T1: Introduction

1.1 What is computer network?

- Examples of computer network
- The Internet
- Network structure: edge and core

1.2 Why computer networks

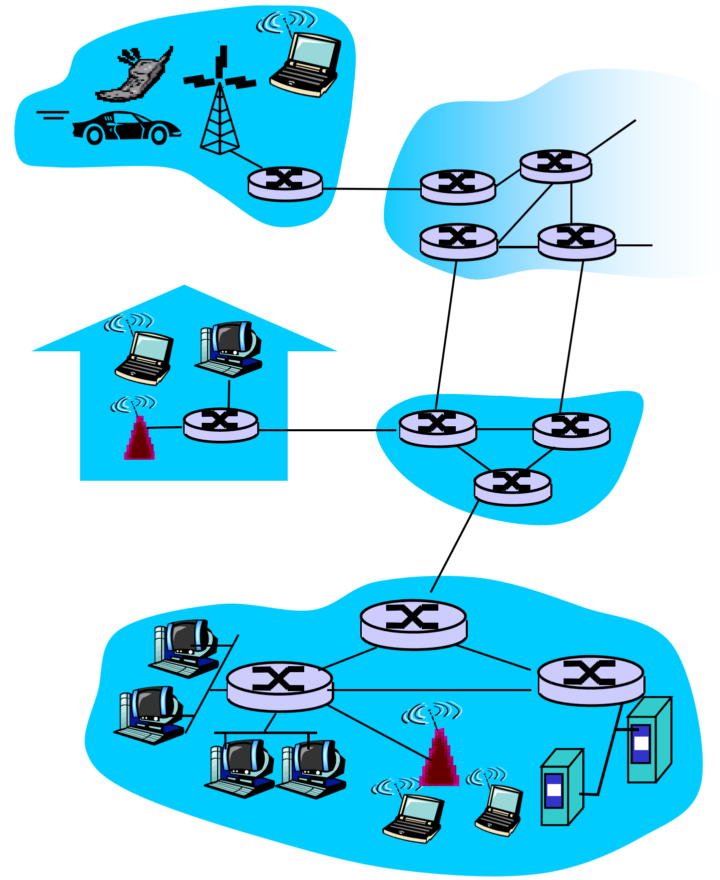
1.3 The way networks work

1.4 Performance metrics:

- Delay, loss and throughput in packet-switched networks

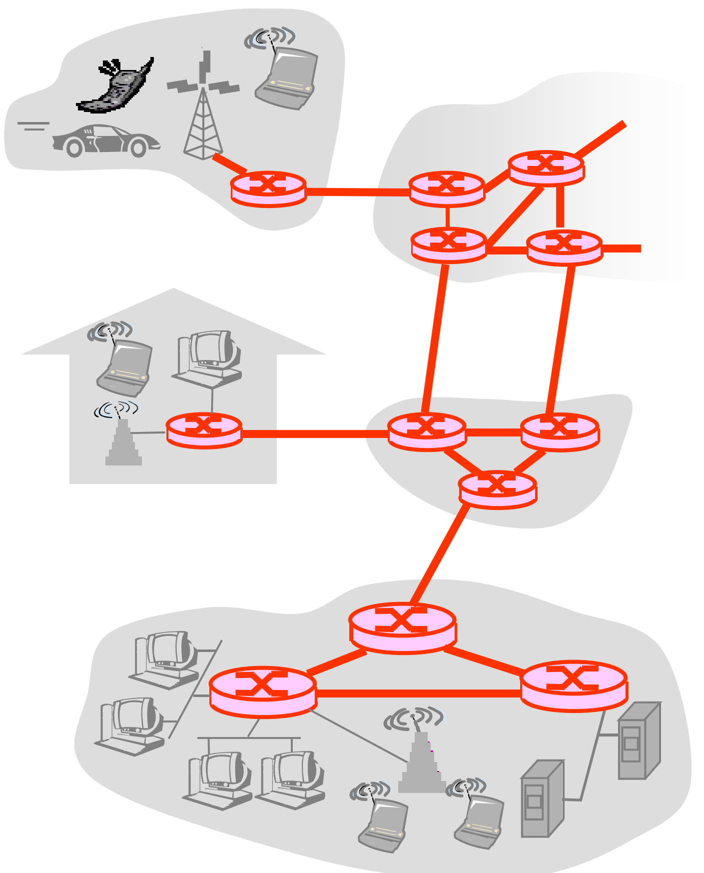
A closer look at network structure:

- **Network edge:**
  applications and hosts
- **Access networks, physical media:**
  wired, wireless communication links
- **Network core:**
  - interconnected routers
  - network of networks

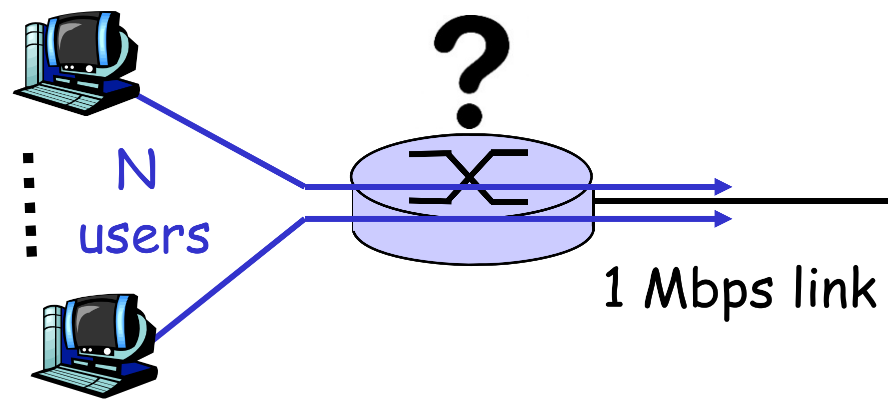


The Network Core

- Mesh of interconnected routers



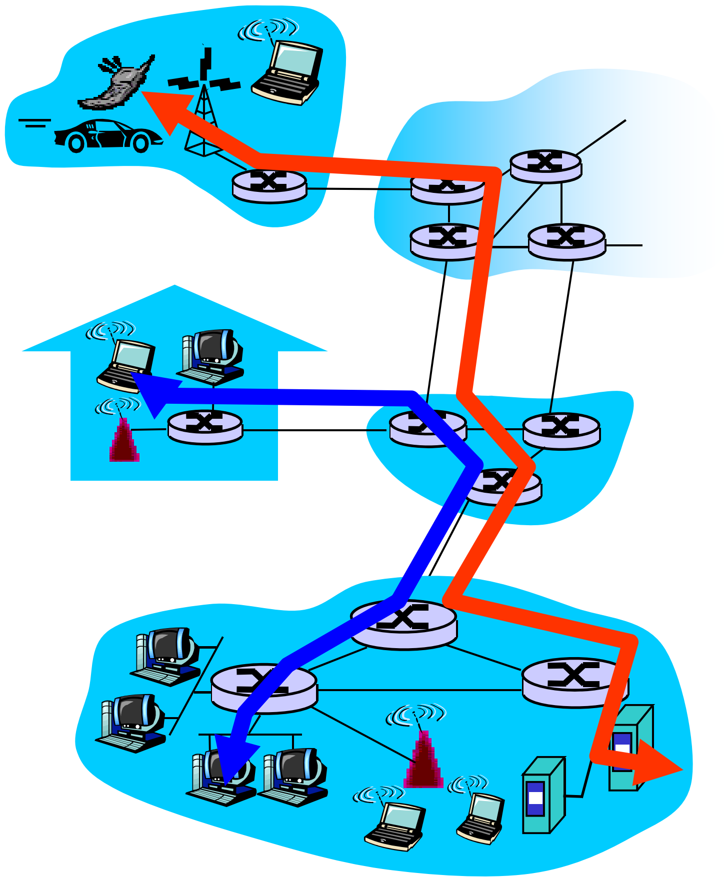
The fundamental question:
how is data (from multiple users) transferred through net?



Network Core: Circuit Switching

End-end resources reserved for "call"

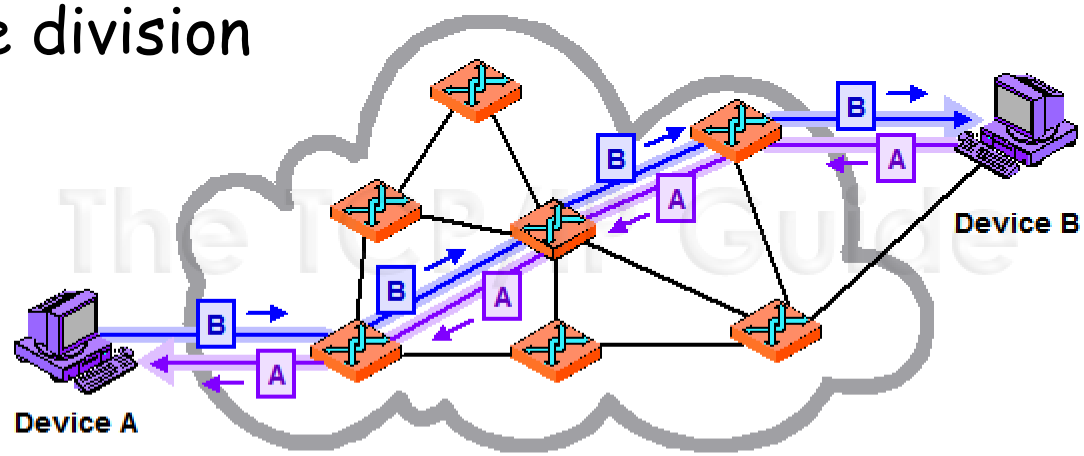
- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required
- Example: telephone net



Network Core: Circuit Switching

Network resources (e.g., bandwidth) divided into "pieces"

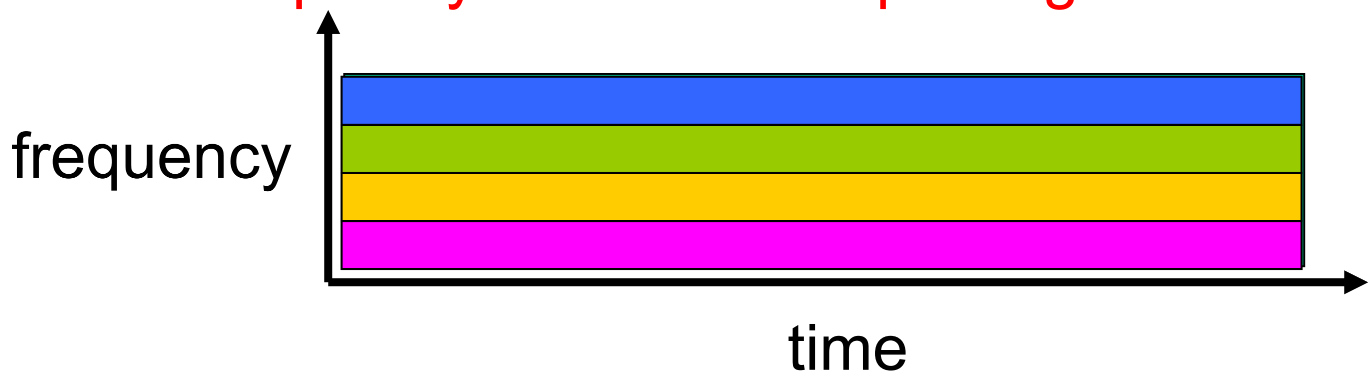
- pieces allocated to calls
- resource piece idle if not used by owning call (no sharing)
- dividing link bandwidth into "pieces"
  - frequency division
  - time division



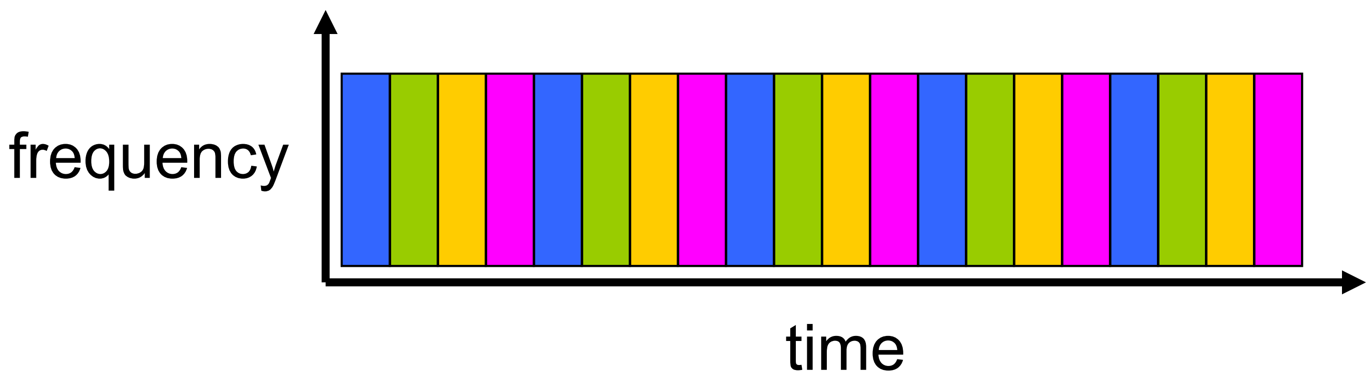
Circuit Switching: FDM and TDM

Example: 4 users

FDM: Frequency-Division Multiplexing



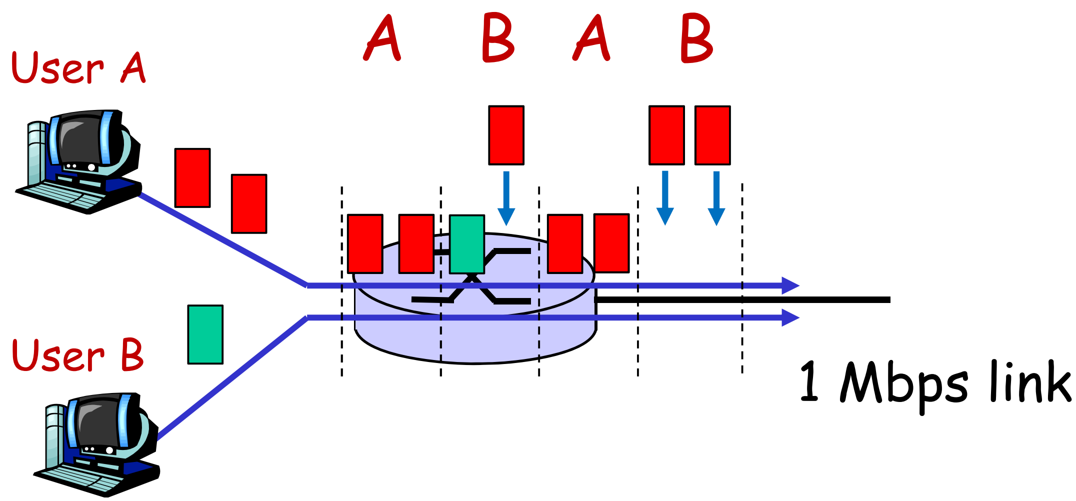
TDM: Time-Division Multiplexing



Network Core: Packet Switching

Dynamic sharing

- Multiple sessions can share one link
- Resources used as needed



Packetizing/Data segmentation   statistical multiplexing

Packet Switching: Packetizing

- A message is segmented into blocks of data called packets. A packet is a group of bits, typically from a few hundreds to thousands.

Header	Data	Trailer

- A packet consists of a header, user information, and a trailer. The header usually contains the addresses of the destination and of the source of the packet; it may also include a sequence number that the destination users to verify that all the packets were received or to reorder them. The trailer contains error control bits that the nodes use to verify that they received the packet correctly.

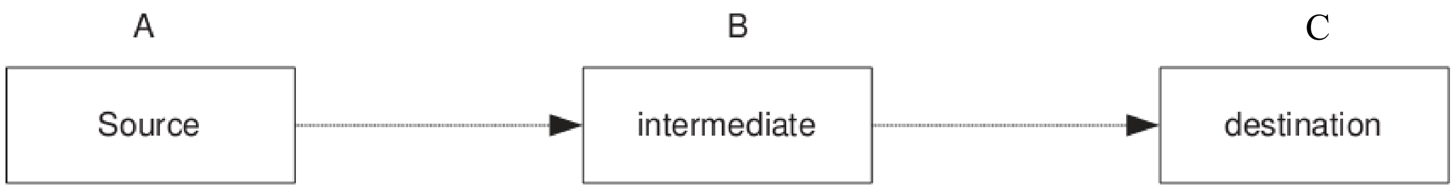
Packetizing: Pipelining gain

- Packets are transmitted without prior reservation of link capacity. When a packet is received at a switch, it is inspected to determine the appropriate output link. If the output link is available, it is transmitted. Otherwise, it is stored and then forwarded to the next switch on its way to the destination (store & forward network)

- The transmission of messages as small packets is called store-and-forward packet switching. The store-and-forward packet switching can reduce the message delivery time. The reduction in the delivery time is called the pipelining gain .

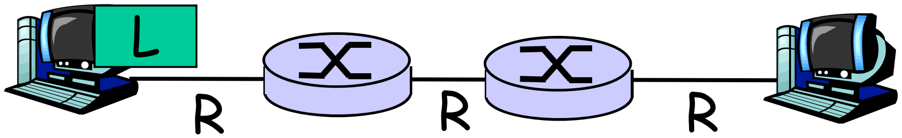
Packetizing: Pipelining gain

- Example: Transmission of a message on a direct link from A to B or B to C takes 1 minute. The total transmission time takes 2 minutes.



If the message is decomposed into 60 packets that takes 1 second each to be transmitted on a direct link, then during the first second packet 1 is sent from A to B , during the next second packet 2 is sent from A to B and packet 1 is sent from B to C , and so on. After 1 minute and 1 second, the complete message is received by C .

Packetizing: store-and-forward



- takes L/R seconds to transmit (push out) packet of L bits on to link at R bps
- store and forward: entire packet must arrive at router before it can be transmitted on next link
- delay = 3L/R (assuming zero propagation delay)

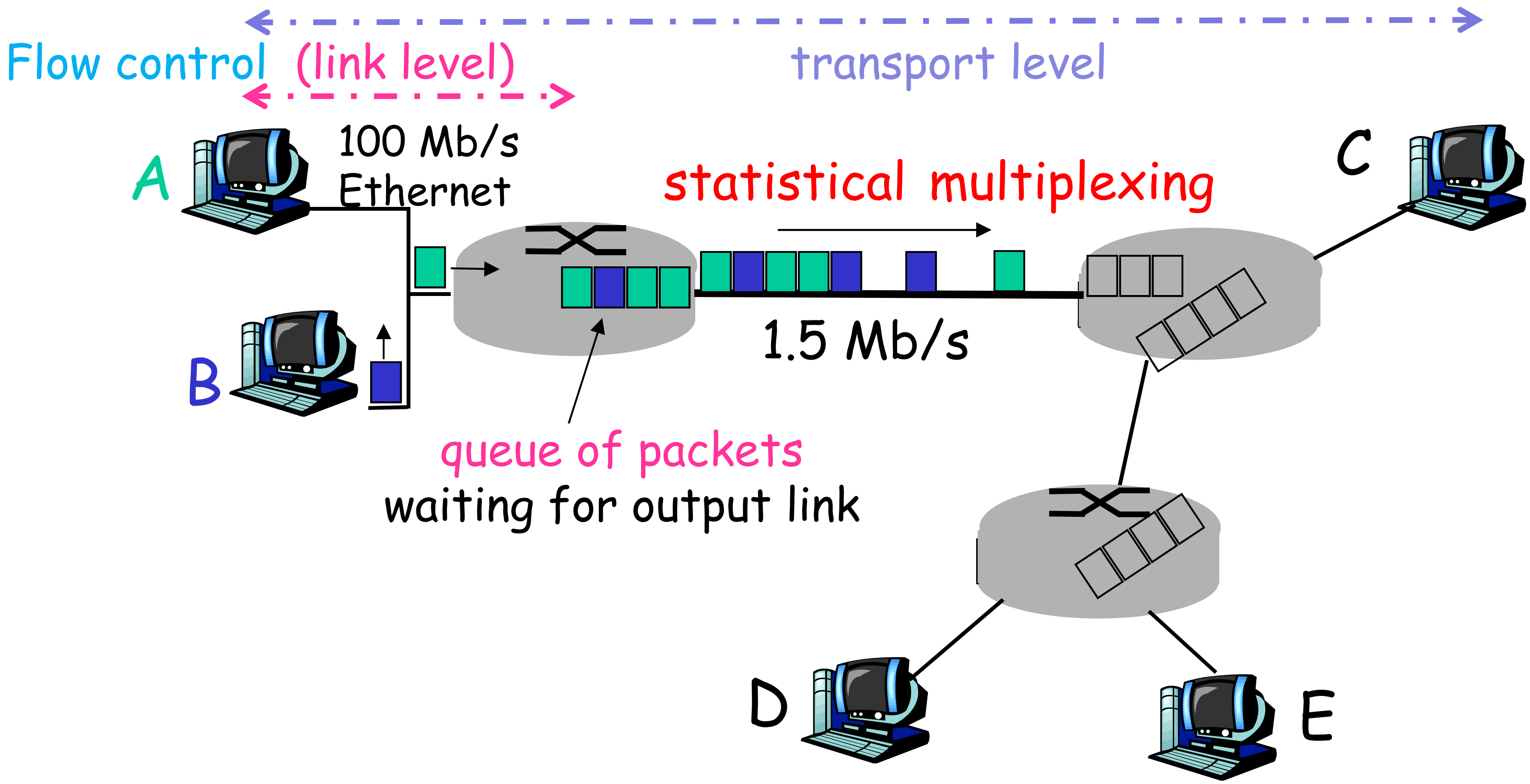
more on delay shortly …

Example:

- L = 7.5 Mbits

  (Note: packets are not that long! ~1.5KB is very common)
- R = 1.5 Mbps
- transmission delay (3L/R) = 15 sec

Packet Switching: Statistical Multiplexing

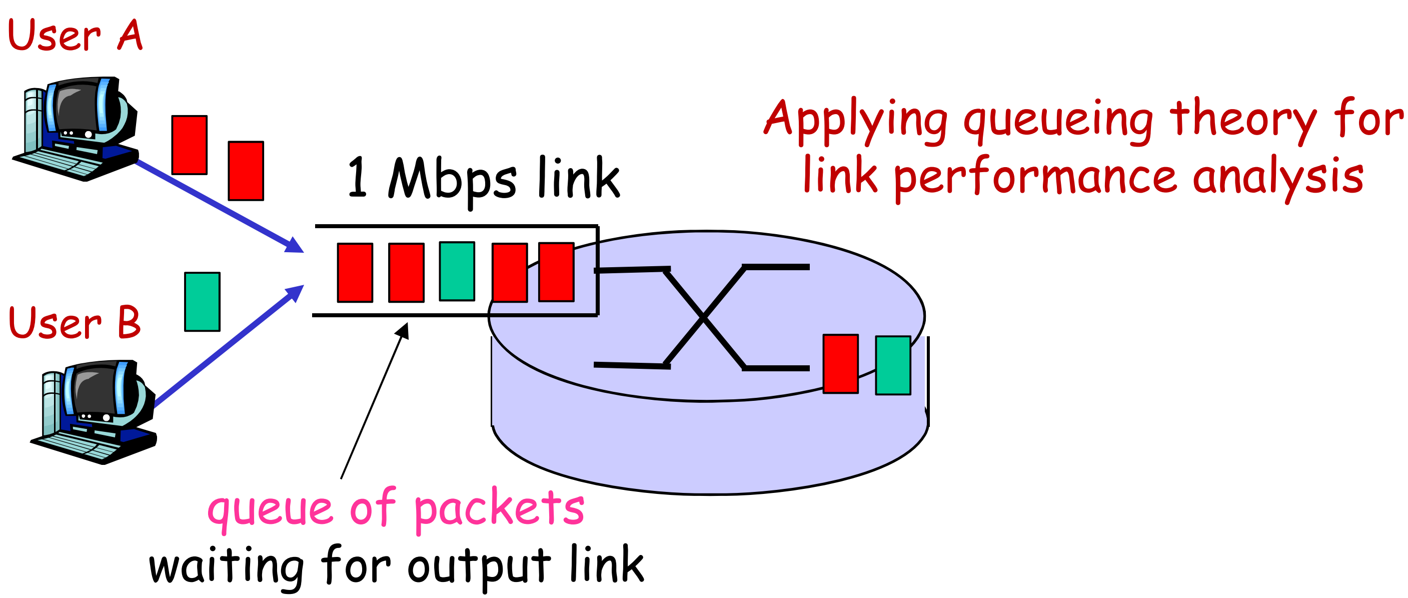


- sequence of A & B packets has no fixed timing pattern
  - bandwidth shared on demand: statistical multiplexing.

Packet Switching: Resource contention

Resource contention:

- Aggregate resource demand can exceed amount available
- Congestion: packets queue, wait for link use



Packet switching versus circuit switching

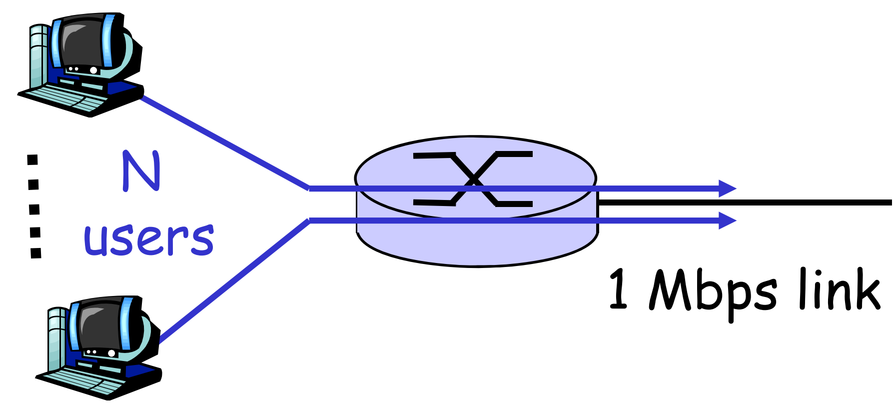
Packet switching allows more users to use network!

Example:

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

- circuit-switching:
  - 10 users

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



Circuit Switching Advantages

- Debate runs far and deep
  - Guarantee of Quality
- Circuit switching
  - Each session has a dedicated circuit
  - Throughput and delay performance will not change
- Packet switching
  - Best-effort service: no guarantees
  - Links get congested, messages arrive out of order, …

Packet Switching Advantages

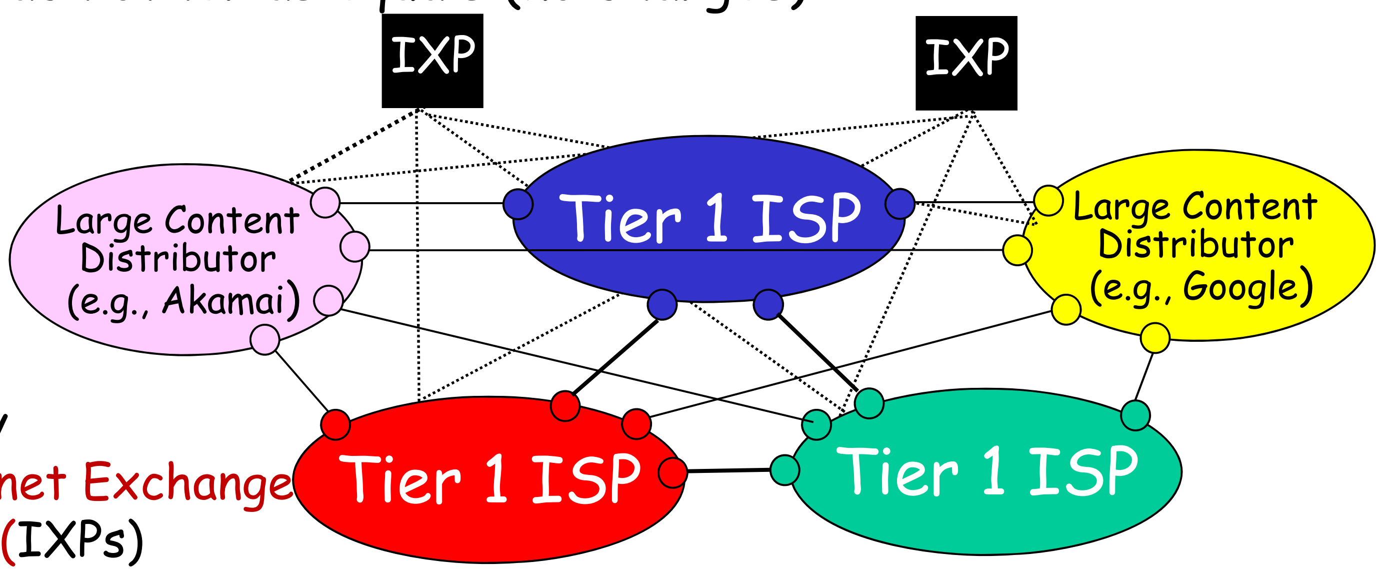
- Ease of connectivity
  - No need to allocate resources first
  - Transmit at will, as long as protocols are allowed
- Scalability
  - Large number of diverse sessions
  - Obtained through high efficiency
    - Statistical multiplexing

A comparison of circuit-switched and packet-switched networks

Item	Circuit switching	Packet switching
Dedicated path	Yes	No
Bandwidth available	Fixed	Dynamic
Potentially wasted	Yes	No
Store-and-forward trans.	No	Yes
Call setup	Yes	Not needed
When can congestion occur	At setup time	On every packet
Charging	Per minute	Per packet
Each packet follows the same route	Yes	No

Internet structure: network of networks

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



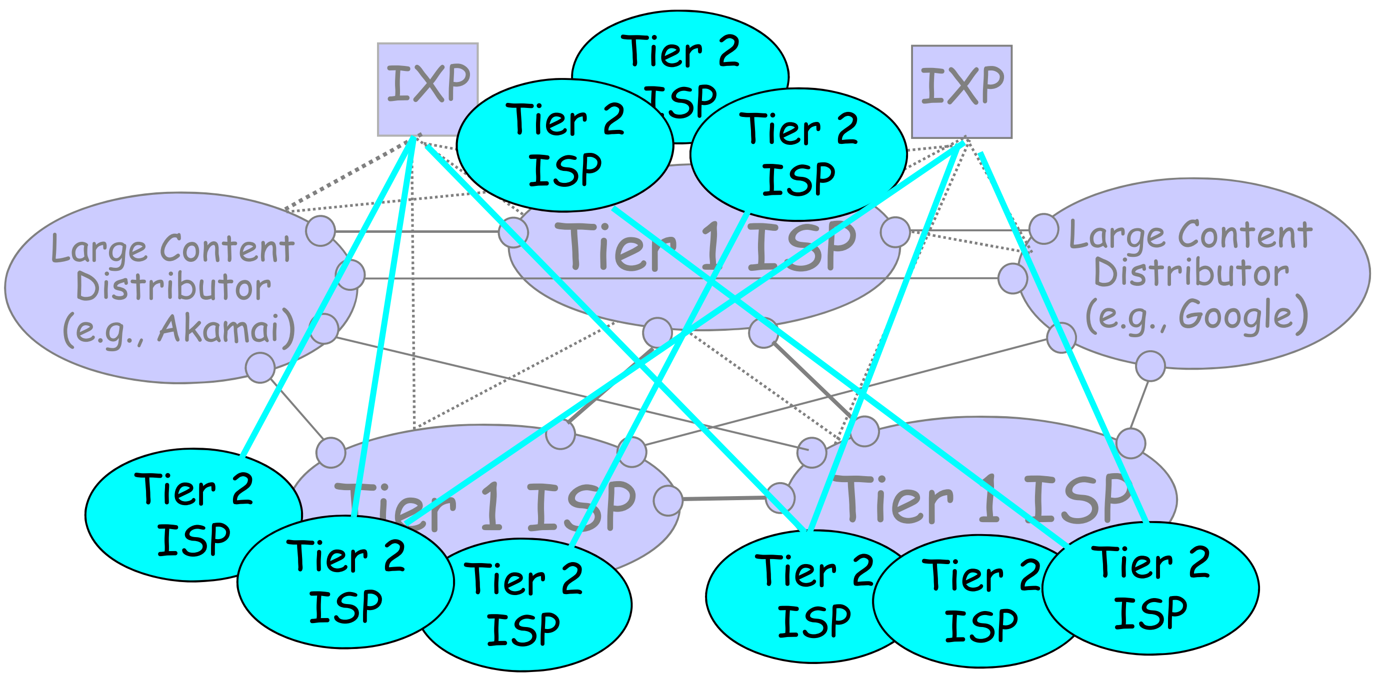
Tier-1 ISPs & Content Distributors, interconnect (peer) privately … or at Internet Exchange Points (IXPs)

Canadian Tier-1: MTS Allstream
(MTS: Manitoba Teleco Service)

Internet structure: network of networks

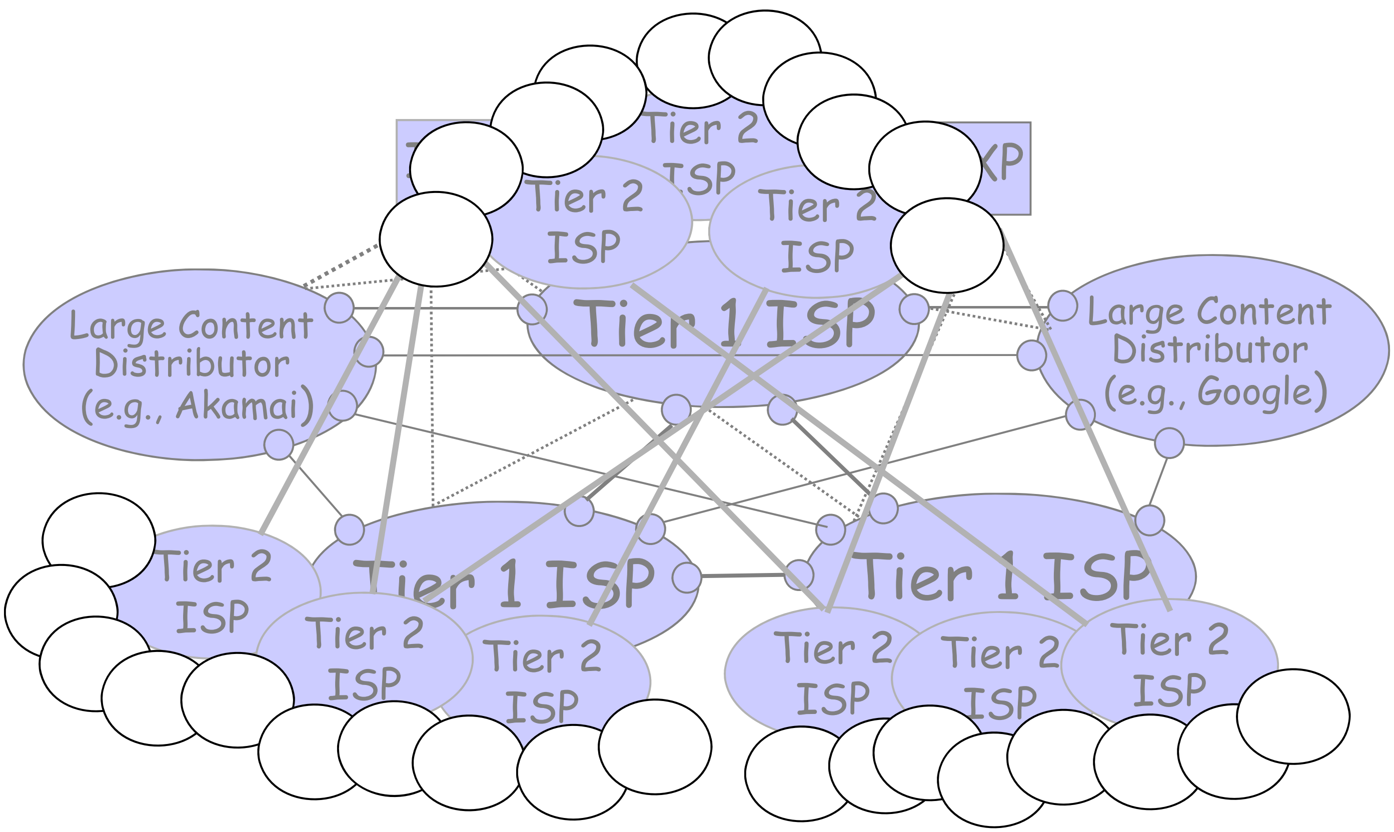
"tier-2" ISPs: smaller (often regional) ISPs

- 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



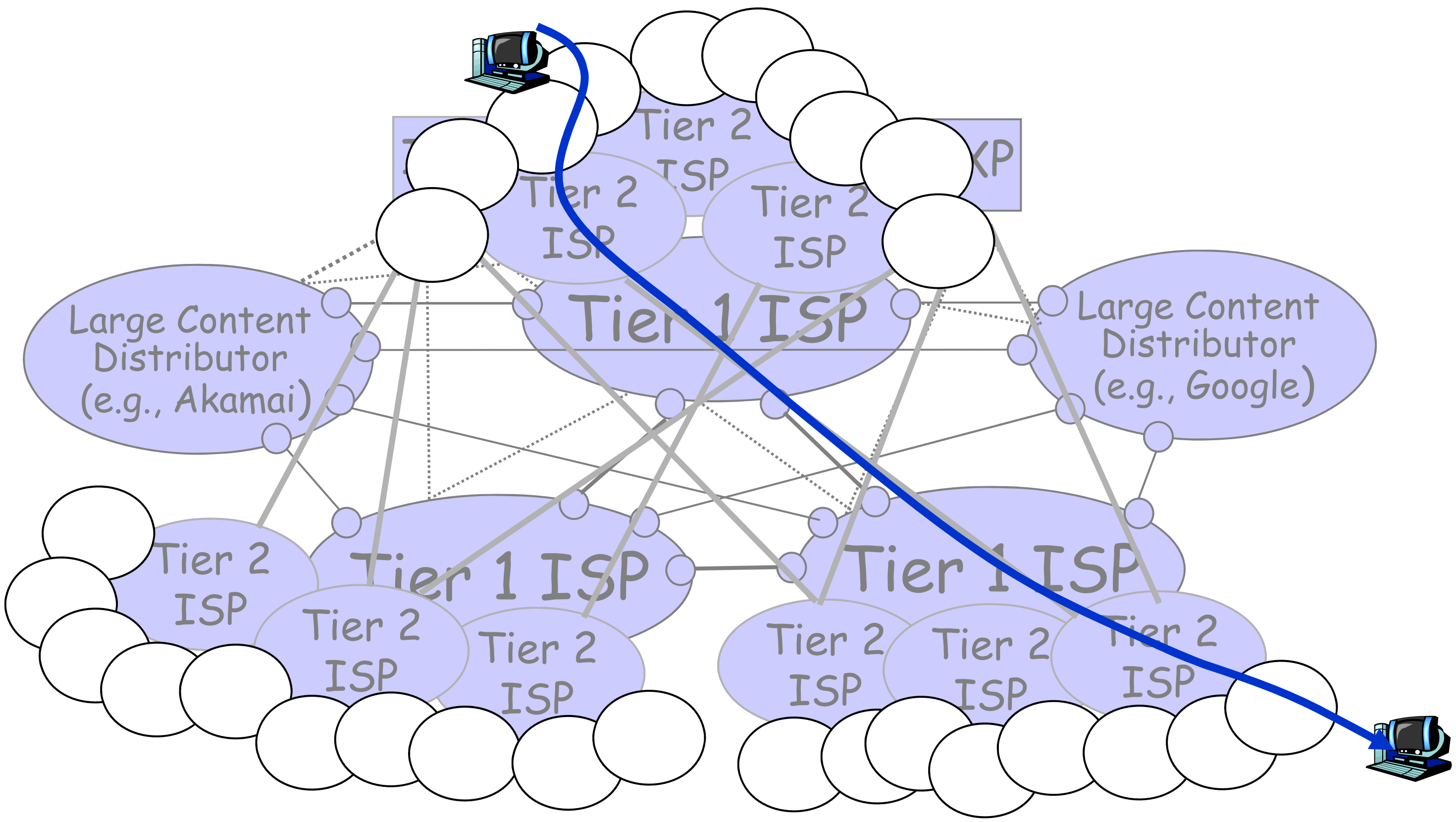
Internet structure: network of networks

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



Internet structure: network of networks

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



T1: Introduction

1.1 What is computer network?

- Examples of computer network
- The Internet
- Network structure: edge and core

1.2 Why computer networks

1.3 The way networks work

1.4 Performance metrics:

- Delay, loss and throughput in packet-switched networks

Why Computer Networks

- Resource sharing
- Efficiency
- High reliability
- Access to remote information
- Person to person communication
- Interactive entertainment
- Others ...

T1: Introduction

1.1 What is computer network?

Examples of computer network

The Internet

Network structure: edge and core

1.2 Why computer networks

1.3 The way networks work

1.4 Performance metrics:

Delay, loss and throughput in packet-switched networks

The Way Networks Work

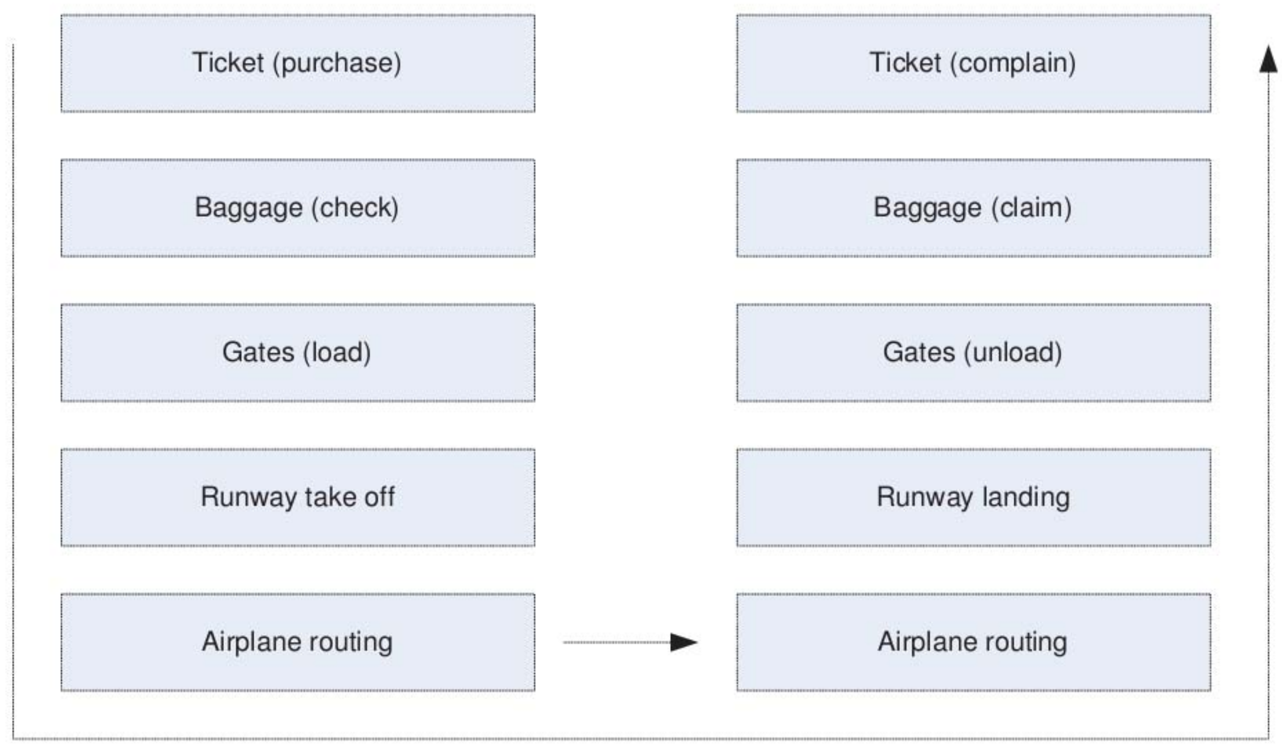
- Network functions are organized in a layered architecture. The services of one layer are implemented on top of the services provided by the layer immediately below.

- The different layers can be designed more or less independently, which greatly simplifies network design.

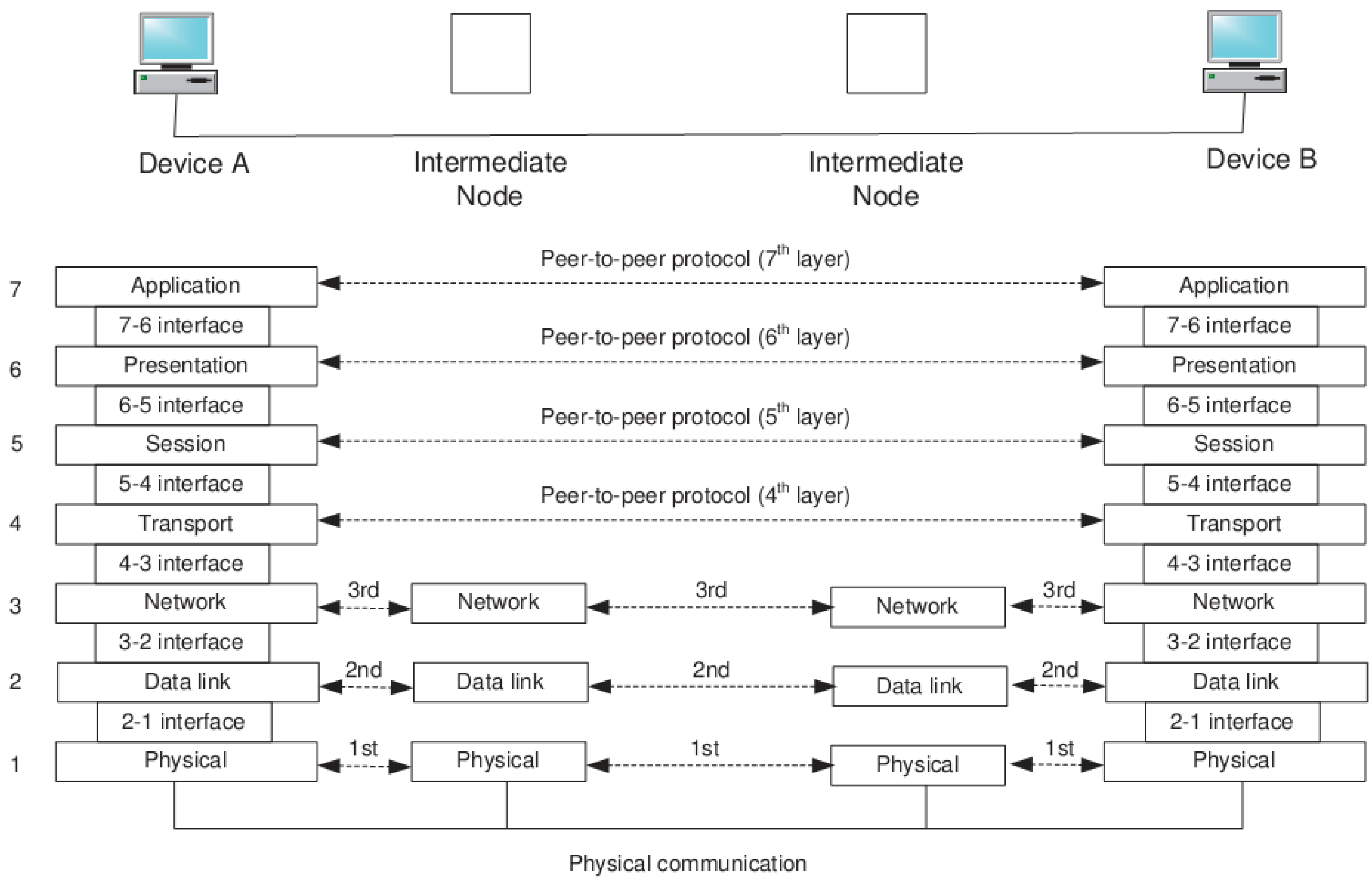
- Another advantage is the compatibility derived from the independence of the layers. For instance, when different networks are interconnected, a computer on one network can access computers on all the networks, independently of the specific implementations of the different networks.

The way Networks Work

The operation is organized in a layered architecture. Each layer implements a service via its own internal-layer actions relying on services provided by layer below.



- The International Organization for Standardization (ISO) proposed the open systems interconnection reference model (OSI model), which is a layered architecture with seven layers.



Definitions

- Protocol: a set of rules that governs communications. (A protocol defines what is communicated, how and when it is communicated);
- Layering: it achieves a functional level of modularity;
- Network architecture: a set of layers and protocols;
- Peer-to-peer protocols: protocols which make the layer N of the source and destination (counterpart) conceptual understanding
- Interface: it defines what information and services a layer must provide for the layer above it

Note:

- Peer-to-peer protocol is achieved using the service provided by their lower layer entities
- Communication functions are partitioned into a vertical set of layers
- Ideally, upper layer entities are independent of the details of the lower layer entities, this is easy to modify protocol at one layer, independent of other layers.
- Allows heterogeneous users to communicate (open system)

Functions of layers

- Application layer implements commonly used communication services e.g., file transfer, directory services, virtual terminal.
- Presentation layer takes care of data compression, security, and format conversions so that nodes using different representations of information can communicate efficiently and securely.
- Session layer uses the transmission layer services to set up and supervise connections between end systems
- Transport layer supervises end-to-end transmission of packets. This layer may arrange for retransmission of erroneous packets.
- Network layer guides the packets from their source to their destination, along a path that may comprise a number of links
- Data link layer provides reliable transmission between nodes that are attached to the same physical link.
- Physical layer transmits raw bit stream over channel

Summary of the functions of layers

The seven layers can be thought of as belonging to three subgroups.

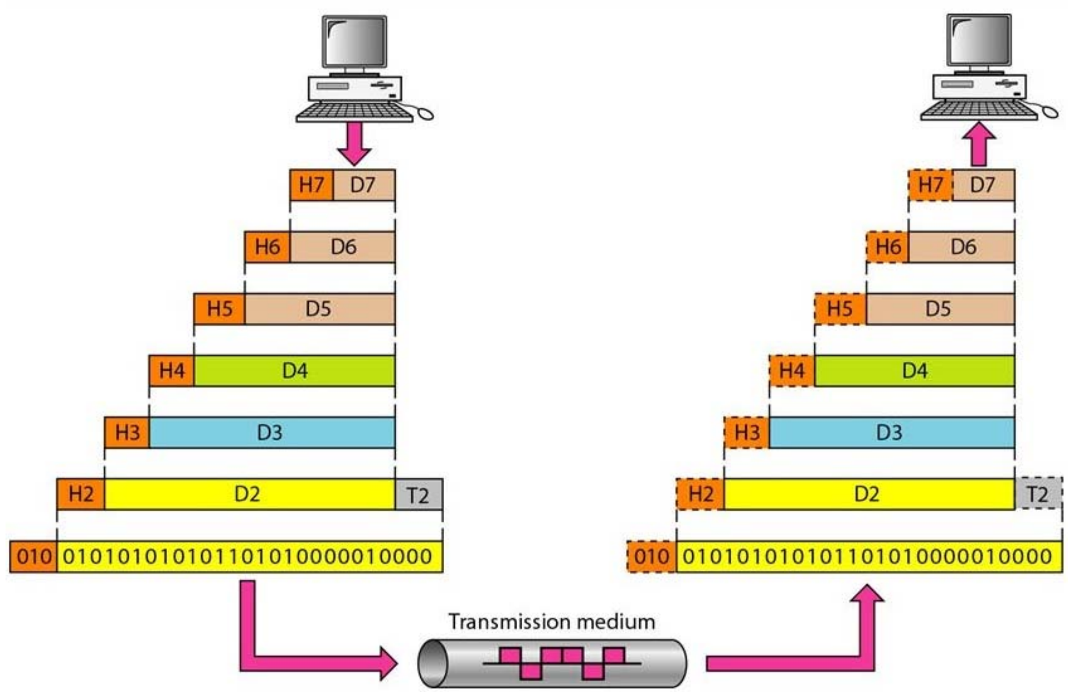
- Layers 1, 2, and 3 are the network support layers. They deal with the physical aspects of moving data from one device to another (such as specifications, physical connections, physical addressing, and transport timing and reliability).

- Layers 5,6, and 7 are the user support layers. They allow interoperability among unrelated software systems.

- Layer 4 links the two subgroups and ensures that what the lower layers have transmitted is in a form that the upper layers can use.

Headers and trailers

H2: physical address of the most recent node and next intended node;
T2: error control information;
H3: source and destination addresses;
H4: sequence number;
H5: connection control information (password and login verification);
H6: type and length of the transmission (type of code, etc)



An exchange using the OSI model

What's a protocol?

human protocols:

- "what's the time?"
- "I have a question"
- introductions

… specific msgs sent

… specific actions taken when msgs received, or other events

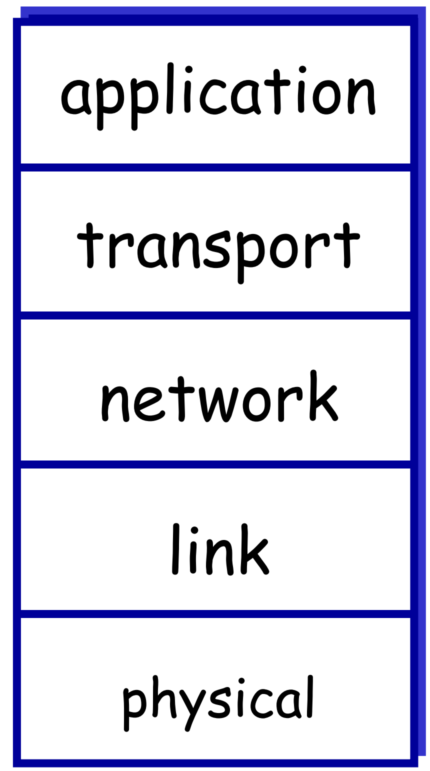
network protocols:

- machines rather than humans
- all communication activity in Internet governed by protocols

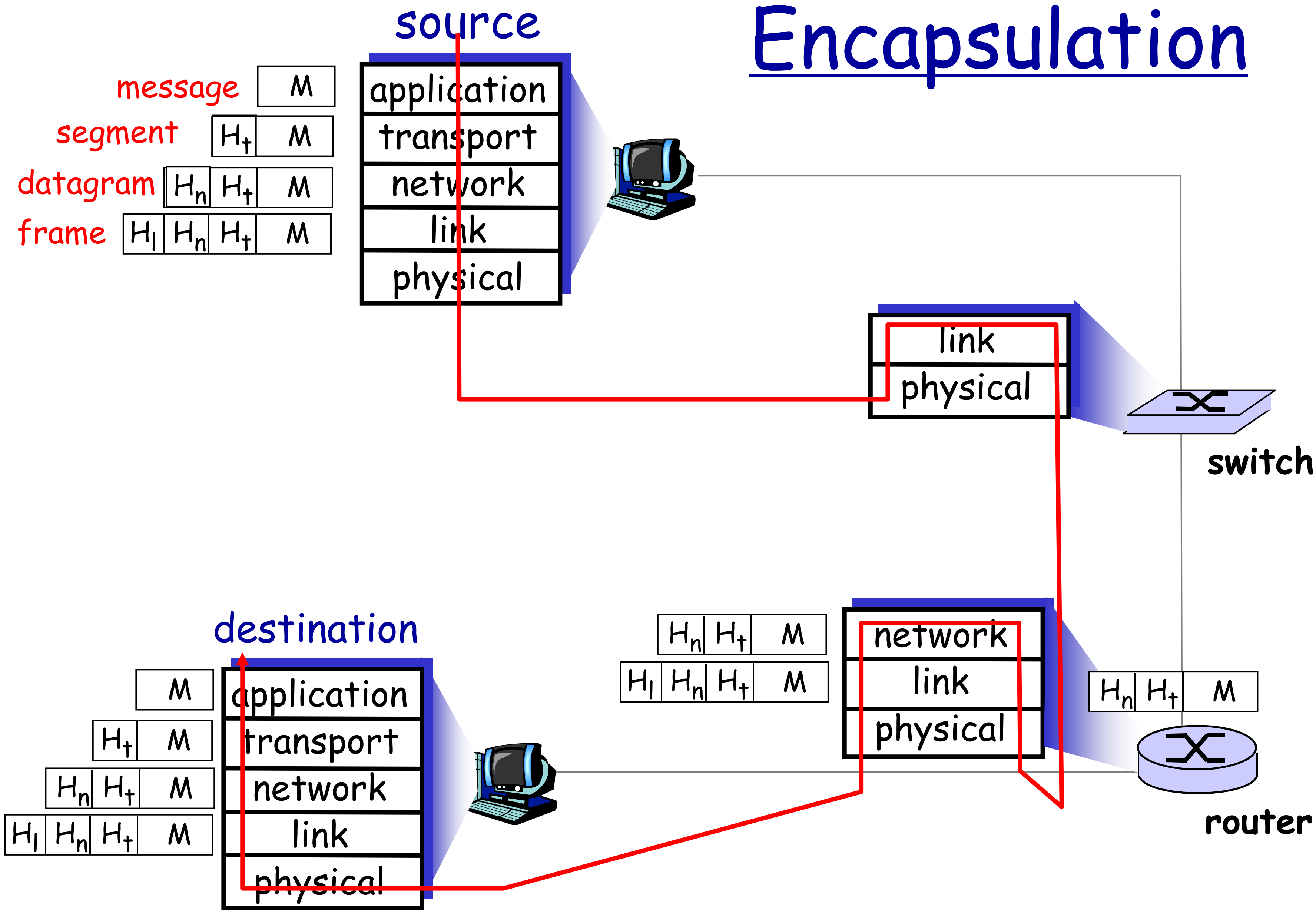
protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt

Internet protocol stack

- application: supporting network applications
  - Skype, SMTP, HTTP
- transport: process-process data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements (MAC/Link)
- physical: bits "on the wire": Ethernet, 802.11 (WiFi)



Encapsulation



T1: Introduction

1.1 What is computer network?
- Examples of computer network
- The Internet
- Network structure: edge and core

1.2 Why computer networks

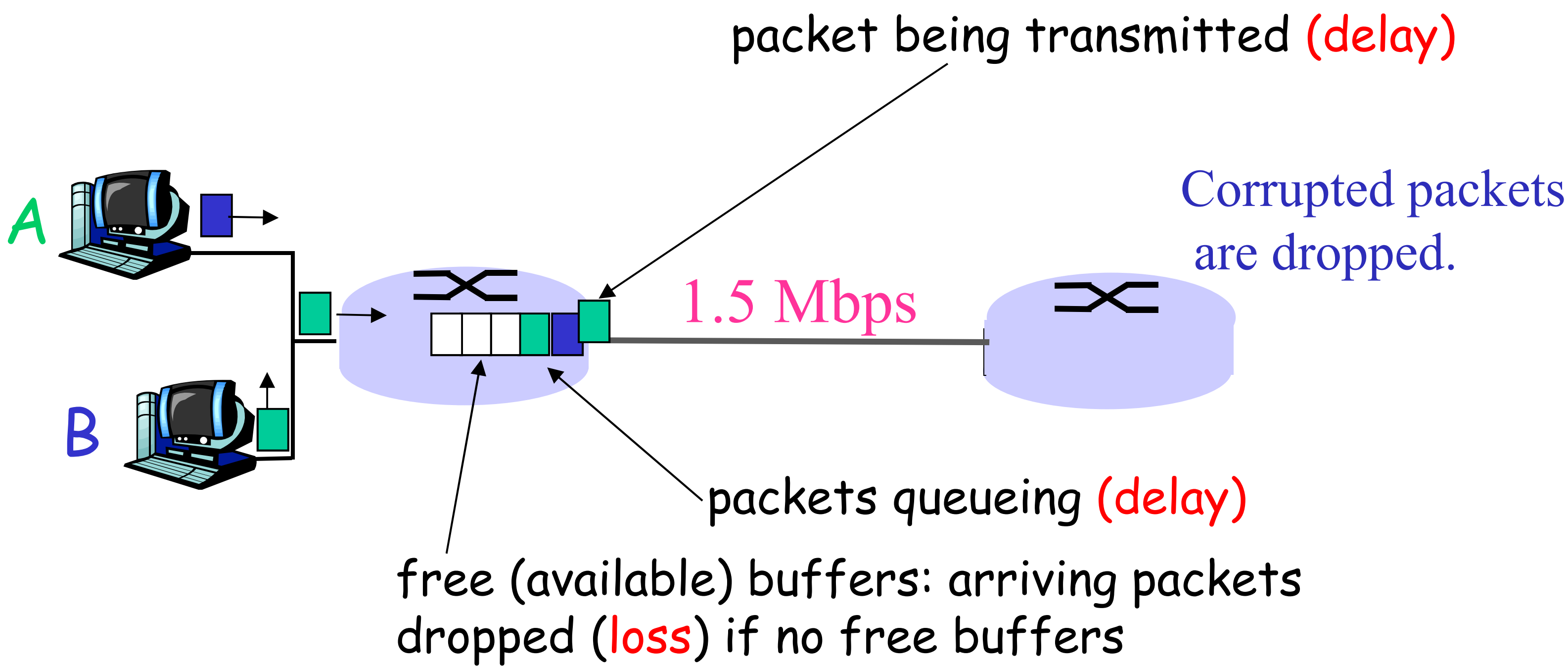
1.3 The way networks work

1.4 Performance metrics:
- Delay, loss and throughput in packet-switched networks

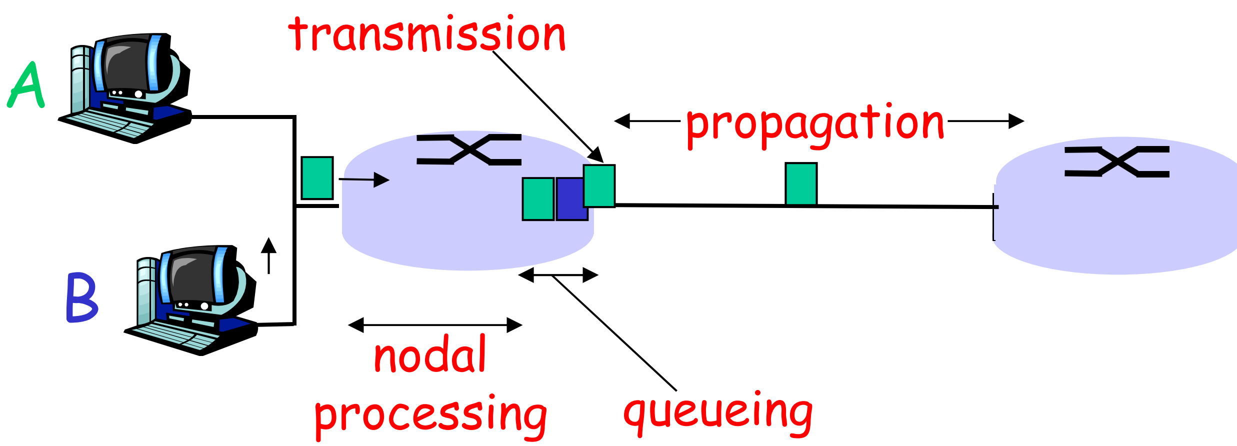
How do loss and delay occur?

packets queue in router buffers (i.e., memory)

- packet arrival rate to link exceeds output link capacity
- packets queue, wait for turn



Four sources of packet delay

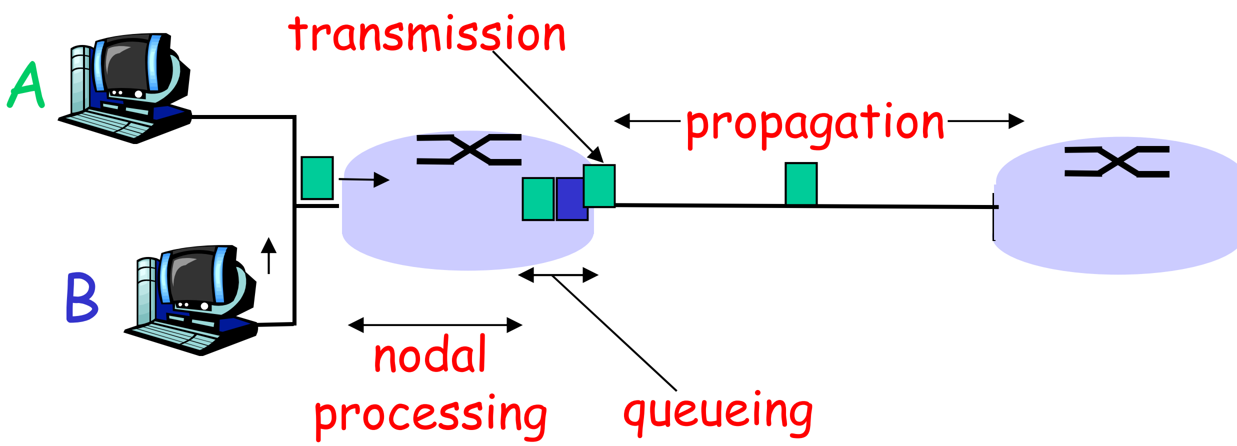


$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{proc}: nodal processing
- check bit errors
- determine output link
- typically < msec

d_{queue}: queueing delay
- time waiting at output link for transmission
- depends on congestion level of router

Four sources of packet delay



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

d_{trans}: transmission delay:

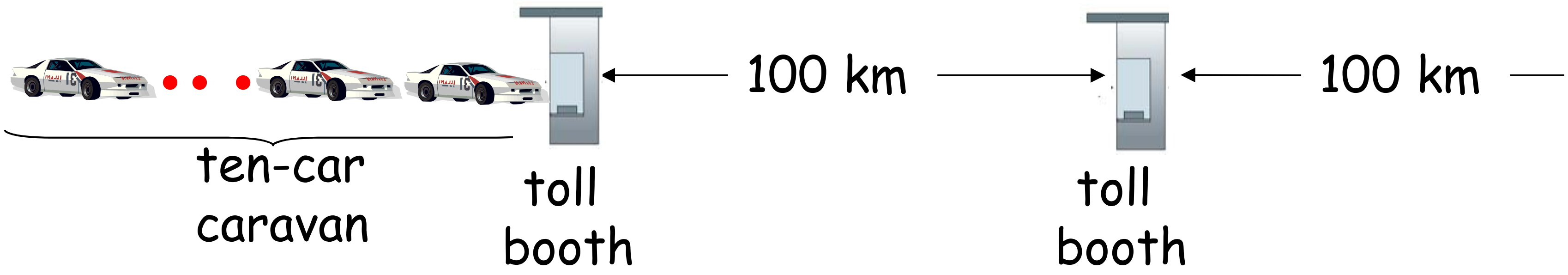
- L: packet length (bits)
- R: link bandwidth (bps)
- $d_{trans} = L/R$

d_{prop}: propagation delay:

- d: length of physical link
- s: propagation speed in medium (~2×10^8 m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop} very different

Caravan analogy

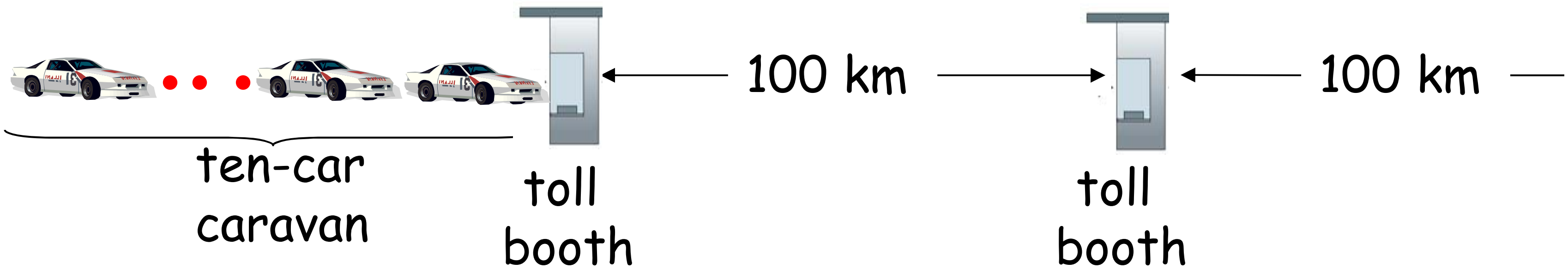


- cars "propagate" at 100 km/hr
- toll booth takes 12 sec to service car (transmission time)
- car~bit; caravan ~ packet

- Q: What is the "queueing delay" experienced by the last car at the first toll booth?

  12*9 = 108 sec

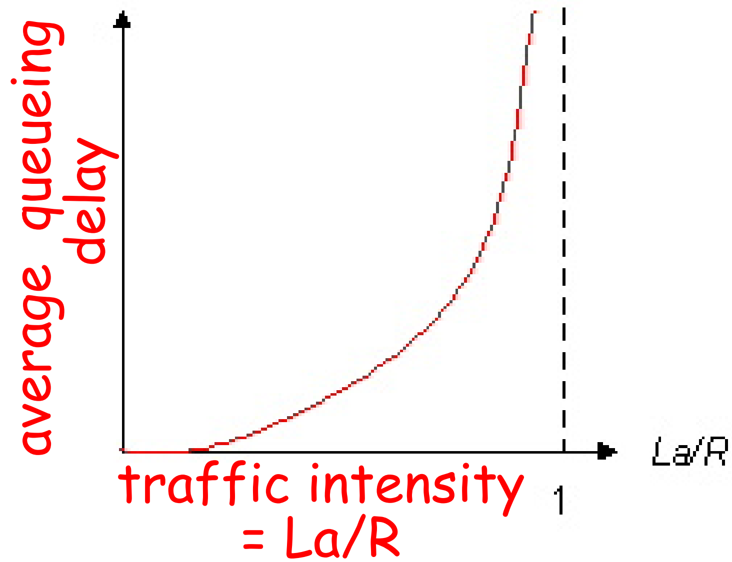
Caravan analogy



- cars now "propagate" at 1000 km/hr
- toll booth now takes 1 min to service a car
- Q: Will cars arrive to 2nd booth before all cars serviced at 1st booth?
  - A: Yes! After 7 min, 1st car arrives at second booth; three cars still at 1st booth.
  - 1st bit of packet can arrive at 2nd router before packet is fully transmitted at 1st router!

Queueing delay

- R: link bandwidth (bps)
- L: packet length (bits)
- a: average packet arrival rate (#packets/s)



- La/R ~ 0: avg. queueing delay small
- La/R -> 1: avg. queueing delay large
- La/R > 1: more "work" arriving

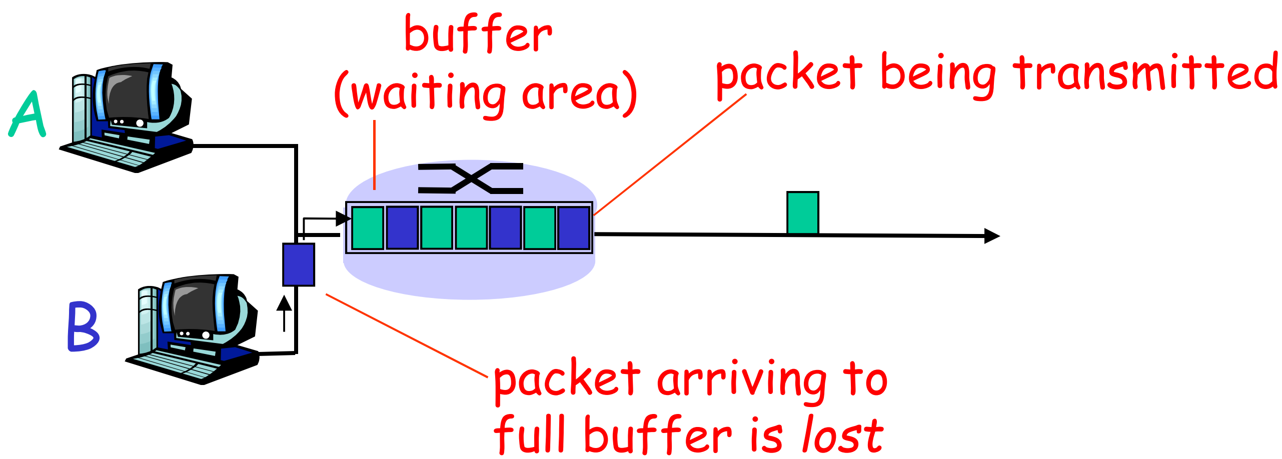
  than that can be serviced, average delay infinite!

La/R ~ 0

La/R -> 1

Packet loss

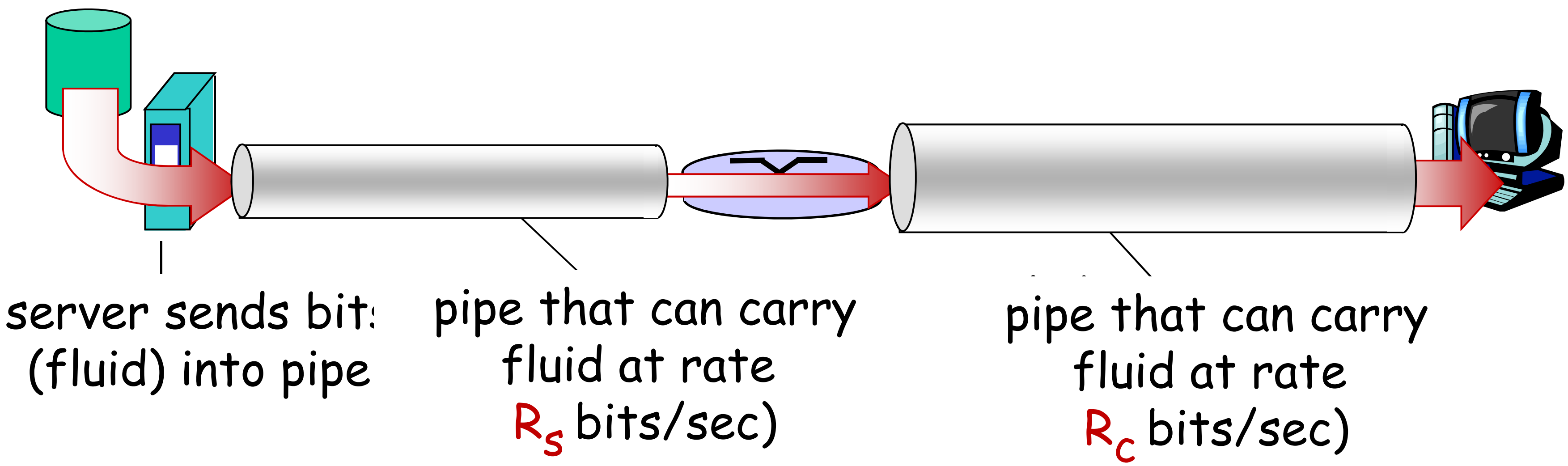
- queue (i.e., buffer) preceding link has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



Throughput

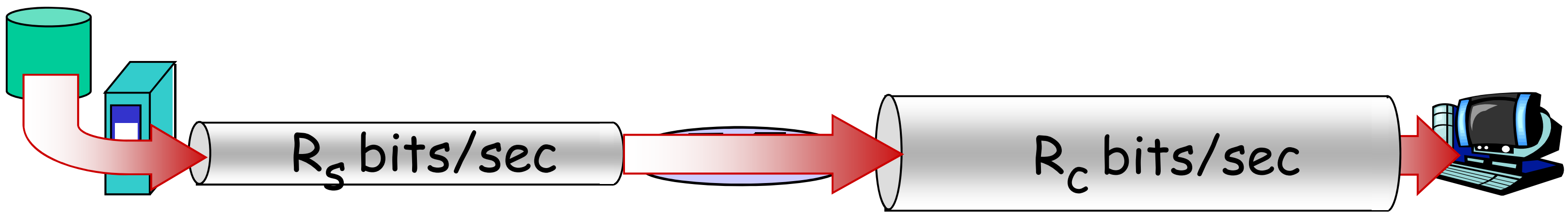
(the output rate of an input/output system)

- throughput: rate (bits/time unit) at which bits transferred between sender/receiver
  - instantaneous: rate at given point in time
  - average: rate over longer period of time

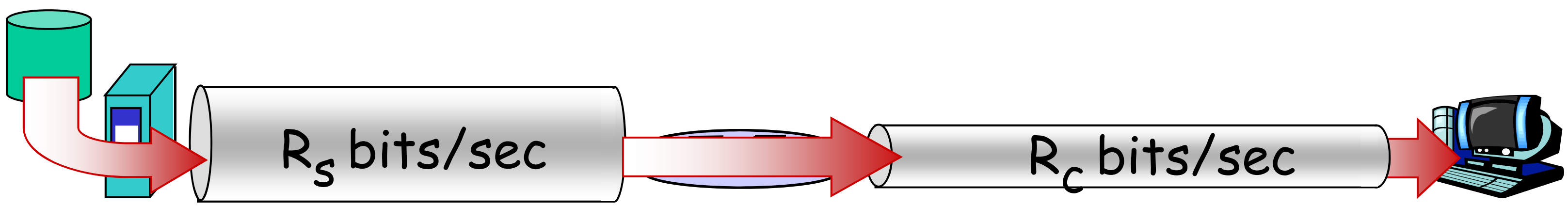


Throughput

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



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

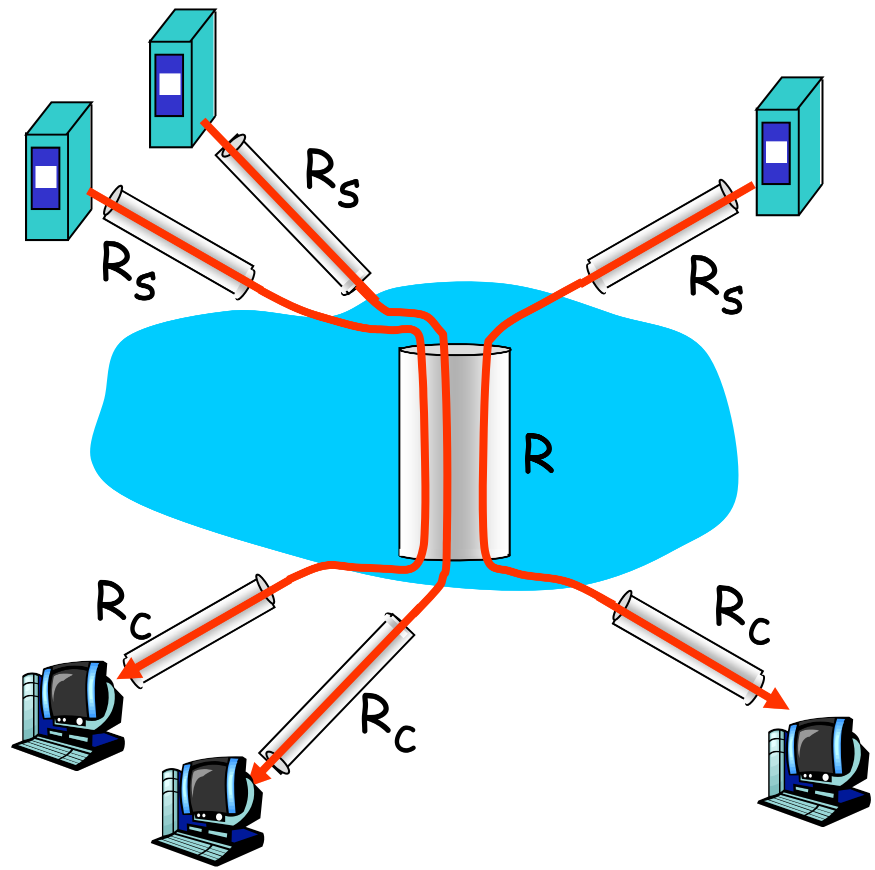


bottleneck link

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

Throughput: Internet scenario

- per-connection end-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 bits/sec

Summary of T1

1.1 Definition of computer network/the Internet.

Network edge

- end systems, access networks, links

Network core

- circuit switching, packet switching, network structure

1.2 Benefits of computer networks

1.3 The way networks work

1.4 Performance metrics:

Delay, loss and throughput in packet-switched networks

What are we going to learn?

- T2 Network Analysis and Queuing Theory
- T3 Physical layer: digital transmission
- T4 Data link layer: Error detection and correction, Retransmission, Multiple Access protocols (MAC): Aloha, CSMA/CD and CSMA/CA.
- T5 Network layer: IP addressing, fragmentation, routing
- T6 Transport layer: TCP and UDP, Flow control and Congestion control
- T7 Application layer: HTTP, MQTT
- ST Engineering Data Center Networks