

# **COMP 3721数据通信导论**

**10a - 第10周 - 第1部分**

# 学习成果

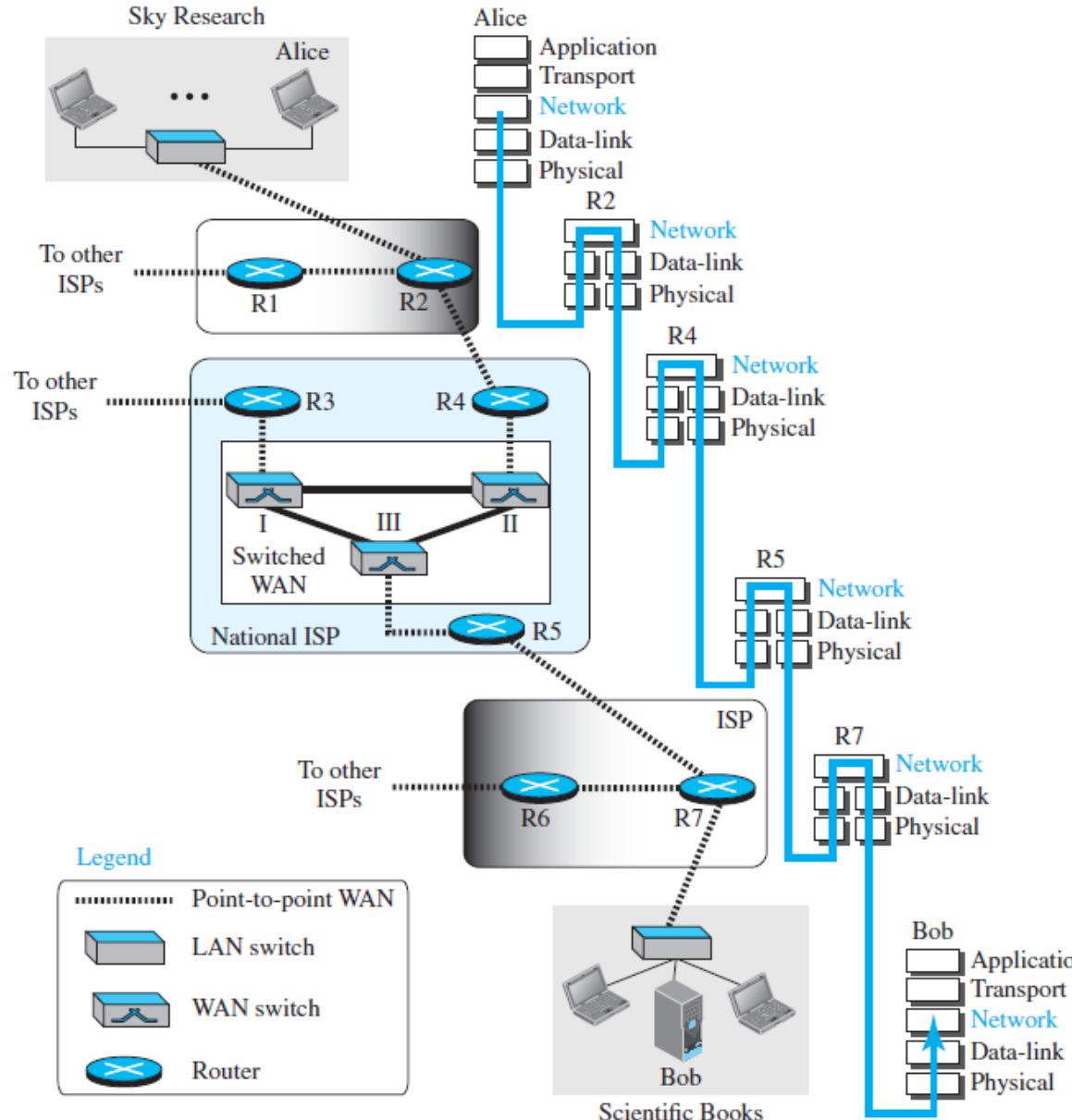
- 到本讲结束时，你将能够
  - 描述什么是分组交换。
  - 解释网络层提供哪些服务。
  - 描述路由器的工作原理。
  - 解释网络层的不同性能度量。

# 简介

- 在 TCP/IP 协议**主机到主机** 传递**数据报** 的网络层负责。
- 它为**传输层提供服务**， 并从**数据链路层接收服务**。

# 主机到主机的网络通信

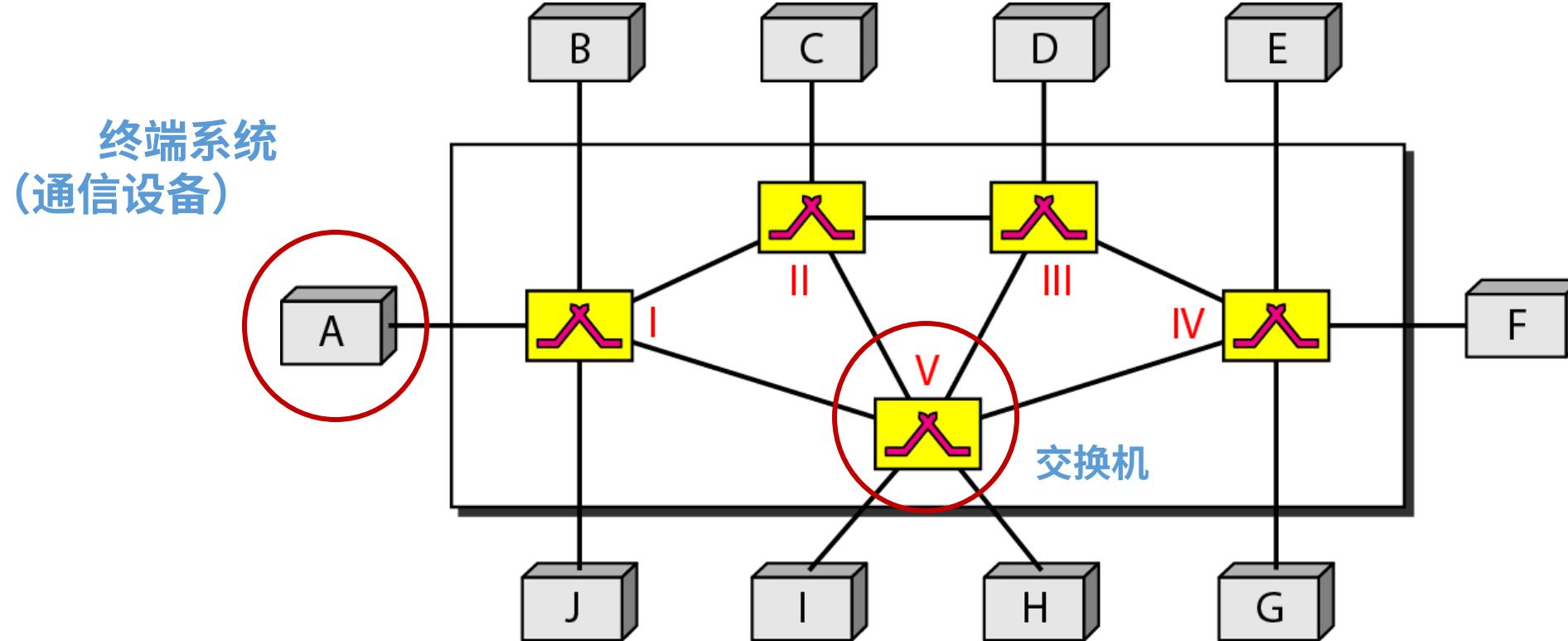
## 层



# 介绍

- 如何将多个设备连接到网络中以实现一对一通信?
- 可能的解决方案
  - 点对点 连接 (网状 或 星形 拓扑)
    - 不切实际 且在应用于 非常大的网络时浪费资源
      - 不具成本效益 → 需要太多基础设施
      - 大多数链路大部分时间将处于空闲状态
    - 多点 连接 (总线 拓扑)
      - 排除了 → 当设备之间的距离和设备总数超过介质和设备的容量时
  - 有什么更好的解决方案吗?

# 交换



注意此处“交换机”是一个通用术语。交换机是能够在连接到交换机的两个或更多设备之间创建临时连接的设备（即，交换机用于转发/路由）。

# 分组交换网络

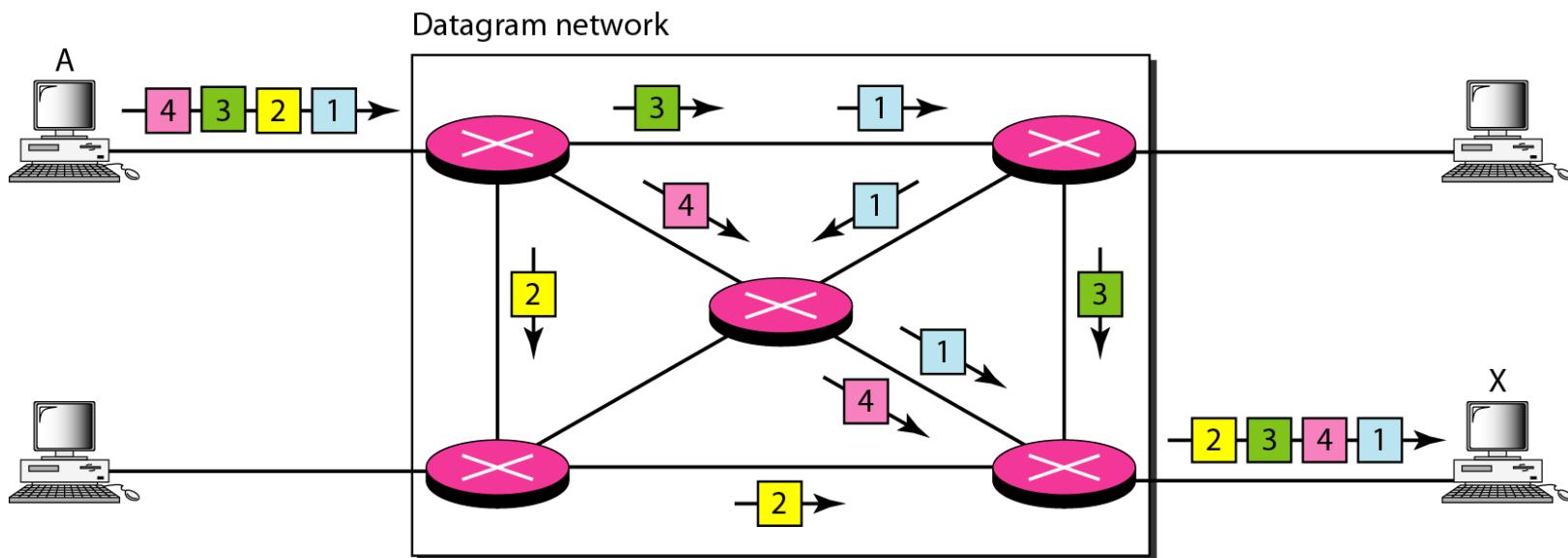
- 从一个端系统发送到另一个端系统的消息被划分为**数据包**。
- 数据包的**大小**由网络和所使用的协议决定。
- **无资源预留**（资源是在需要时分配）。
  - 链路上没有保留的带宽。
  - 没有为每个数据包安排的处理时间。
- 当交换机接收一个数据包时，**该数据包**必须**等待**，如果有**其他数据包**正在**处理**。
- 一个**子类别**:
  - **数据报网络**，例如互联网 → Internet 协议 (IP)

# 数据报网络

- 数据包 被称为 **数据报**，交换机 被称为 **路由器**。
- 数据报交换通常在 **网络层** 进行。
- 同一消息的分组可能经过 **不同路径** 到达目的地。
  - 消息分组的 非均匀延迟
- 可能出现由上层协议（通常是传输层）处理的 **乱序/丢失/丢弃的数据包** →。
- 没有设置或拆卸阶段。
- 目标**地址**及相应的转发**输出端口**记录在路由器的**路由 (转发) 表中**。

# 数据报网络

- 网络层提供**无连接服务**：
  - 路由器不保留关于连接状态的信息。
  - 每个分组都是一个**独立实体**。
  - **没有关系** 对于属于同一消息的分组之间



# 网络层的服务

- 分组化
- 路由与转发
- 差错控制
  - 在数据报中添加校验和字段以控制报头中的任何损坏，但不覆盖整个数据报。
- 流量控制
  - 通常在传输层实现，而不是在网络层。
- 拥塞控制
  - 通常在传输层实现，而不是在网络层。

# 分组化

- 定义：
  - 将 上层接收到的负载（数据）封装 到网络层数据包中，在 源端，并在 目的端 从网络层数据包中解封装。
- 换句话说，网络层的一项职责是 携带负载 从 源端 到 目的端，而不对其进行更改或使用。

# 分组化

- 源主机

- 从上层协议接收有效载荷。
- 添加首部（包括src 和dst IP 地址）。
- 将数据包传递到数据链路层。
- 可能对数据包进行分片。

# 分组化

- 源主机

- 从上层协议接收负载。
- 添加头部（包括 源 和 目的 IP 地址）。
- 将分组交付到数据链路层。
- 可能会对分组进行分片。

- 路径上的路由器

- 不得更改源或目的地网络地址（即 IP 地址）。
- 可能对分组进行分片。

# 分组化

- 源主机

- 从上层协议接收有效载荷。
- 添加首部（包括 源 和 目的 IP 地址）。
- 将分组交付到数据链路层。
- 可能对分组进行分片。

- 路径上的路由器

- 不允许更改 `src` 或 `dst` 网络地址（即 IP 地址）。
- 可能会对分组进行分片。

- 目的主机

- 从其数据链路层接收网络层数据包。
- 对数据包进行去封装。
- 将有效载荷传递到对应的上层协议。
- 如果数据包被分片，在交付到上层之前先重新组装它们。

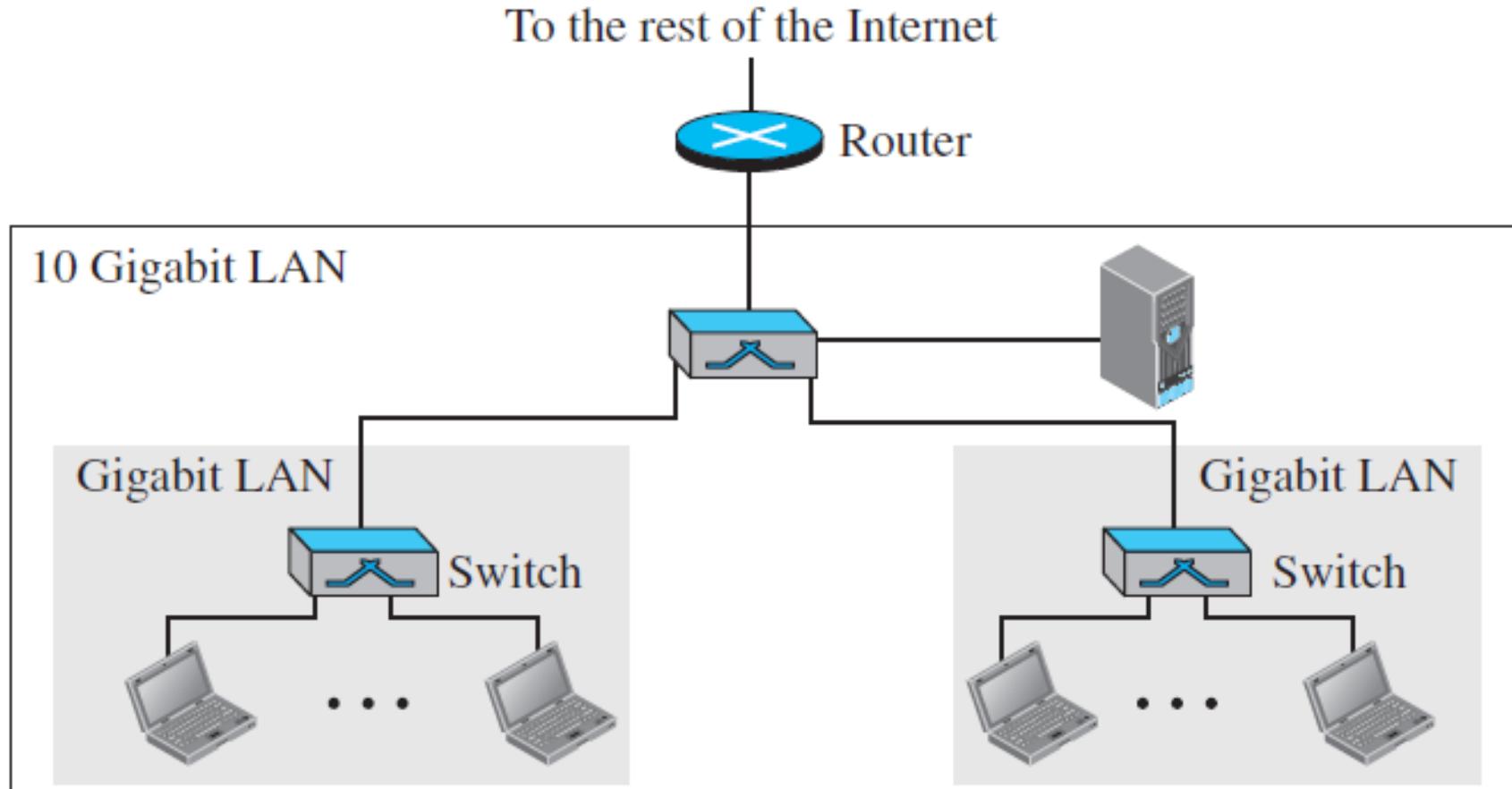
# 路由

- 将数据包从源发送到目的地的路由
  - 使用路由协议在所有可能路径中选择最佳路径
- 路由器实现物理层、数据链路层和网络层（TCP/IP 协议栈的第 1、2 和 3 层）。
- 路由器连接网络；因此，它是一个互连网络设备。

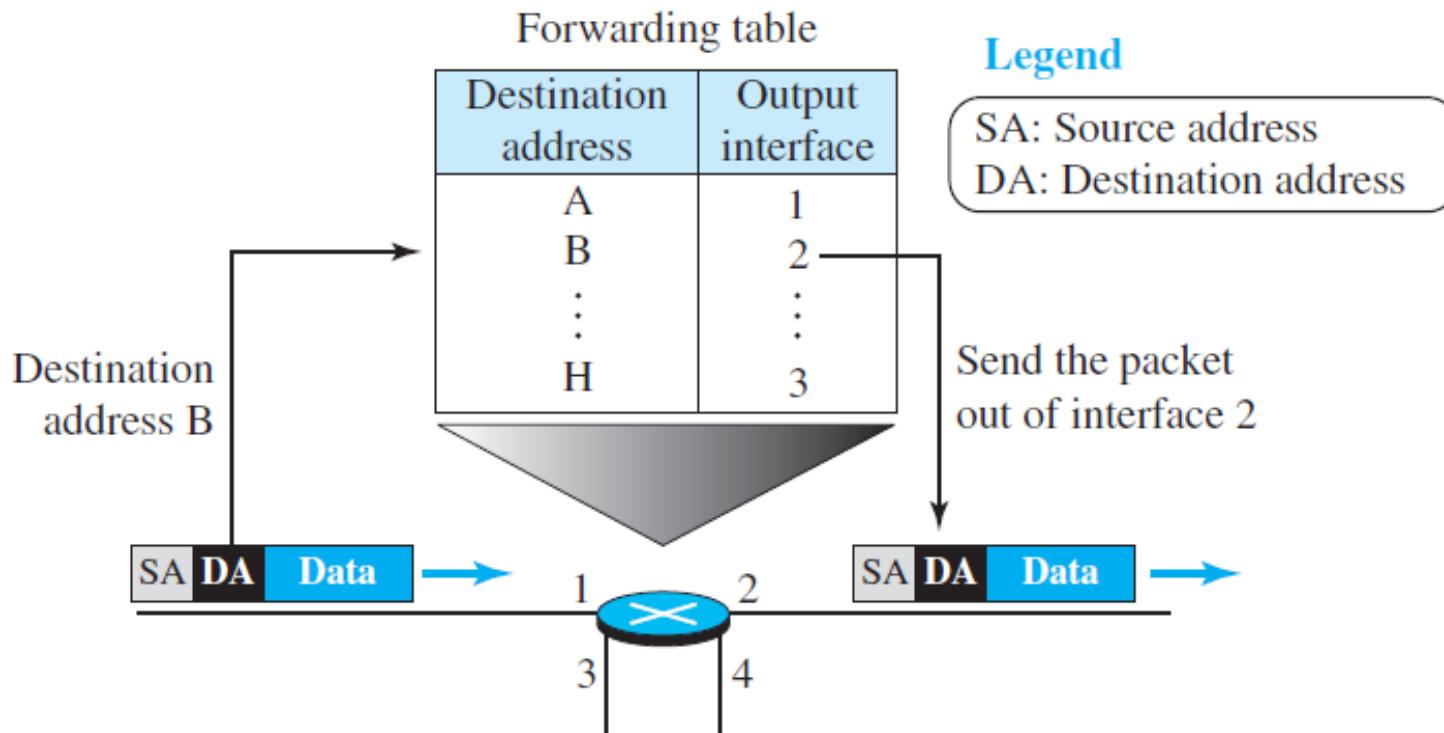
# 路由

- 路由器有一个 **链路层地址** (即 **MAC 地址**) 以及每个 **IP 地址** 对于 **它的** 接口。
  - **原因**: 每条链路可能使用各自的数据链路或物理协议。
  - 路由器在转发数据包之前会**修改** 数据包的 **源** 和 **目的地 链路层 (MAC) 地址** (因为 MAC 地址仅在 **本地范围内有效** ) 。
  - 路由器仅对那些其 **链路层目的地址** 与数据包到达的接口的 **链路层地址相匹配** 的数据包起作用。

# 路由器作为网络层设备



# 路由 (转发) 表 – 高层描述



转发决策基于数据包的目的地 IP 地址。

# 路由（转发）表

- 路由表是动态的，并且会定期更新。
- 在数据包整个传输过程中，目的 IP 地址保持不变。
- 为什么需要源 IP 地址？
  - 以便在数据包被丢弃时向源主机发送错误消息。

# 差错控制

- 校验和 字段位于 数据报的首部，用于控制任何 损坏 在首部中，而不是在整个数据报上。
- 为什么不对整个数据报进行差错控制？
  - 一个原因是：网络层的数据包 可能在路由器处被分片，这会使错误检测变 得低效。

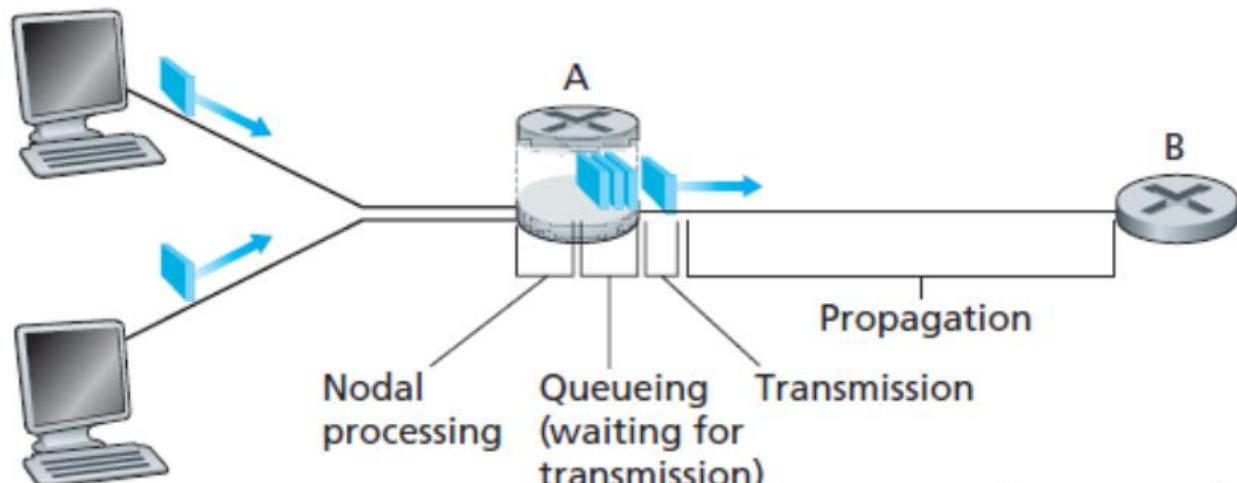
# 性能

- 衡量网络层性能的三项指标

1. 延迟 (也称为时延)
2. 吞吐量
3. 丢包率

# 延迟（时延）

- 一种度量，从源端发送第一比特起，到整条消息完全到达目的地所需时间的长短。

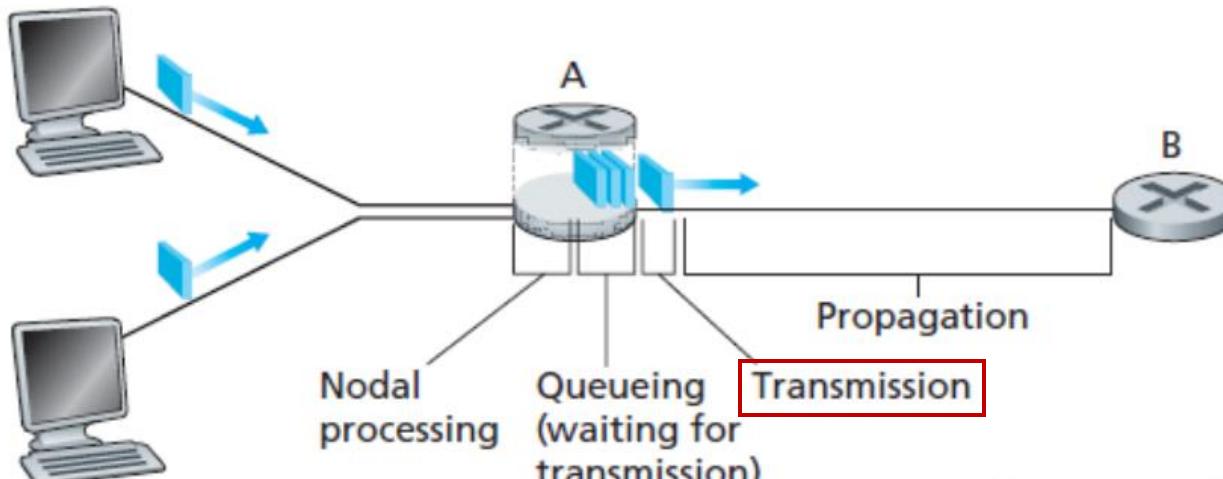


(Diagram from Kurose & Ross)

# 传输延迟

- **传输延迟 (传输时间)**

- 将消息/数据包的所有比特推送到链路上所需的时间。
- 取决于**消息的大小**和信道的带宽 (**传输速率**) 。



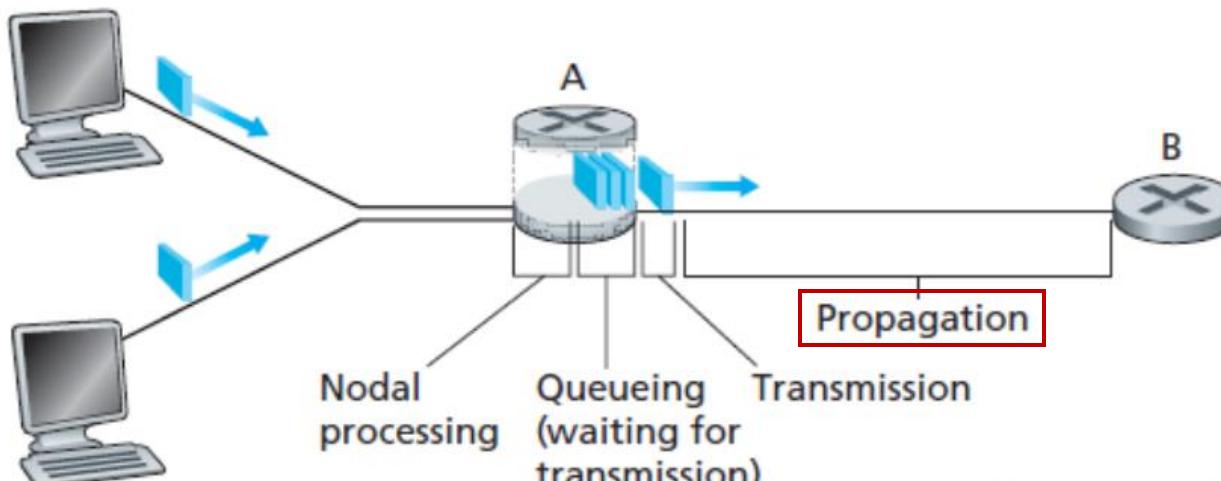
(Diagram from Kurose & Ross)

$$\text{延迟}_{\text{tr}} = \text{分组长度} / \text{传输速率}$$

# 传播延迟

- **传播延迟 (传播时间)**

- 比特从源头到达目的地所需的时间。
- 取决于传输**介质**。
- 在真空中，速度为  $3 \times 10^8$  m/s（在空气和电缆中较低）。



(Diagram from Kurose & Ross)

$$\text{延迟}_{pg} = \frac{\text{距离}}{\text{传播速度}}$$

# 传播时间和传输时间（示例）

- 如果网络带宽为 1 Gbps, 3.5 kB (千字节) 消息 (例如电子邮件) 的传播时间和传输时间分别是多少? 假设发送方与接收方之间的距离为 15,000 km, 光速为  $2.4 \times 10^8$  m/s。

## • 答案:

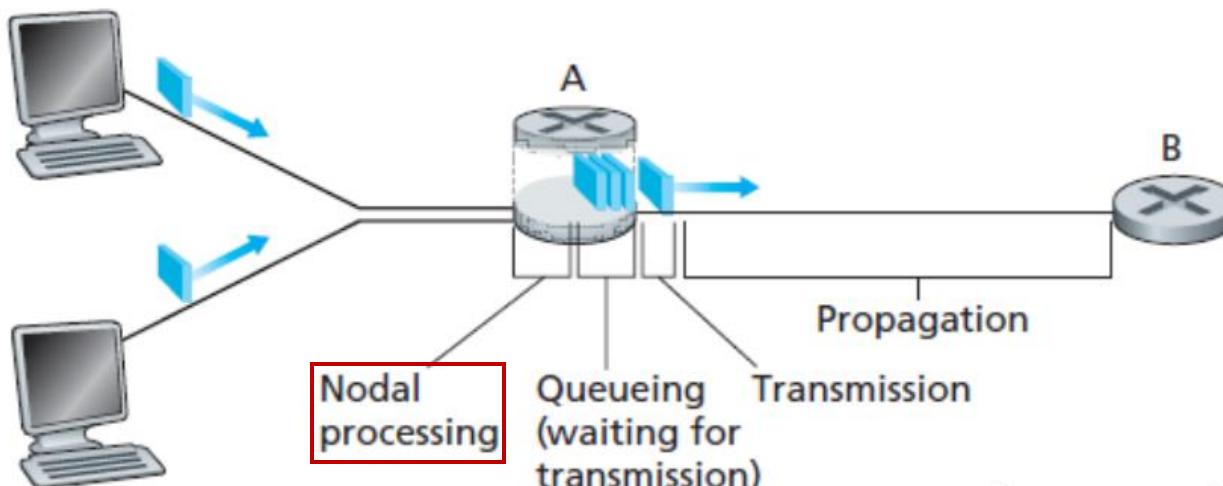
$$\text{Propagation time} = \frac{(15000 \times 1000)}{2.4 \times 10^8} = 0.0625 = \mathbf{62.5 \text{ ms}} = \text{Delay}_{\text{pg}}$$

$$\text{传输时间} = \frac{3.5 \times 1000 \times 8}{1 \times 10^9} = 0.000028 = \mathbf{28 \mu\text{s}} = \text{延迟}_{\text{tr}}$$

# 处理延迟

- **处理延迟 (处理时间)**

- 在**路由器处理数据包或在目的主机中的所需时间。**
- 路由器或目的主机从其输入端口接收数据包，移除头部，执行错误检测过程，然后将数据包发送到输出端口（在路由器的情况下）或交付给上层协议（在目的主机的情况下）。
- 对于每个数据包可能**不同**，但通常按**平均值**来计算。



(Diagram from Kurose & Ross)

# 排队/等待 延迟

- **排队/等待 延迟**

- 中间或终端设备为保存消息而需要的时间 在消息可以被处理之前（通常指的是 等待时间，数据包 在 输入/输出队列 中的 路由器）。
- 随网络负载变化，例如当网络上流量繁重时会增加排队时间 → 不固定
- 路由器在其每个输入端口都设有一个与该输入端口连接的输入队列，用于存放等待处理的数据包。
- 路由器还在其每个输出端口设有 与该输出端口连接的输出队列，用于存放等待发送的数据包。

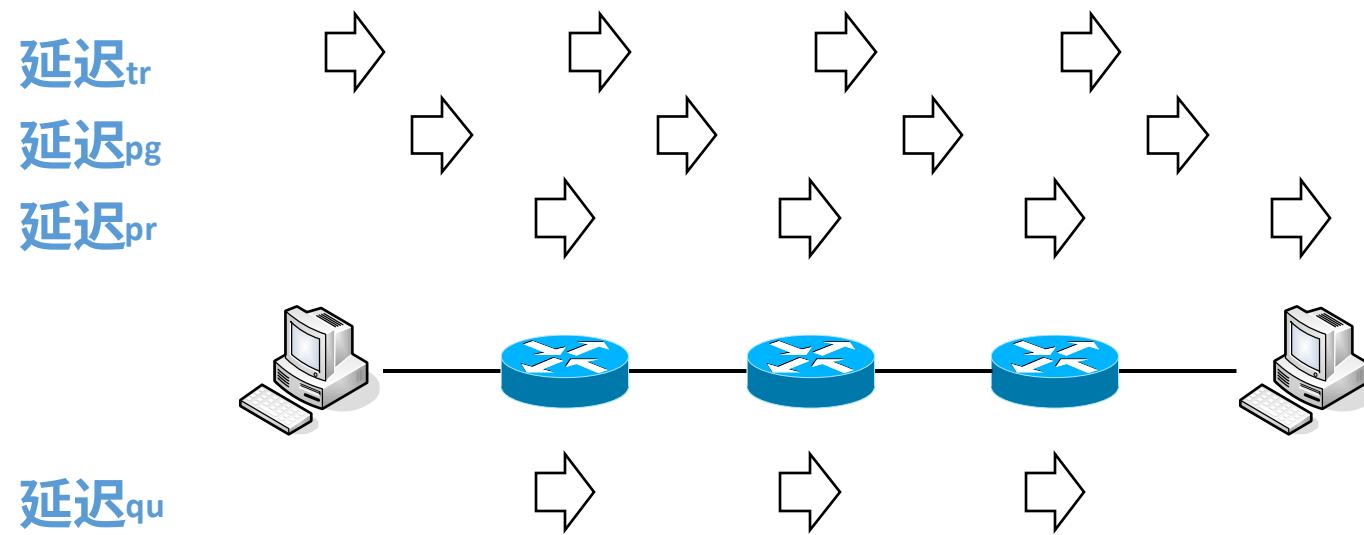
# 总延迟（时延）

- 数据包的总延迟（从源到目的地的延迟）
  - 假设发送方、路由器和接收方的延迟相等。
  - $n$  是路由器的数量  $\rightarrow n + 1$  条链路
    - $(n + 1)$  与  $n$  个路由器 和 源 相关的传输延迟.
    - $(n + 1)$  与  $(n + 1)$  条链路相关的传播延迟.
    - $(n + 1)$  处理延迟 与  $n$  路由器 和 目的地。
    - $n$  排队延迟 与  $n$  路由器。

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

# 总延迟（时延）

$$\text{总延迟} = (n + 1)(\text{延迟}_{tr} + \text{延迟}_{pg} + \text{延迟}_{pr}) + (n)(\text{延迟}_{qu})$$



# 吞吐量

- 衡量我们能多快地 **实际** 通过网络发送数据。
- 不同于以比特每秒为单位的带宽。
  - 带宽是对链路的 **潜在** 度量。
  - 一条链路的带宽可能是  $B$  bps，但我们只能通过该链路发送  $T$  bps ( $T \leq B$ ) 。
  - **最大吞吐量** 本质上等同于 **数字带宽 容量**。

# 吞吐量 – 示例

- **示例**: 一个带宽为 5 Mbps 的网络每分钟只能传输平均 15,000 个数据包，每个数据包平均携带 10,000 比特。该网络的吞吐量是多少？

# 吞吐量 – 示例

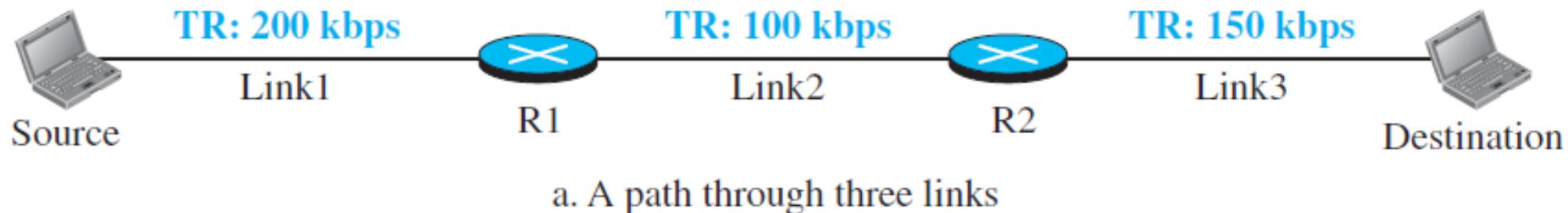
- **示例**: 一个带宽为 5 Mbps 的网络，在每个数据包平均承载 10,000 比特的情况下，每分钟只能传输平均 15,000 个数据包。该网络的吞吐量是多少？

- **答案**:

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

# 吞吐量

- 如何确定从源到目的地的整条路径的吞吐量?
  - 由**瓶颈链路**决定，即具有最小数据率（传输速率）的链路。
  - 对于一条有  $n$  条链路的路径：**吞吐量 = 为  $\{TR_1, TR_2, \dots, TR_n\}$  中的最小值**

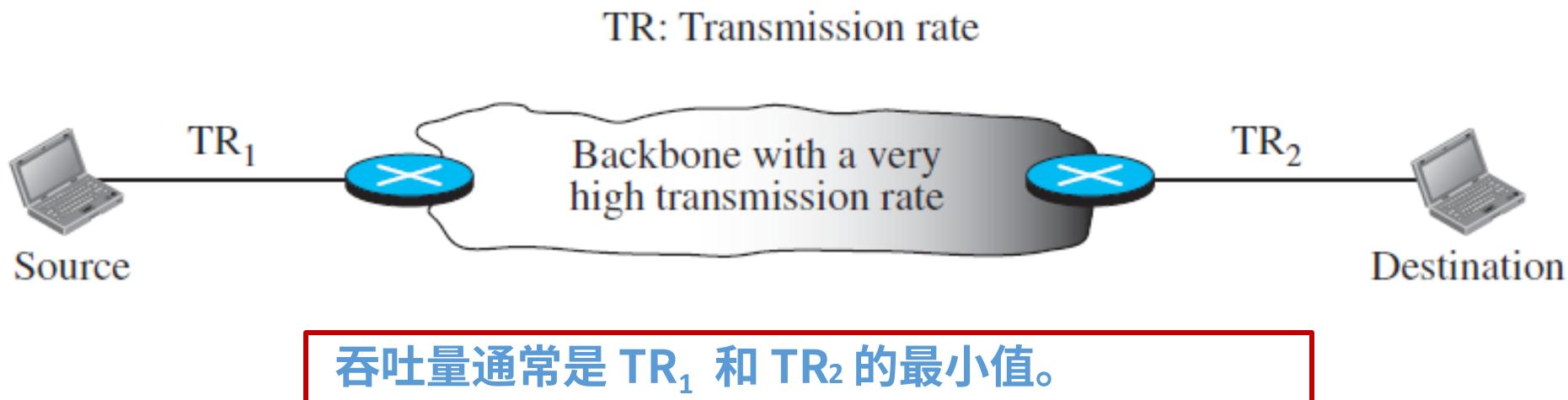


TR: Transmission rate

吞吐量 = 100 kbps

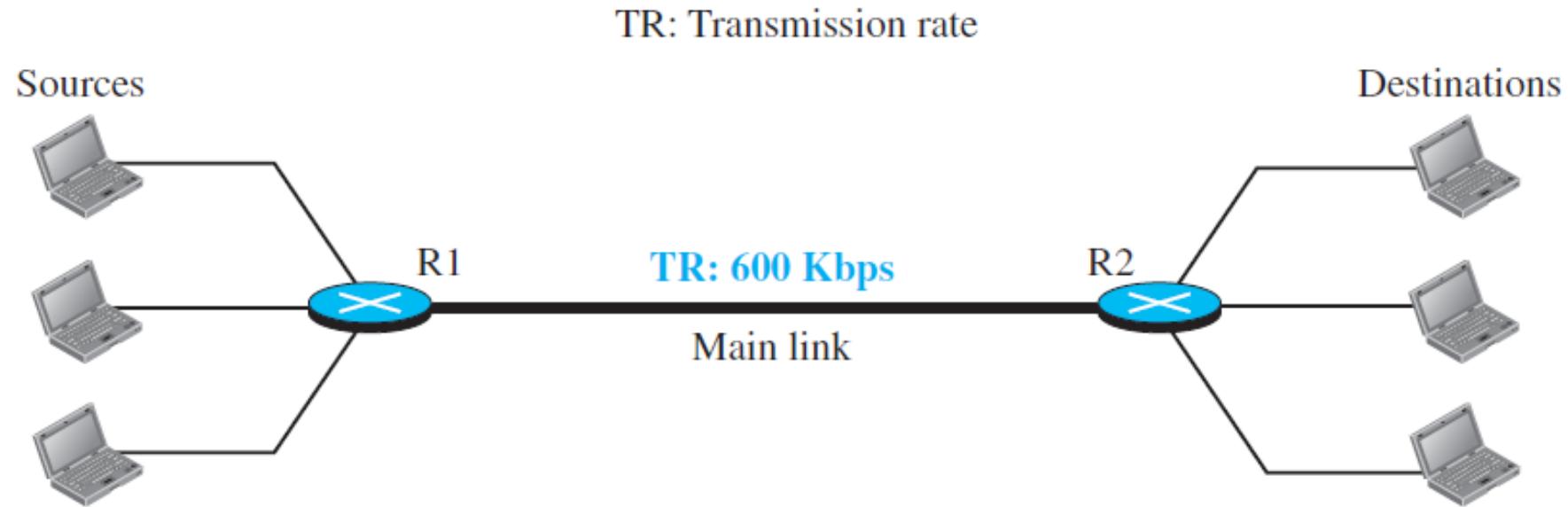
# 吞吐量

- 在互联网中，数据通常通过两个 **接入点** 网络与 **互联网骨干**。



# 共享链路的吞吐量

- 如果两个路由器之间的一条链路的传输速率是共享的，会怎样？在多个流之间？



在吞吐量计算中，主链路的传输速率只有200 kbps，因为该链路在三个流之间共享。

# 数据包丢失

- 丢包数量会严重影响性能。
- 路由器的输入缓冲区（队列）大小是有限的。
- 如果**缓冲区已满**，下一个数据包就需要被丢弃。
- 丢包对互联网网络层的影响：
  - 数据包需要重发，这反过来可能导致**溢出并引起更多丢包**。

# 抖动

- 延迟的变化。
  - 如果不同数据包遭遇不同的延迟，而接收端使用这些数据的应用对时间敏感（例如音频和视频数据），那就是一个问题。
  - 例如：若第一个数据包的延迟为20毫秒，第二个为45毫秒，第三个为40毫秒，那么使用这些数据包的实时应用就会遭遇抖动。

# 摘要

- 数据报网络提供 **无连接服务**。
- 互联网是一个 **分组交换网络**。
- 网络层的主要服务： **分组化**, **路由** 和 **转发**。
- 总时延包含四个部分： **传输时延**, **传播时延**, **处理时延**, 和 **排队时延**。
- 端到端**吞吐量**受路径上**最慢链路**的限制。

# 参考文献

- [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.

# 阅读

- 教材第7章，第7.1-7.3节。
- 教材第7章，第7.8节（练习测试）