COMP 3721
Introduction to Data Communications

10a - Week 10 - Part 1

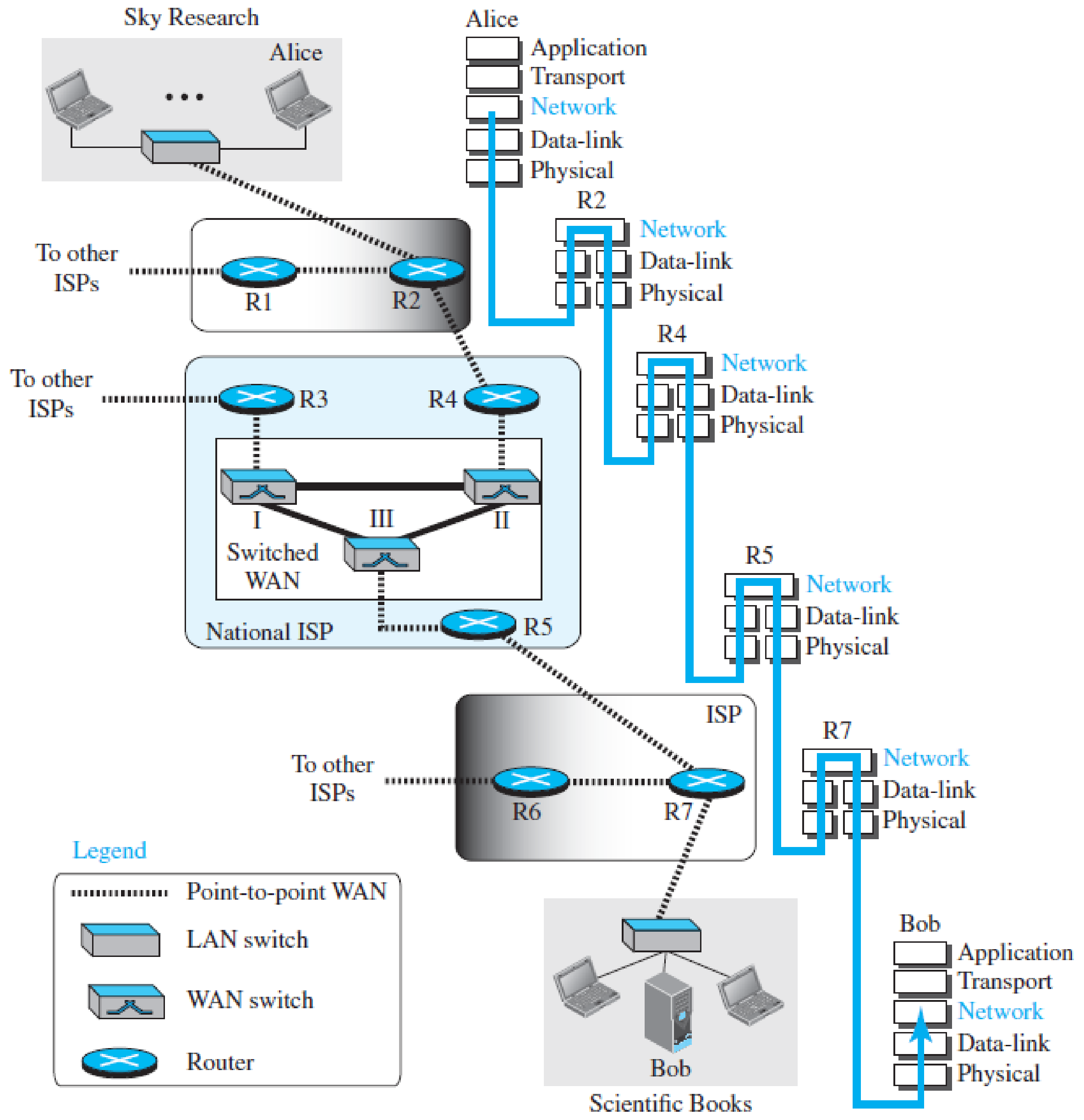
Learning Outcomes

- By the end of this lecture, you will be able to
  - Describe what is packet switching.
  - Explain what are the network-layer services.
  - Describe how a router works.
  - Explain different performance metrics at network layer.

Introduction

- The network layer in the TCP/IP protocol suite is responsible for the **host-to-host** delivery of **datagrams**.
- It provides **services to the transport layer** and receives **services from the data-link layer**.

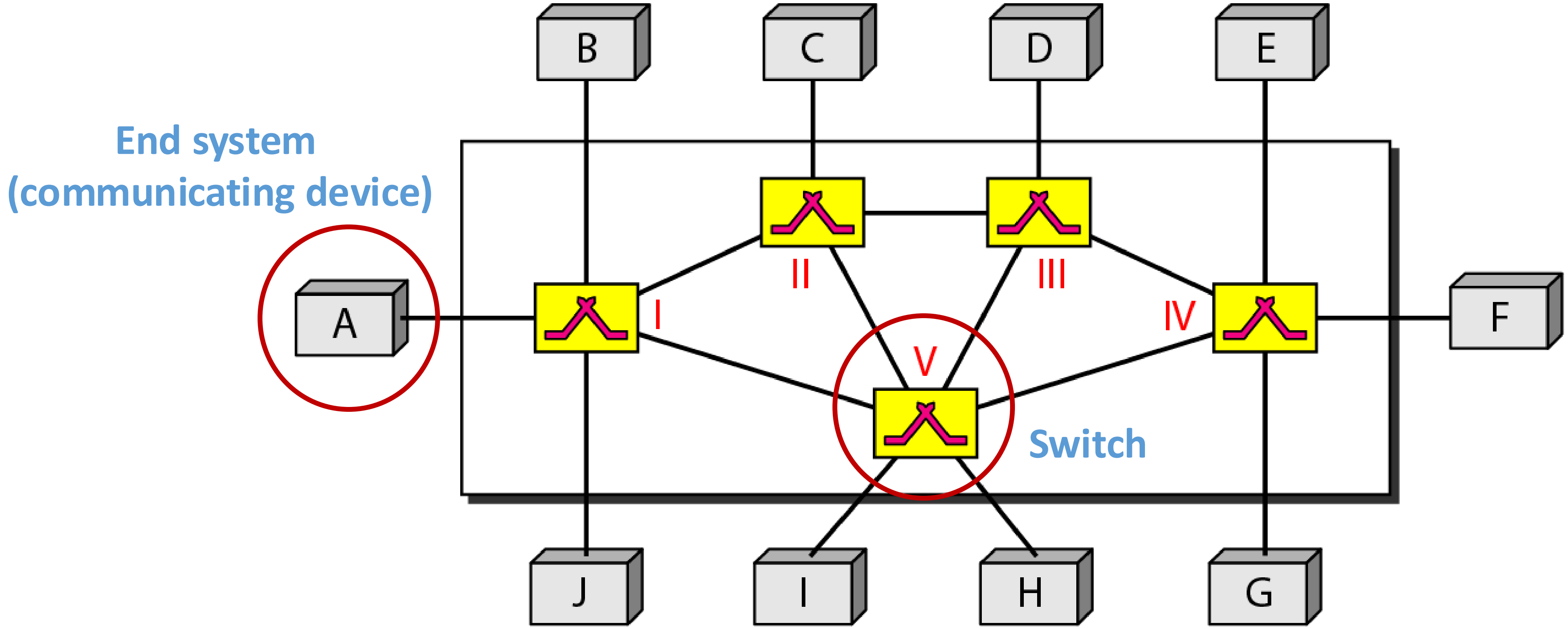
Host-to-Host Communication at the Network Layer



Introduction

- How to connect multiple devices in a network to make one-to-one communication possible?
- Possible solutions
  - **Point-to-point** connection (**mesh** or **star** topology)
    - **Impractical** and wasteful when applied to **very large networks**
      - Not cost-efficient → too much infrastructure
      - Majority of the links would be idle most of the time
  - **Multipoint** connection (**bus** topology)
    - Ruled out → the distances between devices and the total number of devices increase beyond the capacities of the media and equipment
- Any better solution?

Switching



Note that “switch” is a general term here. Switches are devices capable of creating temporary connections between two or more devices linked to the switch (i.e., switches are used for routing).

Packet-Switched Networks

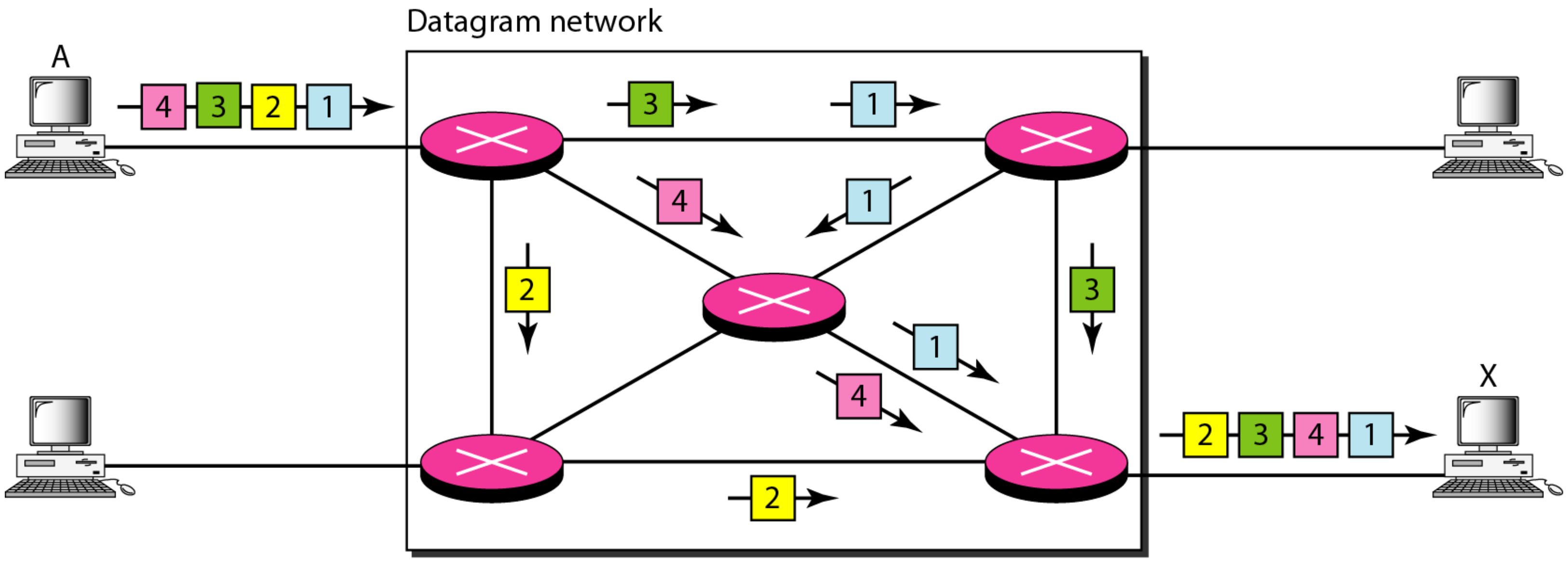
- A message that is sent from an end system to another is divided into **packets**.
- The **size of the packet** is determined by the network and the governing protocol.
- **No resource reservation** (resources are allocated **on demand**).
  - No reserved bandwidth on the links.
  - No scheduled processing time for each packet.
- When a switch receives a packet, the **packet** must **wait** if there are **other packets** being **processed**.
- One **subcategory**:
  - **Datagram networks**, e.g., the Internet → Internet Protocol (IP)

Datagram Networks

- **Packets** are called **datagrams** and **switches** are called **routers**.
- Datagram switching is normally done at the **network layer**.
- Packets of the same message may travel **different paths** to reach their destination.
  - **Non-uniform delay** for packets of a message.
- Possibility of **out-of-order**/**lost**/**dropped packets** → handled by upper-layer protocols (usually transport-layer).
- No setup or teardown phases.
- The **destination addresses** and the corresponding forwarding **output ports** are recorded in the **routing** (**forwarding**) **tables** of routers.

Datagram Networks

- The network layer provides a **connectionless service**:
  - The router does not keep information about the connection state.
  - Each packet is an **independent entity**.
  - **No relationship** between packets belonging to the same message



Services of Network Layer

- **Packetizing**
- **Routing and forwarding**
- Error control
  - Checksum field to the datagram to control any corruption in the header, but not in the whole datagram.
- Flow control
  - Usually implemented in transport layer, not network layer.
- Congestion control
  - Usually implemented in transport layer, not network layer.

Packetizing

- Definition:
  - **Encapsulating** the payload (data received from upper layer) in a network-layer packet at the **source** and **decapsulating** the payload from the network-layer packet at the **destination**.
- In other words, one duty of the network layer is to **carry a payload** from the **source** to the **destination** without changing it or using it.

Packetizing

- **Source host**
  - Receiving payload from an upper layer protocol.
  - Adding a header (including **src** and **dst** IP addresses).
  - Delivering the packet to the data-link layer.
  - May fragment packets.

Packetizing

- **Source host**
  - Receiving payload from an upper layer protocol.
  - Adding a header (including **src** and **dst** IP addresses).
  - Delivering the packet to the data-link layer.
  - May fragment packets.
- **Routers on the path**
  - Not allowed to change the **src** or **dst** network addresses (i.e., IP addresses).
  - May fragment packets.

Packetizing

- **Source host**
  - Receiving payload from an upper layer protocol.
  - Adding a header (including **src** and **dst** IP addresses).
  - Delivering the packet to the data-link layer.
  - May fragment packets.
- **Routers on the path**
  - Not allowed to change the **src** or **dst** network addresses (i.e., IP addresses).
  - May fragment packets.
- **Destination host**
  - Receiving the network-layer packet from its data-link layer.
  - Decapsulating the packet.
  - Delivering the payload to the corresponding upper-layer protocol.
  - If packets are fragmented, reassemble them before delivery to upper-layer.

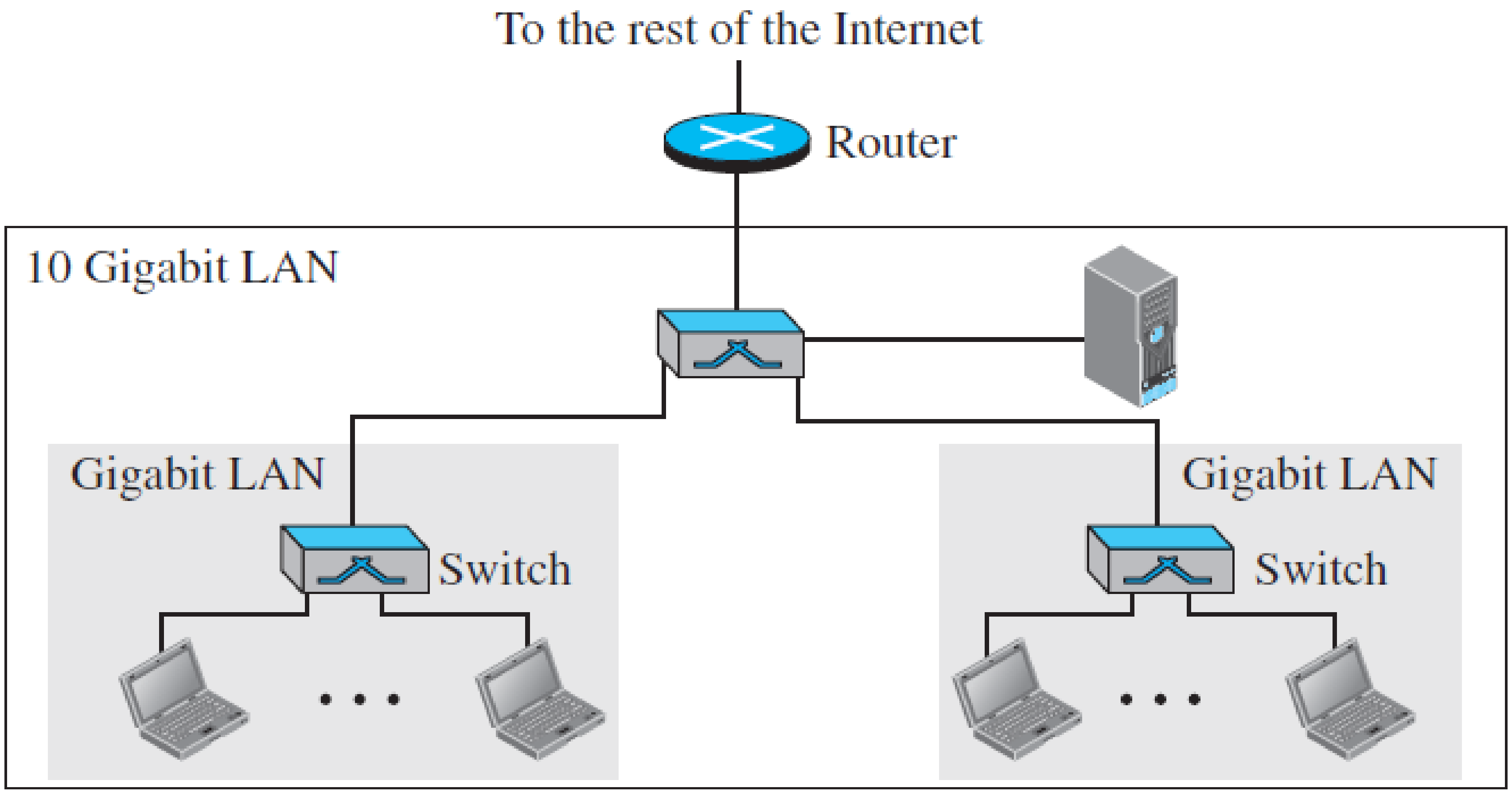
Routing

- Routing the packets from source to destination
  - Choosing the **best route** among all possible routes by using routing protocols
- **Routers** implement the physical layer, data-link layer, and network layer (layers 1, 2, and 3) of the TCP/IP protocol stack.
- A **router connects networks**; so, it is an inter-networking device.

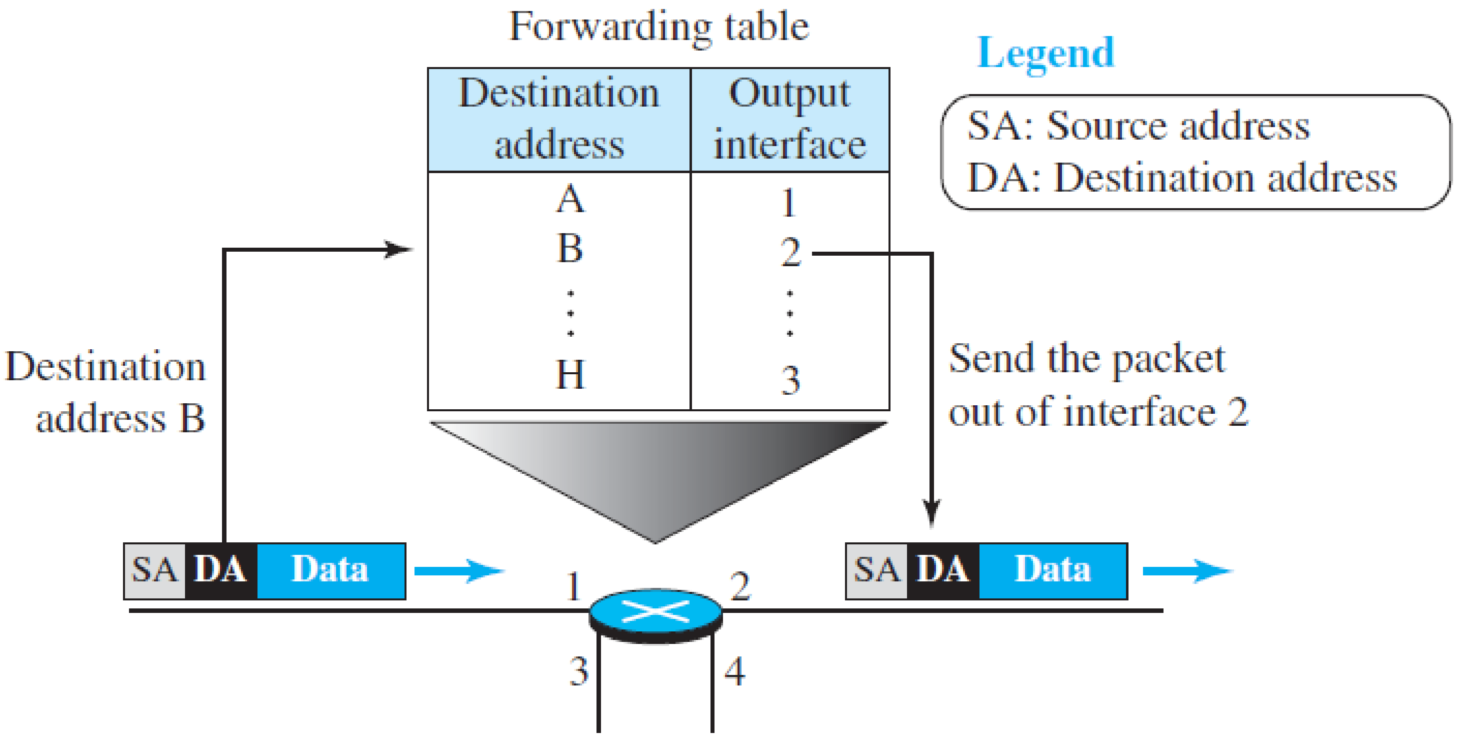
Routing

- A router has a **link-layer address** (i.e., **MAC address**) as well as an **IP address** for **each** of its **interfaces**.
  - **Reason**: each link may use its own data-link or physical protocol.
- The router **modifies** the **source** and **destination link-layer** (MAC) **addresses** of the packet before forwarding the packet (because MAC addresses have only **local jurisdictions**).
- A router ONLY works on packets whose **link-layer destination address** matches the **link-layer address of the interface** at which the **packet arrives**.

Router as a Network-Layer Device



Routing (Forwarding) Table – High Level Description



The forwarding decision is based on the destination IP address of the packet.

Routing (Forwarding) Table

- The **routing tables** are **dynamic** and are **updated periodically**.
- The **destination IP address** is fixed during the entire journey of the packet.
- Why is the **source IP address** is needed?
  - To send error messages to the source if the packet is discarded.

Error Control

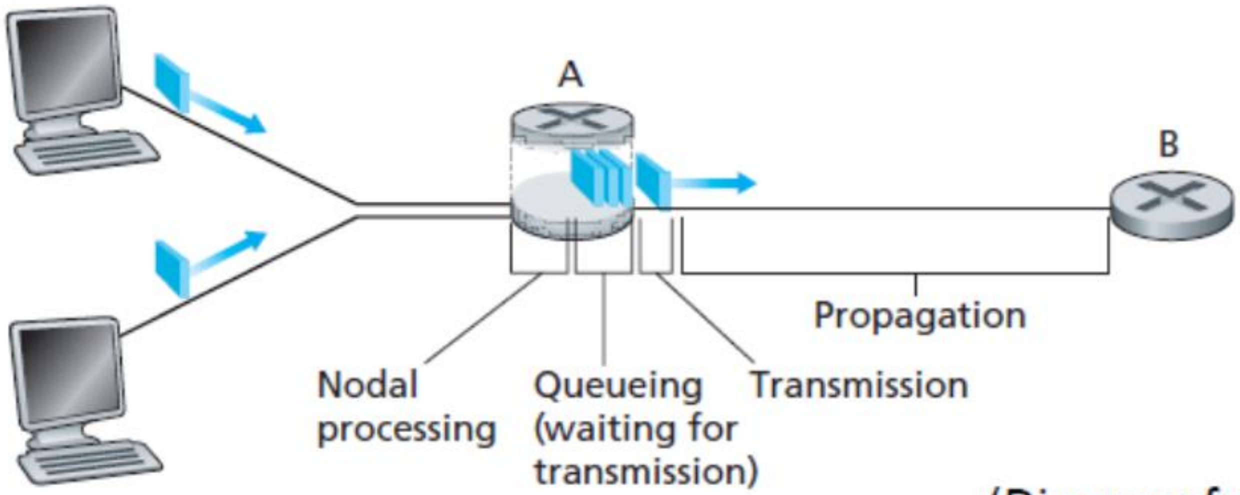
- **Checksum** field in the **header of datagram** to control any **corruption in the header**, not in the whole datagram.
- No error control for the whole datagram, why?
  - One reason: the packet in the network layer **may be fragmented** at the router, which makes error checking inefficient.

Performance

- Three metrics to measure the performance of the network layer
    1. **Delay** (also called **latency**)
    2. **Throughput**
    3. **Packet loss**

Delay (Latency)

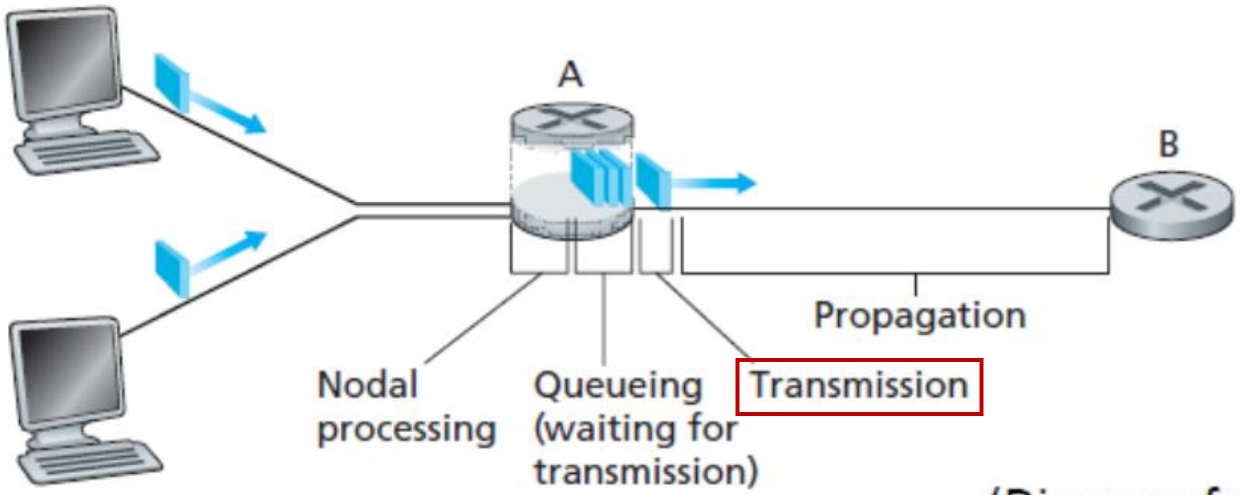
- A measure of **how long it takes for an entire message to completely arrive at the destination** from the time the first bit is sent out from the source.



(Diagram from Kurose & Ross)

Transmission Delay

- **Transmission delay** (**transmission time**)
  - Time taken to push all the bits of a message/packet onto the link.
  - Depends on the **size of the message** and the bandwidth (**transmission rate**) of the channel.

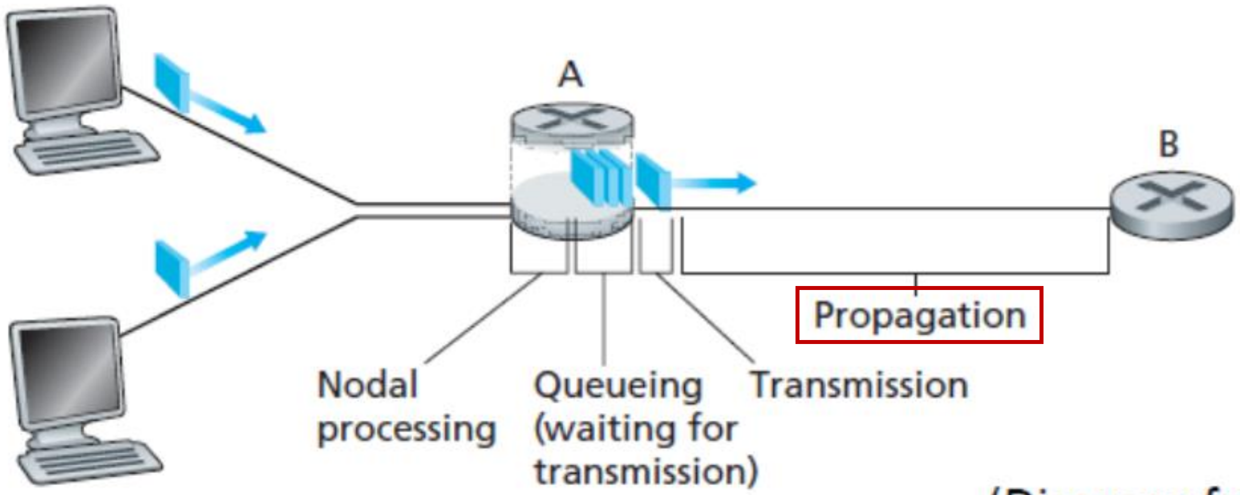


(Diagram from Kurose & Ross)

Delay_{tr} = Packet Length / Transmission Rate

Propagation Delay

- **Propagation delay** (**propagation time**)
  - The time required for a bit to travel from the source to the destination.
  - Depends on the **medium**.
  - In a vacuum, with a speed of 3×10^8 m/s (lower in air and cable).



(Diagram from Kurose & Ross)

Delay_{pg} = Distance / Propagation Speed

Propagation and Transmission Time (Example)

- What are the **propagation time** and the **transmission time** for a 3.5-kB (kilobyte) message (an email) if the bandwidth of the network is 1 Gbps? Assume that the distance between the sender and the receiver is 15,000 km and that light travels at 2.4×10^8 m/s.

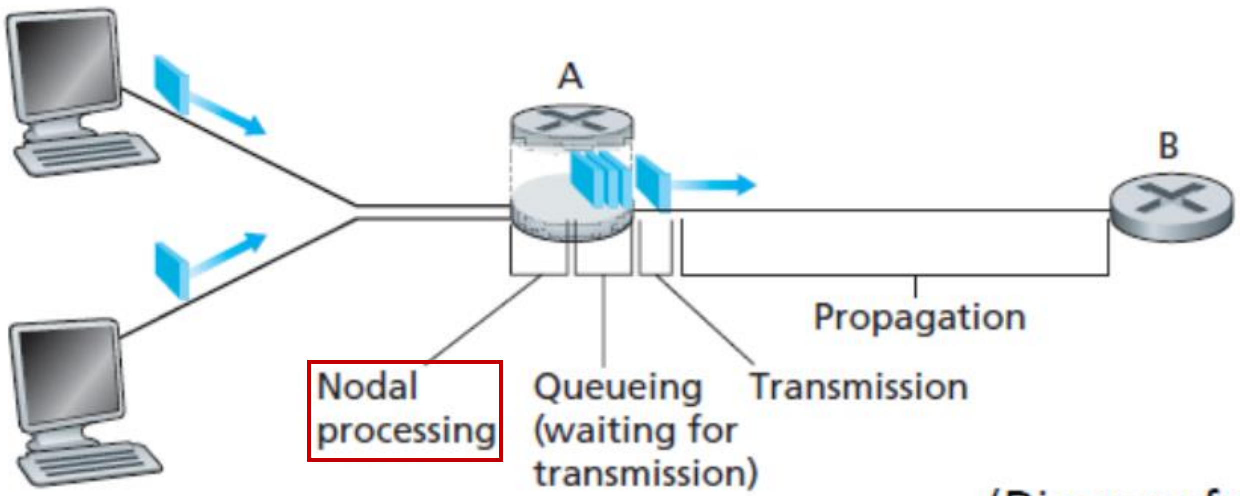
- **Answer**:

- Propagation time = $\frac{(15000 \times 1000)}{2.4 \times 10^8} = 0.0625 = \mathbf{62.5\ ms} = \mathrm{Delay_{pg}}$

- Transmission time = $\frac{(3.5 \times 1000 \times 8)}{1 \times 10^9} = 0.000028 = \mathbf{28\ \mu s} = \mathrm{Delay_{tr}}$

Processing Delay

- Processing delay (processing time)
  - The time required to process a packet in a router or destination host.
  - A router or a destination host receives a packet from its input port, removes the header, perform an error detection procedure, and delivers the packet to the output port (in the case of a router) or deliver the packet to the upper-layer protocol (in the case of the destination host).
  - Can be different for each packet, but normally is calculated as an average.



(Diagram from Kurose & Ross)

Queuing/Waiting Delay

- **Queuing**/**Waiting delay**
  - The time needed for each intermediate or end device to hold the message before it can be processed (mostly referred to the **waiting time** of a **packet** in **input/output queues** of a **router**).
  - Changes with the load incurred on the network, e.g., heavy traffic on the network increases the queuing time → **not fixed**
  - A router has an **input queue connected to each of its input ports** to store packets waiting to be processed.
  - The router also has an **output queue connected to each of its output ports** to store packets waiting to be transmitted.

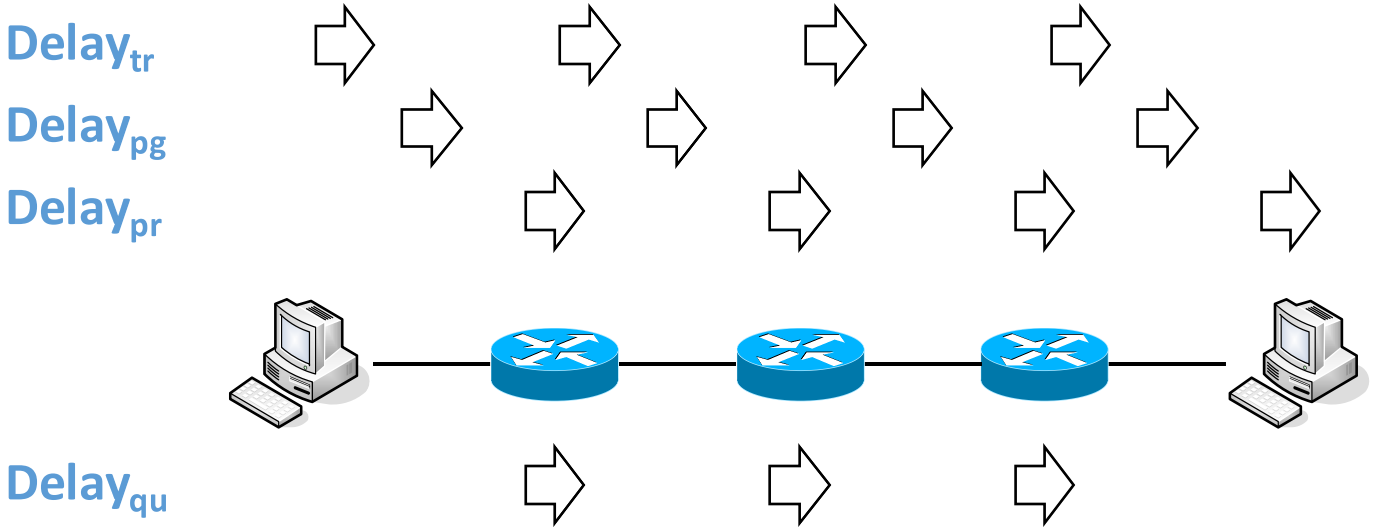
Total Delay (Latency)

- Total delay (source-to-destination delay) for a packet
  - Assume equal delays for the sender, routers, and receiver.
  - *n* is the number of routers → *n* + 1 links
    - (*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**.
    - *n* **queuing delays** related to *n* **routers**.

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

Total Delay (Latency)

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



Throughput

- A measure of how fast we can **actually** send data through a network.
- Different from bandwidth in bits per second.
  - The bandwidth is a **potential** measurement of a link.
  - A link may have a bandwidth of B bps, but we can only send T bps through this link ($T \leq B$).
  - **Maximum throughput** is essentially synonymous to **digital bandwidth capacity**.

Throughput – Example

- **Example**: A network with bandwidth of 5 Mbps can pass only an average of 15,000 packets per minute with each packet carrying an average of 10,000 bits. What is the throughput of this network?

Throughput – Example

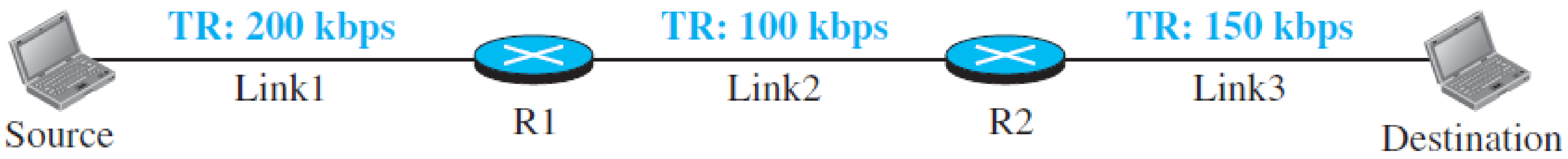
- **Example**: A network with bandwidth of 5 Mbps can pass only an average of 15,000 packets per minute with each packet carrying an average of 10,000 bits. What is the throughput of this network?

- **Answer**:

$$10000\,\frac{\mathrm{b}}{\mathrm{p}} \times 15000\,\frac{\mathrm{p}}{\mathrm{min}} \times \frac{1\ \mathrm{min}}{60\ \mathrm{s}} = 2500000\ \mathrm{bps} = \mathbf{2.5\ Mbps}$$

Throughput

- How to determine the throughput of the whole path from source to destination?
  - It is indicated by the **bottleneck link**, which is the link with the minimum data rate (transmission rate).
  - For a path with *n* links: **Throughput = minimum $\{TR_1, TR_2, \ldots TR_n\}$**



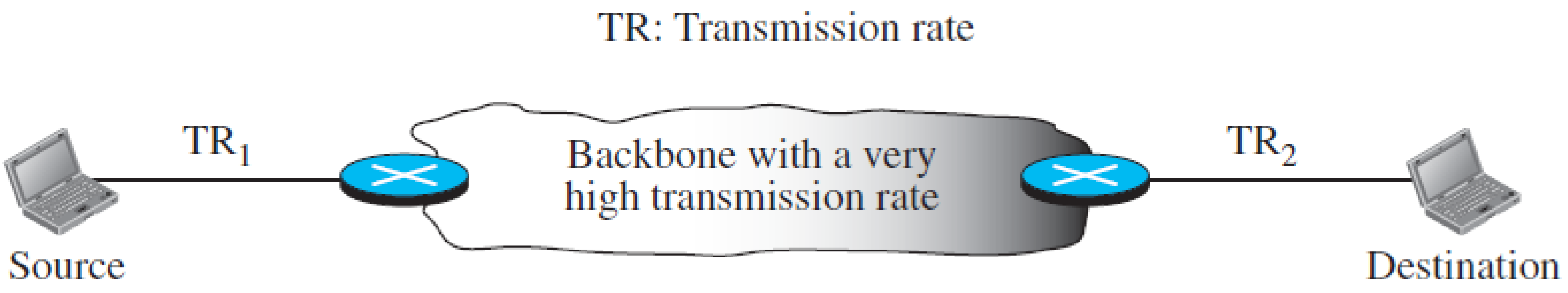
a. A path through three links

TR: Transmission rate

Throughput = 100 kbps

Throughput

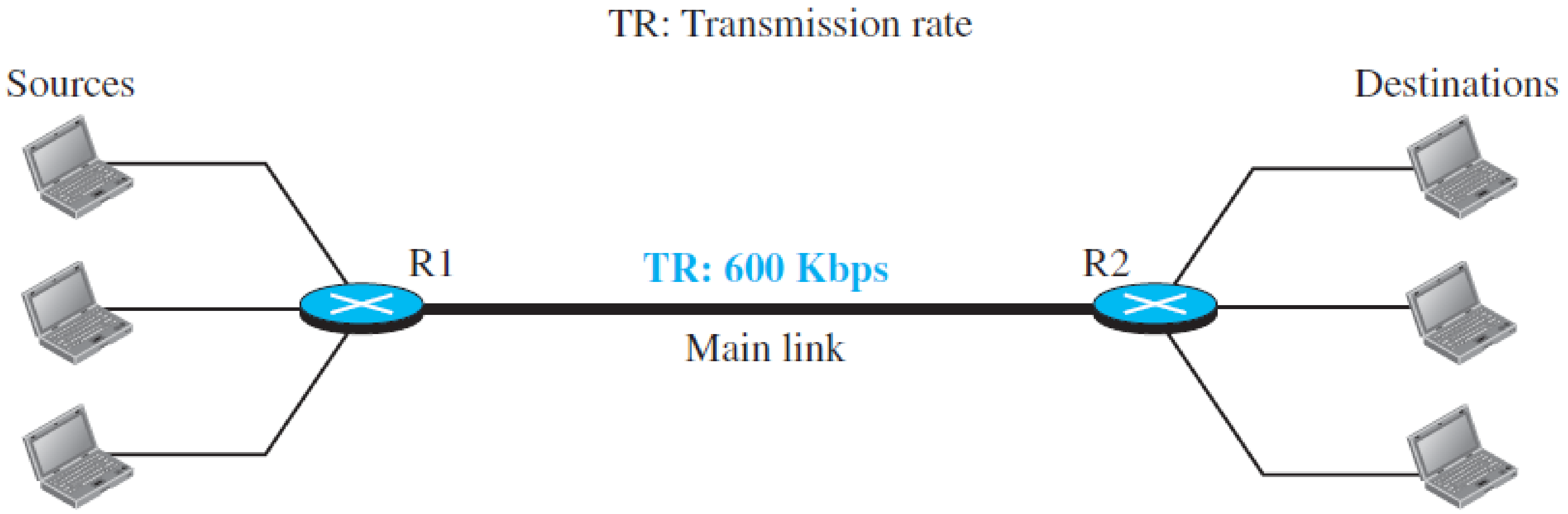
- In the Internet, the data normally passes through two **access networks** and the **Internet backbone**.



Throughput is usually the minimum of TR_1 and TR_2.

Throughput in a Shared Link

- What if the transmission rate of a link between two routers is shared between multiple flows?



The transmission rate of the main link in the calculation of the throughput is only 200 kbps because the link is shared between three flows.

Packet Loss

- The number of packets lost can severely affect performance.
- The size of the input buffer (queue) of the router is limited.
- If the **buffer** is **full**, the next packet needs to be dropped.
- Impact of the packet loss on the Internet network layer:
  - The packet needs to be re-sent, which in turn may create **overflow** and cause **more packet loss**.

Jitter

- The **variations** in **delay**.
  - It is a problem if different packets of data encounter different delays and the application using the data at the receiver site is time-sensitive (e.g., audio and video data).
  - Example: if the delay for the first packet is 20 ms, for the second is 45 ms, and for the third is 40 ms, then the real-time application that uses the packets endures jitter.

Summary

- Datagram networks provide **connectionless service**.
- Internet is a **packet-switched network**.
- Main services of network layer: **packetizing**, **routing** and **forwarding**.
- Total delay has four components: **transmission delay**, **propagation delay**, **processing delay**, and **queuing delay**.
- The end-to-end **throughput** is limited by the **slowest link** on the path.

References

[1] Behrouz A.Forouzan, Data Communications & Networking with TCP/IP Protocol Suite, 6th Ed, 2022, McGraw-Hill companies.

[2] J.F. Kurose, K.W. Ross, Computer Networking: A Top-Down Approach, 7th Ed, 2017, Pearson Education, Inc.

Reading

- Chapter 7 of the textbook, sections 7.1 – 7.3.
- Chapter 7 of the textbook, section 7.8 (Practice Test)