521150A
Introduction to Internet

Lecture 2 – Architecture, design principles and performance

Erkki Harjula 2023

Schedule of the course

PART I: Basics of networking and Internet, data link layer

Mon 13.3. 10-12 L6/Zoom	Lecture 1: Introduction & motivation
Tue 14.3. 10-12 L6/Zoom	Lecture 2: Architecture & design principles
Wed 15.3. 10-12 L5/Zoom	Lecture 3: Data link layer – basics part I
Thu 16.3. 10-12 L4/Zoom	Exercise session 1A
Mon 20.3. 10-12 L6/Zoom	Lecture 4: Data link layer – basics part II
Tue 21.3. 10-12 L6/Zoom	Exercise session 1B
Tue 21.3. 14-18 AT122/Zoom	Lab exercise 1 – group 3
Wed 22.3. 10-12 L5/Zoom	Lecture 5: Data link layer – Wired networks
Wed 22.3. 14-18 AT122/Zoom	Lab exercise 1 – group 4
Thu 23.3. 14-18 AT122/Zoom	Lab exercise 1 – group 2
Fri 24.3. 12-16 AT122/Zoom	Lab exercise 1 – group 1
Mon 27.3. 14-16 L5/Zoom	Exercise session 1C
Tue 28.3. 10-12 L6/Zoom	Lecture 6: Data link layer – Wireless networks
Wed 29.3. 10-12 Moodle	Theory exam 1

PART II: Network and transport layers

Thu 30.3. 8-10 L5/Zoom	Lecture 7: Network layer part I
Mon 3.4. 14-16 L5/Zoom	Exercise session 2A
Tue 4.4. 10-12 L6/Zoom	Lecture 8: Network layer part II
Wed 5.4. 10-12 L5/Zoom	Lecture 9: Transport layer part I; Course work intro
Tue 11.4. 10-12 L6/Zoom	Lecture 10: Transport layer part II
Tue 11.4. 14-18 AT122/Zoom	Lab exercise 2 – group 3
Wed 12.4. 10-12 Moodle	Theory exam 2
Wed 12.4. 14-18 AT122/Zoom	Lab exercise 2 – group 4
Thu 13.4. 8-10 L5/Zoom	Exercise session 2B
Thu 13.4. 14-18 AT122/Zoom	Lab exercise 2 – group 2
Fri 14.4. 12-16 AT122/Zoom	Lab exercise 2 – group 1

PART III: Application layer, network security and multimedia

Fri 14.4. 10-12 L5/Zoom	Lecture 11: Networking applications
Mon 17.4. 14-16 L5/Zoom	Lecture 12: Network security
Tue 18.4. 10-12 L6/Zoom	Exercise session 3A
Wed 19.4. 10-12 L5/Zoom	Lecture 13: Multimedia and QoS
Mon 24.4. 14-16 L5/Zoom	Exercise session 3B
Tue 25.4. 14-18 AT122/Zoom	Lab exercise 3 – group 3
Wed 26.4. 10-12 L5/Zoom	Lecture 14: Challenges&Future Internet trends
Wed 26.4. 14-18 AT122/Zoom	Lab exercise 3 – group 4
Thu 27.4. 14-18 AT122 /Zoom	Lab exercise 3 – group 2
Fri 28.4. 12-16 AT122 /Zoom	Lab exercise 3 – group 1
Wed 3.5. 10-12 Moodle	Theory exam 3
Thu 25.5. 16-19 L1	Final exam

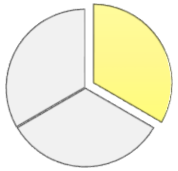
Course work (independent work)

Main learning objectives of this lecture

1. Know the architecture and building blocks of the Internet
2. Understand the design principles of the Internet and their realization
3. Understand the main differences between Packet switching and circuit switching networks
4. Be aware of key performance attributes of packet switched networks

Architecture and Building blocks of the Internet

Formal definition of ”Internet” by US Federal Networking Council (1995)



"Internet" refers to the global information system that:

i. Is logically linked together by a **globally unique address space based on the Internet Protocol (IP)** or its subsequent extensions/follow-ons;

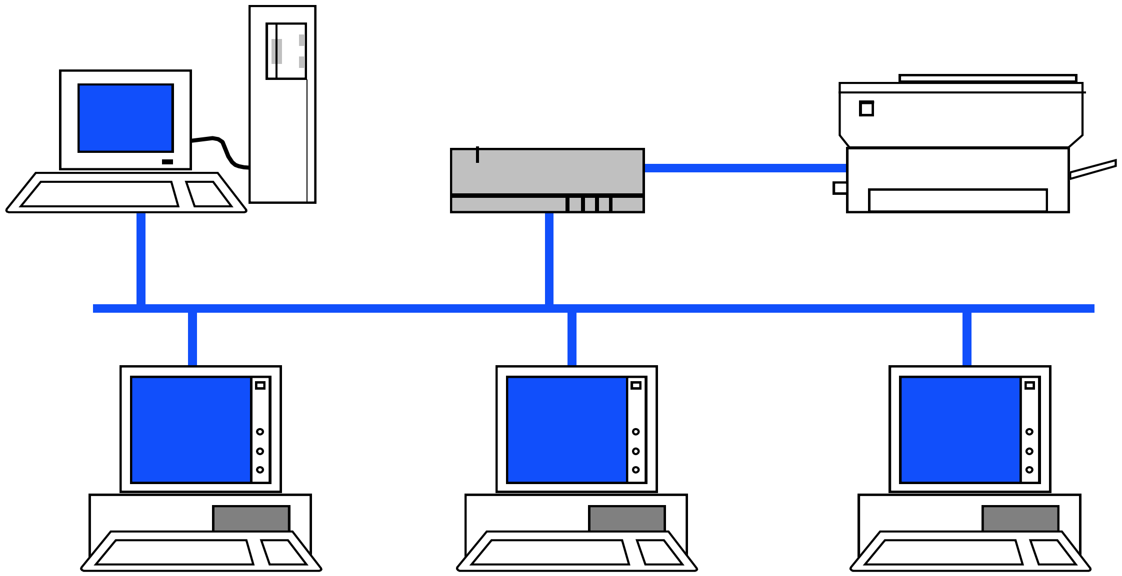
ii. Is able to support communications using the **Transmission Control Protocol/Internet Protocol (TCP/IP) suite** or its subsequent extensions/follow-ons, and/or other IP-compatible protocols; and

iii. Provides, uses or makes accessible, either publicly or privately, **high level services layered on the communications and related infrastructure** described herein

Internet is a large computer network

– **Computer network** (Tanenbaum) ~ ”A collection of autonomous computers interconnected by a single technology.”

= ”collection of data links”



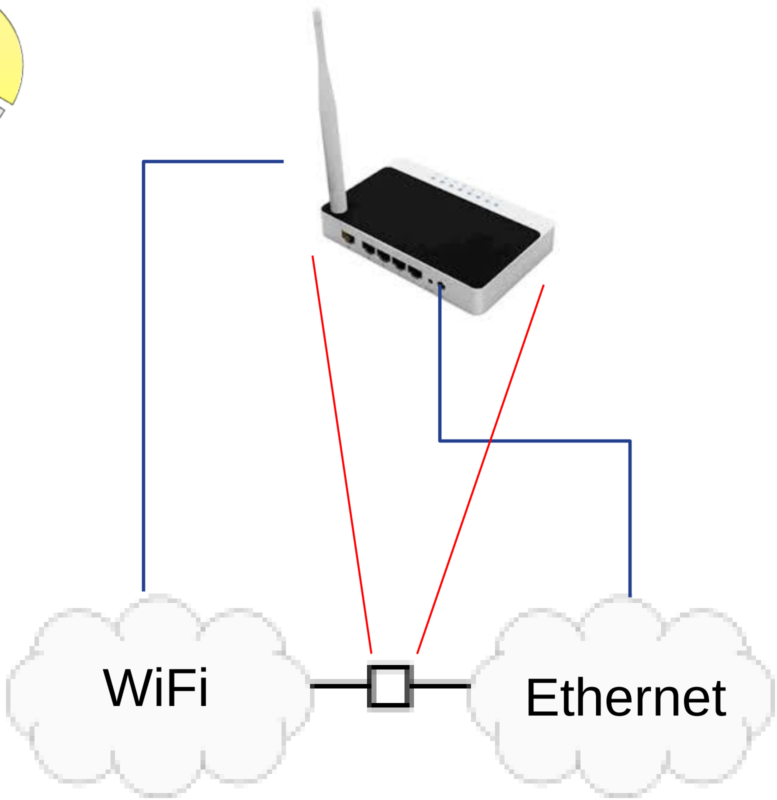
– **Distributed system** ~ ”A collection of independent computers appearing to its users as a single coherent system, e.g. WWW.”

Definitions & concepts

- **Definition**: A system of connected physical networks is known as an internet(work).
- **Motivation**: No single networking technology is best for all needs.
- **The concept of universal service**: A communication system that supplies universal service allows arbitrary pairs of computers to communicate.
- **Universal service in a heterogeneous world**: Although universal service is highly desirable, incompatibilities among network hardware and physical addressing prevent from building a bridged network that includes arbitrary technologies.

=> Need for higher-layer protocols

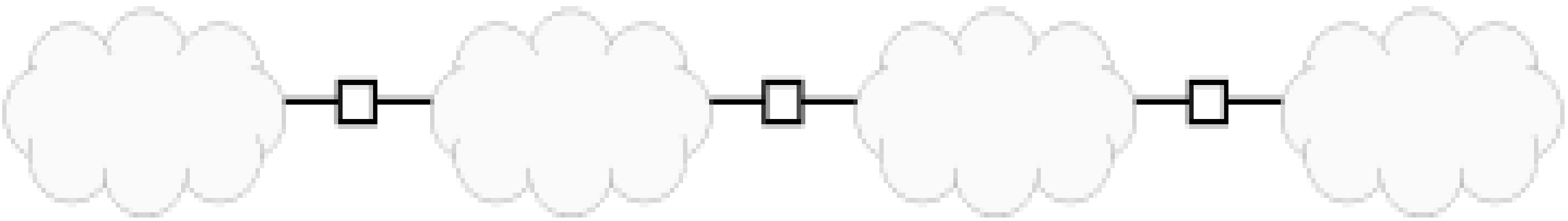
Physical network connection with routers



- **Router**
  - A special purpose system dedicated to the task of interconnecting networks.
  - A router can interconnect networks that use different technologies, including different:
    - media,
    - physical addressing schemes, and
    - frame formats.
- **Example (on the left):**
  - Two physical networks connected by a router, which has a separate interface for each network connection.
    - Computers can attach to each network.
    - E.g. Ethernet/WiFi router

Internet architecture

- Internet consists of a set of networks interconnected by routers.
- Internet scheme allows organizations to choose:
  - The **number** of and **type** of networks,
  - The number of **routers** to use to interconnect them, and
  - The interconnection **topology**.

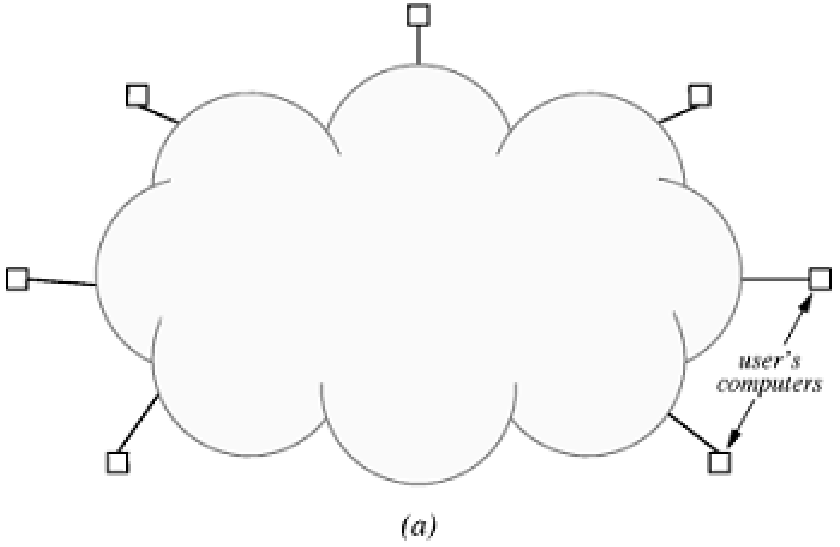


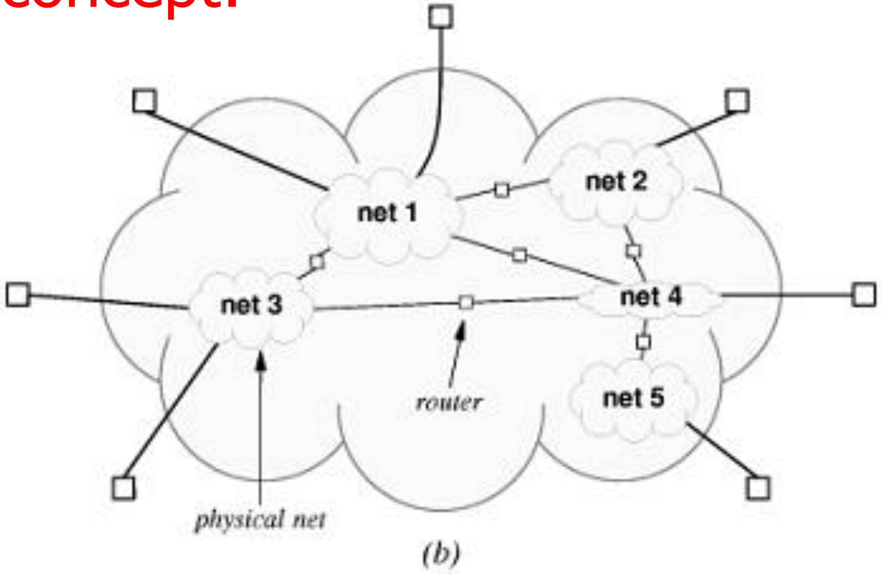
An internetwork formed using three routers to interconnect four physical networks.

Virtual network

- Internet provides the appearance of a single seamless communication system (”universal service”) to which many computers attach.
- An internet is a **virtual network** system because the communication system is an abstraction – no uniform network system exists.

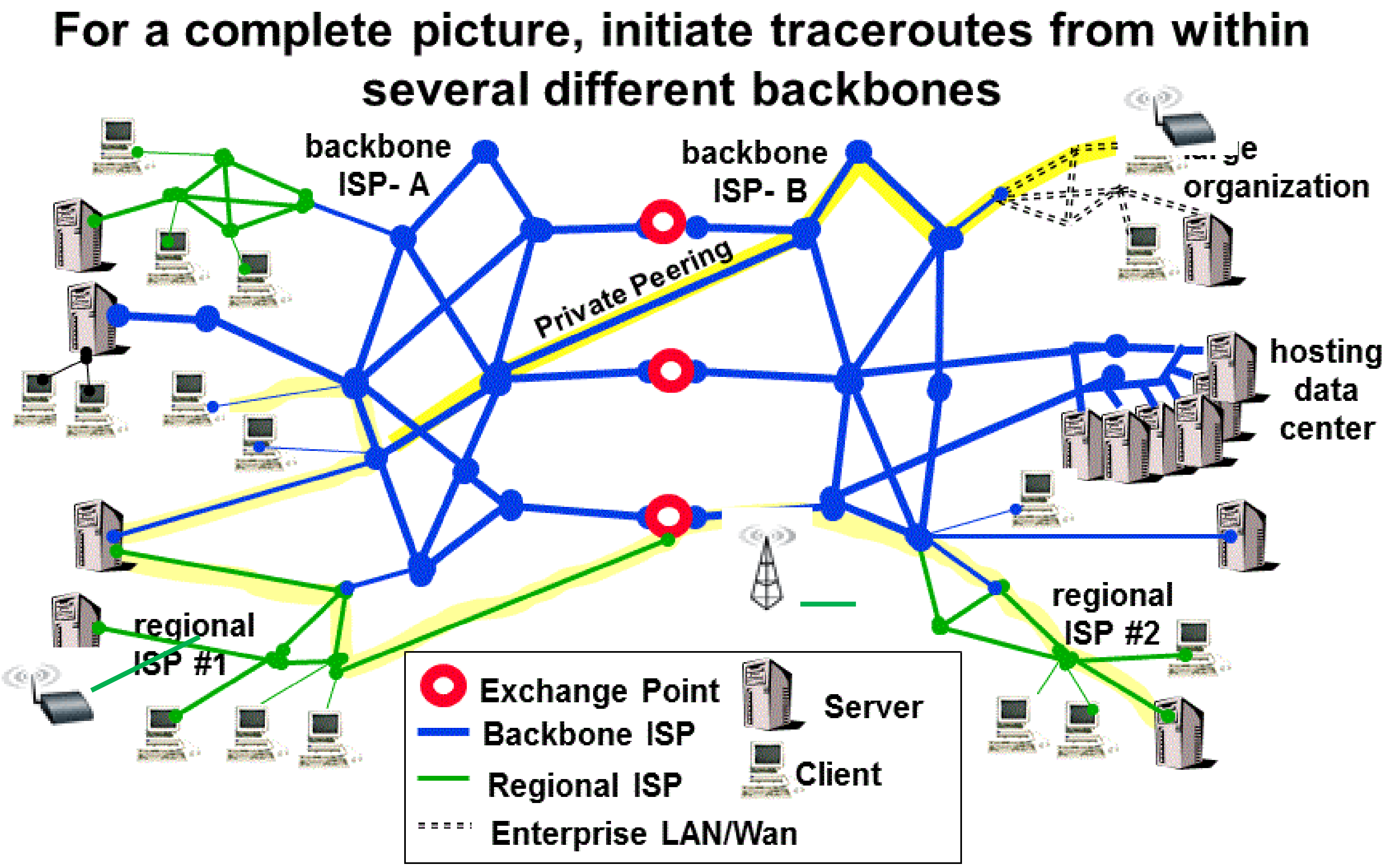
The internet concept:





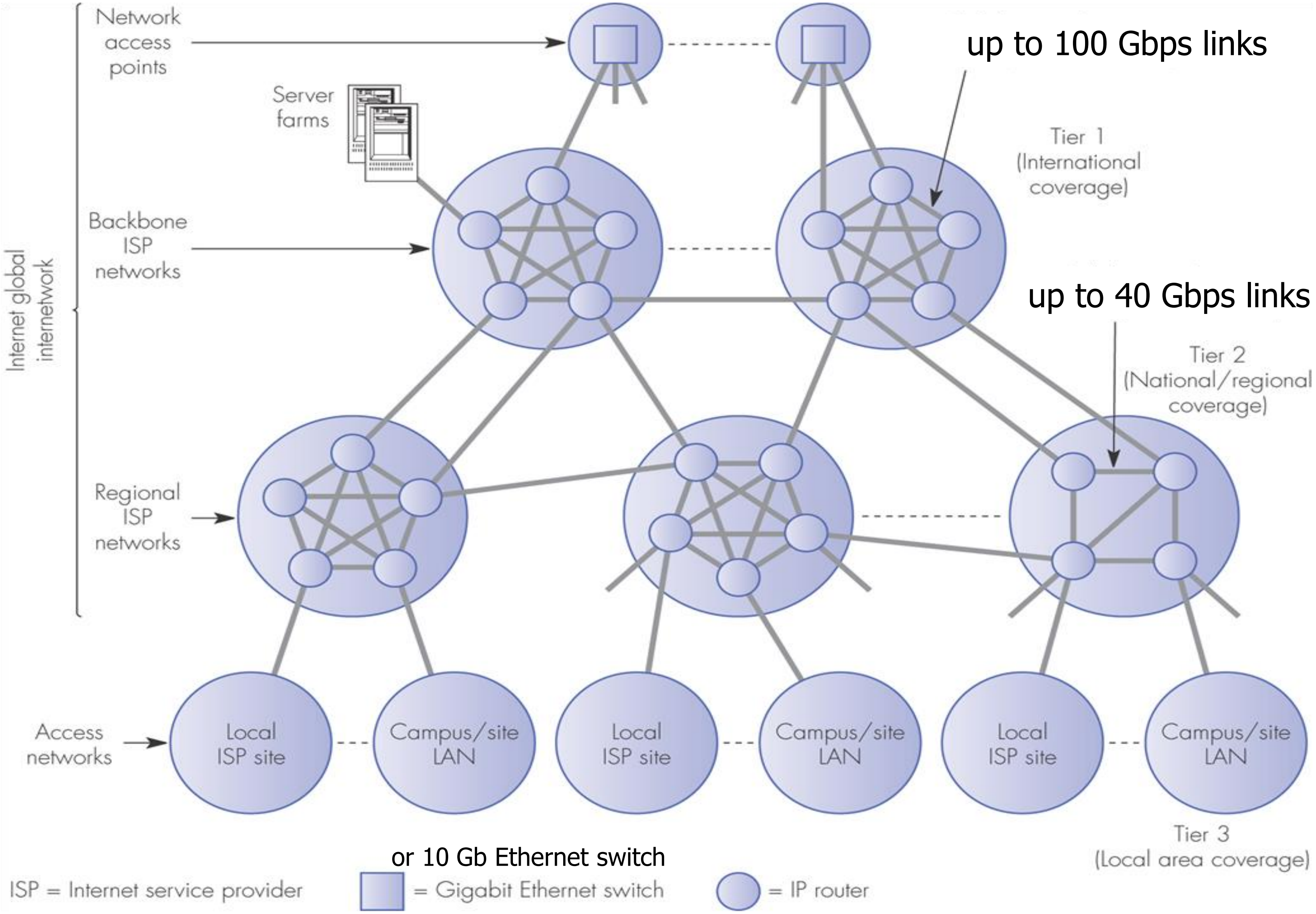
(a) the illusion of a single network; (b) the underlying physical structure.

Simplified general Internet architecture

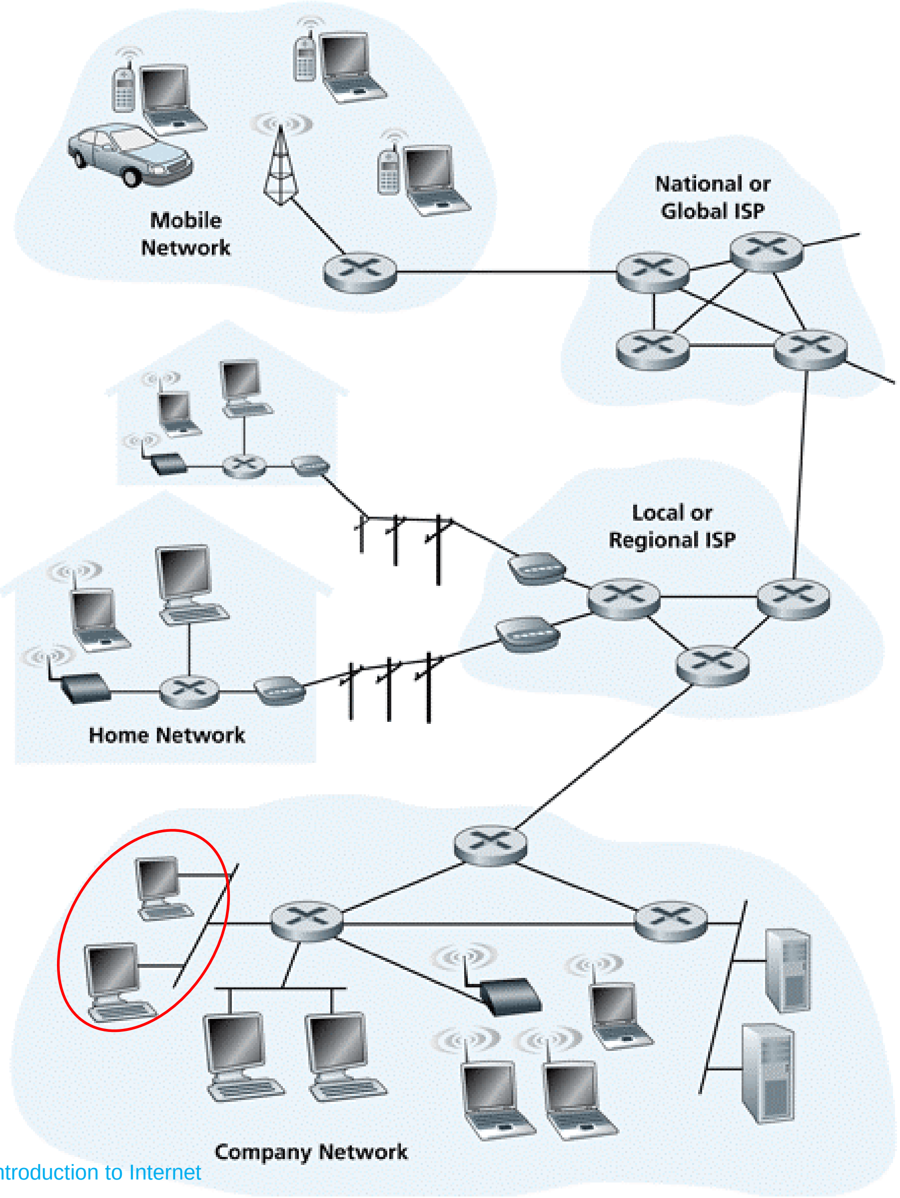


Source: *Information Navigators, Russ Raynal*

Hierarchical architecture of public Internet



Internet building blocks



- **Hosts**
  - Millions of connected computing devices (hosts, nodes)
  - Run networking applications

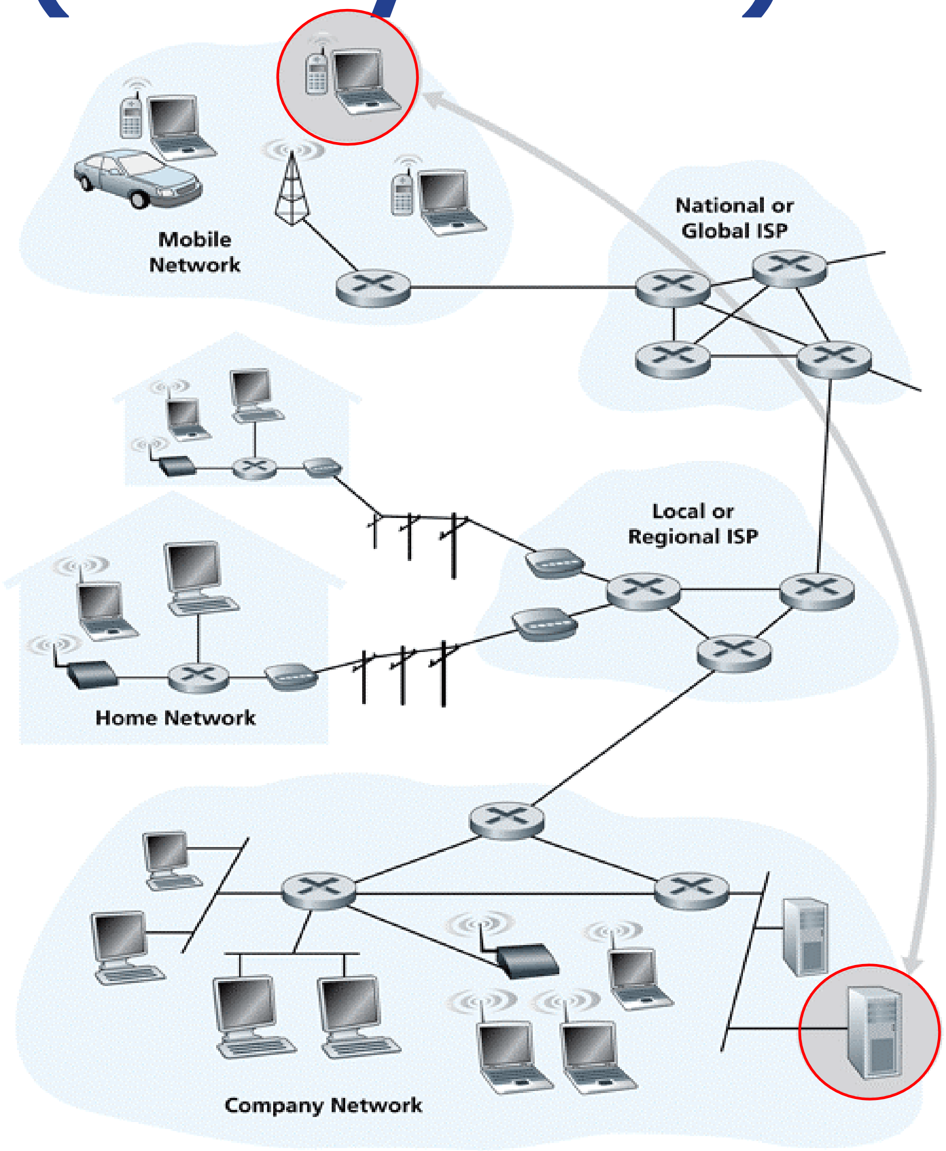
- **Links**
  - Physical medium (fiber, copper, coaxial, radio) comprising communication links between devices
  - Transfer data (bits) back and forth
- **Routers**
  - Interconnect networks
  - Forward packets (chunks of data)

Network edge: Hosts (end systems)

- **Hosts run application programs**
  - E.g. web, email
  - Communicate by sending messages using a well-defined protocol

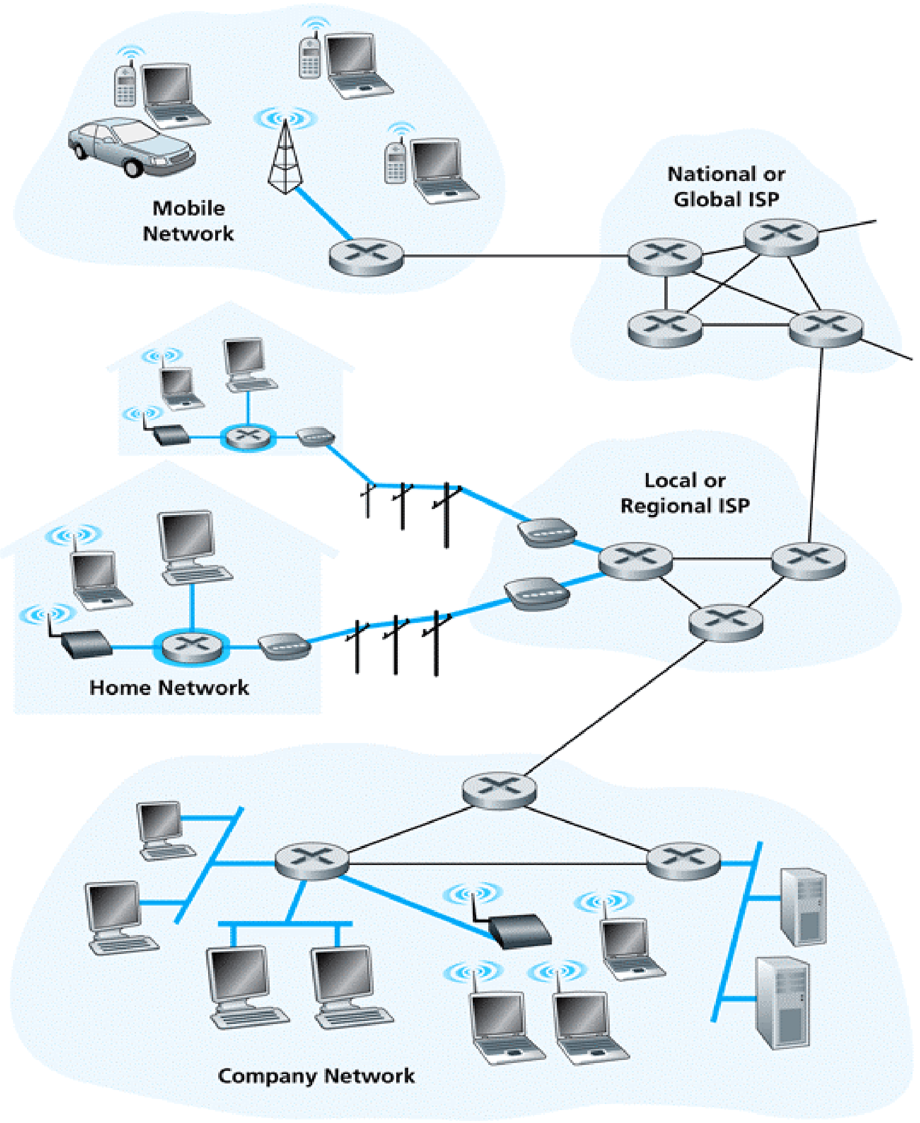
- **Client/server model:**
  - Client host requests, receives service from always-on server
  - E.g. web browser/server; email client/server

- **Peer-to-peer model:**
  - Hosts simultaneously act as clients and servers
  - E.g. BitTorrent, KaZaA, DC++



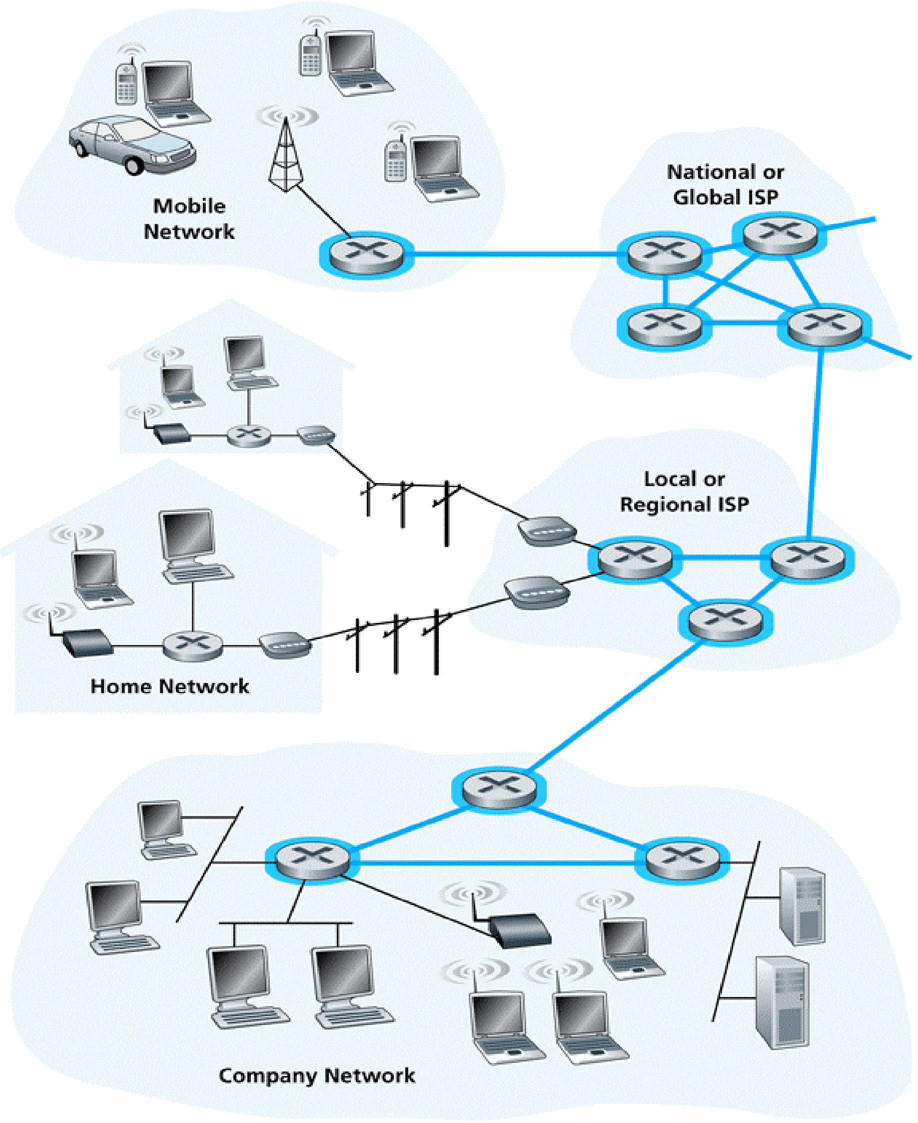
Network edge: Access networks

- **Access networks connect hosts to edge router**
- **Wired access networks**
  - Copper (xDSL, cable modem, Ethernet)
  - Optics (FTTH)
- **Wireless access networks**
  - Wireless: WiFi, Mobile networks (3G, 4G/LTE, 5G)
- **Keep in mind…**
  - Data rate (bandwidth) of access network?
  - Quality of access network?
  - Shared or dedicated?



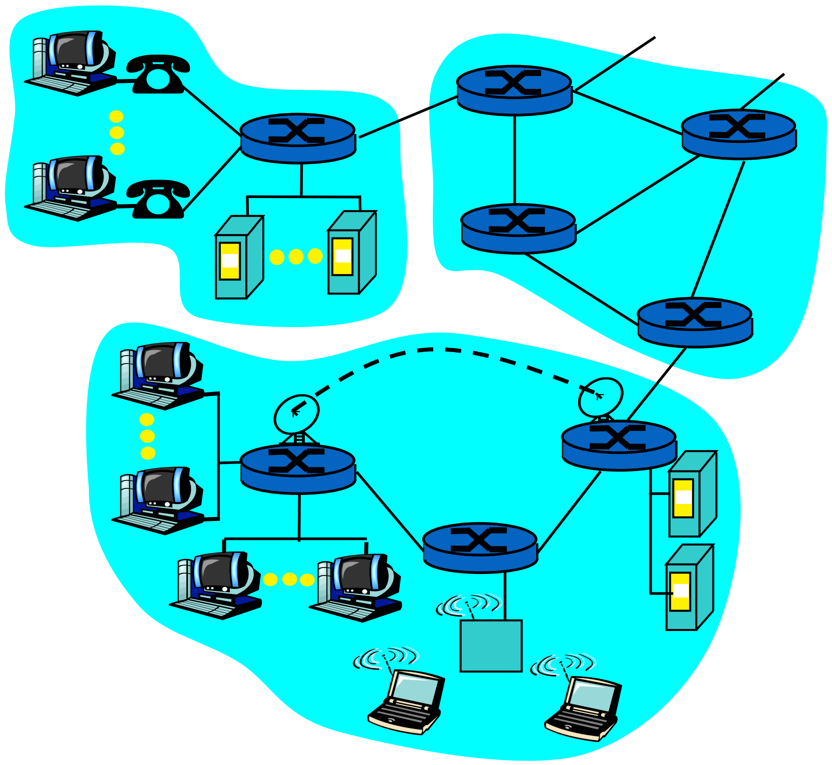
Network core

- **Core network connects access networks with each other through mesh of interconnected routers**
- **Internet(working)**
  - Network of networks
- **Fundamental question: how is data transferred through the network?**
  - Circuit switching
  - Packet switching
  - We'll return to this later …



Service viewpoint

- **Communication infrastructure enables distributed applications**
  - Provides a well-defined **API** (Application Programming Interface)
  - Web, email, games, e-commerce, database, voting, file sharing, etc.



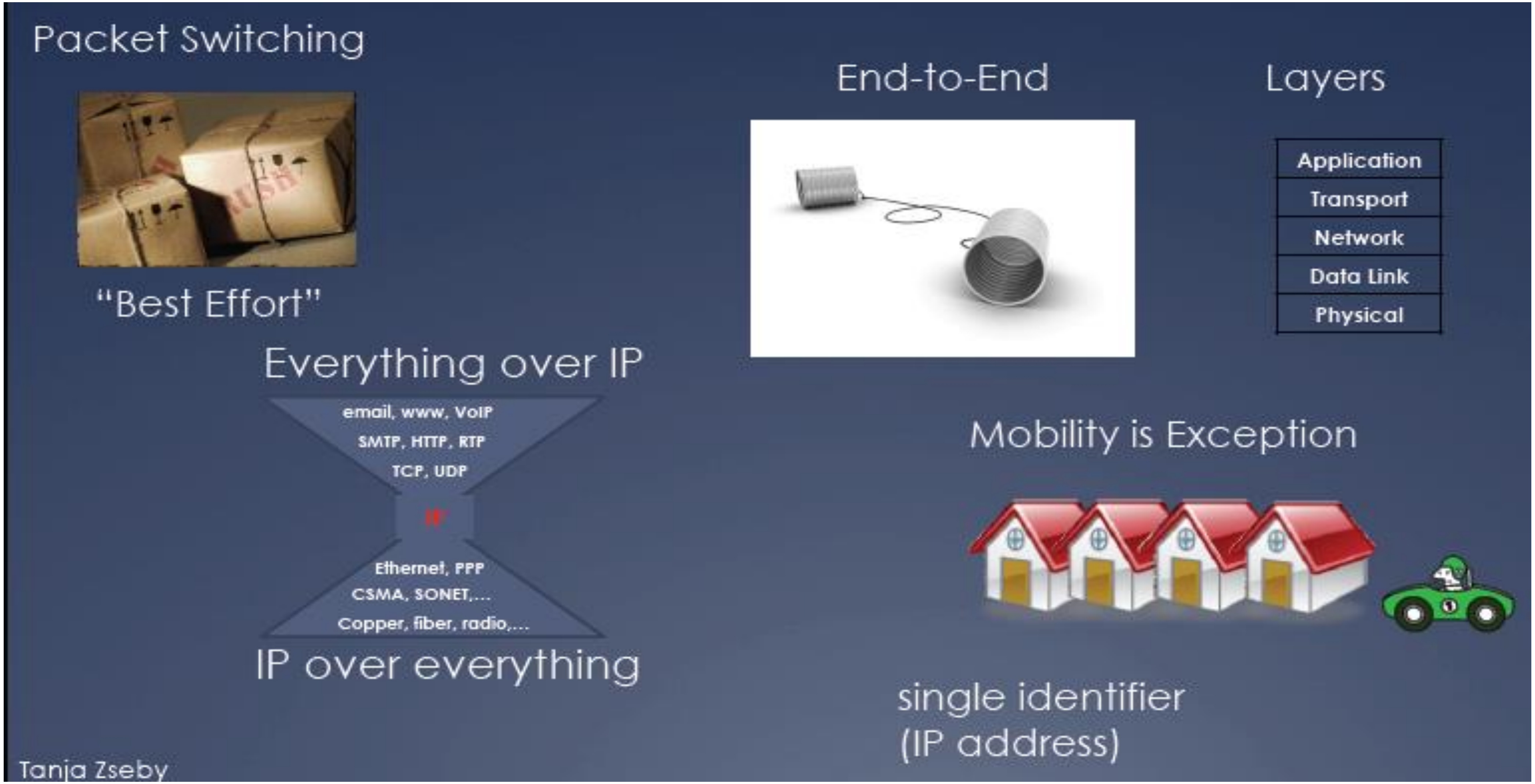
- **Communication services provided to applications**
  - Reliable vs unrealiable
  - Connection-oriented vs connectionless

- Cyberspace (Gibson):

  "A consensual hallucination experienced daily by billions of operators, in every nation, ..."

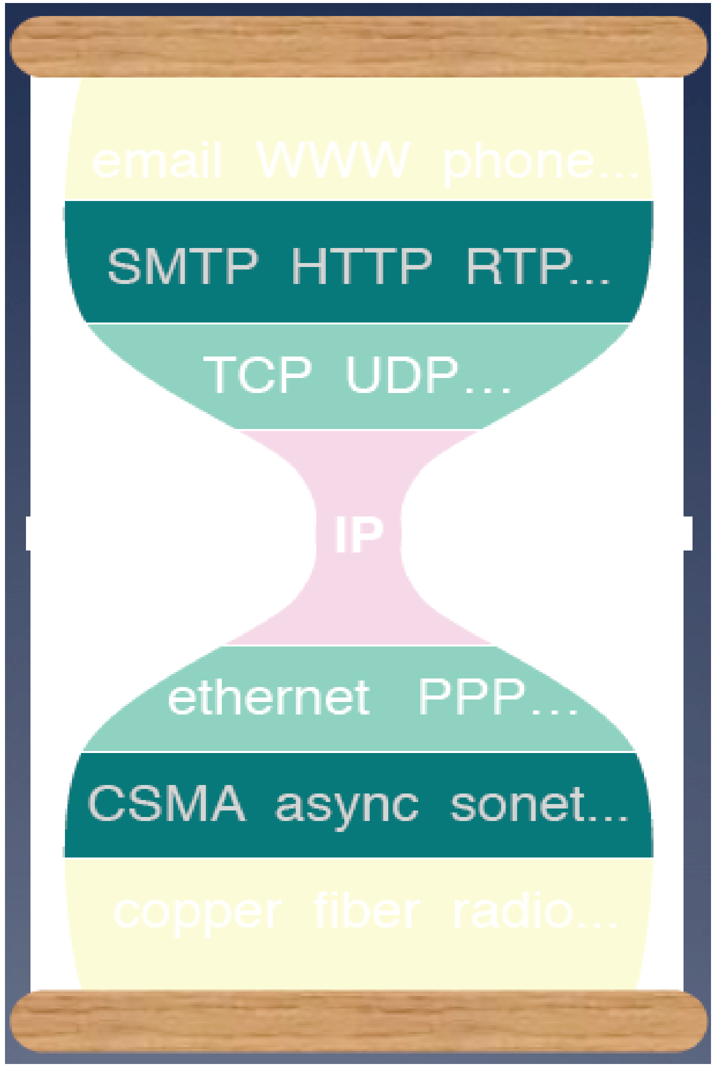
Internet design principles & layered architecture

Internet's original design principles



IP: Everything over IP & IP over everything

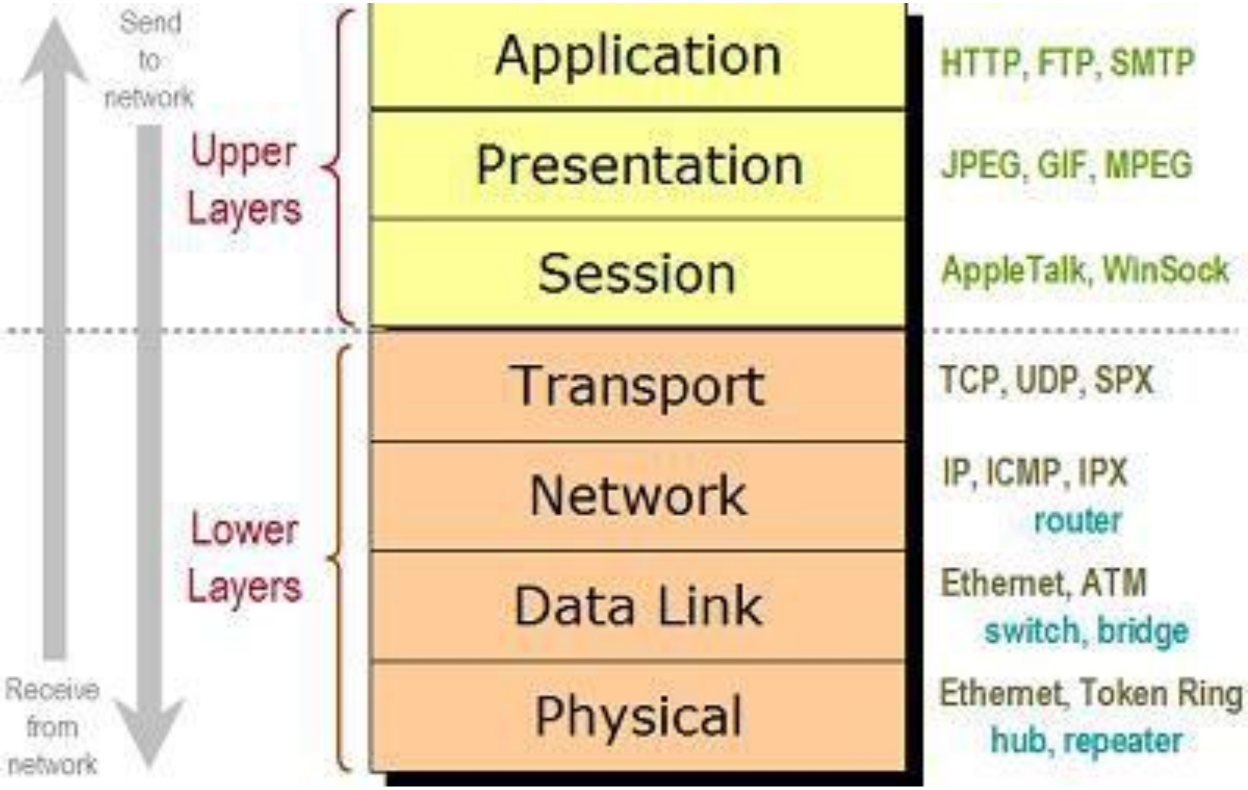
- **Why an internet layer?**
  - Make a bigger network
  - Global addressing
  - Virtualize network to isolate end-to-end protocols from network details/changes
- **Why a single internet protocol?**
  - Maximize interoperability
  - Minimize number of service interfaces
- **Why a narrow internet protocol?**
  - Assumes least common network functionality to maximize number of usable networks



(Deering 1998)

Layered design

- **For the purpose of mastering the complexity, most networks are organized as a stack of layers or levels**
  - Layer N uses services provided by lower layer N-1 and provides services to upper layer N+1
  - Peers in a given layer communicate using a well-defined **protocol**



OSI model (as example)

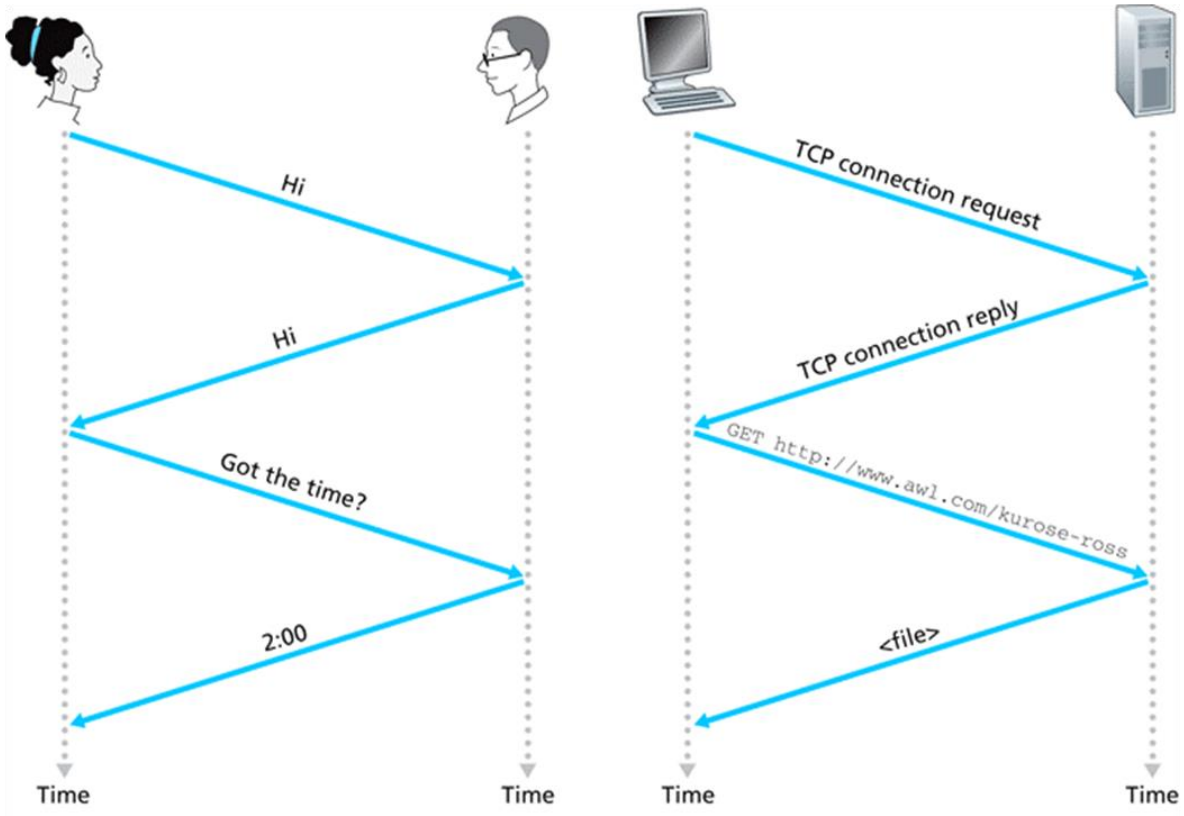
- Explicit structure allows identification & relationship of complex system's pieces
- Modularization eases maintenance & updating of systems
  - Changes within one layer transparent for the rest of system

Layered design: Protocols

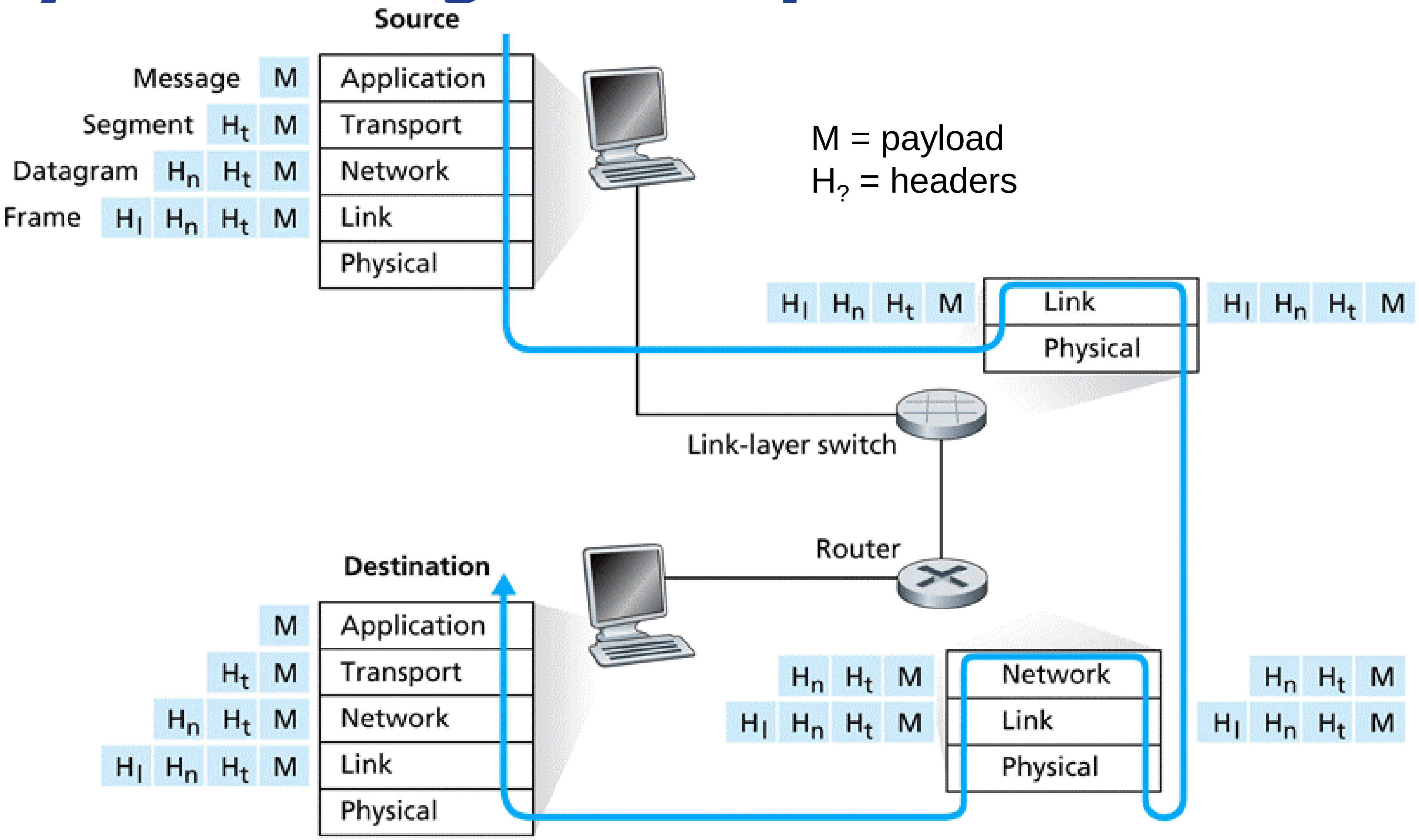
– Protocol defines:

- The format and the order of messages exchanged between two or more communicating entities
  - Message (PDU, protocol data unit) comprises of application data (payload) and protocol control information (header)
- The actions taken on the transmission and/or receipt of a message or other event
  - Protocol is a state machine !

Header	*Payload (application specific)*



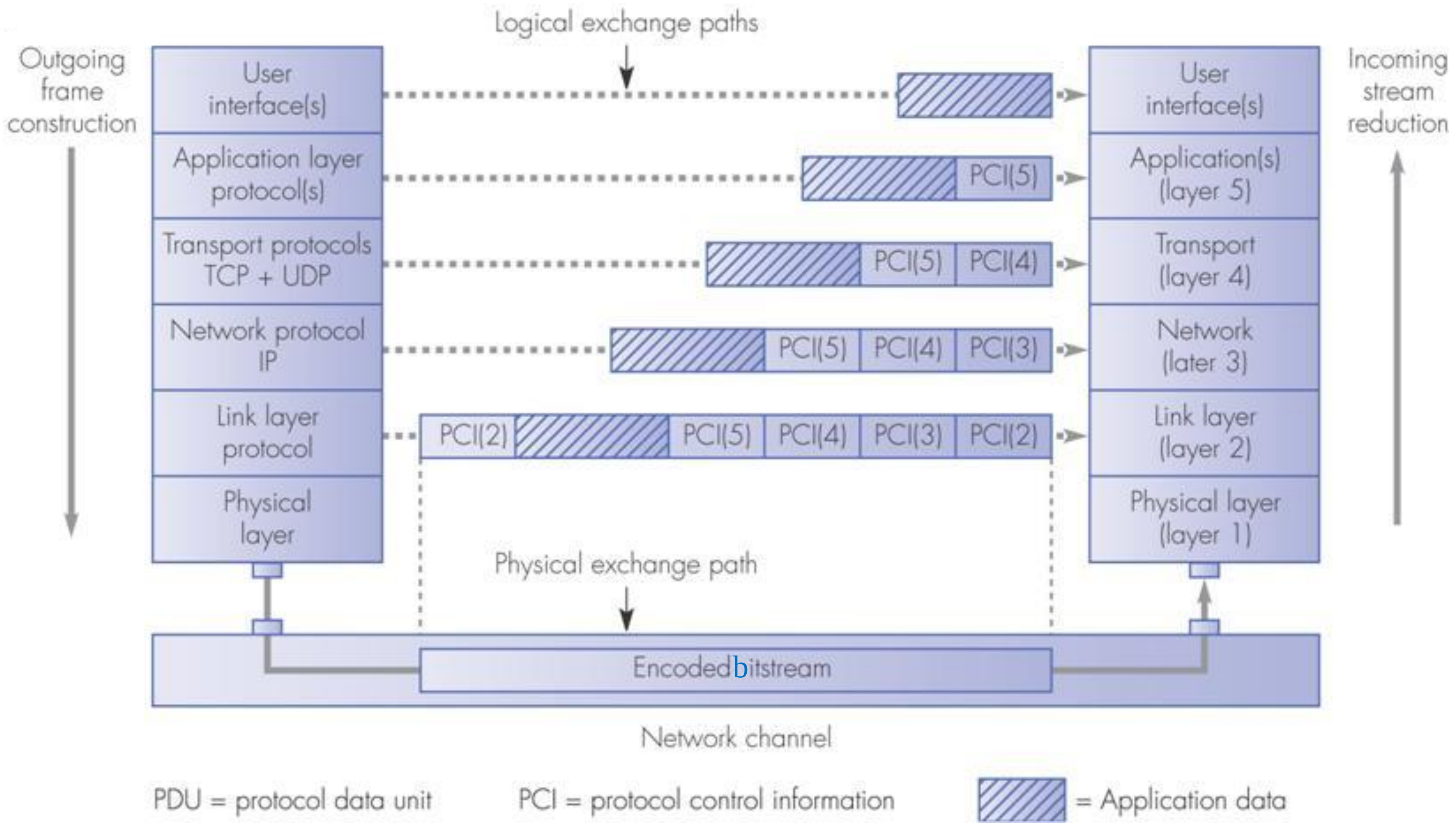
Layered design: Encapsulation



M = payload

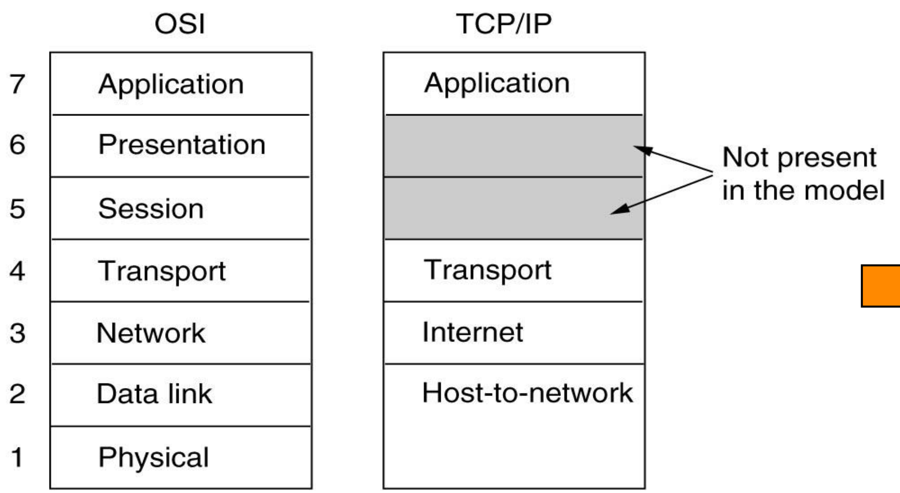
$H_?$ = headers

Layered design: Internet protocol stack



Layered design: Networking reference models

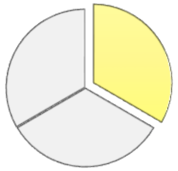
- **OSI (Open Systems Interconnection) reference model**
  - Useful model
  - Protocols did not become popular for various reasons
- **TCP/IP reference model (Cerf and Khan 1974)**
  - Nonexistent model
  - Protocols widely used

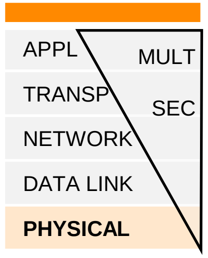


Internet reference model used in this course

5	Application layer
4	Transport layer
3	Network layer
2	Data link layer
1	Physical layer

Physical layer (L1, PHY): Bits in the wire(less)



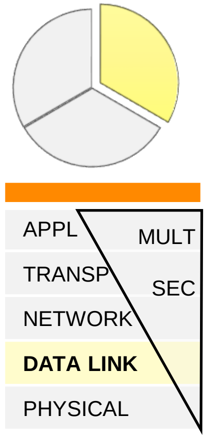


- **Two principal types of transmission medium**
  - Guided media: copper, coax, optical fiber
  - Unguided media: wireless (radio, IR/visible light)
- **Key properties** are data rate and distance

NIC is identified by a **physical (MAC) address**, e.g.
00:A0:C9:14:C8:29

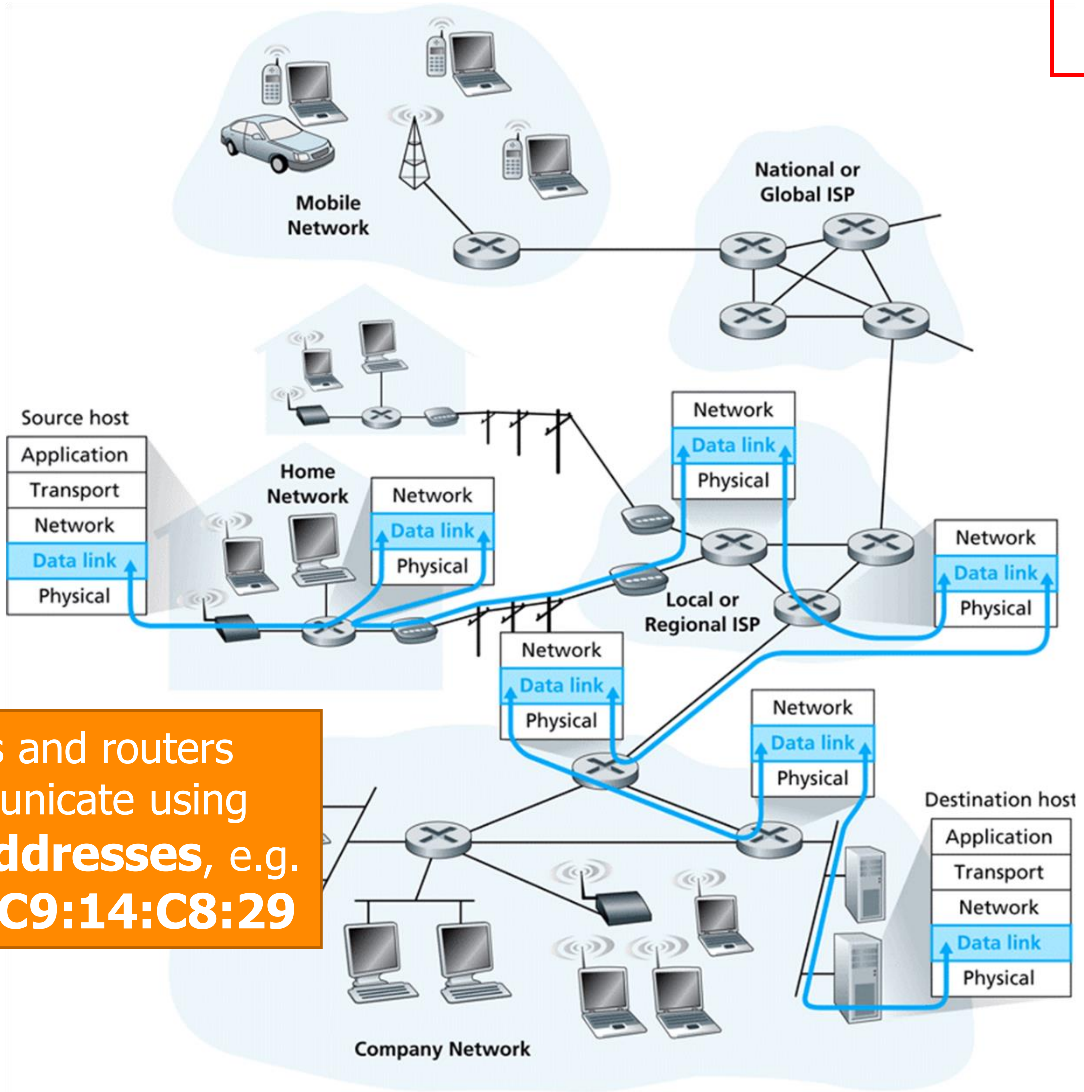
- **Each host has one or more NIC** (Network Interface Card), also called as network adapter **(routers have two or more NIC)**
  - Connects host to a communication link
  - NIC: hardware & control software
  - When sending:
    1. Transforms binary data from upper layers into electric current changing at a given rate (baud rate)
    2. Ejects the current into the transmission media (wired/wireless)
  - When receiving:
    1. Transforms received electric current into binary data
    2. Sends to upper layers

Data link layer (L2)



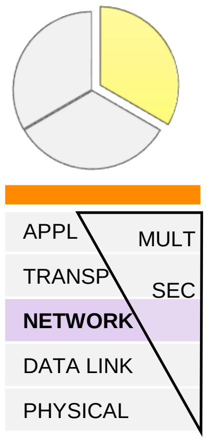
Data link layer delivers a frame (encapsulated datagram) from a node to adjacent node over a link

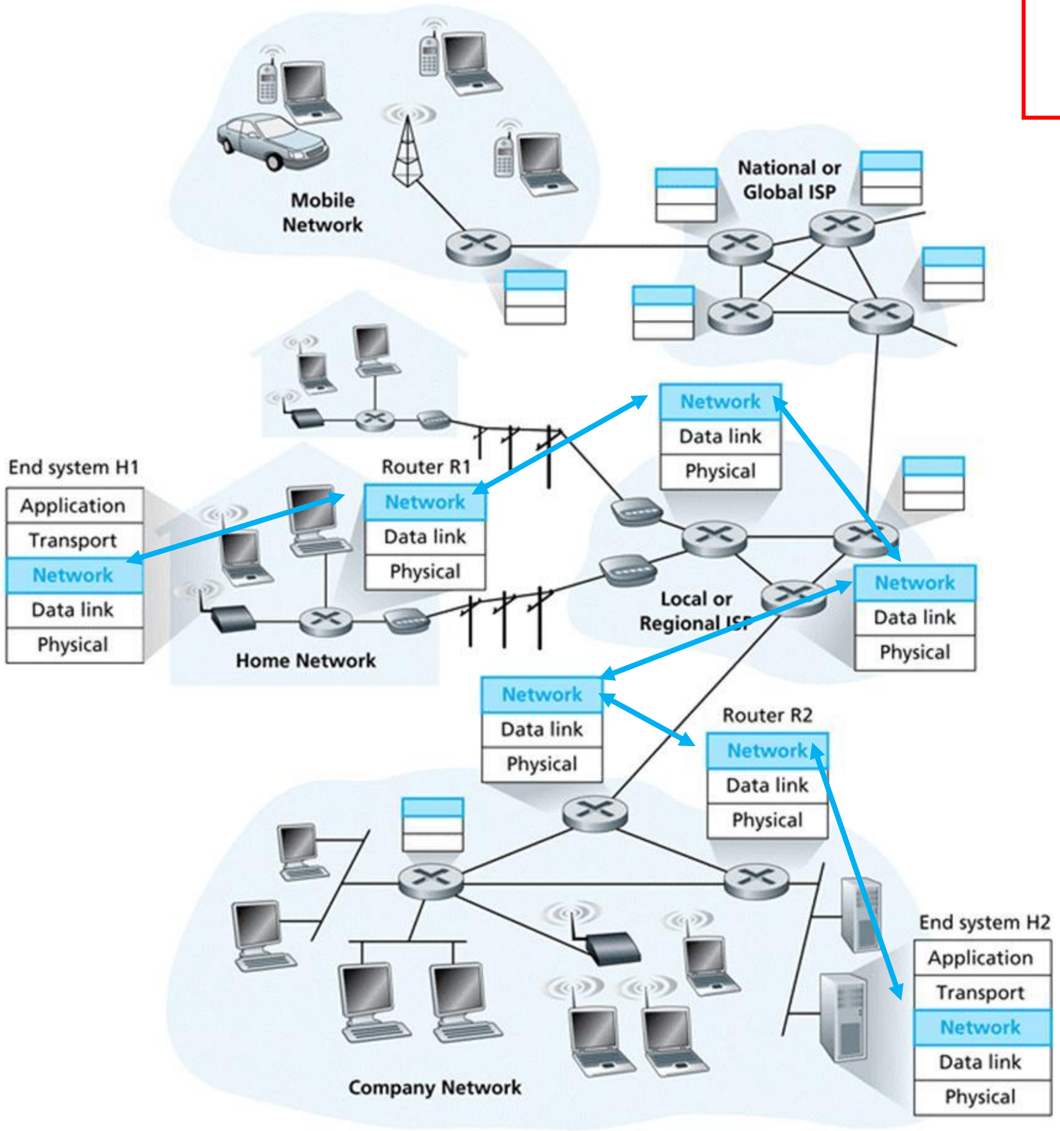
- **Nodes:** Hosts and routers
- **Links:** Communication channels that connect adjacent nodes along communication path
  - Wired or wireless
  - E.g. Ethernet, WLAN, LTE
  - Key feature: data rate ("bandwidth") expressed in bps (bits per second)
- **Frames:** L2 packets which encapsulate datagram provided by L3 (network layer)
- Data link layer functionality is implemented in NIC



Hosts and routers communicate using **MAC-addresses**, e.g. **00:A0:C9:14:C8:29**

Network layer (L3)



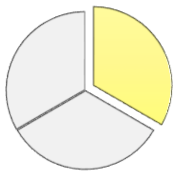


Network layer delivers a datagram (encapsulated segment) from sending host to receiving host

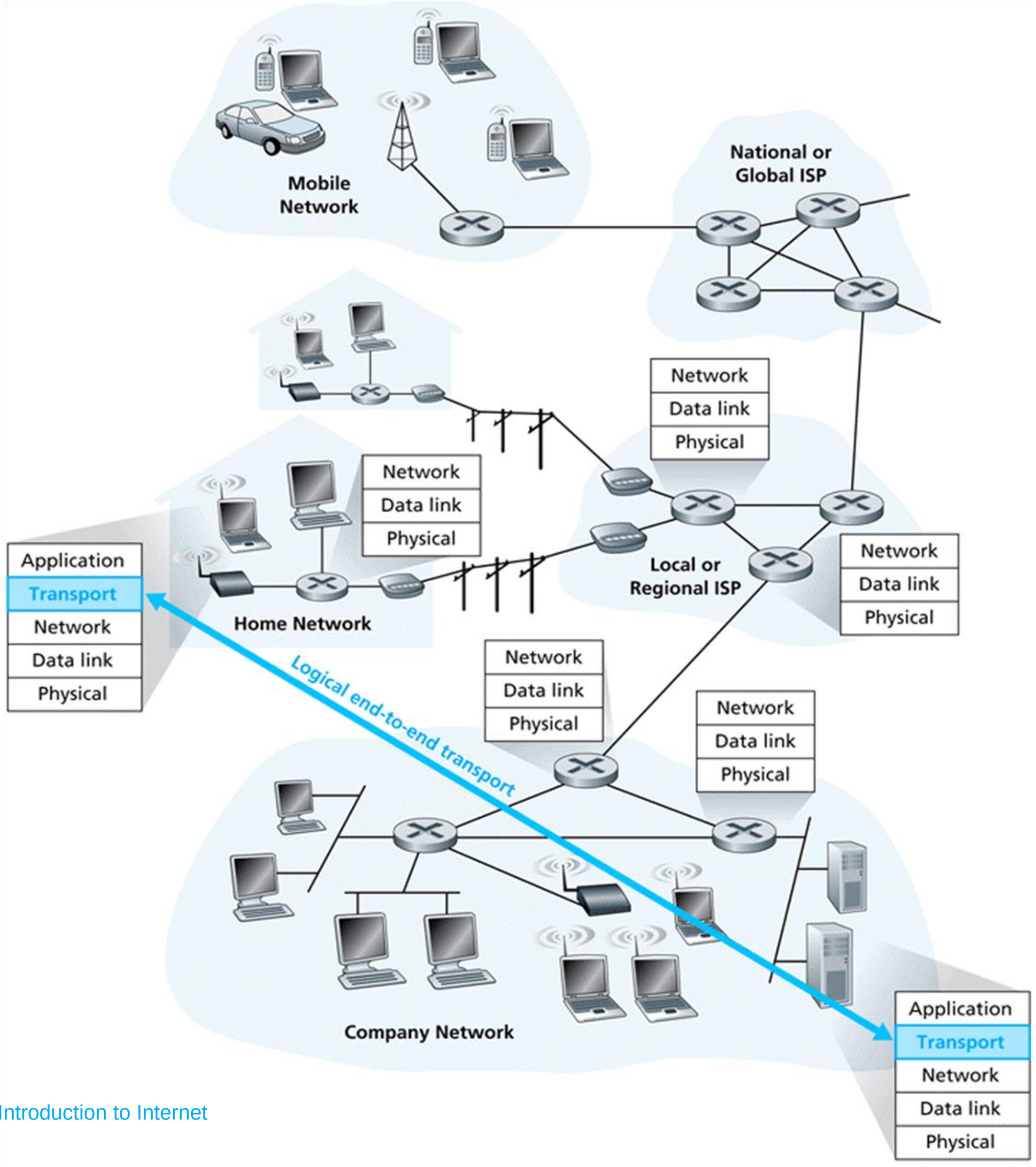
- **Logical end-to-end communication between two hosts**
  - On sending host encapsulates segments into datagrams
  - On receiving host, delivers segments to transport layer
- **Network layer protocol in *every* host and router**
- **Router examines header fields in all datagrams passing through it**
- **Key protocols: IP, ICMP, routing protocols**

Internet nodes communicate using **IP-addresses**, e.g.
98.139.180.149 (IPv4) or
2002:4559:1fe2::4559:1fe2 (IPv6)

Transport layer (L4)



APPL MULT
TRANSP SEC
NETWORK
DATA LINK
PHYSICAL

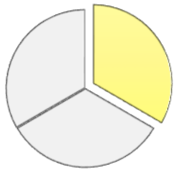


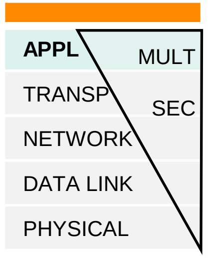
Transport layer transports a segment (and encapsulated message) from a process on a host to another process on a different host

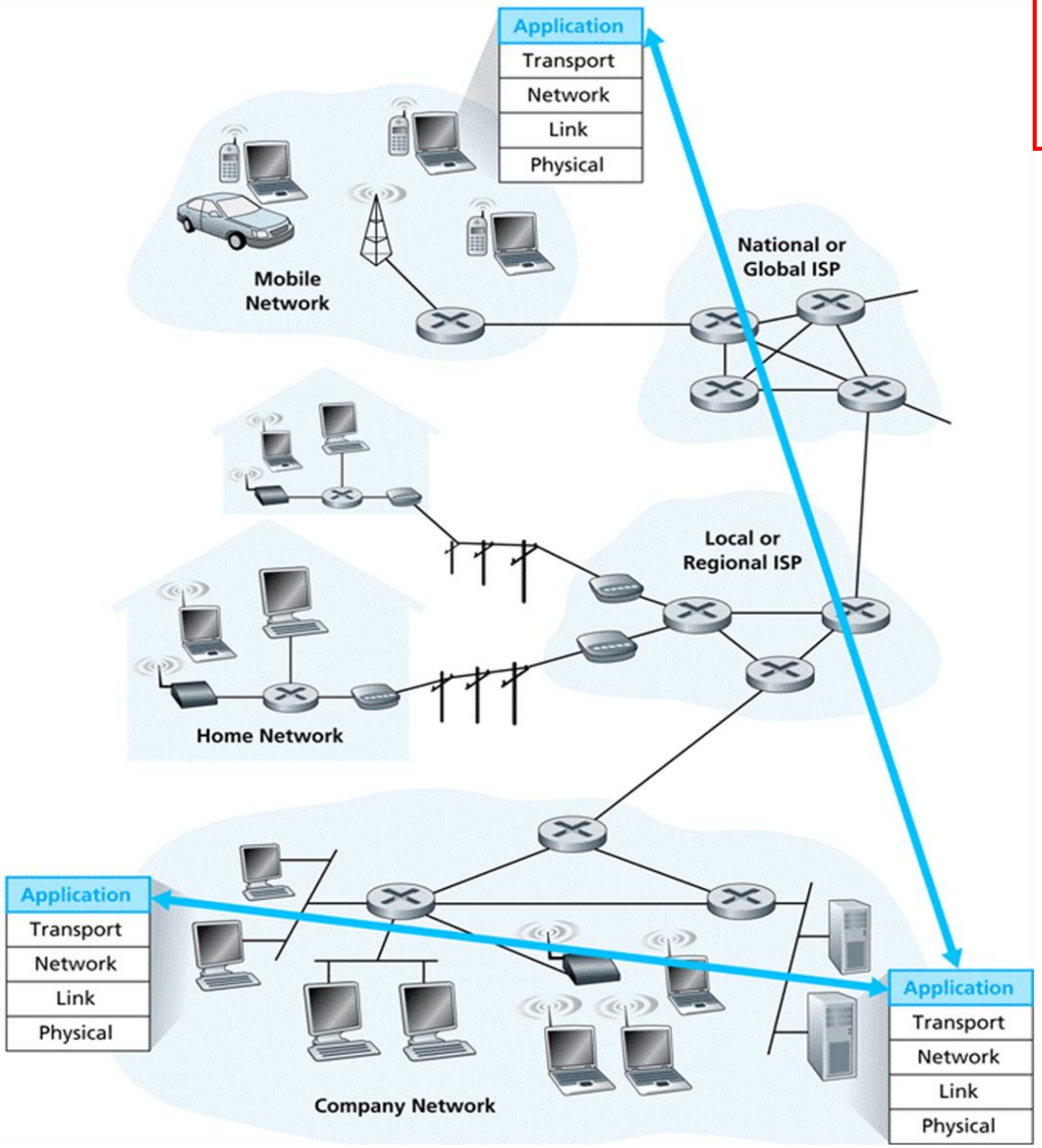
- **Logical end-to-end communication between application processes running on different hosts**
- **Transport protocol runs in hosts**
  - Sender: breaks application messages into segments, passes to network layer
  - Receiver: reassembles segments into messages, passes to application layer
- **Different transport protocols available to applications**
  - Internet: TCP and UDP primary
  - Implemented in hosts' operating system

Processes are identified using **port numbers**, e.g.
22 (SSH), 80 (HTTP), 143 (IMAP), etc.

Application layer (L5)







Application layer allows networking applications to communicate by exchanging messages

- **Many application layer protocols supporting networking applications**
  - E.g. HTTP, SMTP, SNMP, FTP
- User applications
  - Run on different hosts
    - Allows for rapid development and propagation of applications
  - Communicate over a network
    - E.g. WWW: web browser software communicates with web server software
- Network core devices do not run user application code

Applications and services communicate using **application layer addresses**, e.g. **http://www.google.com (Web URL)**, or **info@google.com (email address)**

Principles of Packet-switched networks

Internet is a packet switching datagram network

- **There are two fundamental approaches for moving data through a network of links and switches**

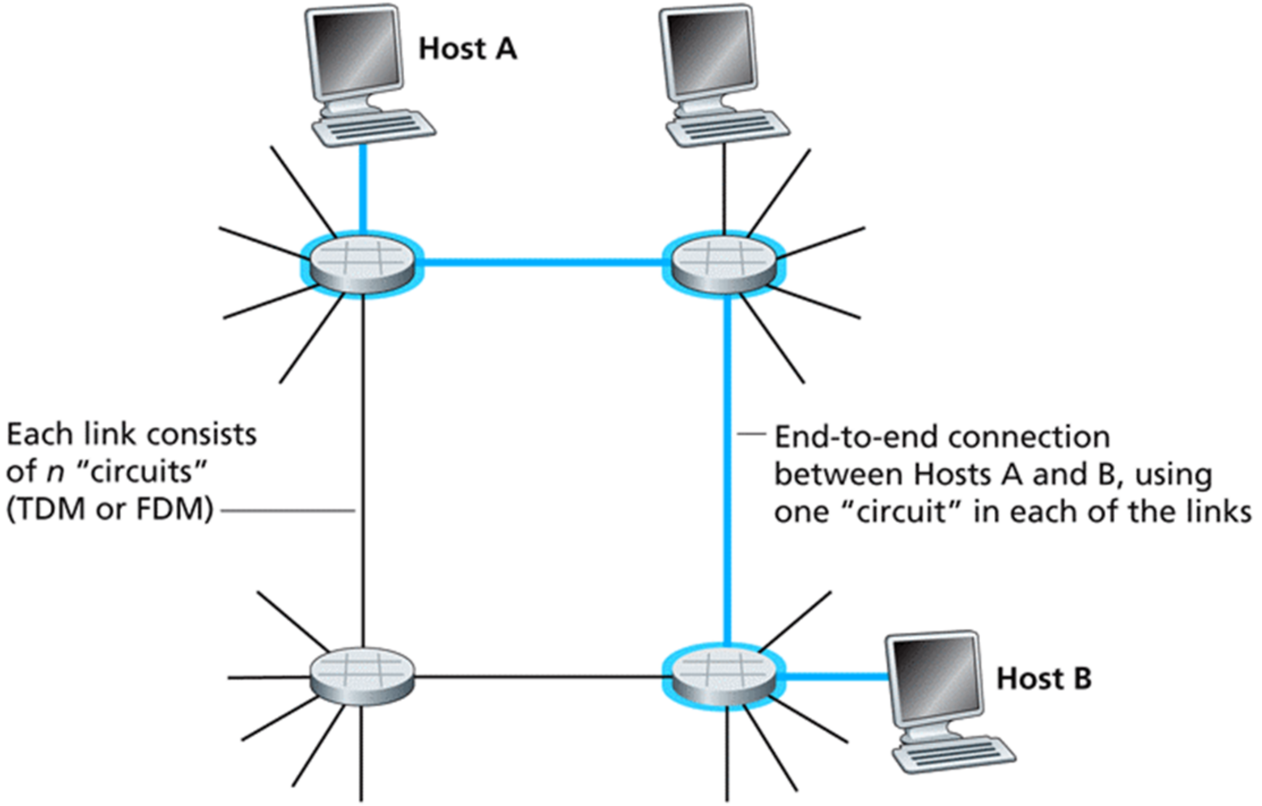
- **Circuit switching**
  - Network resources (link bandwidth, switch capacity) are divided into pieces
  - Dedicated pieces are allocated for the communication session for the duration of the session (”call”)

- **Packet switching**
  - Network resources are not reserved but shared
  - Communication session uses resources on demand and may have to compete/wait for them

Circuit switching (1)

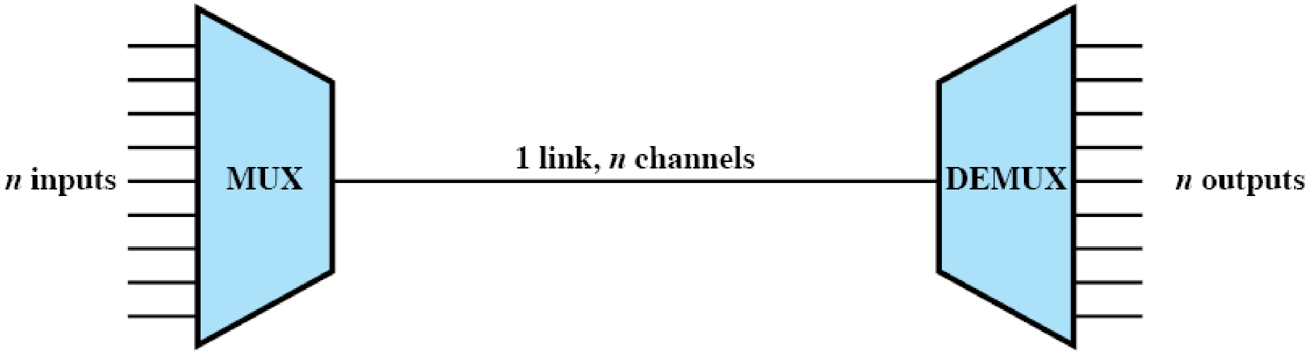
– **End-to-end resources reserved for communication session ("call")**
  - Link bandwidth, switch capacity
  - Dedicated resources, no sharing
  - Circuit-like (guaranteed) performance
  - Session setup required
  - Motivation
    - Quality of service (QoS)
    - Billing

– **E.g. plain old telephone service (POTS)**



Circuit switching (2)

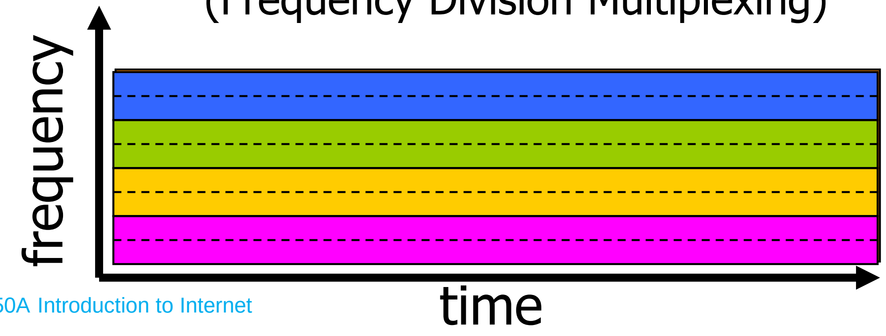
- **Network resources (e.g. bandwidth) divided into “pieces”**
  - Pieces allocated to sessions (“calls”)
  - Resource piece idle if not used by owning session (no sharing)
  - Dividing link bandwidth into “pieces” by multiplexing
    - One link carries n separate logical channels
    - Each channel is allocated a piece of the link for exclusive use



Example: 4 users

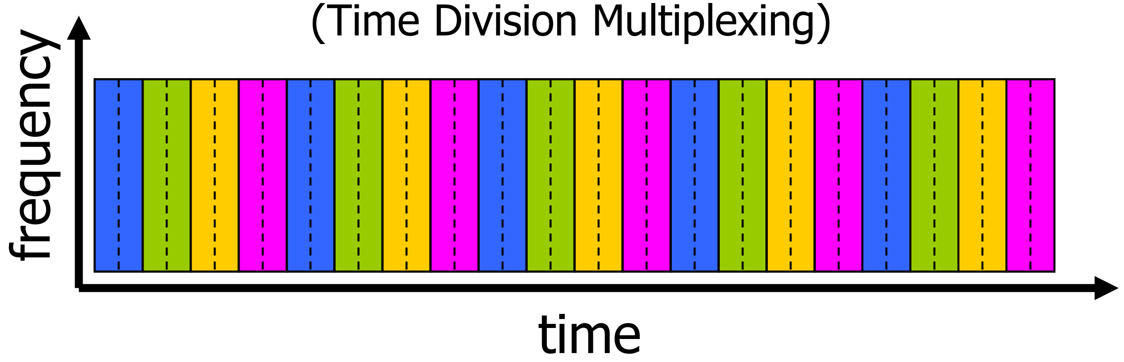
FDM

(Frequency Division Multiplexing)

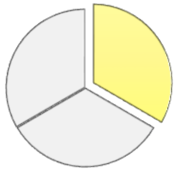


TDM

(Time Division Multiplexing)



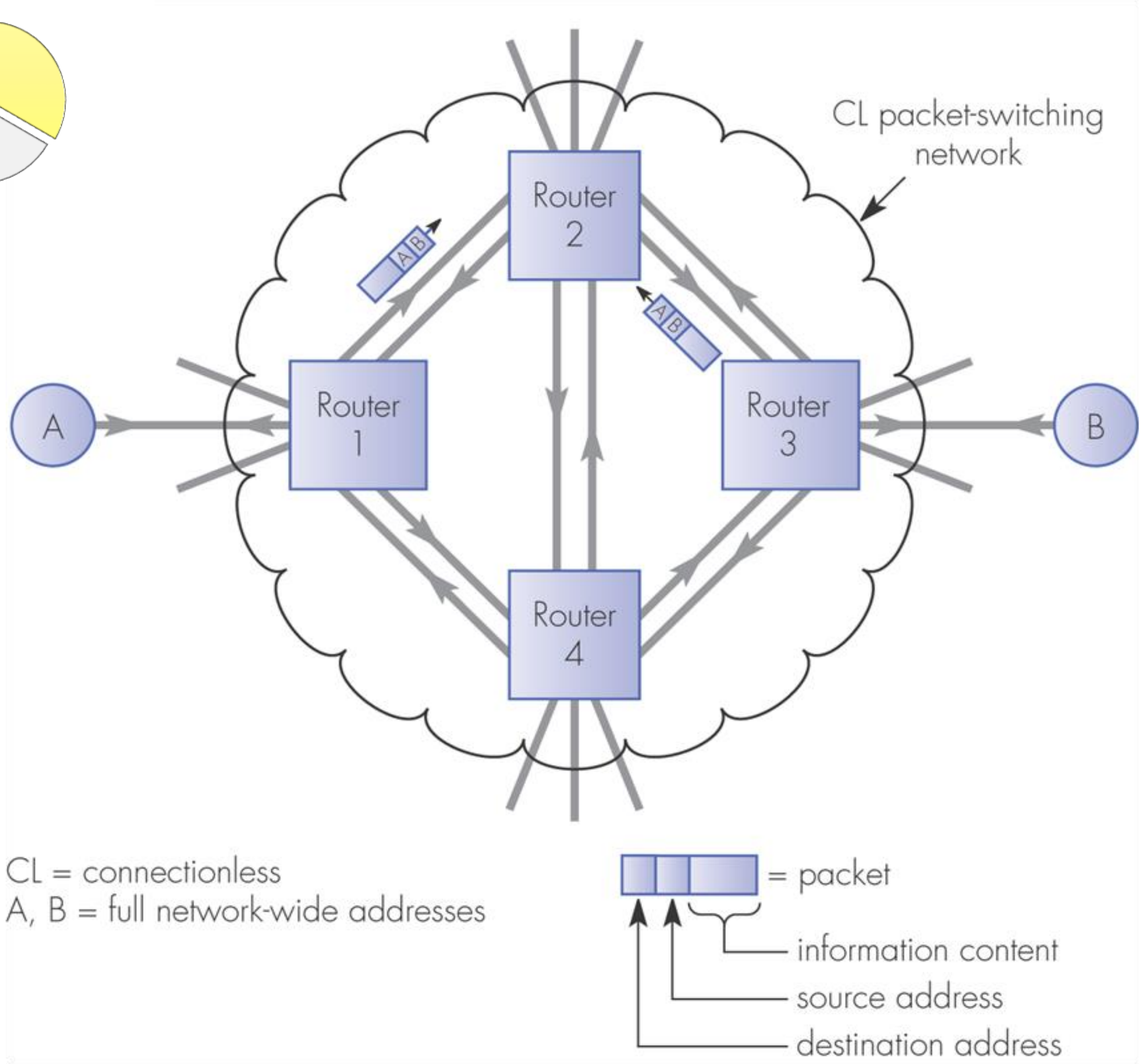
Packet switching



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

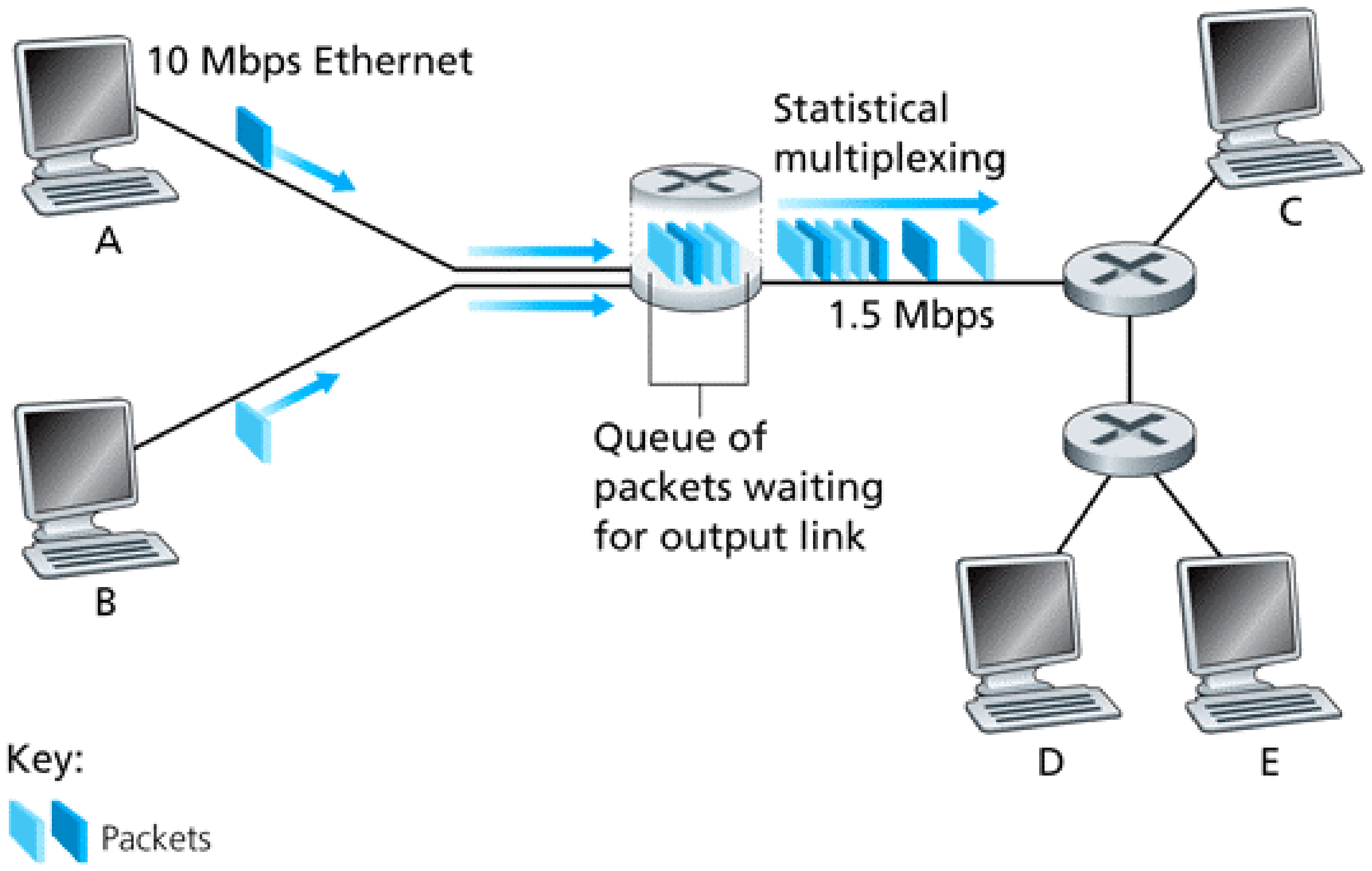
~~Bandwidth division into "pieces"~~
~~Dedicated allocation~~
~~Resource reservation~~

Packet switching: Datagram network



- **Destination address in packet determines next hop**
- **Routes may change during session**
- **E.g. IP**

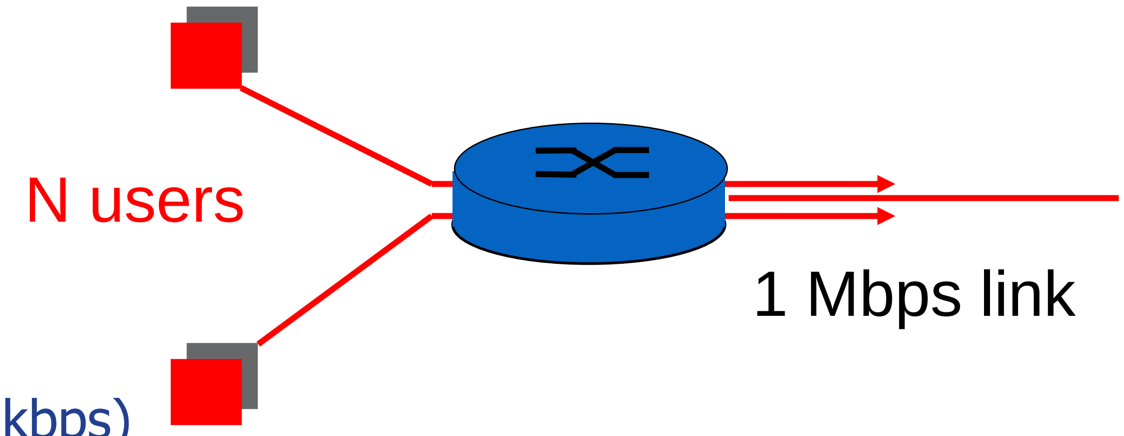
Packet switching: Statistical multiplexing



Sequence of A & B packets does not have fixed pattern, which is called *statistical multiplexing*

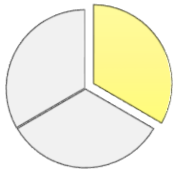
Packet switching vs circuit switching (1)

- **1 Mbps link**
- **Each user**
  - 100 kbps when “active”
  - Active 10% of time
- **Circuit switching**
  - Max. 10 users (1 Mbps / 100 kbps)
- **Packet switching**
  - With 35 users, probability of >10 active users is less than .0004
- **Packet switching leads to more efficient network utilization**



→ Allows more users per network!

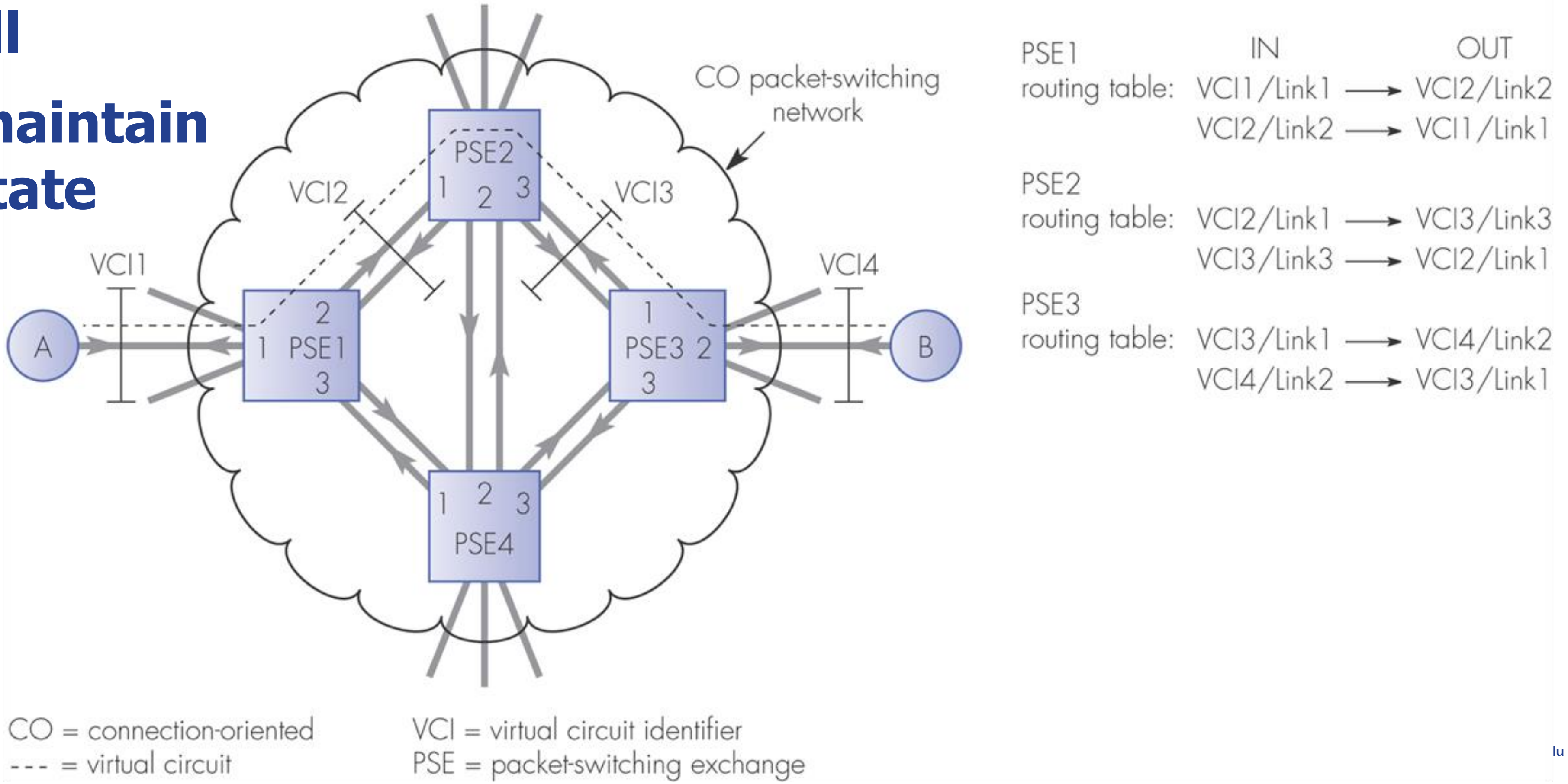
Packet switching vs circuit switching (2)



– **Packet switching is great for bursty data**
  - Resource sharing on demand, *"best-effort network"*
  - More simple, no call setup
– **Excessive congestion: packet delay and loss**
  - Protocols needed for reliable data transfer, congestion control
– **How to provide circuit-like behavior?**
  - QoS guarantees needed for real-time multimedia applications
  - Some solutions:
    - QoS-aware protocols and routers
    - Virtual circuit networks

Packet switching: Virtual circuit network

- Each packet carries tag (virtual circuit ID) which determines next hop
- Fixed path determined at call setup time, remains fixed during call
- Routers maintain per-call state
- E.g. ATM



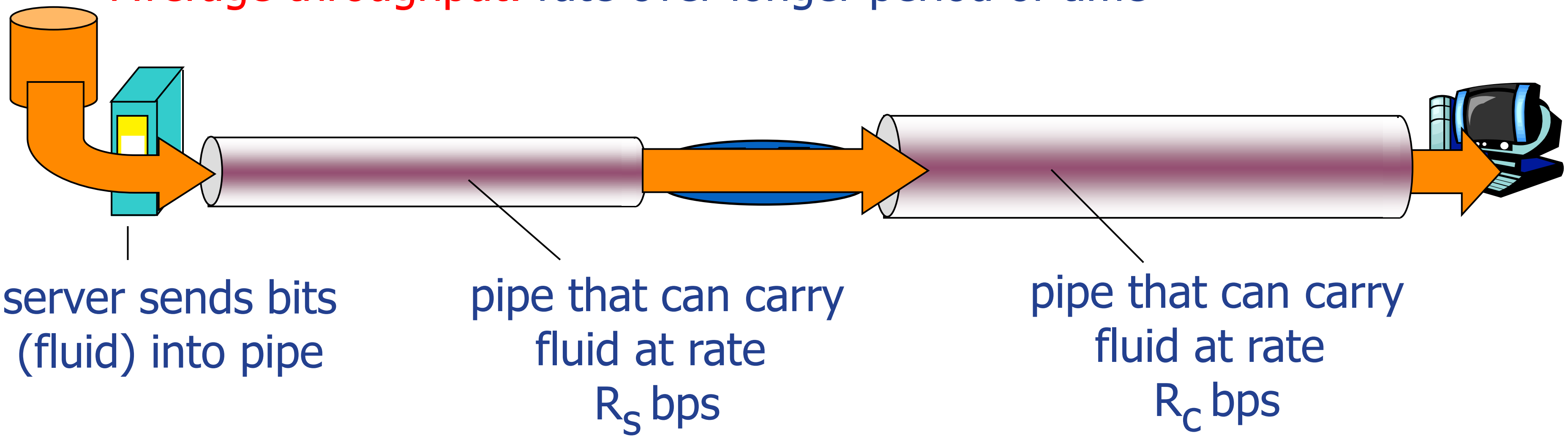
Performance of Packet-switched networks

Key network performance attributes from application/service point of view

- **Throughput: what is the amount of data per second (bits per second ~ bps) that can be transferred?**
- **Delay: how long does it take to transfer certain amount of data between two end systems?**
  - Router delays
  - End system delays (modulation, encoding, media packetization)
- **Packet loss: how much of the data is lost during transfer?**

Throughput (1)

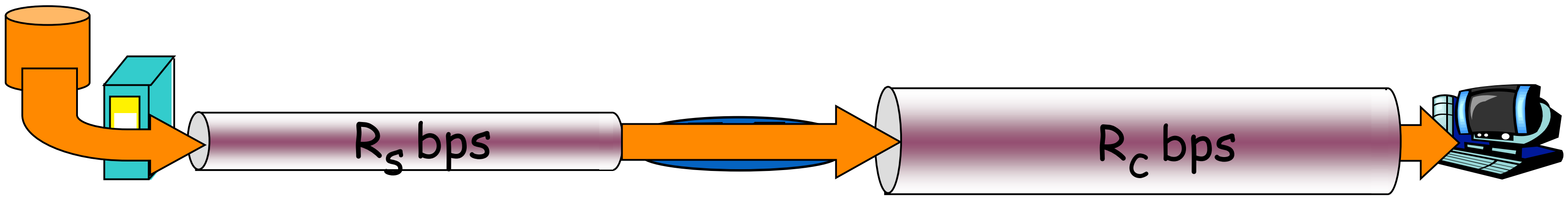
- **Throughput: rate (bits/s, bps) at which bits transferred between sender/receiver**
  - Instantaneous throughput: rate at given point in time
  - Average throughput: rate over longer period of time



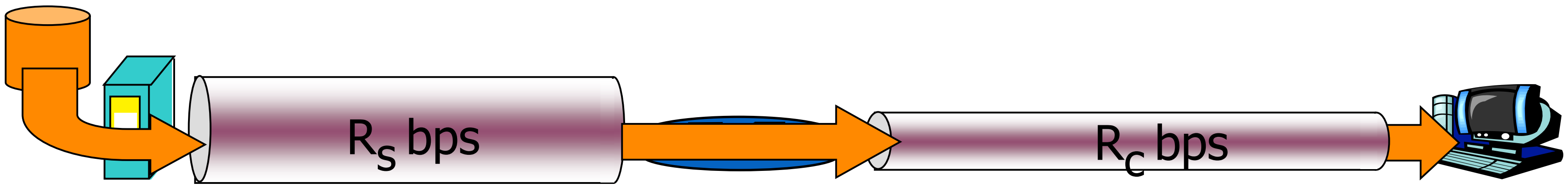
- Bandwidth x delay (d_{prop}) product (bdp)
  - Amount of data that "fills the pipe", transmitted before the first bit arrives at destination

Throughput (2)

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



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

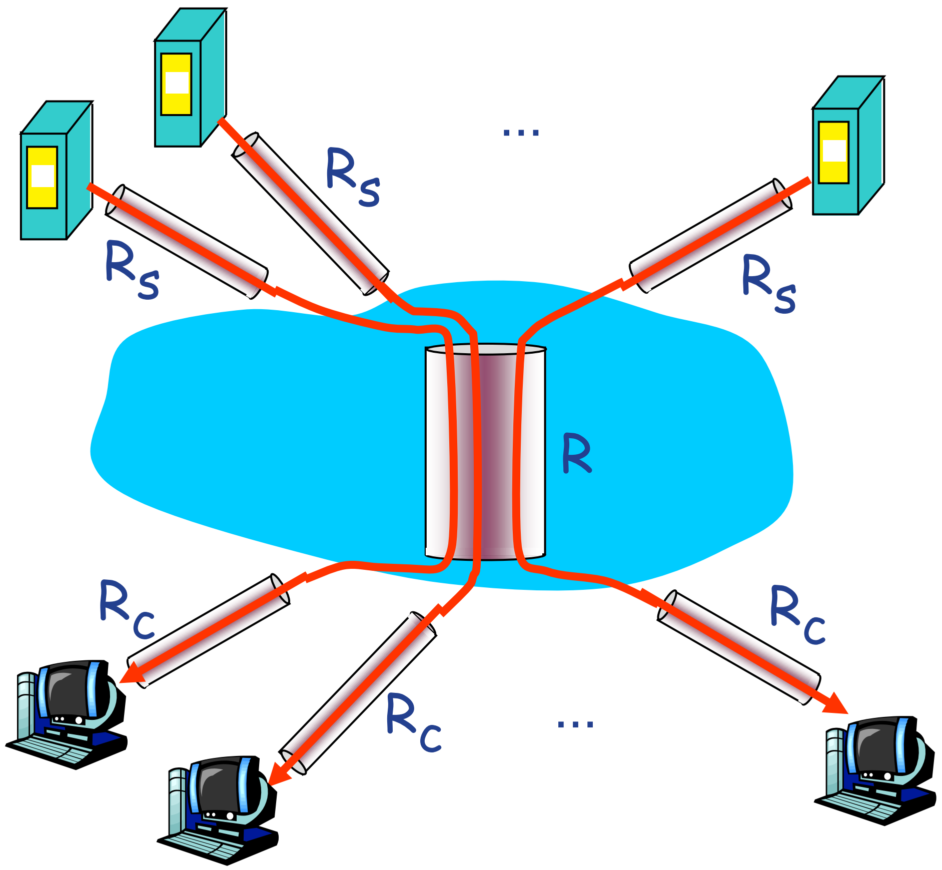


Bottleneck link

Link on end-end path that limits end-end throughput

Throughput (3)

- Internet scenario: 10 connections through link R
- Per-connection end-to-end throughput: MIN(R_c, R_s, R/10)
- In practice R_c or R_s is often bottleneck

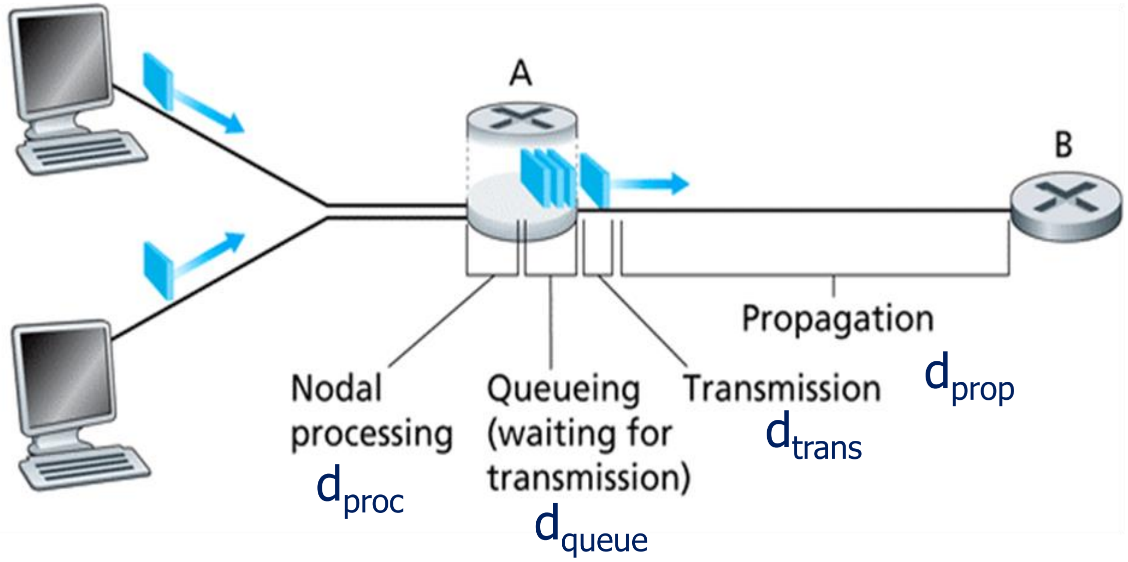


10 connections (fairly) share backbone bottleneck link R bps

Delay in packet switched networks (1)

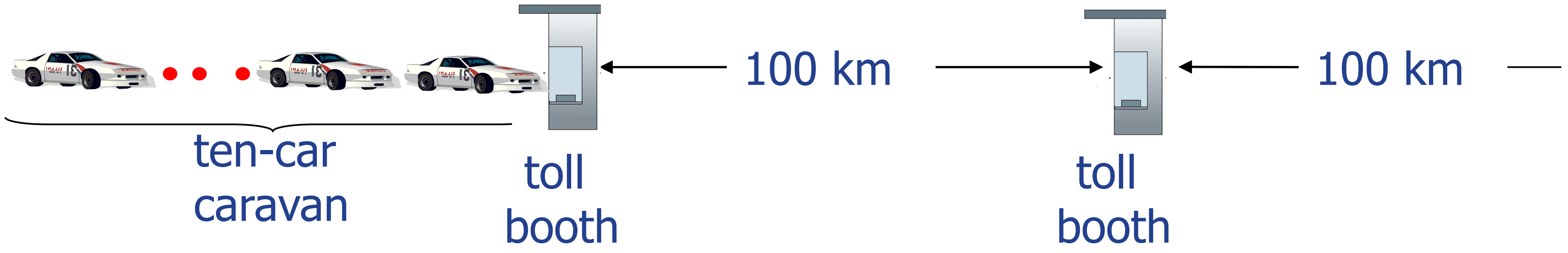
- **Packet suffers different delays along its end-to-end path**
- **Four important delay components in a link/router**

1. Nodal (node internal) processing delay (d_{proc})
   – E.g. header processing, error checking, typically few µs or less
2. Queuing delay (d_{queue})
   – Time waiting at output link for transmission, depends on congestion level
3. Transmission delay (d_{trans})
   – L/R, time needed to send a packet of L bits into a link of data rate R bps, significant for low data rate links
4. Propagation delay (d_{prop})
   – d/v, time a bit needs to propagate the length d of the link, when the propagation speed in the transmission medium is v (~$2 \cdot 10^8$ m/s), from Few µs to hundreds of ms



Delay in packet switched networks (2)

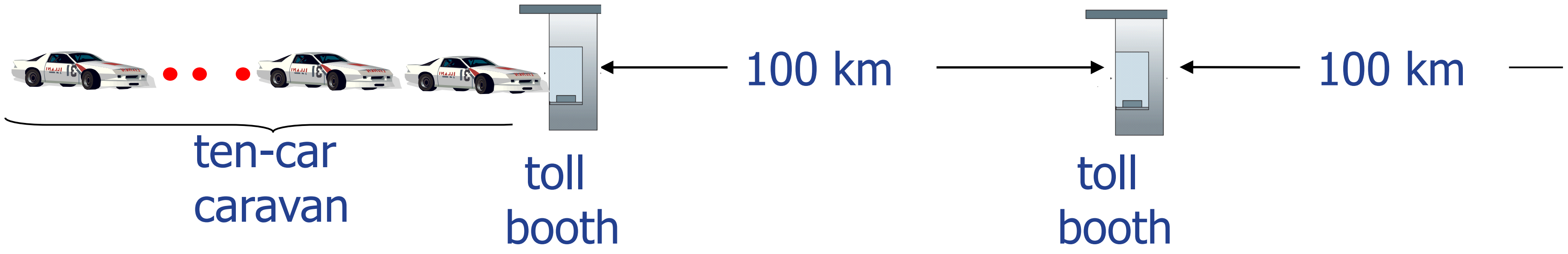
Caravan analogy (1)



- Car ~ bit, Caravan ~ packet
- Cars "propagate" at 100 km/h
- Toll booth takes 12 s to service car (transmission time)
- Q: How long until caravan is lined up before 2nd toll booth?
- Time to "push" entire caravan through toll booth onto highway = 12x10 = **120 s**
- Time for last car to propagate from 1st to 2nd toll both: 100km/(100km/h) = **1 h**
- A: 1h + 120s (2min) = **62min**

Delay in packet switched networks (3)

Caravan analogy (2)



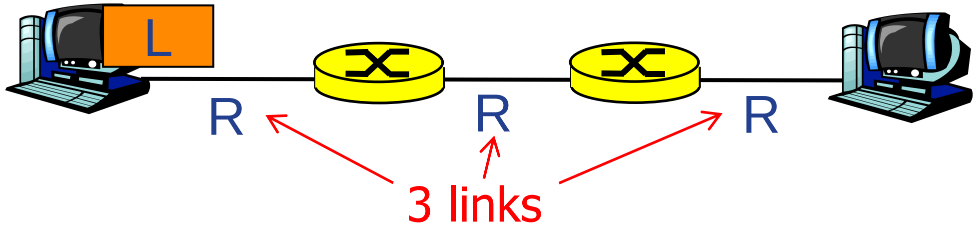
- Now cars "propagate" at 1000 km/h
- Toll booth now takes 1 min to service a car
- Q: Will cars arrive to 2nd booth before all cars serviced at 1st booth?

- Yes! E.g. after 7 min, 1st car at 2nd booth and 3 cars still at 1st booth.

→ 1st bit of packet can arrive at 2nd router before packet is fully transmitted from 1st router!

Delay in packet switched networks (4)

Transmission delay in store-and-forward switching

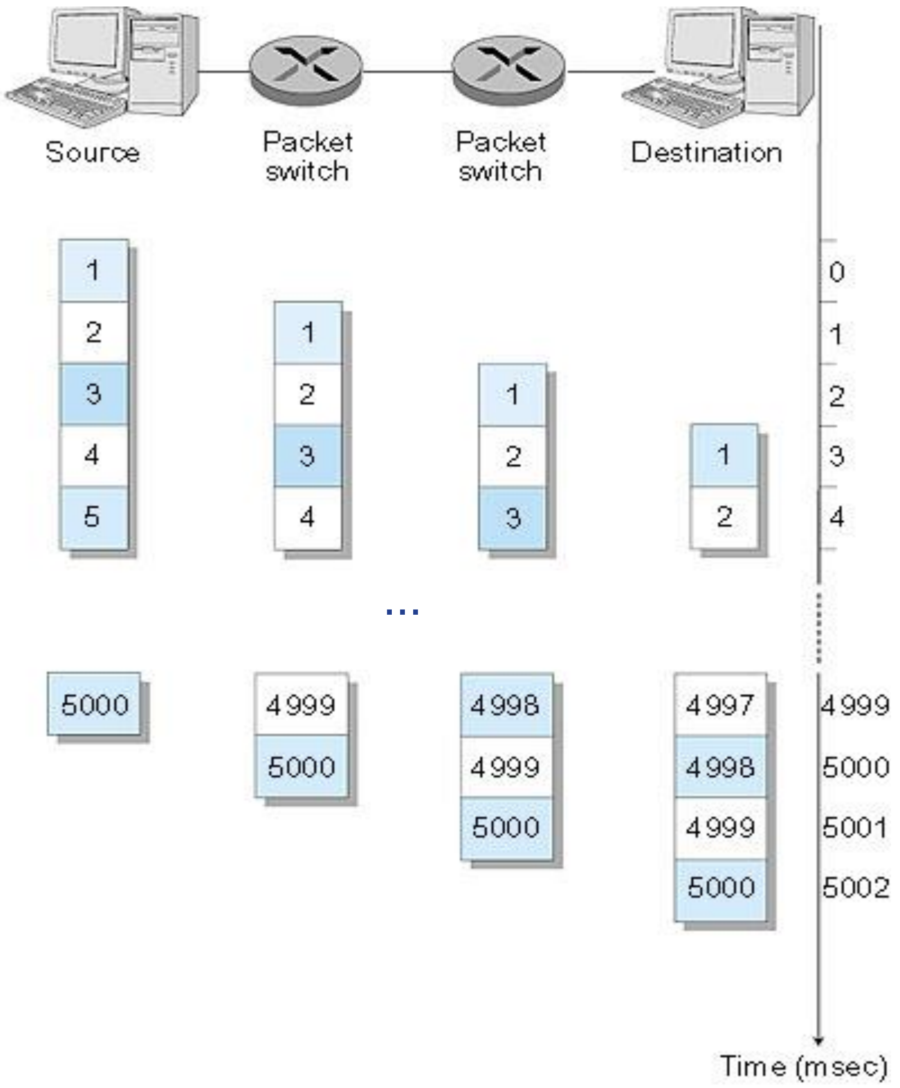


R=link bandwidth (bps)
L=packet length (b)

- Takes L/R seconds to transmit (push out) packet of L bits onto link of R bps
- **Store-and-forward switching:** entire packet must arrive at router before packet can be transmitted on next link
- In the network above, the total transmission delay = 3 x L/R
- **Example:** L = 7.5 Mb, R = 1.5 Mbps

  Delay = 3L/R = 3 x 7.5 Mb / 1.5 Mbps = **15 s**

Delay in packet switched networks (5)



Transmission delay in store-and-forward switching with **message segmenting:**

- Break up message into 5000 packets of 1500 bits
- 1 ms to transmit packet on one link
- Pipelining: each link works in parallel
- Total transmission delay reduced from 15 s to **5.002 s**:

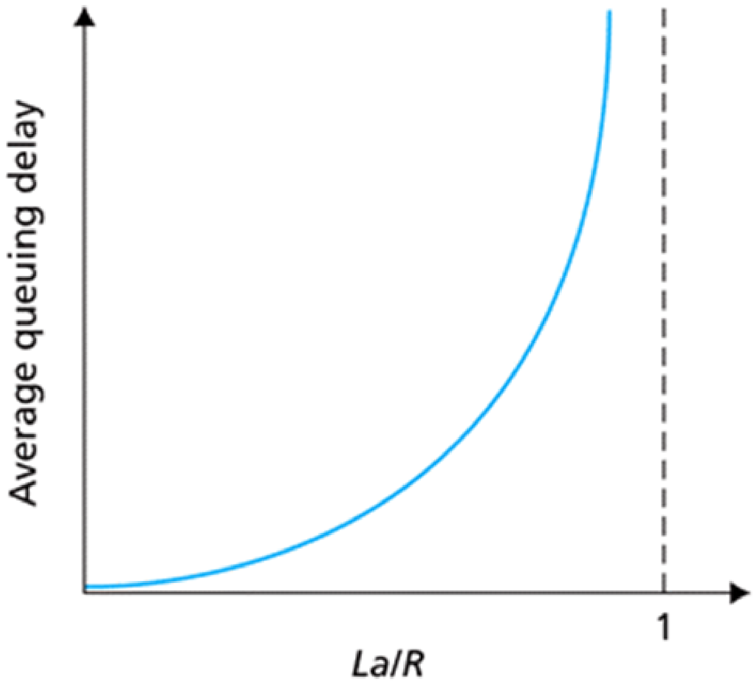
Delay in packet switched networks (6)

Transmission delay in cut-through switching

- Cut-through switching: H (H<<L) header bits of the packet must arrive at a router before packet can be transmitted on next link
- Effectively, (L-H)/R amount of time is saved in comparison to store-and-forward switching
- However, a router may forward a faulty packet

Queuing delay and traffic intensity

- **R=link bandwidth (bps)**
- **L=packet length (b)**
- **a=average packet arrival rate (pps)**



traffic intensity = La/R

- La/R ~ 0: average queuing delay small
- La/R -> 1: delays become large
- La/R > 1: more "work" arriving than can be serviced, average delay infinite!

Real-life delay in the Internet (1)

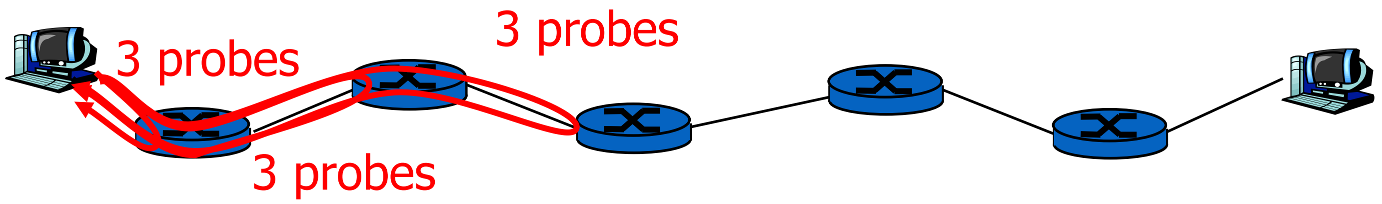
– **Assuming there are N-1 routers between source and destination, then**

$$d_{end\text{-}end} \sim N\,(d_{proc} + d_{queue} + d_{trans} + d_{prop})$$

(See d_{proc}, d_{queue}, d_{trans}, d_{prop} explanations on page 46)

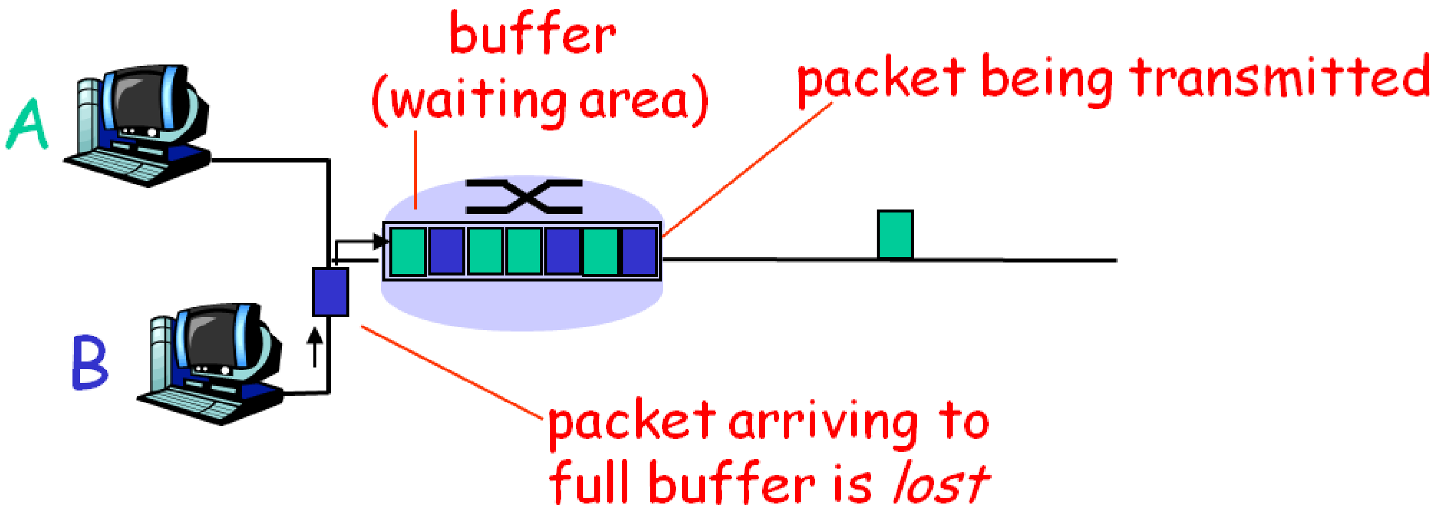
– **Traceroute (Unix/Linux), Tracert (Windows) program for estimating round-trip delay from source to each router on end-to-end path towards destination**

- For each router *n:*
  - Sends three packets that will reach router *n* on path towards destination
  - Router *n* will return packets to sender
  - Sender times interval between transmission and reply



Packet loss (due to queuing)

- Input packet queue (i.e. **buffer**) preceding a link has finite capacity
- Packet arriving to full queue is dropped (aka lost)
- Lost packet may be retransmitted by previous node, by source end system, or not at all



Real-life delay in the Internet (2)

```
traceroute to cs.columbia.edu (128.59.16.20), 30 hops max, 40 byte packets
 1  so-gw.oulu.fi (130.231.48.1)  0.816 ms  0.464 ms  0.479 ms
 2  oy-gw.oulu.fi (130.231.248.1)  0.520 ms  0.701 ms  0.545 ms
 3  oy-oulu-gw.oulu.fi (193.167.221.2)  1.440 ms  0.665 ms  0.735 ms
 4  oulu-funet-gw.oulu.fi (193.167.221.18)  1.379 ms  1.762 ms  1.113 ms
 5  oulu0-g2000-oulu3.funet.fi (193.166.187.117)  1.931 ms  1.768 ms  1.539 ms
 6  abo0-p2000-oulu0.funet.fi (193.166.255.169)  10.306 ms  10.063 ms  10.242 ms
 7  csc0-p2000-abo0.funet.fi (193.166.255.161)  12.686 ms  12.667 ms  12.849 ms
 8  helsinki0-x4100-csc0.funet.fi (193.166.255.154)  17.933 ms  12.984 ms  13.108 ms
 9  se-tug.nordu.net (193.10.68.97)  19.634 ms  29.983 ms  19.329 ms
10  se-fre.nordu.net (193.10.252.85)  20.354 ms dk-uni.nordu.net (193.10.68.18)  35.747 ms se-fre.nordu.net (193.10.252.85)  19.798 ms
11  dk-ore.nordu.net (193.10.68.118)  29.960 ms dk-ore.nordu.net (193.10.68.25)  33.020 ms dk-ore.nordu.net (193.10.68.118)  29.938 ms
12  nordunet.rt1.cop.dk.geant2.net (62.40.124.45)  33.281 ms  32.949 ms  30.001 ms
13  so-4-0-0.rt1.ams.nl.geant2.net (62.40.112.78)  46.296 ms  42.925 ms  62.976 ms
14  so-7-0-0.rt1.nyc.us.geant2.net (62.40.112.134)  132.978 ms  131.073 ms  175.719 ms
15  198.32.11.50 (198.32.11.50)  129.926 ms  126.448 ms  129.660 ms
16  199.109.4.153 (199.109.4.153)  126.185 ms  126.631 ms  129.647 ms
17  columbia.nyc-gsr.nysernet.net (199.109.4.14)  125.760 ms  129.644 ms  129.655 ms
18  cc-core-1-x-nyser32-gw-1.net.columbia.edu (128.59.255.5)  126.896 ms  130.108 ms  126.542 ms
19  mudd-edge-1-x-cc-core-1.net.columbia.edu (128.59.255.86)  136.989 ms  131.990 ms  133.446 ms
20  cs.columbia.edu (128.59.16.20)  129.898 ms  129.874 ms  133.463 ms
```

Compare!

Real-life delay in the Internet (3)

- **Ping program for estimating round-trip delay from source to destination**

```
(tk1)(skidi)(187)(~) ping -s cs.columbia.edu 56 10
PING cs.columbia.edu: 56 data bytes
64 bytes from cs.columbia.edu (128.59.16.20): icmp_seq=0 time=134 ms
64 bytes from cs.columbia.edu (128.59.16.20): icmp_seq=1 time=134 ms
64 bytes from cs.columbia.edu (128.59.16.20): icmp_seq=2 time=130 ms
64 bytes from cs.columbia.edu (128.59.16.20): icmp_seq=3 time=133 ms
64 bytes from cs.columbia.edu (128.59.16.20): icmp_seq=4 time=134 ms
64 bytes from cs.columbia.edu (128.59.16.20): icmp_seq=5 time=130 ms
64 bytes from cs.columbia.edu (128.59.16.20): icmp_seq=6 time=130 ms
64 bytes from cs.columbia.edu (128.59.16.20): icmp_seq=7 time=130 ms
64 bytes from cs.columbia.edu (128.59.16.20): icmp_seq=8 time=130 ms
64 bytes from cs.columbia.edu (128.59.16.20): icmp_seq=9 time=134 ms

----cs.columbia.edu PING Statistics----
10 packets transmitted, 10 packets received, 0% packet loss
round-trip (ms)  min/avg/max/stddev = 130./131.8/134./1.97
```

Key points to remember

1. Know the architecture and building blocks of the Internet
2. Understand the design principles of the Internet and their realization
3. Understand the main differences between Packet switching and circuit switching networks
4. Be aware of key performance attributes of packet switched networks

Thank you!