

Advanced Computer Network

LECTURE 2

Routing and Forwarding
Switching Techniques
Network performance

Content:

- Packet Switching and Network Layer
- Introducing the Routing and Forwarding
- Connectionless vs. Connection-Oriented Services
- Forwarding Process in Routers
- Network Performance Metrics (Delay, Throughput, Packet Loss)
- Throughput Example

Packet Switching and Network Layer

- The network layer in the Internet today is a packet-switched network.
- **Mechanism of Source and Destination Computers**
 - 1) In this type of network, a message from the upper layer is **divided into manageable packets** and each packet is sent through the network.
 - 2) The source of the message sends the packets **one by one as a Flow**;
 - 3) the destination of the message receives the **packets one by one as a Flow**.
 - 4) The **destination waits for all packets** belonging to the same message to arrive before delivering the message to the upper layer.
- **Mechanism of connecting devices (switches and Routers)**
 - 1) The connecting devices in a packet-switching network still need to decide how to route the packets to the final destination.

Introducing the Routing and Forwarding

Duties of the network layer:

They are routing and forwarding, which are **directly related to each other**.

- **Routing**

This means that there is more than one route from the source to the destination, the network layer is responsible for finding the best one. This is done by **running some routing protocols** to:

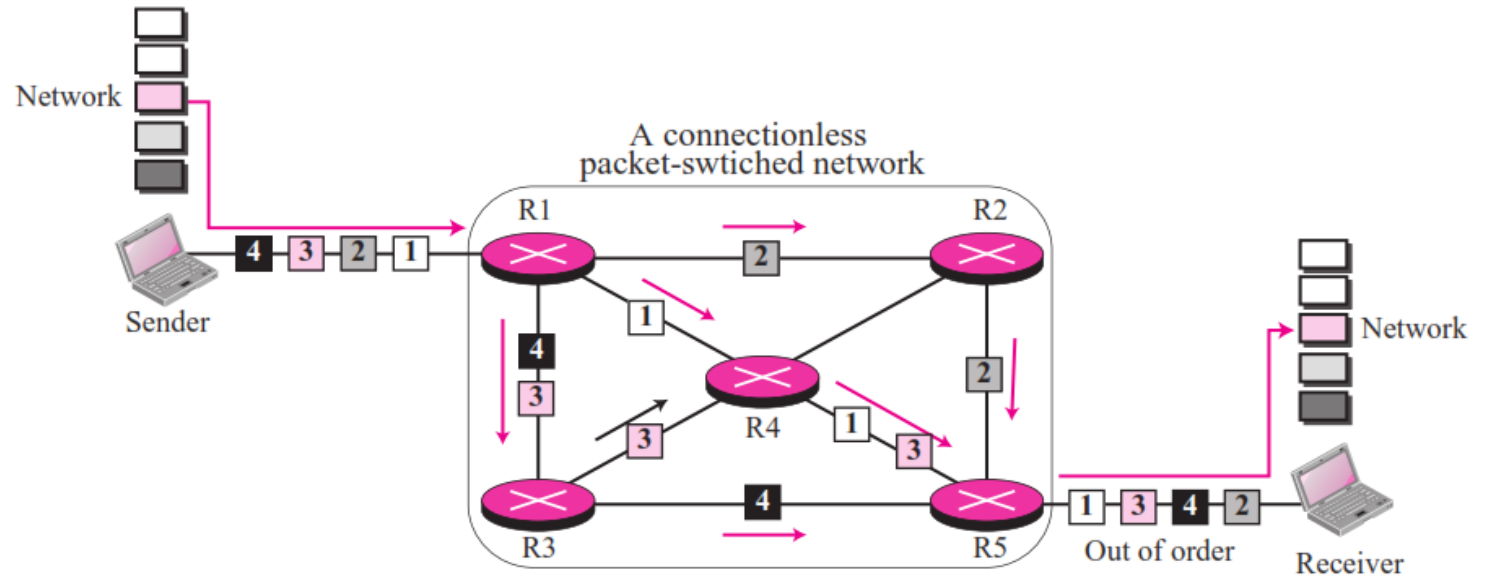
- 1) help the routers **coordinate** their knowledge about the neighborhood (**table convergence**)
- 2) and to **build a consistent tables** to be used when a packet arrives.

- **Forwarding**

The action applied by each router when a packet arrives at one of its interfaces. A router normally use a decision-making table for applying this action. It is called (the forwarding table) or (the routing table).

Connectionless Service (Datagram Approach)

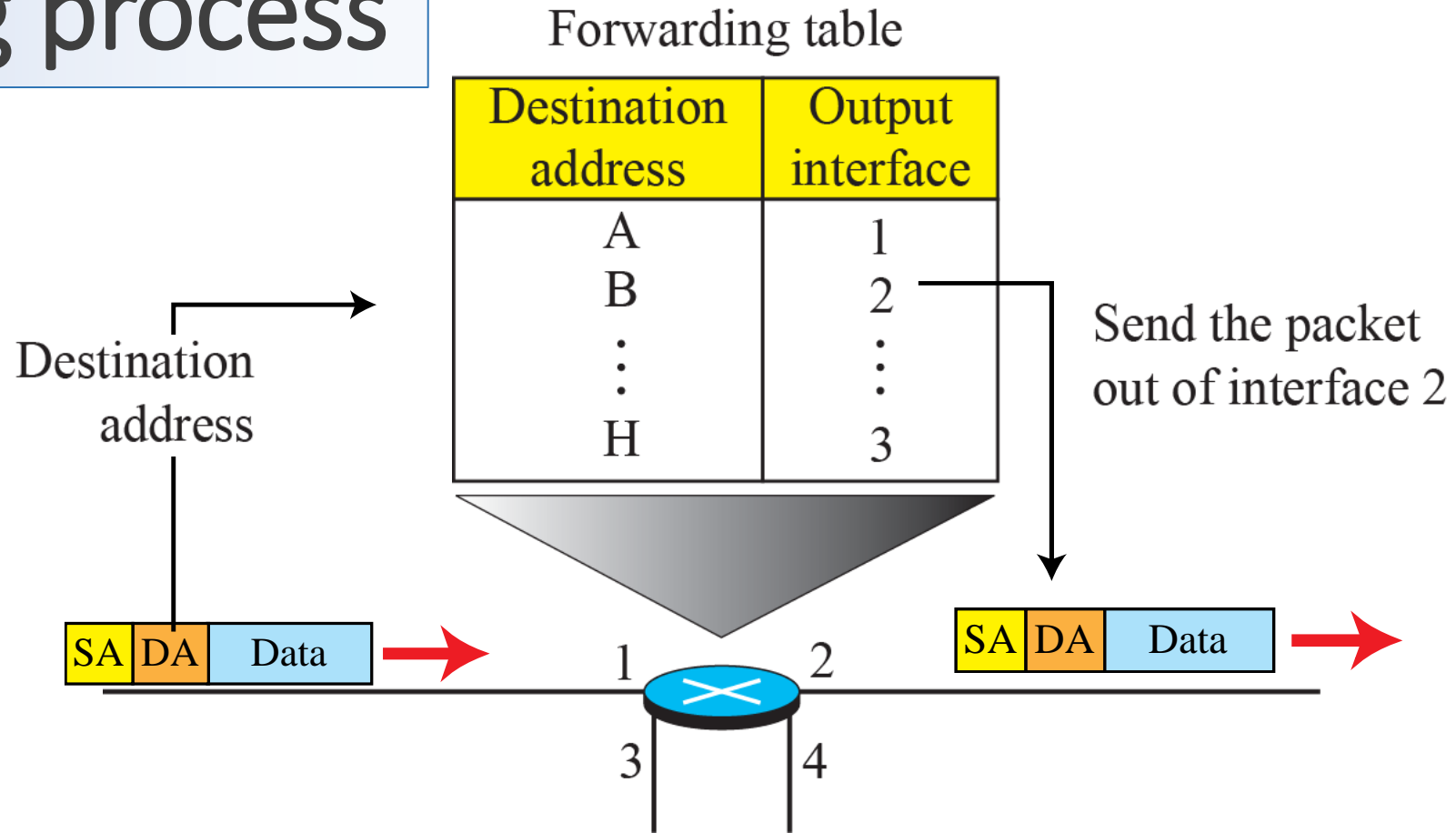
- When the **Internet started**, the network layer was designed to provide a **connectionless service**,
- in which the network layer protocol **treats each packet independently**,
- with each packet having **no relationship** to any other packet.
- The packets in a message may or **may not travel the same path** to their destination.
- When the Internet started, it was decided to make the network layer a connectionless service to **make it simple**.
- The idea was that the network layer is only **responsible for delivery of packets from the source to the destination**.



Advantages:

- **Simplicity:** No need to establish a connection before sending data.
- **Scalability:** Suitable for large networks like the Internet.

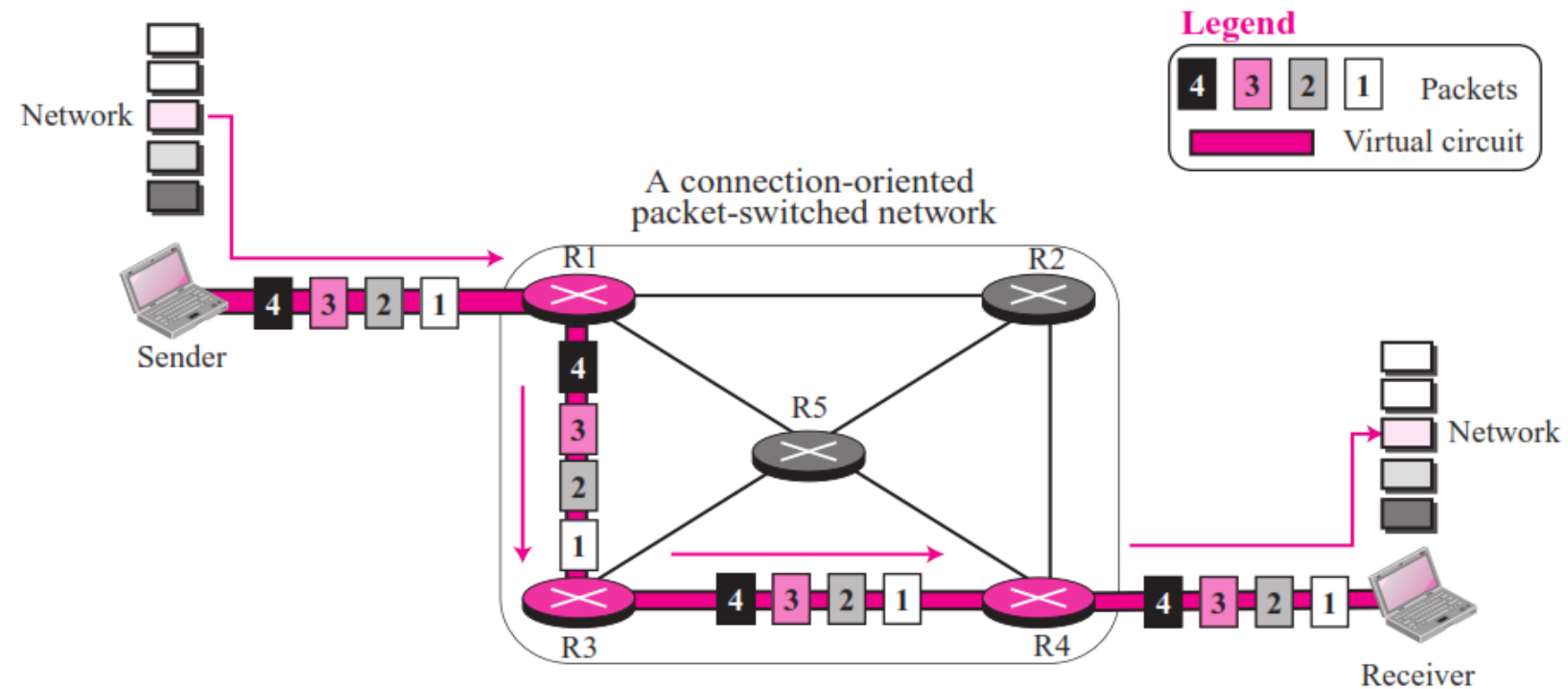
Forwarding process



- When a router receives a packet from **one of its attached networks**, it needs to **forward the packet to another attached network** (in unicast routing) or to some attached networks (in multicast routing).
- To make this decision, the router uses a piece of information in the packet header, which can be the **destination address** or a **label**, to **find the corresponding output** interface number in the forwarding table.

Connection-Oriented Service (Virtual Circuit Approach) (Tunnels)

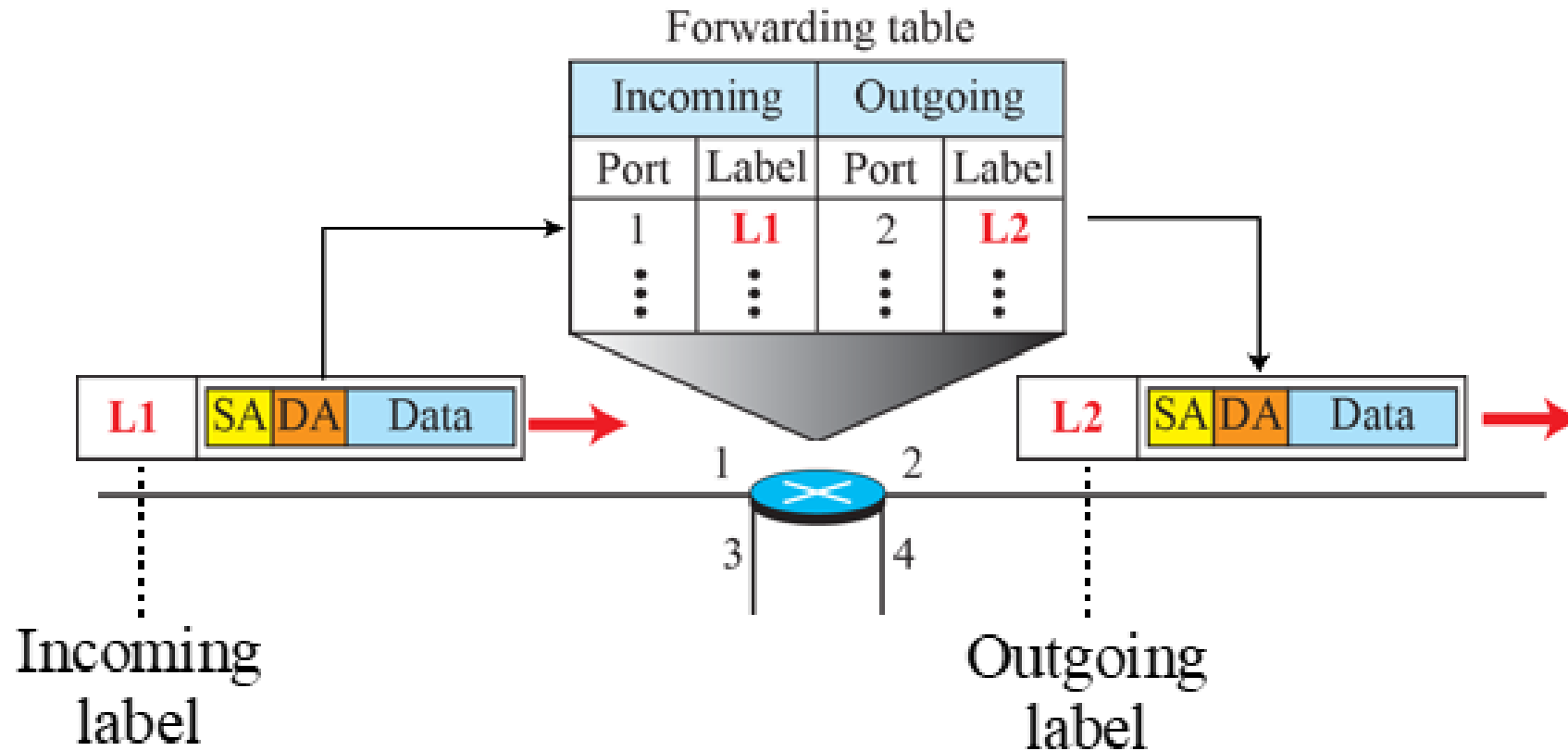
In a connection-oriented service, there is a relation between all packets belonging to a message. Before all datagrams in a message can be sent, a **virtual connection** should be set up to define the path for the datagrams. After connection setup, the datagrams can follow the same path. In this type of service, not only must the packet contain the source and destination addresses, it must also contain a flow label, a virtual circuit identifier that defines the virtual path the packet should follow.



Advantages:

- Predictable path and order of packets.
- Better QoS and Privacy.

Forwarding process in a router when used in a virtual circuit network

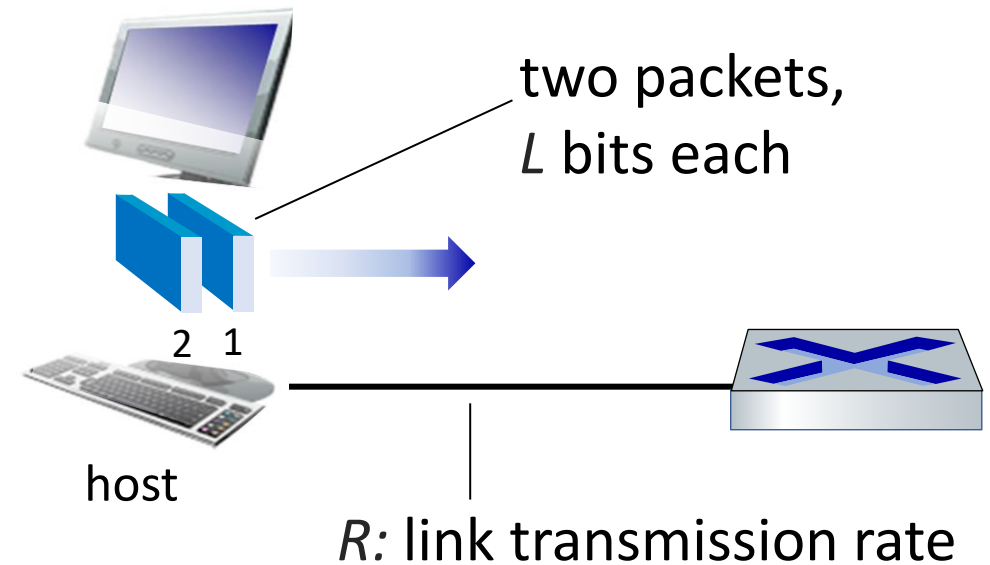


The forwarding decision is based on the value of the label.

Host: sends *packets* of data

host sending function:

- takes application message
- breaks into smaller chunks, known as *packets*, of length L bits
- transmits packet into access network at *transmission rate* R
 - link transmission rate, aka link *capacity*, aka *link bandwidth*



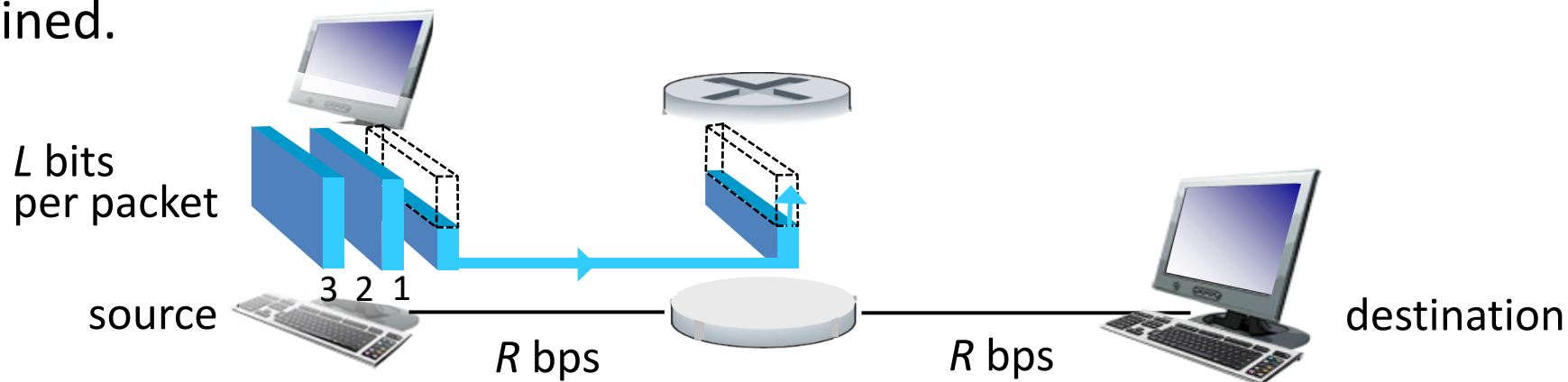
$$\text{packet transmission delay} = \text{time needed to transmit } L\text{-bit packet into link} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

Forwarding Process in Routers

- **Store-and-forward switching**
- **Cut-through switching**
 - 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
 - *Cut-through* starts forwarding a frame (or packet) before the whole frame has been received, normally as soon as the destination address and outgoing interface is determined.

One-hop store and forward numerical example:

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



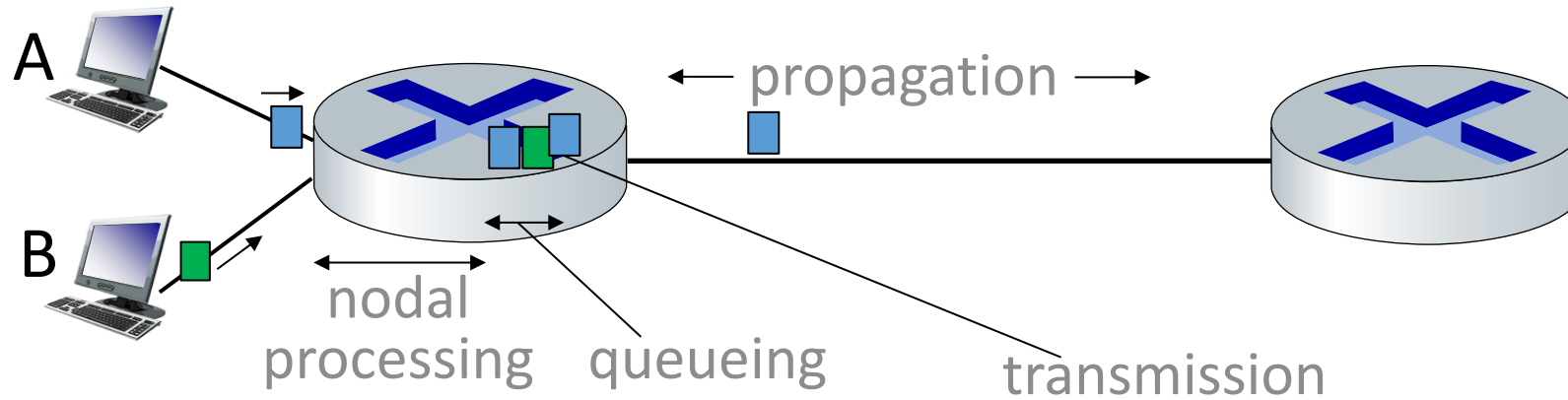
Network performance metrics

Metrics:

- **Delay:** Time taken for a packet to travel from source to destination.
 - Types: Transmission, Propagation, Processing, Queuing.
- **Throughput:** Rate at which data is successfully delivered.
- **Packet Loss:** Percentage of packets lost during transmission.
- **Example:** High queuing delay occurs when a router is congested.

Packet delay: four sources

The delays in a network can be divided into four types:
transmission delay, propagation delay, processing delay, and queuing delay.



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

d_{proc} : nodal processing

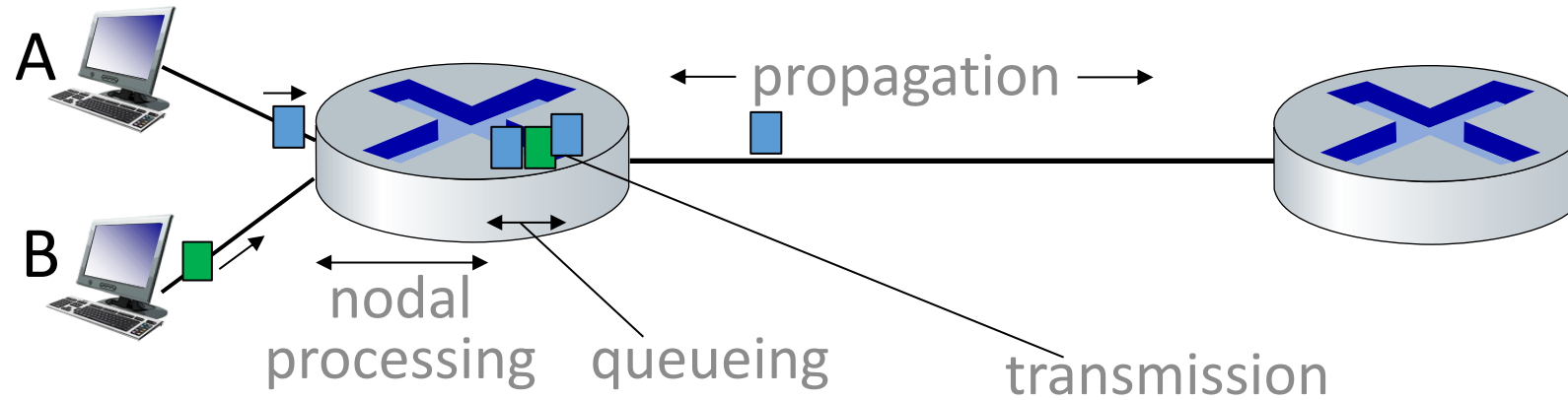
- check bit errors
- determine output link
- typically < microseconds

d_{queue} : queueing delay

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

Packet delay: four sources

The delays in a network can be divided into four types:
transmission delay, propagation delay, processing delay, and queuing delay.



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

d_{trans} : transmission delay:

- L : packet length (bits)
- R : link transmission rate (bps)

$$\text{▪ } d_{\text{trans}} = L/R$$

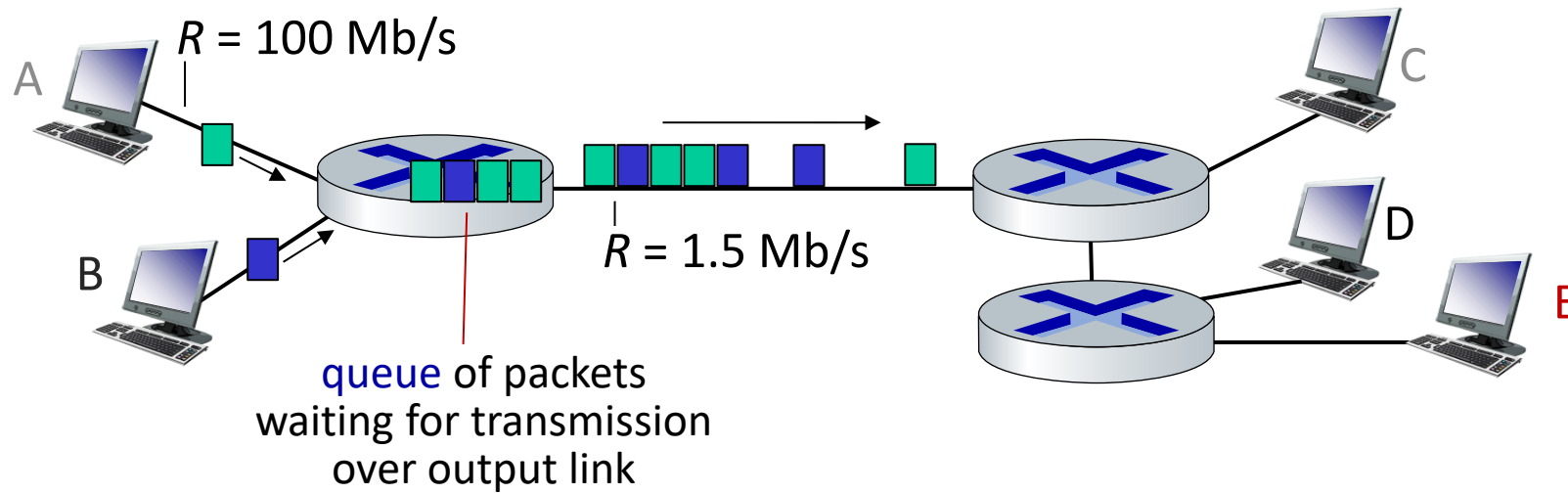
d_{prop} : propagation delay:

- d : length of physical link
- s : propagation speed ($\sim 2 \times 10^8$ m/sec)

$$\text{▪ } d_{\text{prop}} = d/s$$

d_{trans} and d_{prop}
very different

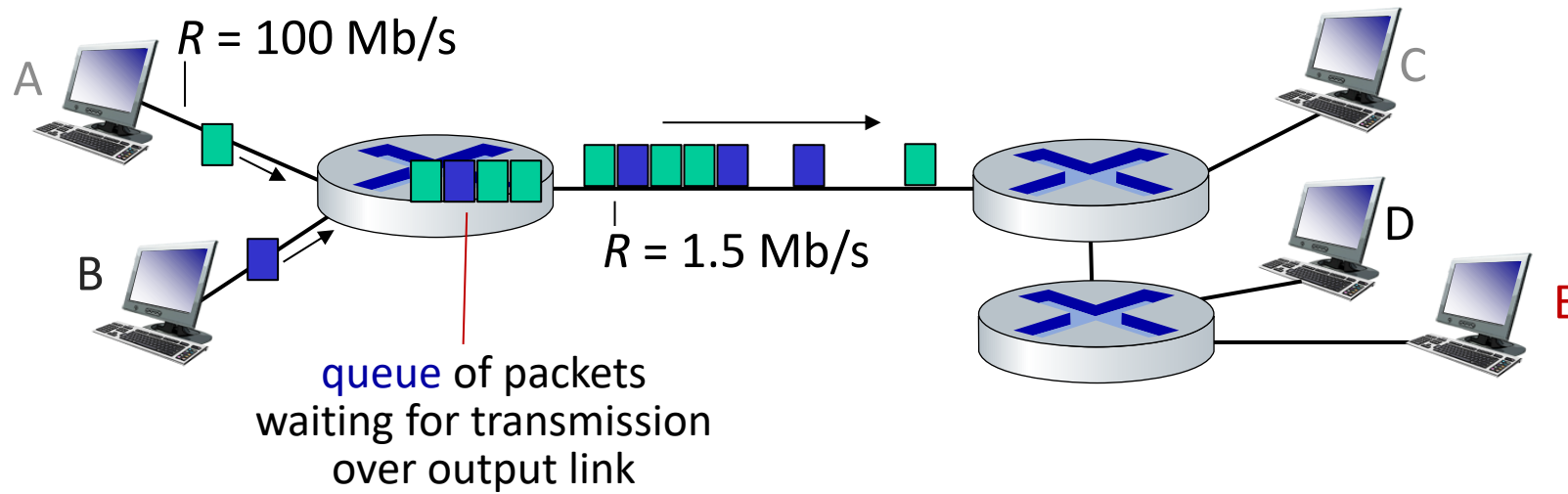
Queueing



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



Queueing

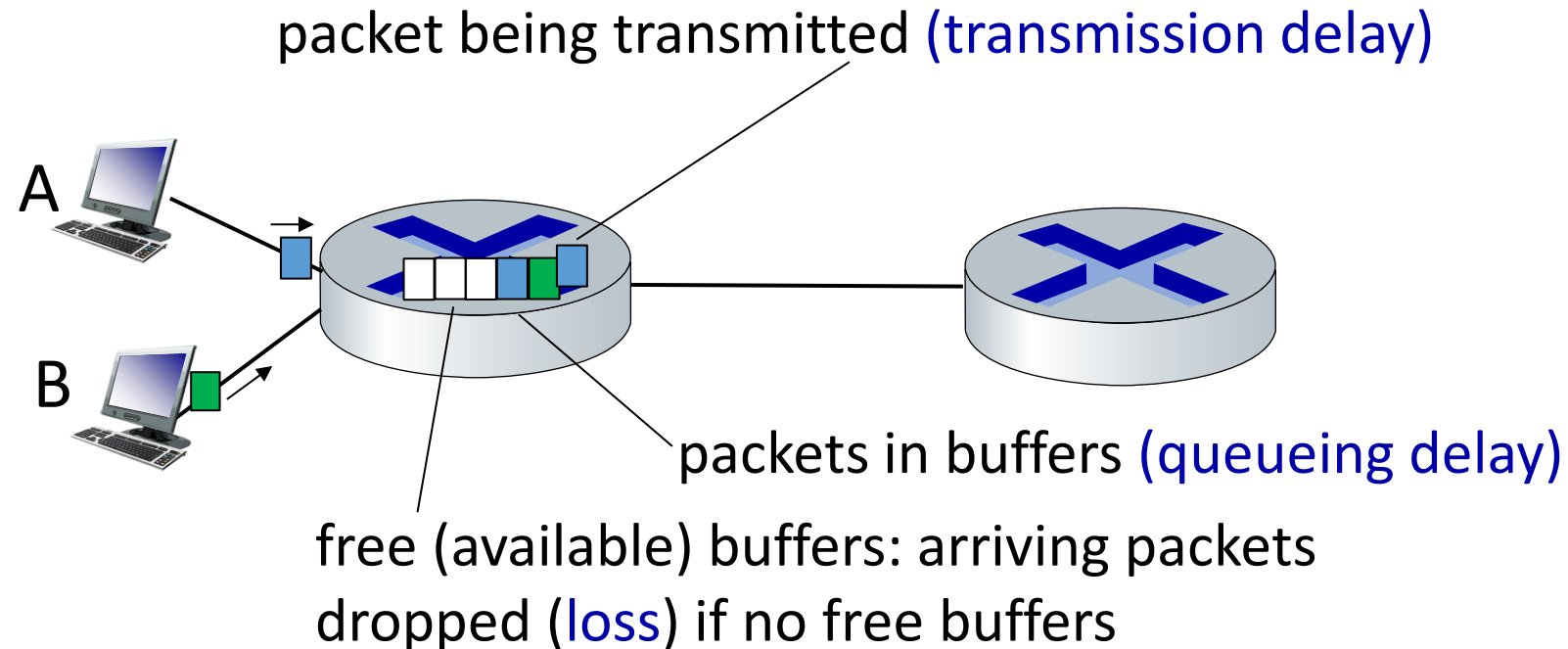


Packet queuing and loss: if arrival rate (in bps) to link exceeds transmission rate (bps) of link for some period of time:

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

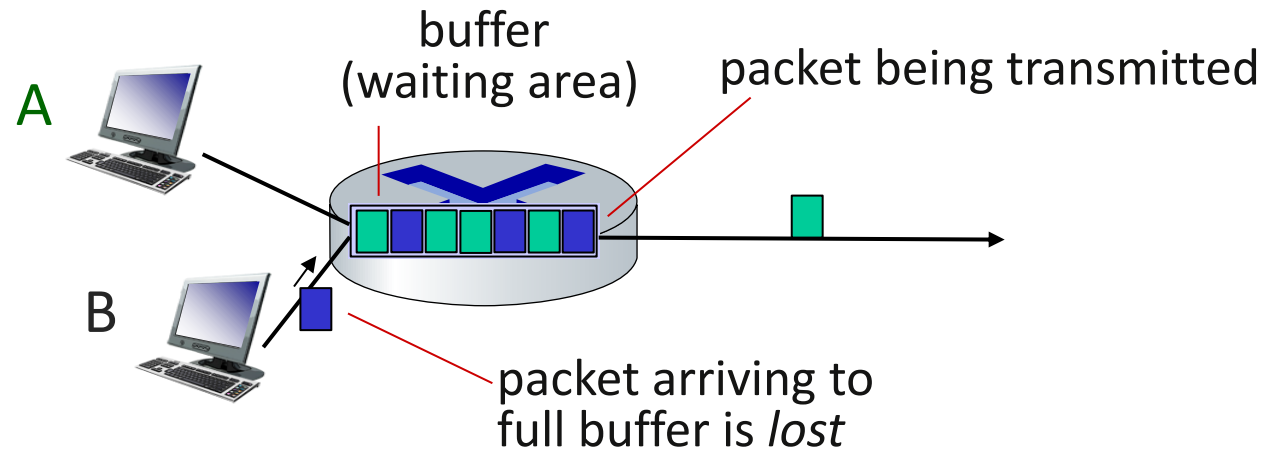
How do packet delay and loss occur?

- 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 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



Example

- **Transmission delay**

For example, in a Fast Ethernet LAN with the transmission rate of 100 million bits per second and a packet of 10,000 bits, it takes $(10,000)/(100,000,000)$ or 100 microseconds for all bits of the packet to be put on the line.

- **Propagation delay**

For example, if the distance of a cable link in a point-to-point WAN is 2000 meters and the propagation speed of the bits in the cable is 2×10^8 meters/second, then the propagation delay is 10 microseconds.

Total delay

Assuming equal delays for the sender, routers, and receiver, the total delay (source-to-destination delay) a packet encounters can be calculated if we know the number of routers, n , in the whole path.

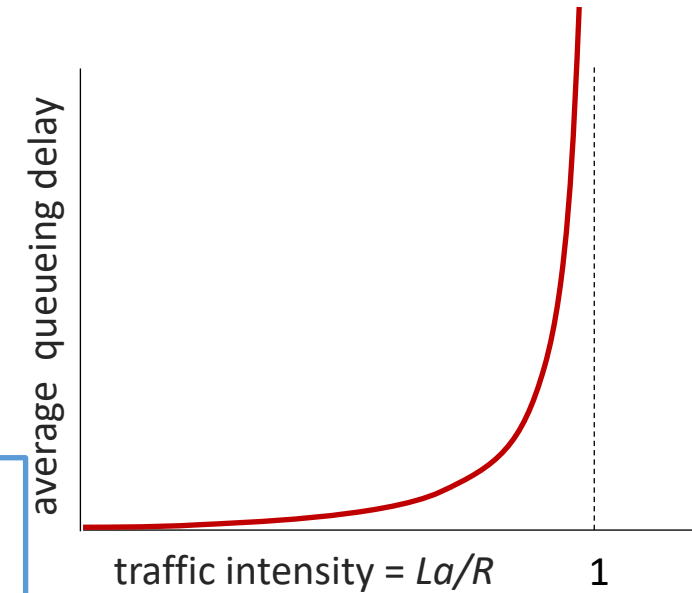
$$\text{Total delay} = (n + 1) (\text{Delay}_{\text{tr}} + \text{Delay}_{\text{pg}} + \text{Delay}_{\text{pr}}) + (n) (\text{Delay}_{\text{qu}})$$

Note that if we have n routers, we have $(n + 1)$ links. Therefore, we have $(n + 1)$ transmission delays related to n routers and the source, $(n + 1)$ propagation delays related to $(n + 1)$ links, $(n + 1)$ processing delays related to n routers and the destination, and only n queuing delays related to n routers.

Packet queueing delay as a function of load

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

$$\frac{L \cdot a}{R} : \frac{\text{arrival rate of bits}}{\text{service rate of bits}} \quad \text{“traffic intensity”}$$



○ (Low Traffic Intensity)

$La/R \sim 0$: avg. queueing delay small

Interpretation: The arrival rate λ is much smaller than the service rate R . The system is underutilized.

Effect: because there is little congestion, packets or tasks are serviced immediately without waiting in the queue.

Example: If $\lambda=1$ packet/sec and $R=10$ packets/sec, then $La/R=0.1$. The system is lightly loaded, and delays are minimal.

○ (High Traffic Intensity)

$La/R \rightarrow 1$: avg. queueing delay large

Interpretation: The arrival rate λ is close to the service rate R . The system is heavily utilized.

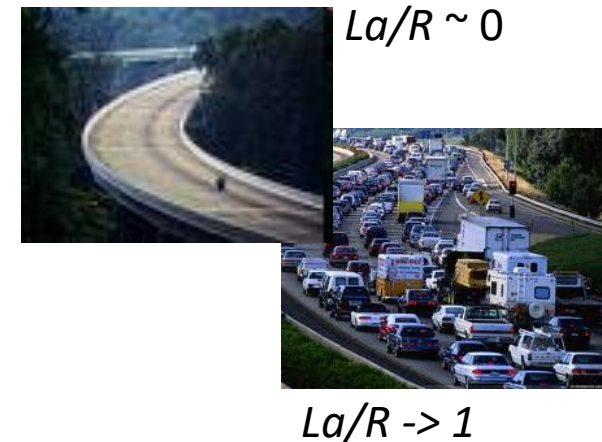
Effect: Packets or tasks spend significant time waiting in the queue before being serviced.

Example: If $\lambda=9$ packets/sec and $R=10$ packets/sec, then $La/R=0.9$. The system is highly congested, and delays grow significantly.

○ (Overloaded System)

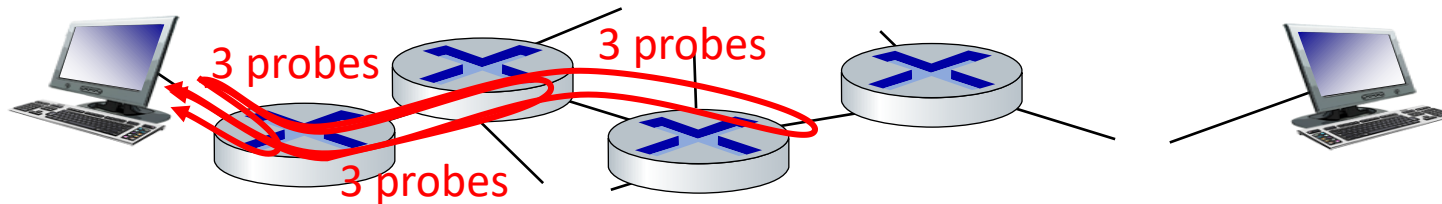
Interpretation: The arrival rate λ exceeds the service rate R . $La/R > 1$: more “work” arriving is more than can be serviced - average delay infinite!
The system is overloaded and cannot keep up with the incoming workload.

Example: If $\lambda=12$ packets/sec and $R=10$ packets/sec, then $La/R=1.2$. The system is overwhelmed, and delays grow indefinitely.



“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

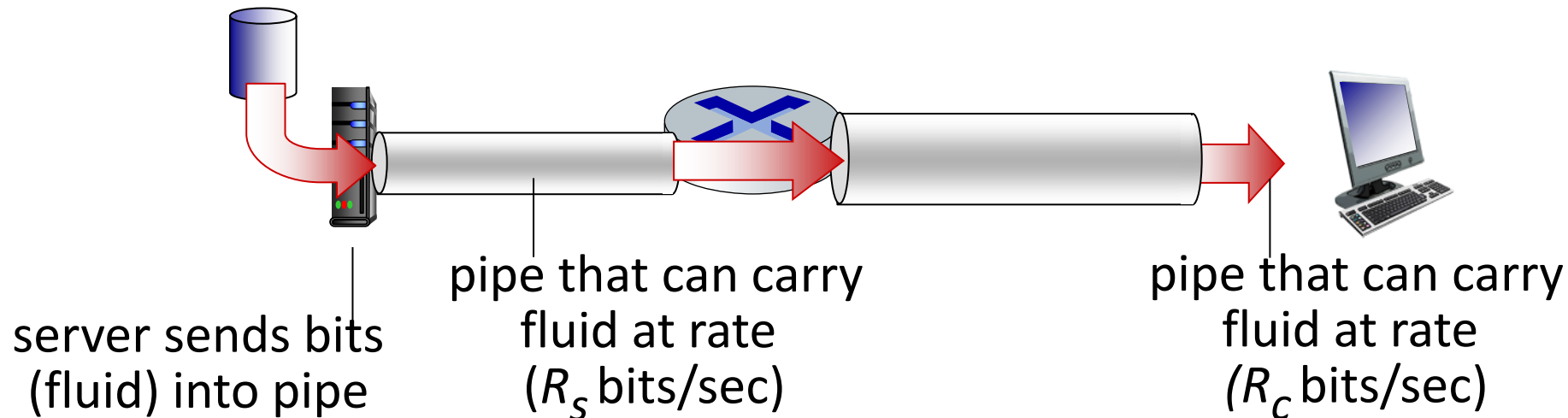


Throughput

- **Throughput:** rate (bits/time unit) at which bits are being sent from sender to receiver
 - **Instantaneous:** rate at given point in time
 - **Average:** rate over longer period of time
 - **Bottleneck Principle:** Throughput is determined by the slowest link in the path.
 - **Bottleneck Formula:** $\text{Throughput} = \min(R_s, R_c)$
 - R_s : Sender's link capacity.
 - R_c : Receiver's link capacity.

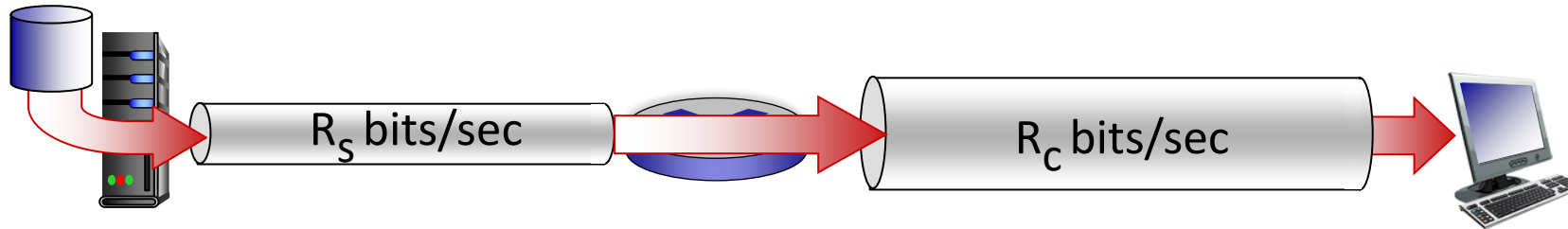
•Examples:

- Sender (R_s) = 10 Mbps, Receiver (R_c) = 5 Mbps \rightarrow Throughput = 5 Mbps.
- Sender (R_s) = 1 Gbps, Receiver (R_c) = 100 Mbps \rightarrow Throughput = 100 Mbps.

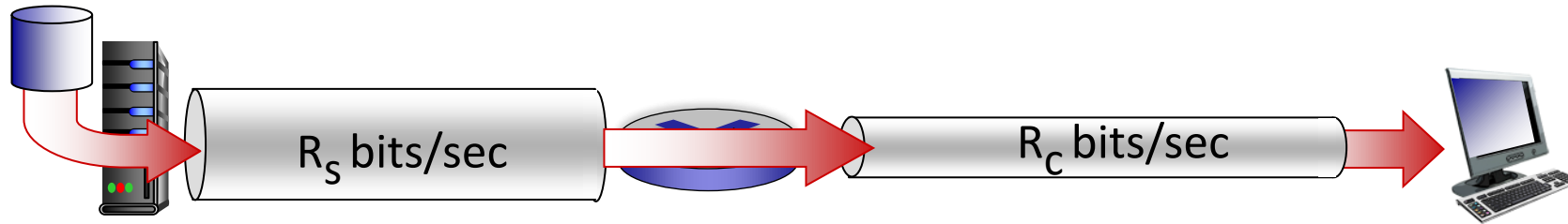


Throughput

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



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

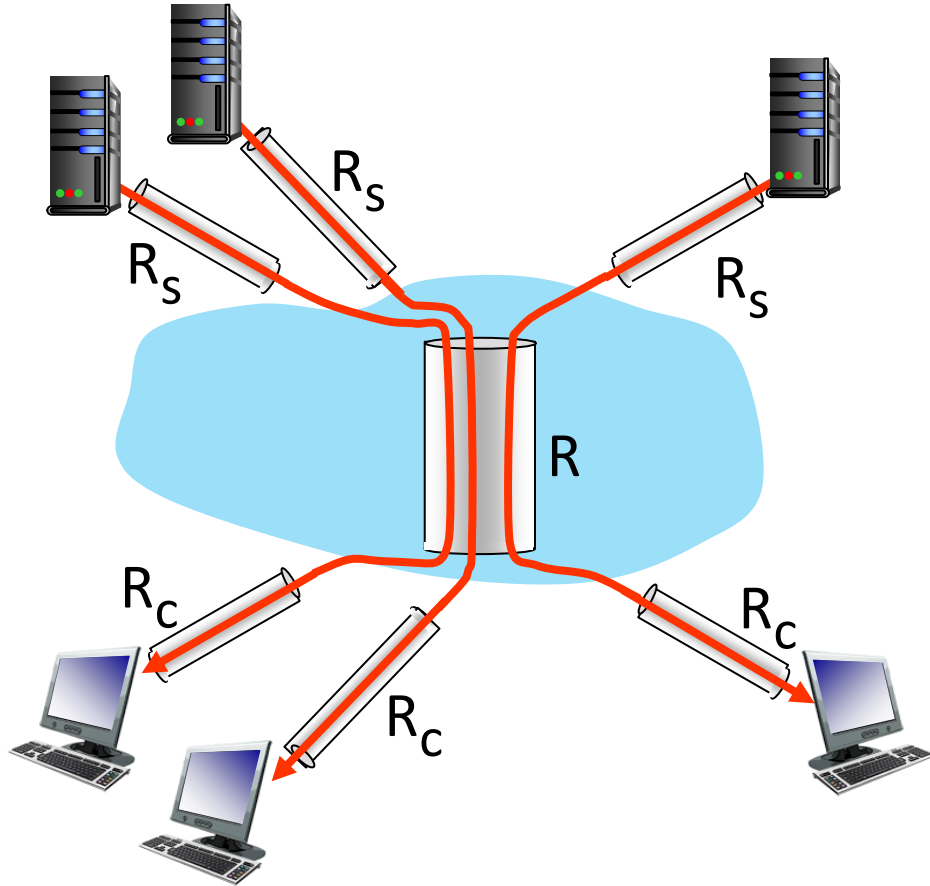


bottleneck link

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

The average throughput is determined by the bottleneck, the pipe with the smallest diameter.

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