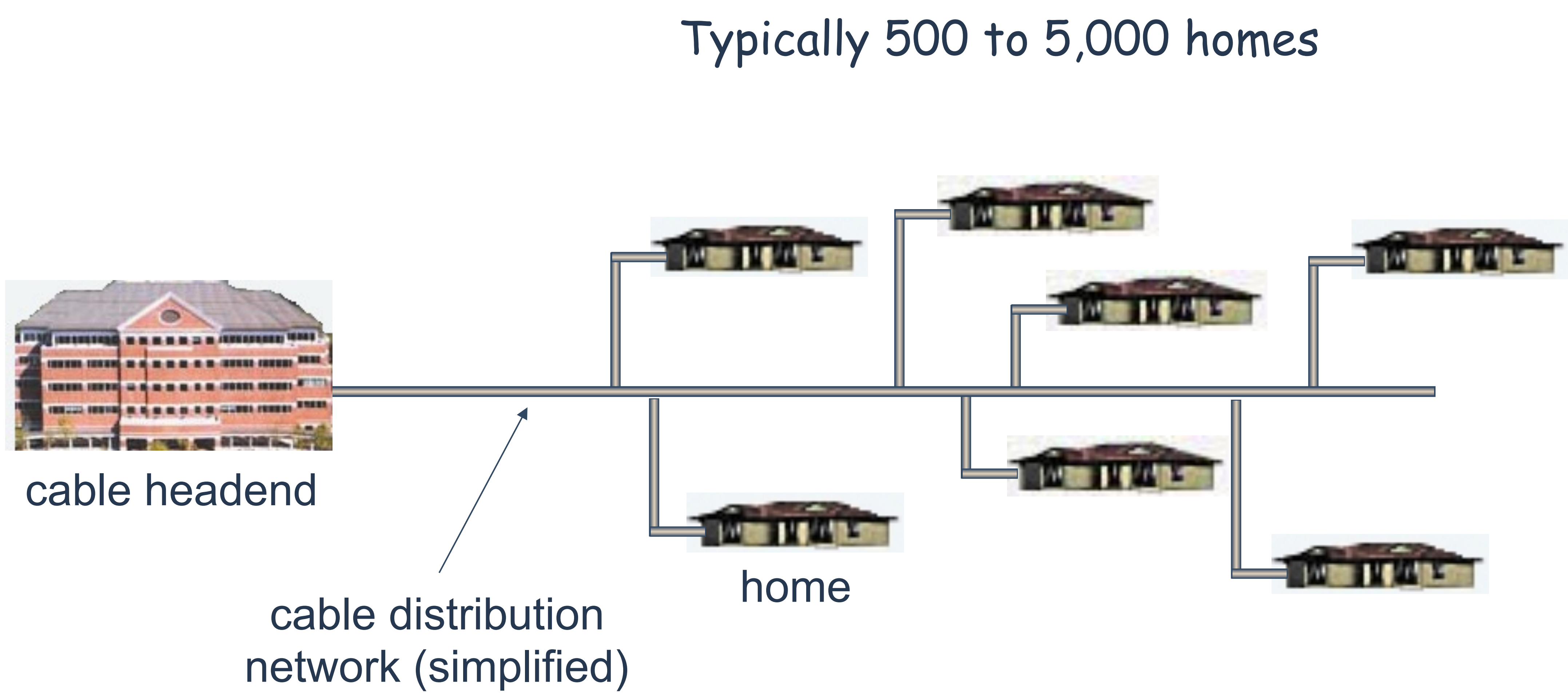
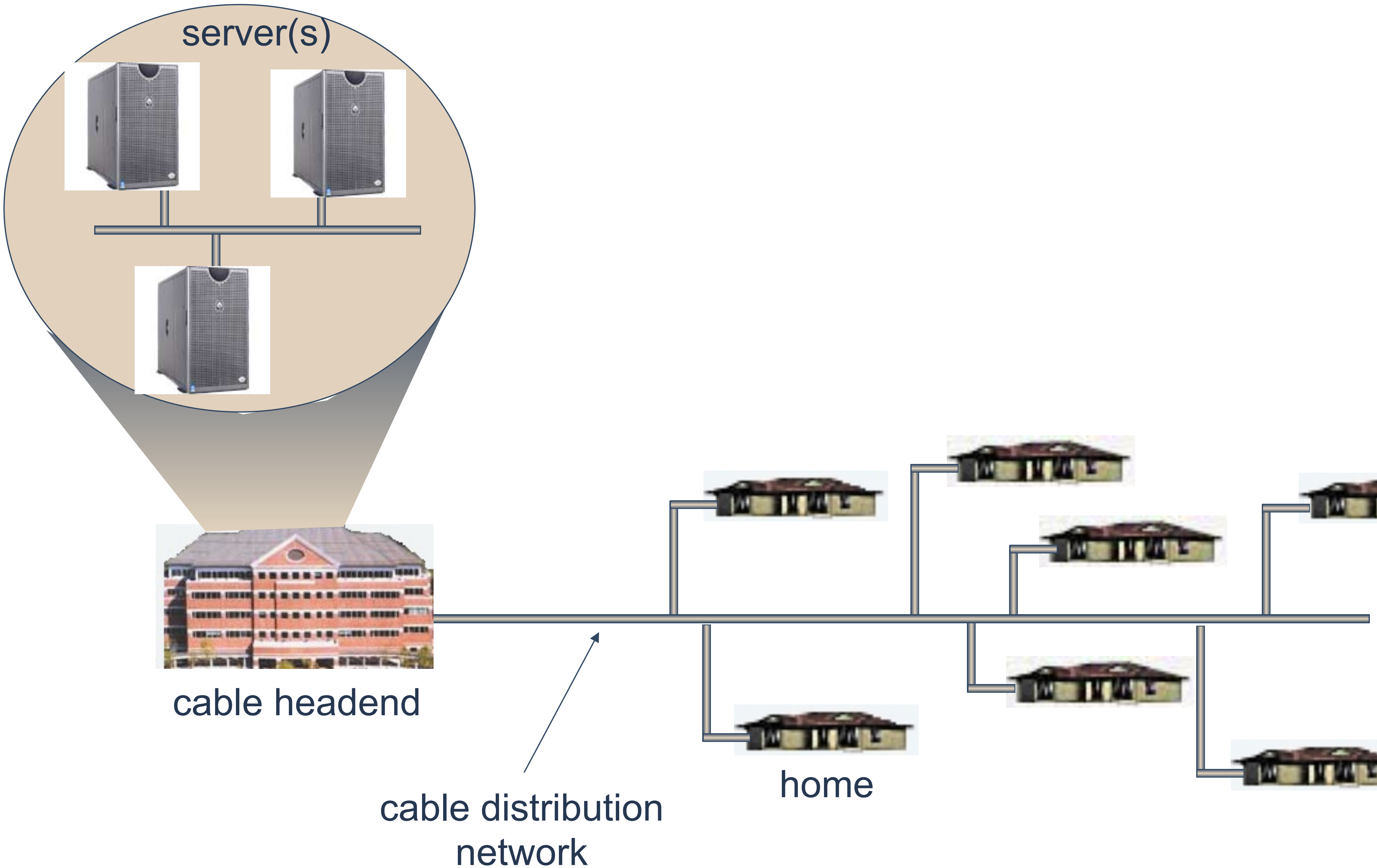


Diagram: http://www.cabledatacomnews.com/cmic/diagram.html

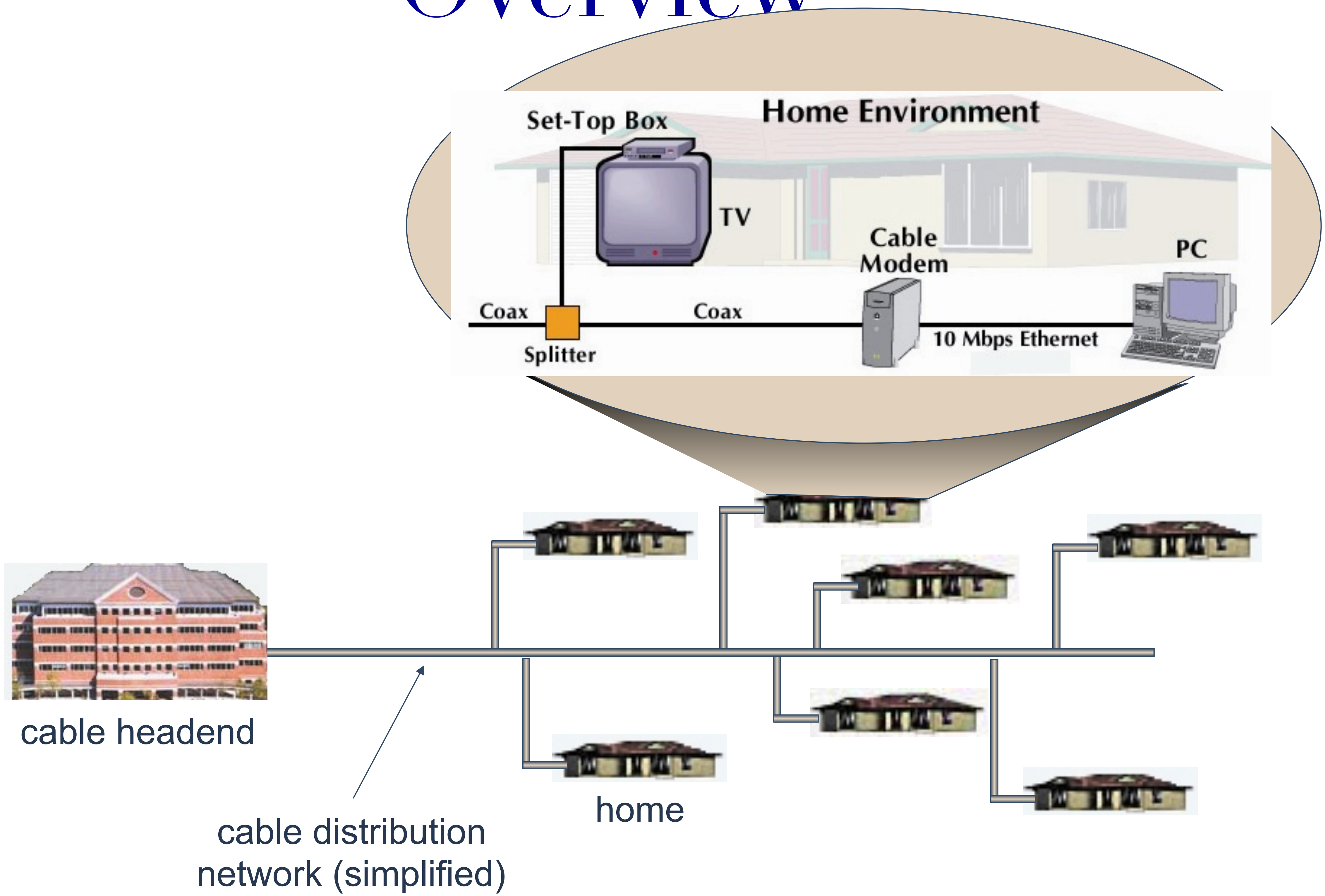
Cable Network Architecture: Overview



Cable Network Architecture: Overview



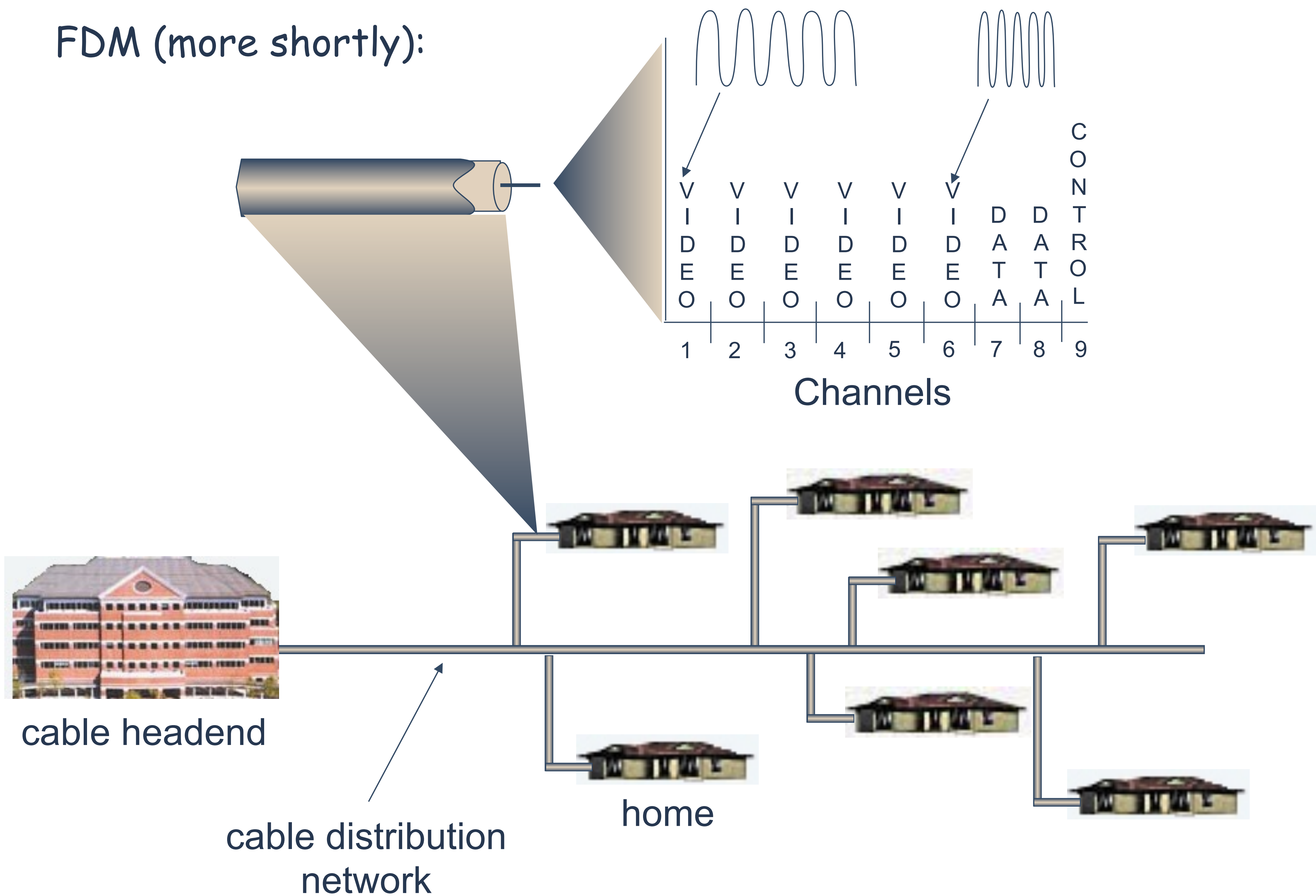
Cable Network Architecture: Overview



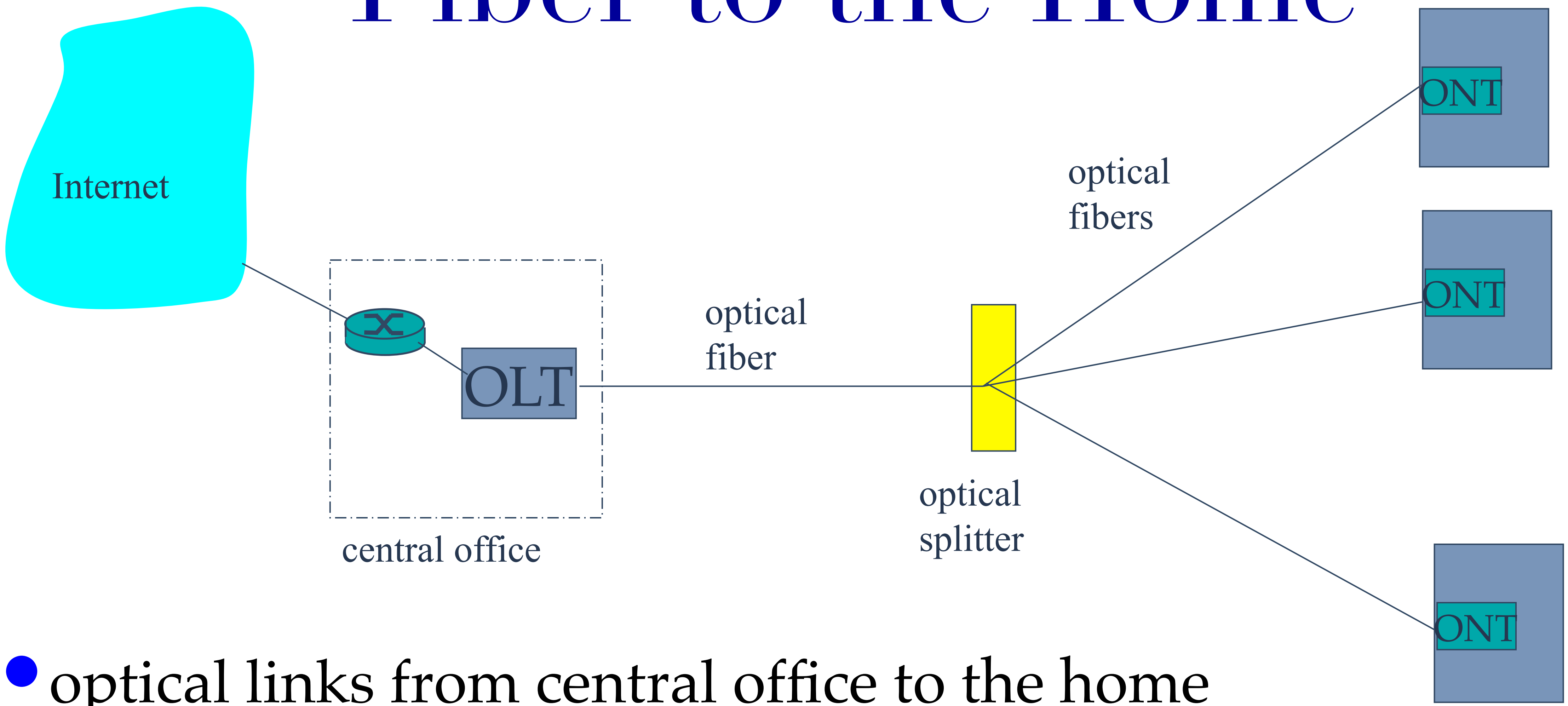
From Kurose & Ross, Computer Networking: A Top-Down Approach, 5e

Cable Network Architecture: Overview

FDM (more shortly):

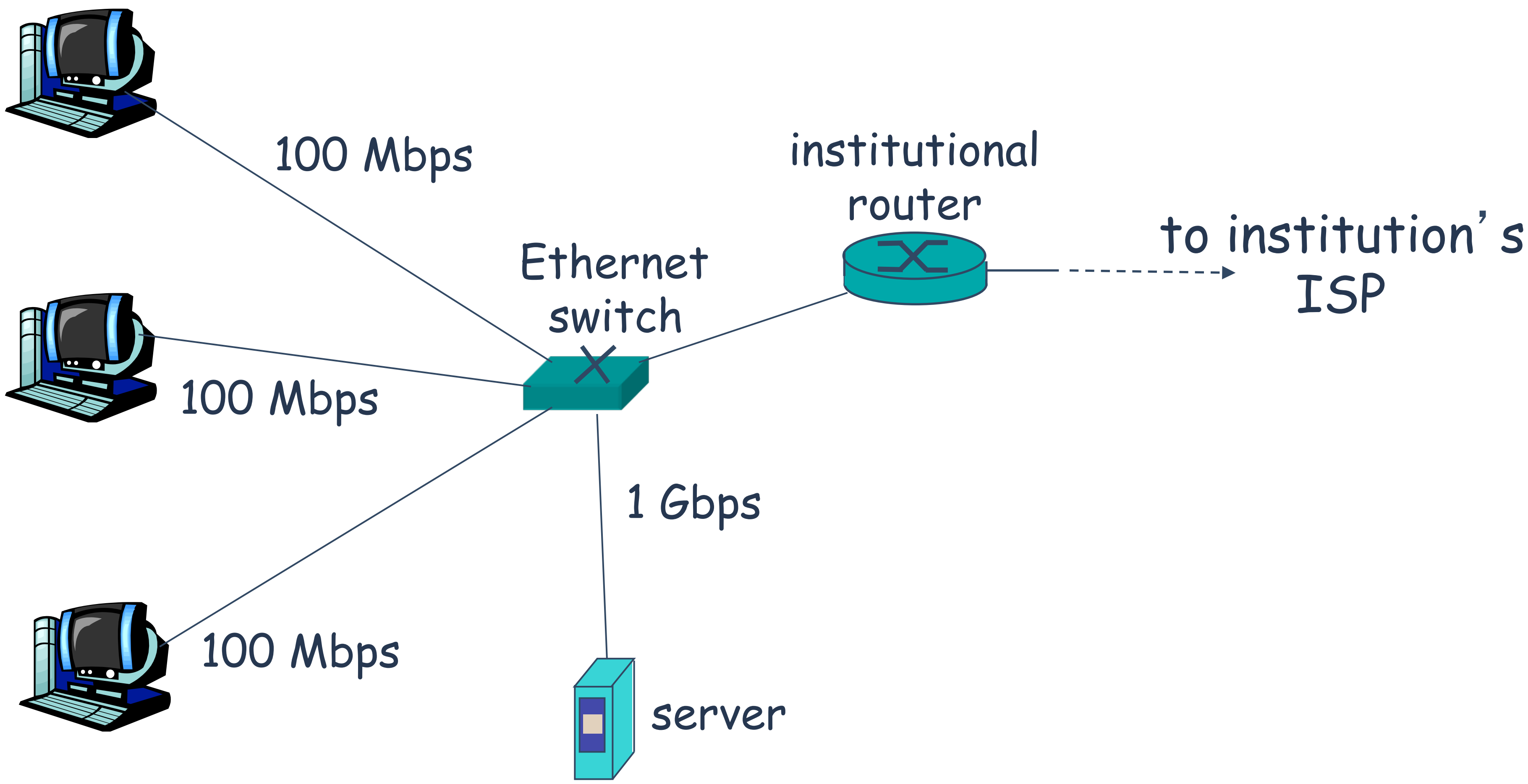


Fiber to the Home



- optical links from central office to the home
- two competing optical technologies:
  - Passive Optical network (PON)
  - Active Optical Network (PAN)
- much higher Internet rates; fiber also carries television and phone services

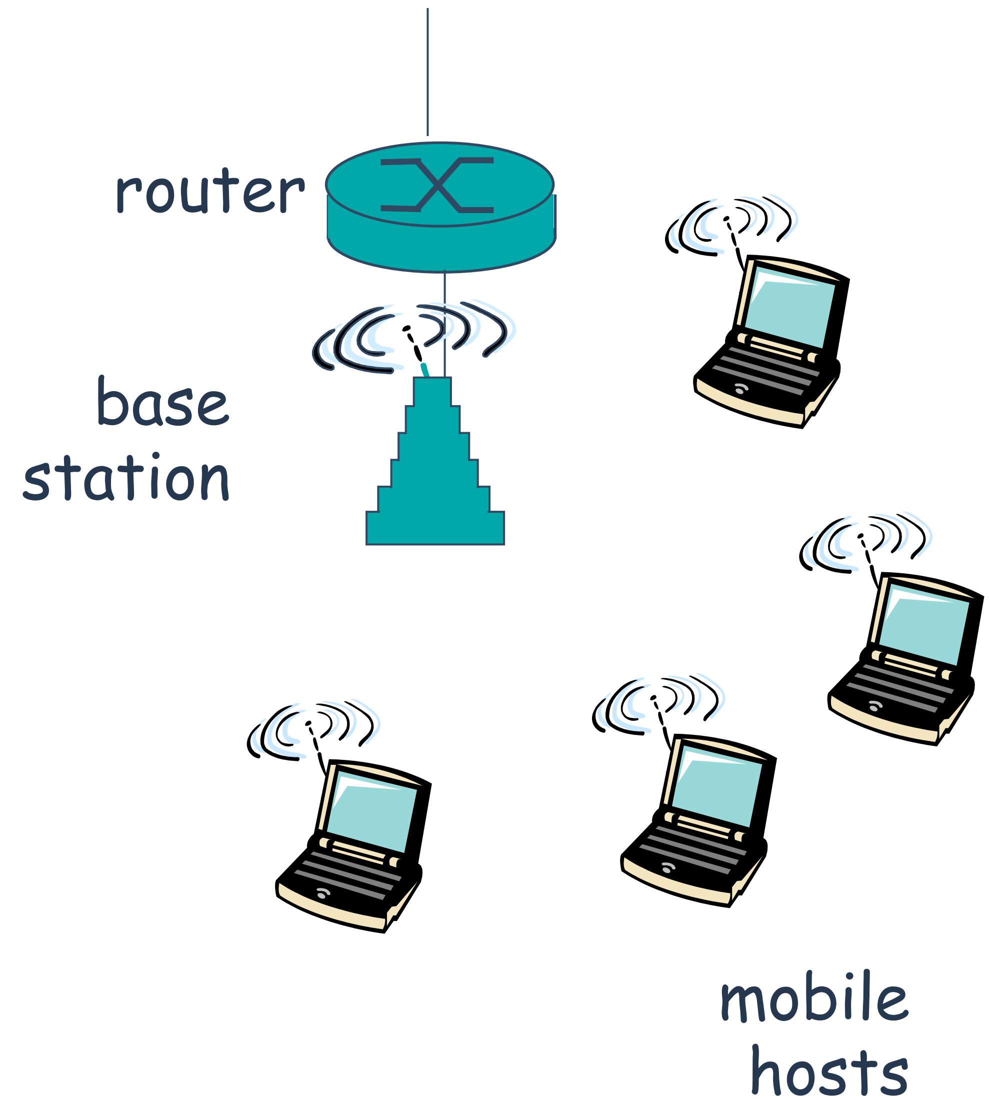
Ethernet Internet access



- typically used in companies, universities, etc
- 10 Mbps, 100Mbps, 1Gbps, 10Gbps Ethernet
- today, end systems typically connect into Ethernet switch

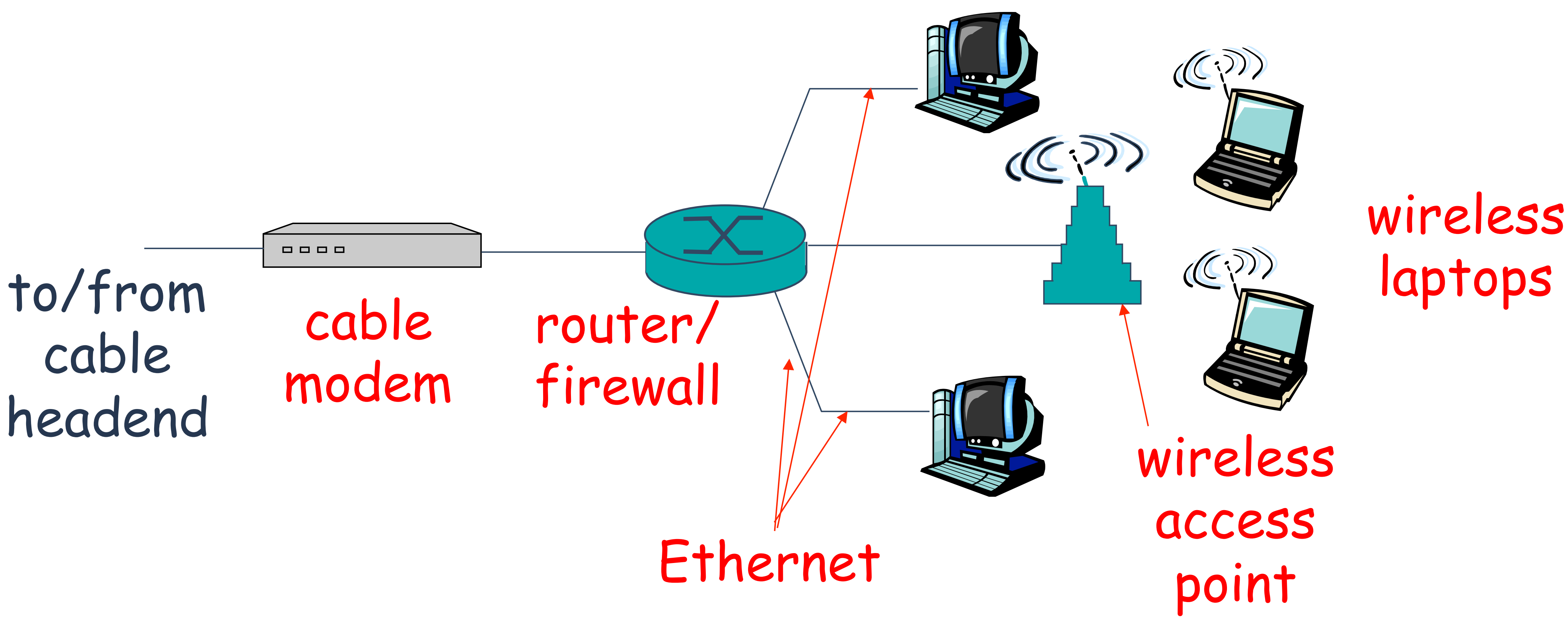
Wireless access networks

- shared *wireless* access network connects end system to router
  - via base station aka "access point"
- wireless LANs:
  - 802.11b/g (WiFi): 11 or 54 Mbps
- wider-area wireless access
  - provided by telco operator
  - ~1Mbps over cellular system (EVDO, HSDPA)
  - next up (?): WiMAX (10's Mbps) over wide area



Home networks

- Typical home network components:
  - ➡ DSL or cable modem
  - ➡ router/firewall/NAT
  - ➡ Ethernet
  - ➡ wireless access point



Physical Media

- bit: propagates between transmitter/rcvr pairs
- physical link: what lies between transmitter & receiver
- guided media:
  - signals propagate in solid media: copper, fiber, coax
- unguided media:
  - signals propagate freely, e.g., radio

Twisted Pair (TP)

- two insulated copper wires
  - Category 3: traditional phone wires, 10 Mbps Ethernet
  - Category 5: 100Mbps Ethernet

Physical Media: coax, fiber

Coaxial cable:

- two concentric copper conductors
- Bidirectional
- Baseband:
  - single channel on cable
  - legacy Ethernet
- Broadband:
  - multiple channels on cable
  - HFC

Fiber optic cable:

- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (e.g., 10's-100's Gpbs)
- low error rate: repeaters spaced far apart ; immune to electromagnetic noise

Physical media: radio

- signal carried in electromagnetic spectrum
- no physical "wire"
- Bidirectional
- propagation environment effects:
  - Reflection
  - obstruction by objects
  - interference

Radio link types:

- terrestrial microwave
  - e.g. up to 45 Mbps channels
- LAN (e.g., WiFi)
  - 11Mbps, 54 Mbps
- wide-area (e.g., cellular)
  - 3G cellular: ~ 1 Mbps
- Satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus low altitude