

# 计算机网络

## 期末复习

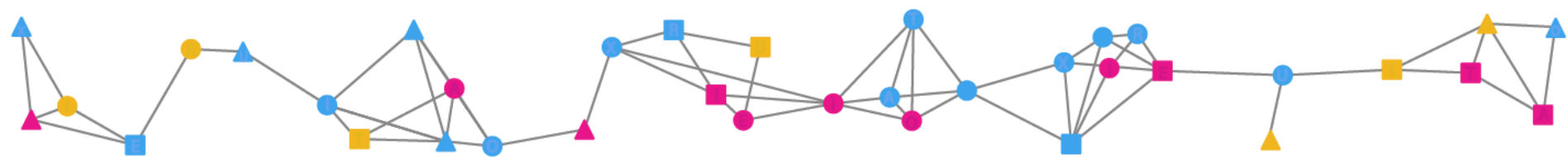
---

谢瑞桃

[xie@szu.edu.cn](mailto:xie@szu.edu.cn)

[rtxie.github.io](https://github.com/rtxie)

计算机与软件学院  
深圳大学



# 计算机网络

## 第一章 概论

---



# 第一章讲解内容

---

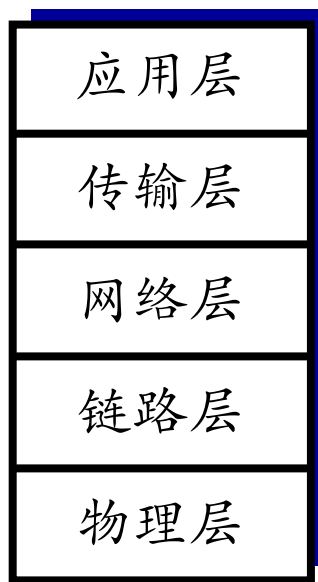
1. 什么是因特网?
2. 网络边缘
  - 端系统，接入网，链路
3. 网络核心
  - 分组交换，电路交换，网络互联
4. 协议分层模型
5. 发展历史



## 4. 因特网协议栈

### TCP/IP模型， TCP/IP协议栈

- **应用层**：支持各种网络应用程序
  - HTTP, SMTP
- **传输层**：进程与进程之间的数据传输
  - TCP, UDP
- **网络层**：将数据报(分组)从源主机路由到目的主机
  - IP, 路由协议
- **链路层**：相邻网络设备之间的数据传输
  - 以太网(Ethernet), WiFi
- **物理层**：利用传导介质传输电磁信号



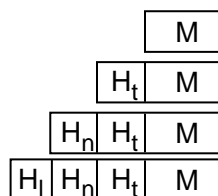
# 4.封装

报文message

报文段segment

数据报datagram

帧frame



源主机

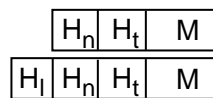
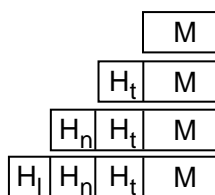


链路层  
物理层

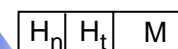


交换机

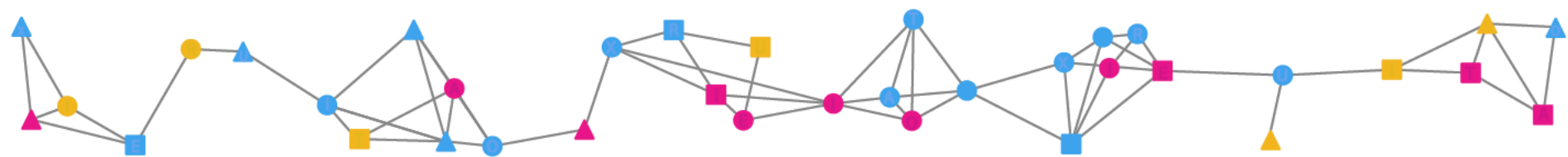
目的主机



网络层  
链路层  
物理层



路由器



# 计算机网络

## 第二章 应用层

---



# 第二章讲解内容

---

## 1. 应用层概述

- 客户/服务器架构，P2P架构，套接字

## 2. Web 和 HTTP

- 非持续HTTP，持续HTTP，响应时间，请求/响应报文

## 3. Email

- SMTP，POP3，IMAP

## 4. DNS 域名系统

- 层级结构，迭代查询，递归查询



# 应用程序对网络服务的需求

应用	数据丢失	带宽	时间敏感
文件传输	不能丢失	弹性	不
e-mail	不能丢失	弹性	不
Web	不能丢失	弹性	不
实时多媒体（网络电话，视频会议）	容忍丢失	音频: 5Kbps-1Mbps 视频: 10Kbps-5Mbps	是，几十毫秒
流媒体	容忍丢失	同上	是，几秒
交互式游戏	容忍丢失	Kbps+ 云游戏几十Mbps	是，几十毫秒
讯息	不能丢失	弹性	是或不是





# 应用程序对网络服务的需求

- 两种传输层协议提供两种网络服务
- UDP
  - 不可靠传输
    - 基本裸奔
- TCP
  - 可靠传输
    - 应对丢包、乱序等问题
  - 流量控制
  - 拥塞控制
    - 两把水龙头，控制流入网络的数据量



# 应用程序对网络服务的需求

应用	应用层协议	传输层协议
文件传输	FTP	TCP
e-mail	SMTP, IMAP, POP3	TCP
Web	HTTP/1.1, HTTP/2, HTTP/3	TCP
网络电话	SIP, RTP	TCP/UDP
流媒体	HTTP, DASH	TCP/UDP
交互式游戏	WOW, FPS (proprietary)	TCP/UDP



## 第二章知识点汇总

---

- 应用层概述
  - 理解两种架构
  - 掌握使用套接字进行网络编程的方法（实验）
  - 理解应用程序对网络服务的不同需求



# 第二章讲解内容

---

## 1. 应用层概述

- 客户/服务器架构，P2P架构，套接字

## 2. Web 和 HTTP

- 非持续HTTP，持续HTTP，响应时间，请求/响应报文

## 3. Email

- SMTP，POP3，IMAP

## 4. DNS 域名系统

- 层级结构，迭代查询，递归查询

# 非持续HTTP

- 用户输入URL: `http://www.szu.edu.cn/`



1. 创建TCP套接字。HTTP客户端向服务器`www.szu.edu.cn`的80端口发起TCP连接。

2. 创建TCP套接字。主机`www.szu.edu.cn`的HTTP服务器接受连接，并通知客户端。

3. HTTP客户端向TCP套接字发送包含URL的HTTP请求报文，请求对象/`/`，即缺省主页。

4. HTTP服务器接收请求报文，将所请求的对象封装成响应报文，并发送给TCP套接字。

5. HTTP客户端收到包含HTML文件的响应报文，并显示。

在4结束后，HTTP服务器关闭连接。  
HTML所引用的其他对象用新的TCP连接传输。

time

# 非持续HTTP响应时间

- RTT (Round Trip Time) : 一个非常小的分组从客户端传输到服务器, 再返回来所需要的往返时间

这种方法好吗?

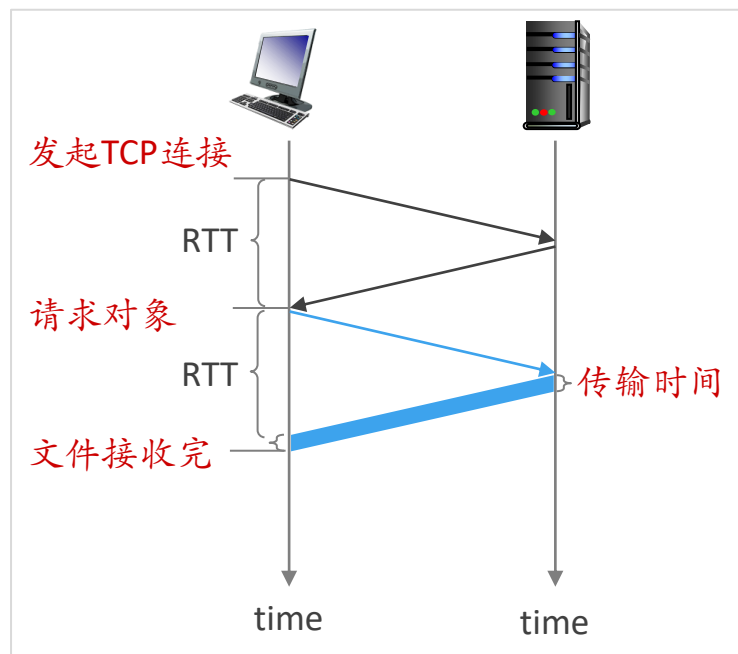
- 每个对象的HTTP响应时间

= 建立TCP连接所需的RTT

+ HTTP请求/响应所需的RTT

+ 对象传输时间

=  $2RTT + \text{传输时间}$





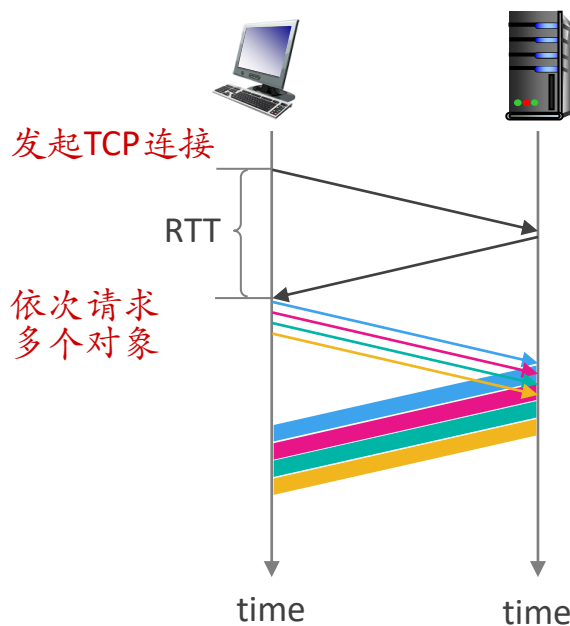
# 持续HTTP (HTTP/1.1)

---

- 一个TCP连接用于请求/响应多个对象
- 每个对象的HTTP响应时间 = RTT + 传输时间

# 持续HTTP (HTTP/1.1)

- 一个TCP连接用于请求/响应多个对象
- 每个对象的HTTP响应时间 = RTT + 传输时间
- 流水线工作方式：在一个对象请求的响应到达之前，发送其他对象的请求。







# 第二章知识点汇总

---

## ■ Web和HTTP

- 理解Web系统的构成
- 理解非持续和持续HTTP
- 掌握响应时间的计算方法
- 了解请求/响应报文的基本结构



# 第二章讲解内容

---

## 1. 应用层概述

- 客户/服务器架构，P2P架构，套接字

## 2. Web 和 HTTP

- 非持续HTTP，持续HTTP，响应时间，请求/响应报文

## 3. Email

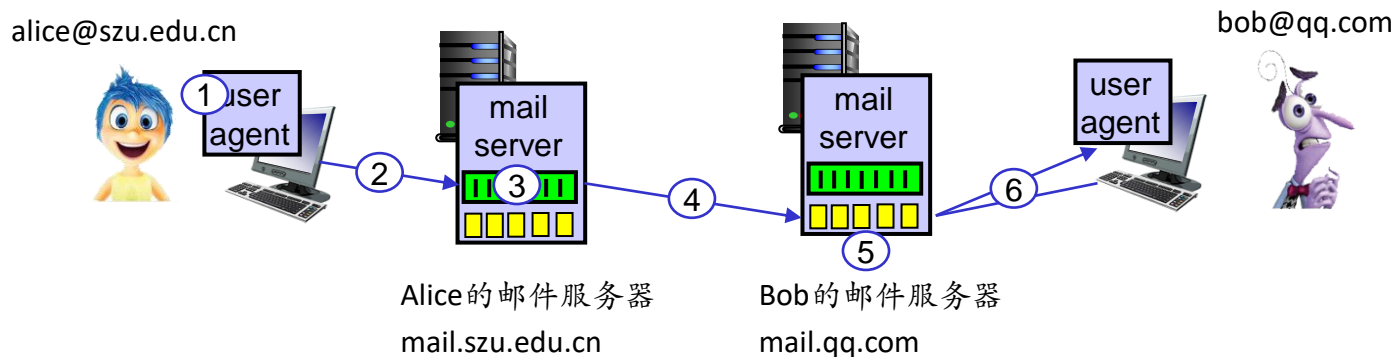
- SMTP, POP3, IMAP

## 4. DNS 域名系统

- 层级结构，迭代查询，递归查询

# Email 传输

- 1) Alice用Outlook写了一封邮件，准备用邮箱alice@szu.edu.cn发给bob@qq.com。
- 2) Alice的Outlook将邮件报文发送给邮件服务器mail.szu.edu.cn，到达报文队列。
- 3) 邮件服务器mail.szu.edu.cn上的SMTP客户端建立一个与服务器mail.qq.com的TCP连接。
- 4) SMTP客户端通过该TCP连接发送邮件报文。
- 5) 邮件服务器email.qq.com将邮件报文存入Bob的邮箱。
- 6) Bob利用他的UA阅读邮件，**注意UA是客户端，先发起TCP连接。**





# 第二章知识点汇总

---

## ■ Email

- 了解邮件系统
- 了解SMTP协议的基本原理



# 第二章讲解内容

---

## 1. 应用层概述

- 客户/服务器架构，P2P架构，套接字

## 2. Web 和 HTTP

- 非持续HTTP，持续HTTP，响应时间，请求/响应报文

## 3. Email

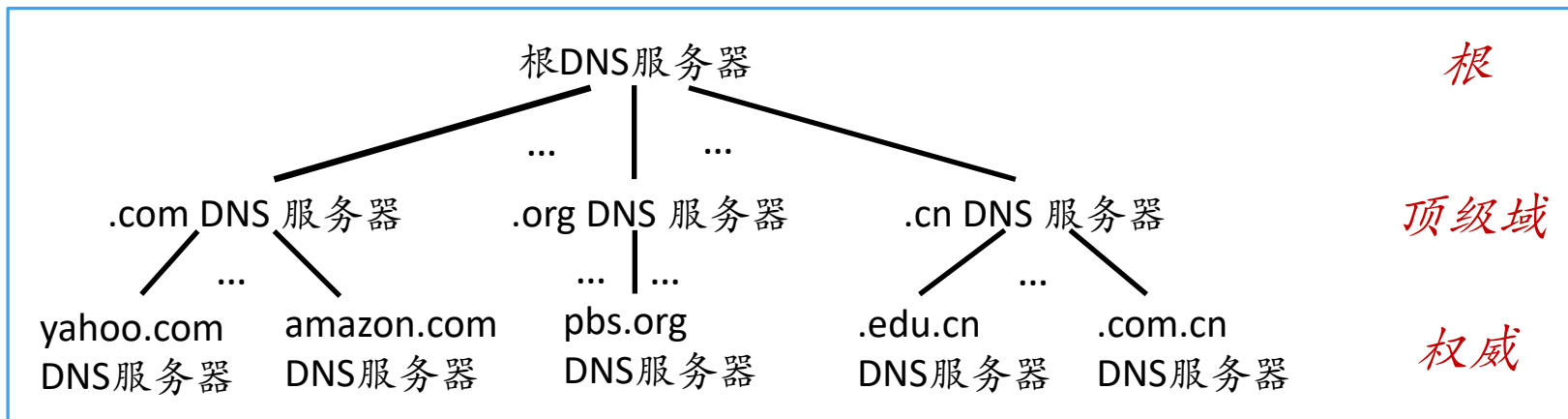
- SMTP，POP3，IMAP

## 4. DNS 域名系统

- 层级结构，迭代查询，递归查询

# DNS 域名系统

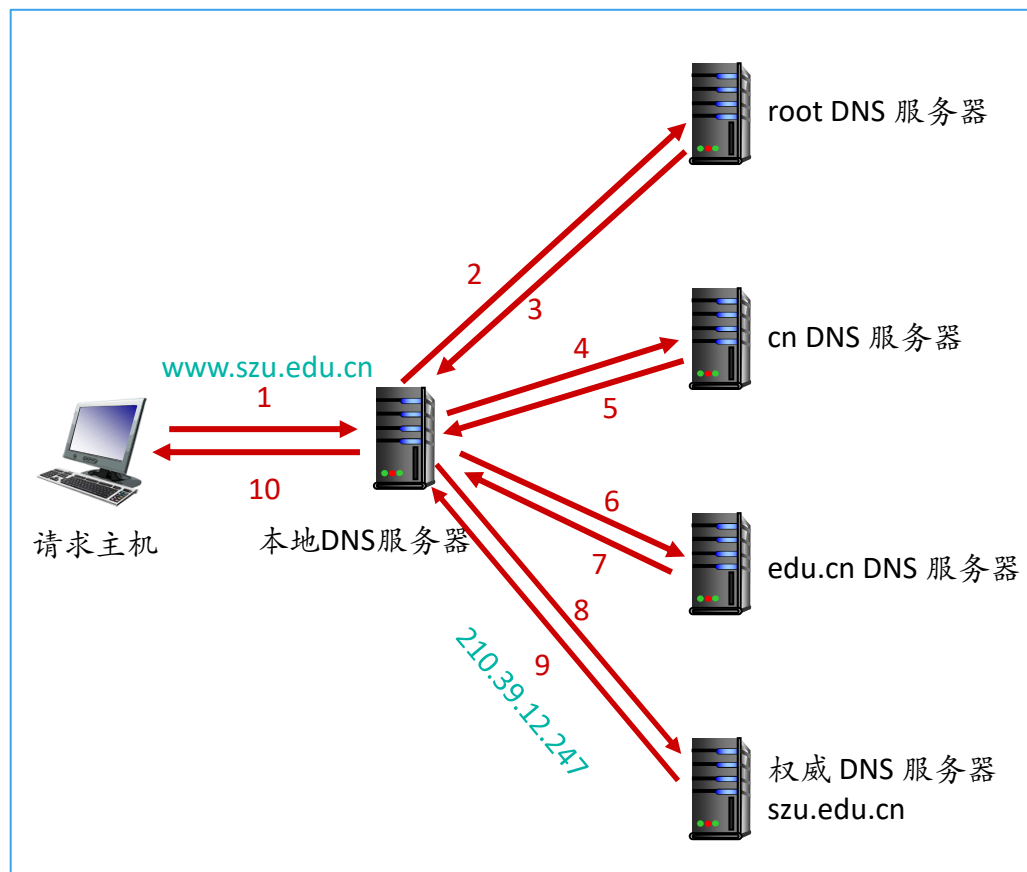
- 一个由众多的DNS服务器实现的分布式数据库



- 本地域名服务器
  - 一般ISP、企业、学校都会配置一个
  - 当主机做DNS查询时，请求会先发往本地域名服务器
    - 有域名地址映射的缓存
    - 代理查询

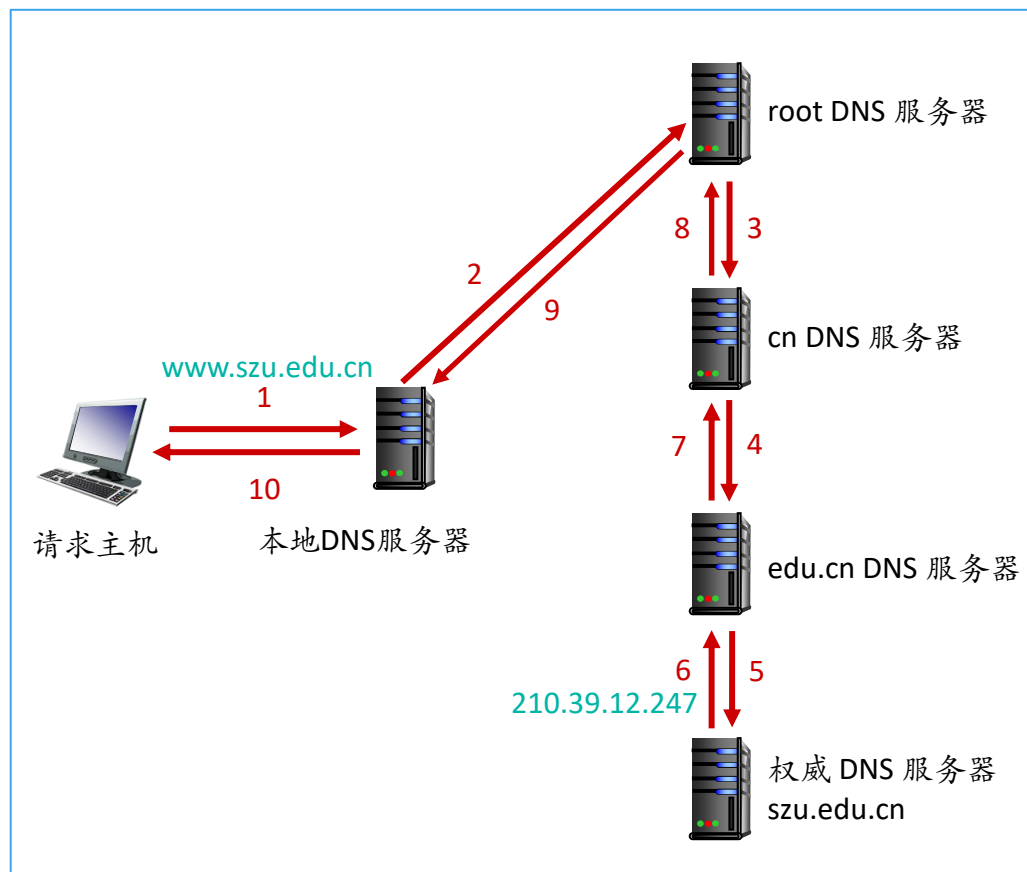
# DNS 域名解析：迭代查询

- 举例：查询 [www.szu.edu.cn](http://www.szu.edu.cn)
- 迭代查询
  - 被询问的服务器返回下一步询问的服务器地址
  - 我不知道，你去问XXX吧



# DNS 域名解析：递归查询

- 举例：查询 [www.szu.edu.cn](http://www.szu.edu.cn)
- 递归查询
  - 被询问的服务器去查询
  - 我不知道，我帮你去问吧



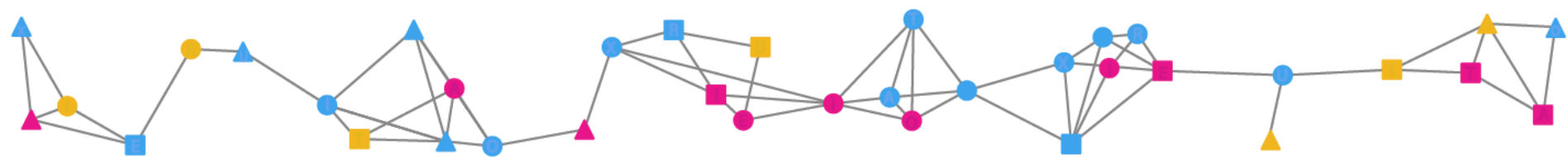




# 第二章知识点汇总

---

- DNS域名系统
  - 理解层级结构
  - 掌握迭代查询和递归查询



# 计算机网络

## 第三章 传输层

---



# 第三章讲解内容

---

## 1. 传输层概述与UDP

- 需求/服务/协议、多路复用/分解、UDP协议

## 2. 可靠传输

- 可靠传输基础知识、TCP可靠传输

## 3. TCP

- 报文段结构、超时时间间隔、流量控制、连接管理

## 4. TCP拥塞控制

- 网络拥塞、TCP拥塞控制、吞吐量分析



# 传输层需求、服务和协议

应用层需求	传输层服务	UDP	TCP
为运行在不同主机上的进程之间提供逻辑通信	进程间交付	✓	✓
检测报文段是否出错	差错检测	✓	✓
解决丢包、差错问题	可靠传输	✗	✓
解决乱序问题	按序交付	✗	✓
解决接收缓存溢出问题	流量控制	✗	✓
应对网络拥塞	拥塞控制	✗	✓

# 多路复用/多路分解

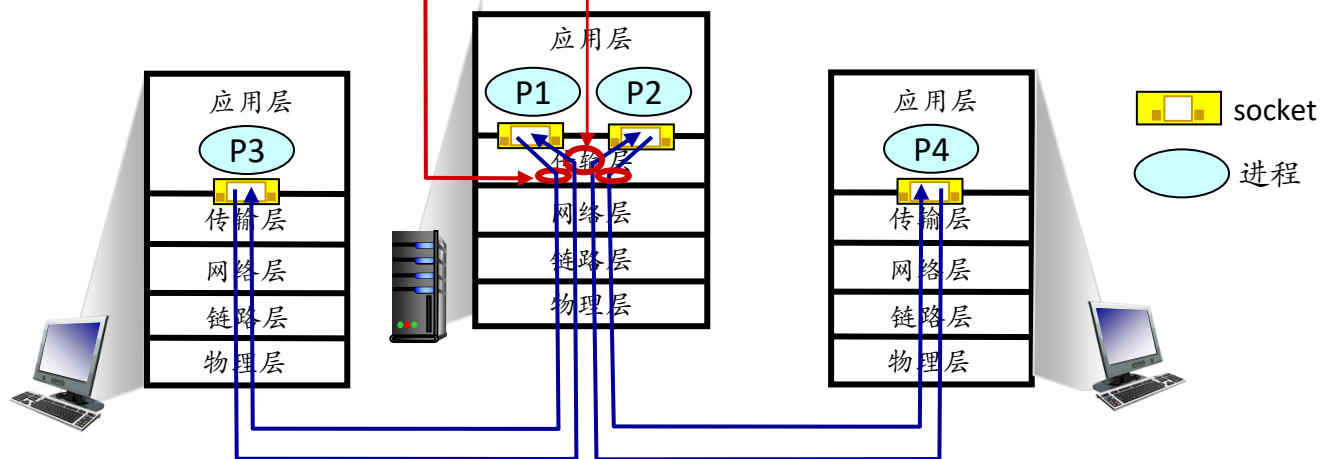
## ■ 解决进程间交付的技术

### 发送方多路复用

处理来自多个套接字的数据，添加传输层首部

### 接收方多路分解

利用报文段的首部，将数据交给正确的套接字





# 多路复用/多路分解

---

- IP地址标识主机
- 端口号标识进程
  - 16比特，0—65535之间
  - 其中0—1023是周知端口号（HTTP80，FTP21）



# 第三章讲解内容

---

## 1. 传输层概述与UDP

- 需求/服务/协议、多路复用/分解、UDP协议

## 2. 可靠传输

- 可靠传输基础知识、TCP可靠传输

## 3. TCP

- 报文段结构、超时间隔、流量控制、连接管理

## 4. TCP拥塞控制

- 网络拥塞、TCP拥塞控制、吞吐量分析



# 可靠传输讲解内容

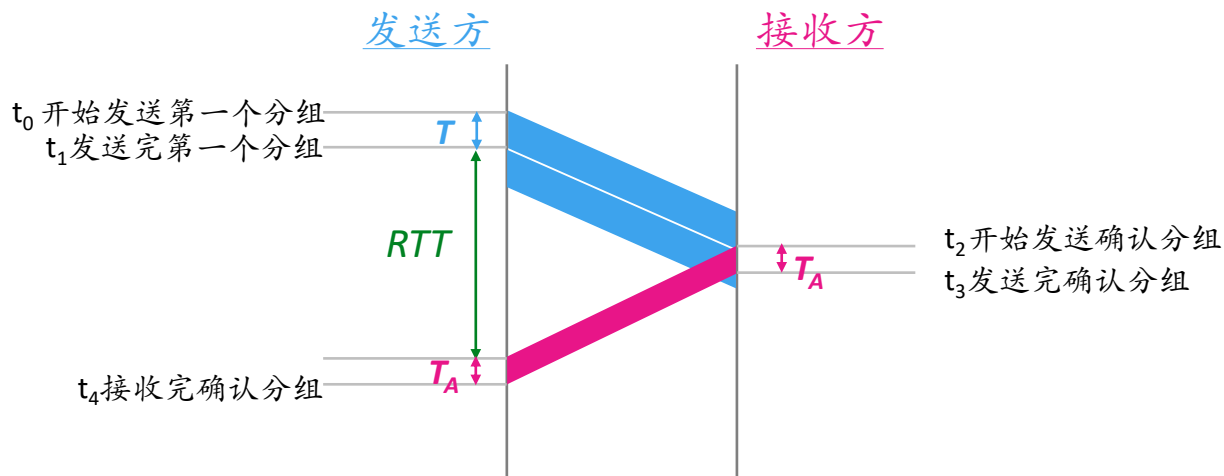
- 解决差错问题
- 解决丢包问题
- 停等协议与流水线协议
- 滑动窗口协议
- 回退N步
- 选择重传
- TCP可靠传输



# 信道利用率

信道利用率

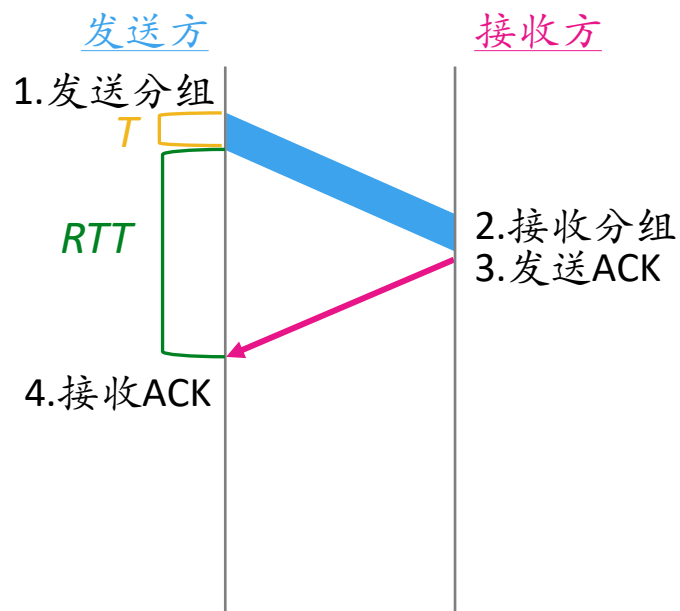
$$\begin{aligned} &= \frac{\text{信道被占用的时间}}{\text{周期}} \\ &= \frac{n \text{ 个分组的传输时间}}{\text{传输一对分组与其确认的时间}} \\ &= \frac{nT}{T + RTT + T_A} \end{aligned}$$



# 可靠传输—性能

- 忽略ACK传输时间的信道利用率：
- 停等协议

$$\frac{T}{RTT+T}$$



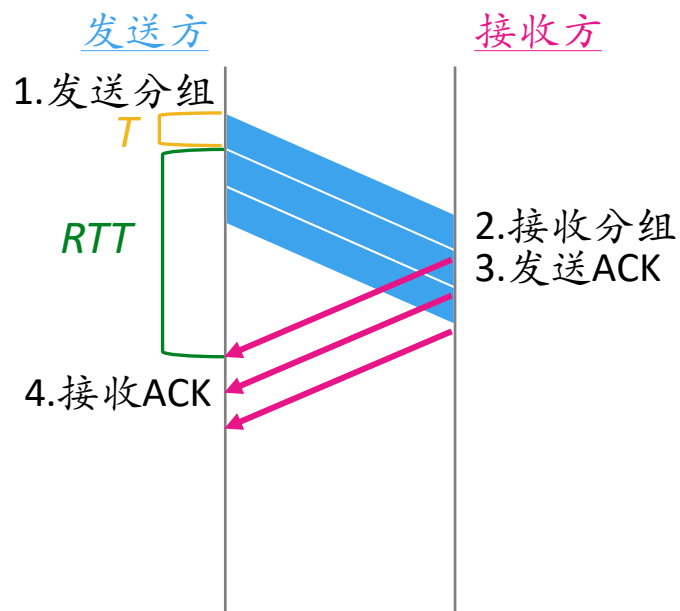
# 可靠传输—性能

- 忽略ACK传输时间的信道利用率：

$$\frac{3T}{RTT+T}$$

- 流水线提高了信道利用率

- 流水线





# 第三章知识点汇总

---

- 掌握可靠传输的信道利用率的计算方法



# 可靠传输讲解内容

- 解决差错问题
- 解决丢包问题
- 停等协议与流水线协议
- 滑动窗口协议
- 回退N步
- 选择重传
- TCP可靠传输



# 回退N步

- 允许未确认的分组数最大为 $N$
- 接收方不缓存失序分组，接收窗口为1，设计简单
- 失序分组都被丢弃，需要重传
- 超时：重传窗口中的所有分组
- 累积确认：对序号为 $n$ 的分组的确认，表示接收方已正确接收到序号小于等于 $n$ 的所有分组。
- 定时器：针对整个窗口，每次窗口滑动就重启定时器



# 第三章知识点汇总

---

- 理解回退N步（滑动窗口）协议的工作原理



# 可靠传输讲解内容

- 解决差错问题
- 解决丢包问题
- 停等协议与流水线协议
- 滑动窗口协议
- 回退N步
- 选择重传
- TCP可靠传输





# 选择重传

- 允许未确认的分组数  $> 1$
- 接收方缓存失序分组，接收窗口  $> 1$ ，设计复杂
- 失序分组被缓存，不需要重传
- 超时：只重传没收到确认的分组
- 逐个确认：对序号为  $n$  的分组的确认，表示接收方已正确接收该分组。
- 定时器：针对每个分组



# 第三章知识点汇总

---

- 理解选择重传协议的工作原理
- 理解回退N步与选择重传两种机制的区别



# 可靠传输讲解内容

- 解决差错问题
- 解决丢包问题
- 停等协议与流水线协议
- 滑动窗口协议
- 回退N步
- 选择重传
- TCP可靠传输



# TCP可靠传输

- 确保进程发送的字节流和通信进程接收的字节流完全相同
- 无损坏、无间隙、非冗余、按序



# TCP可靠传输

- 利用了校验和、确认、序号、定时器、流水线等技术
- **序号**：基于字节流，报文段数据首字节的字节流编号
- **确认号**：已确认的最大序号加一，含义是：期望从对方收到的下一个字节的序号
- 确认机制：**累积确认**（类似回退N步）
- 重传：**只重传丢失或损坏的分组**（类似选择重传）
- 定时器：**针对窗口中最早未确认的分组**
- 回退N步和选择重传的混合体

# TCP可靠传输

## ■ 无丢包

发送方

接收方

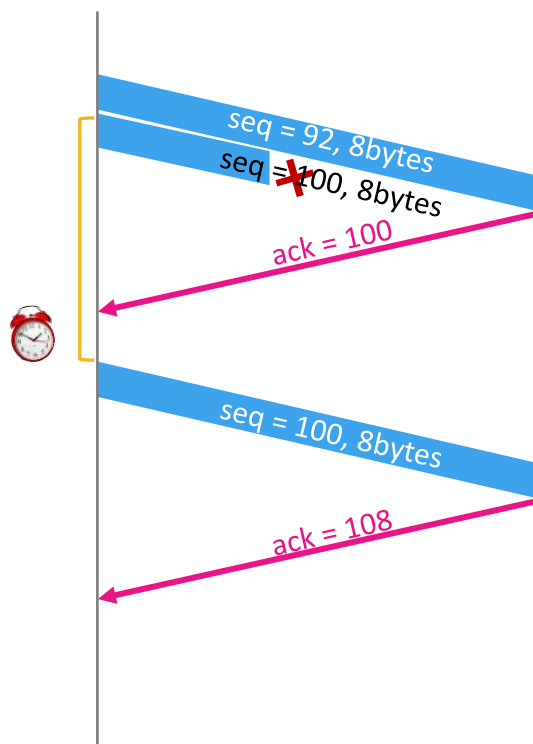


# TCP可靠传输

- 数据丢失
  - 假定没有其他数据发送

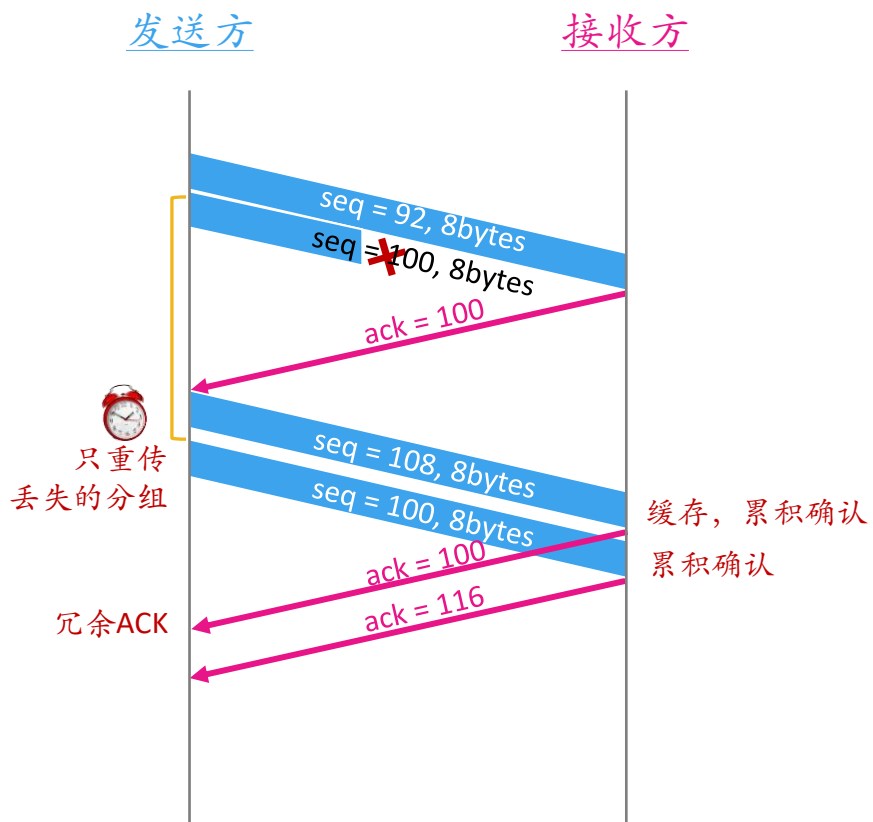
发送方

接收方



# TCP可靠传输

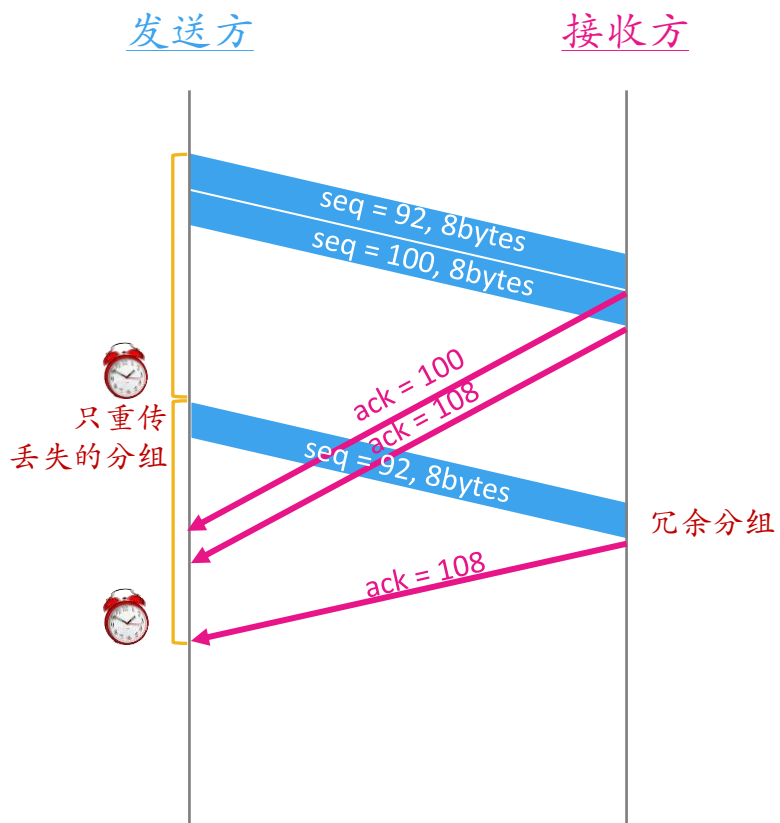
- 数据丢失
  - 假设还有数据发送



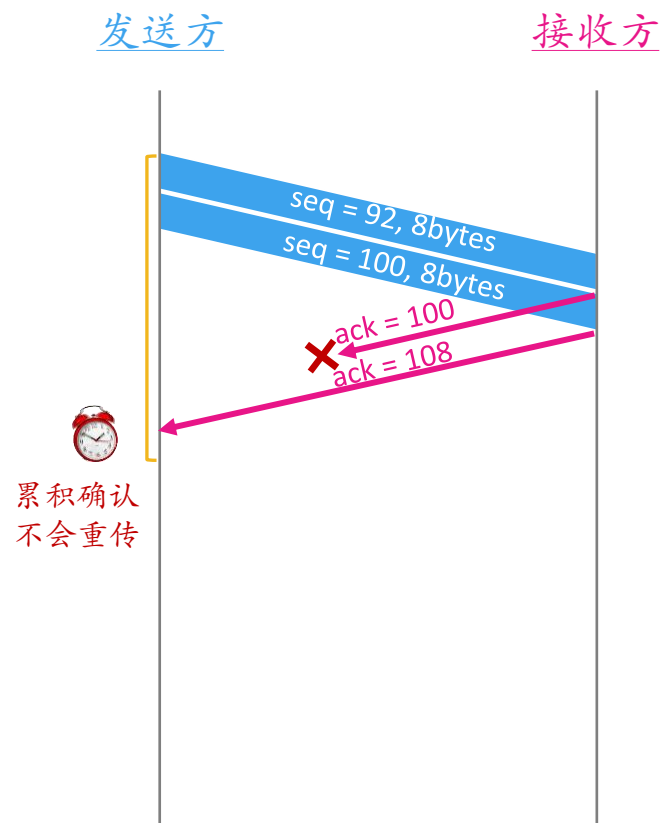


# TCP可靠传输

## ■ 过早重传



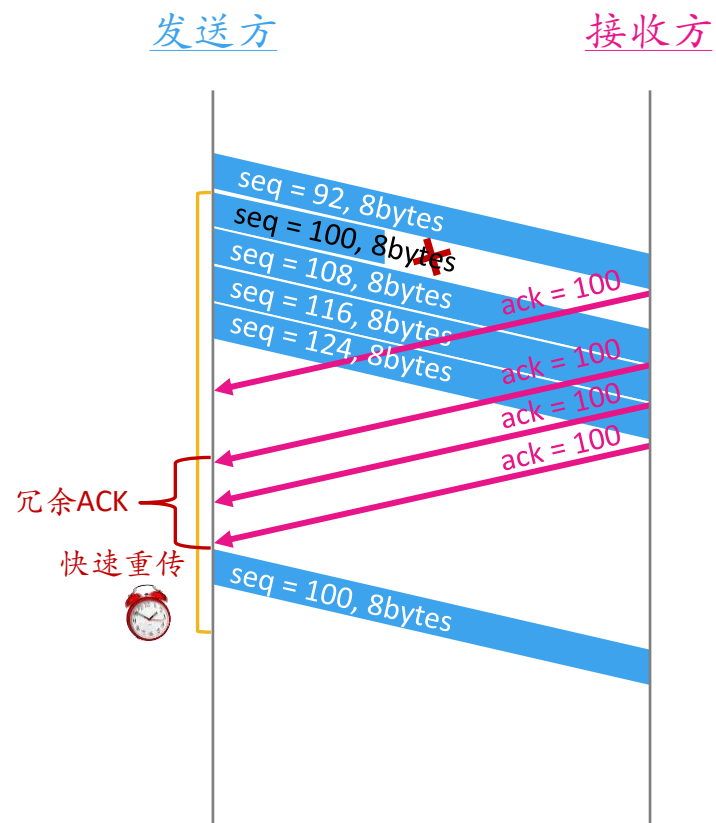
## ■ 累积确认



# TCP可靠传输

## ■ 快速重传

- 如果出现三个冗余ACK，即使定时器还没过期，提早重传





# 第三章知识点汇总

---

- 理解TCP可靠传输的原理
- 理解各种可靠传输技术在TCP中的应用
- 理解TCP快速重传的原理



# 第三章讲解内容

---

## 1. 传输层概述与UDP

- 需求/服务/协议、多路复用/分解、UDP协议

## 2. 可靠传输

- 可靠传输基础知识、TCP可靠传输

## 3. TCP

- 报文段结构、超时间隔、流量控制、连接管理

## 4. TCP拥塞控制

- 网络拥塞、TCP拥塞控制、吞吐量分析



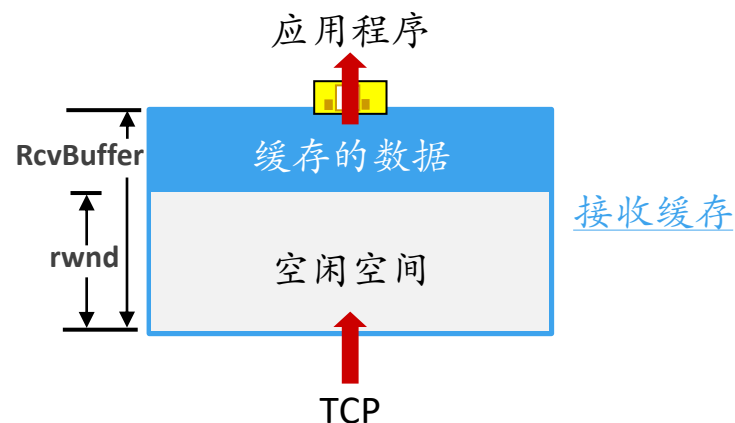
# TCP讲解内容

---

- 报文段结构
- 超时时间间隔
- 流量控制
- 连接管理

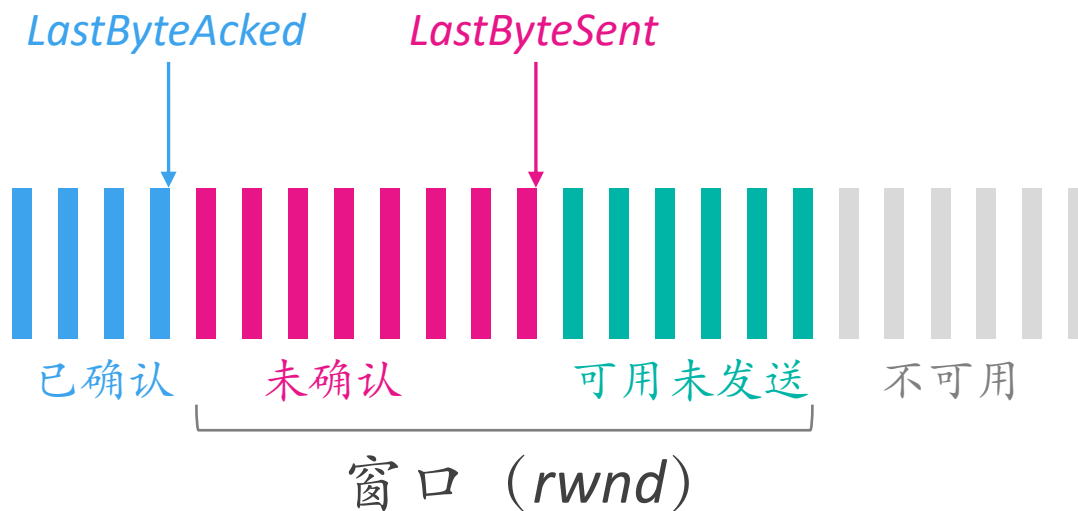
# TCP流量控制

- 接收方计算接收窗口 (rwnd)，将其放在TCP首部随数据发给发送方
- 发送方限制已发送未确认的字节数（窗口）不超过 rwnd
- 接收缓存 (RcvBuffer) 的容量由socket的选项决定



# TCP流量控制

- 已发送未确认的字节数  $\leq rwnd$
- $LastByteSent - LastByteAcked \leq rwnd$



图示：发送方的序号空间



# 第三章知识点汇总

---

- 理解流量控制技术的目的
- 理解流量控制技术的原理





# TCP讲解内容

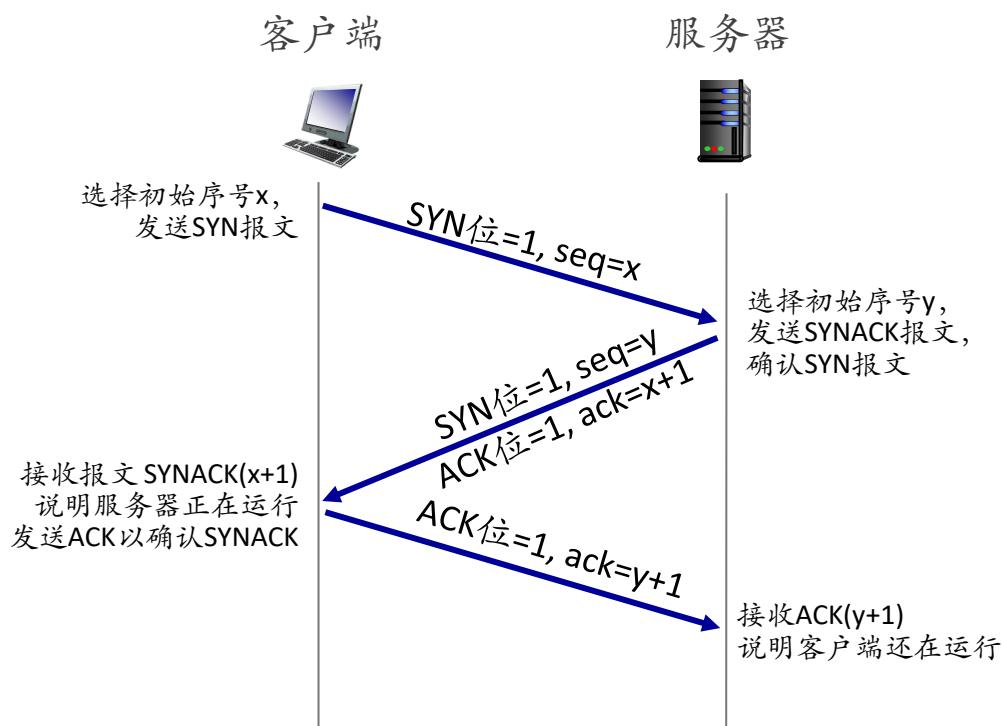
---

- 报文段结构
- 超时时间间隔
- 流量控制
- 连接管理

# TCP 建立连接

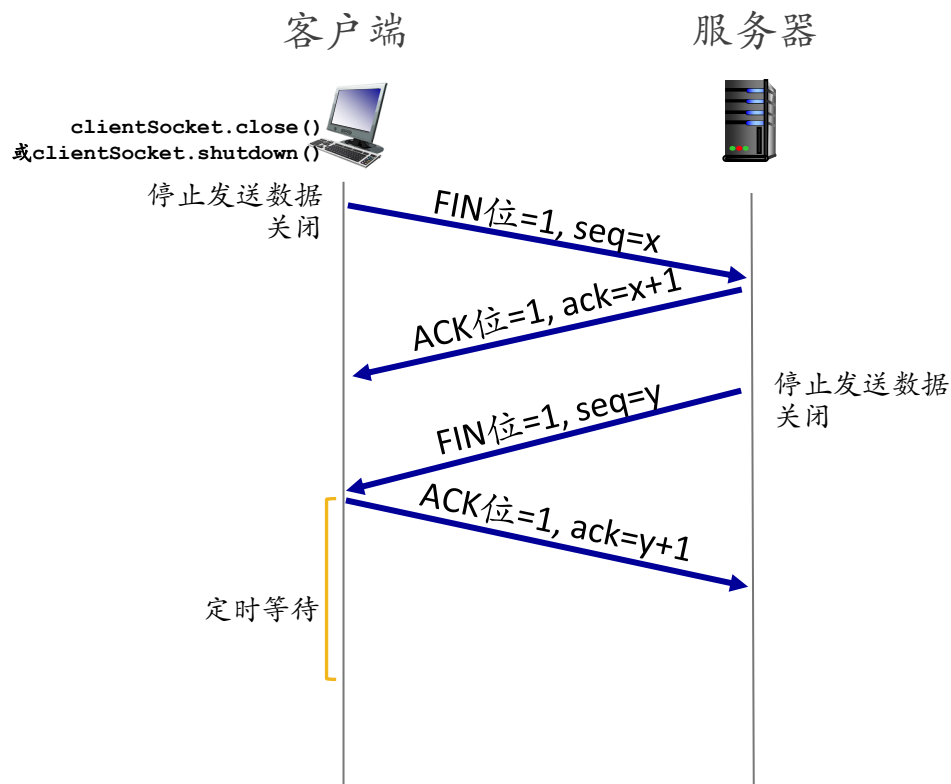
## ■ 三次握手

- SYN报文虽然不带数据，但在计算ack时仍然算一个字节



# TCP 关闭连接

- 双方分别关闭连接
  - ACK分组和FIN分组可以合并





# 第三章知识点汇总

---

- 理解TCP建立连接的步骤
- 理解TCP关闭连接的步骤



# 第三章讲解内容

---

## 1. 传输层概述与UDP

- 需求/服务/协议、多路复用/分解、UDP协议

## 2. 可靠传输

- 可靠传输基础知识、TCP可靠传输

## 3. TCP

- 报文段结构、超时时间间隔、流量控制、连接管理

## 4. TCP拥塞控制

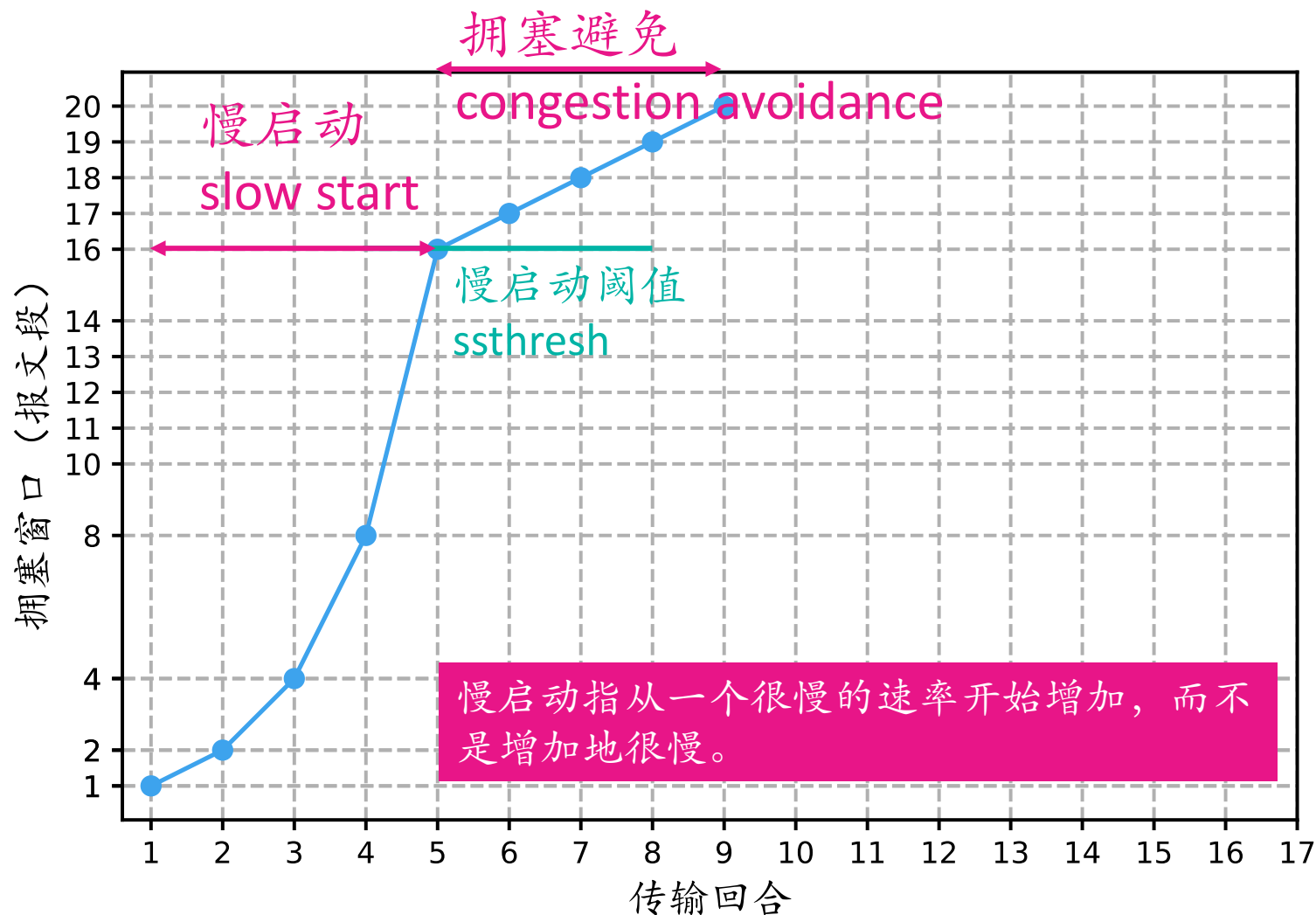
- 网络拥塞、TCP拥塞控制、吞吐量分析



# TCP 拥塞控制

- 问题三： TCP发送方如何动态调节发送速率(cwnd)?
- 子问题2： 如何增加cwnd?
  - 线性增加：
    - 每个RTT增加一个MSS
    - 增长速度慢，但一旦发生拥塞损失小
  - 指数增加：
    - 每个RTT增加一倍MSS
    - 增长速度快，但一旦发生拥塞损失大
- 用哪种方法好呢?
- 先指数增加，超过一定阈值以后线性增加

# TCP 拥塞控制



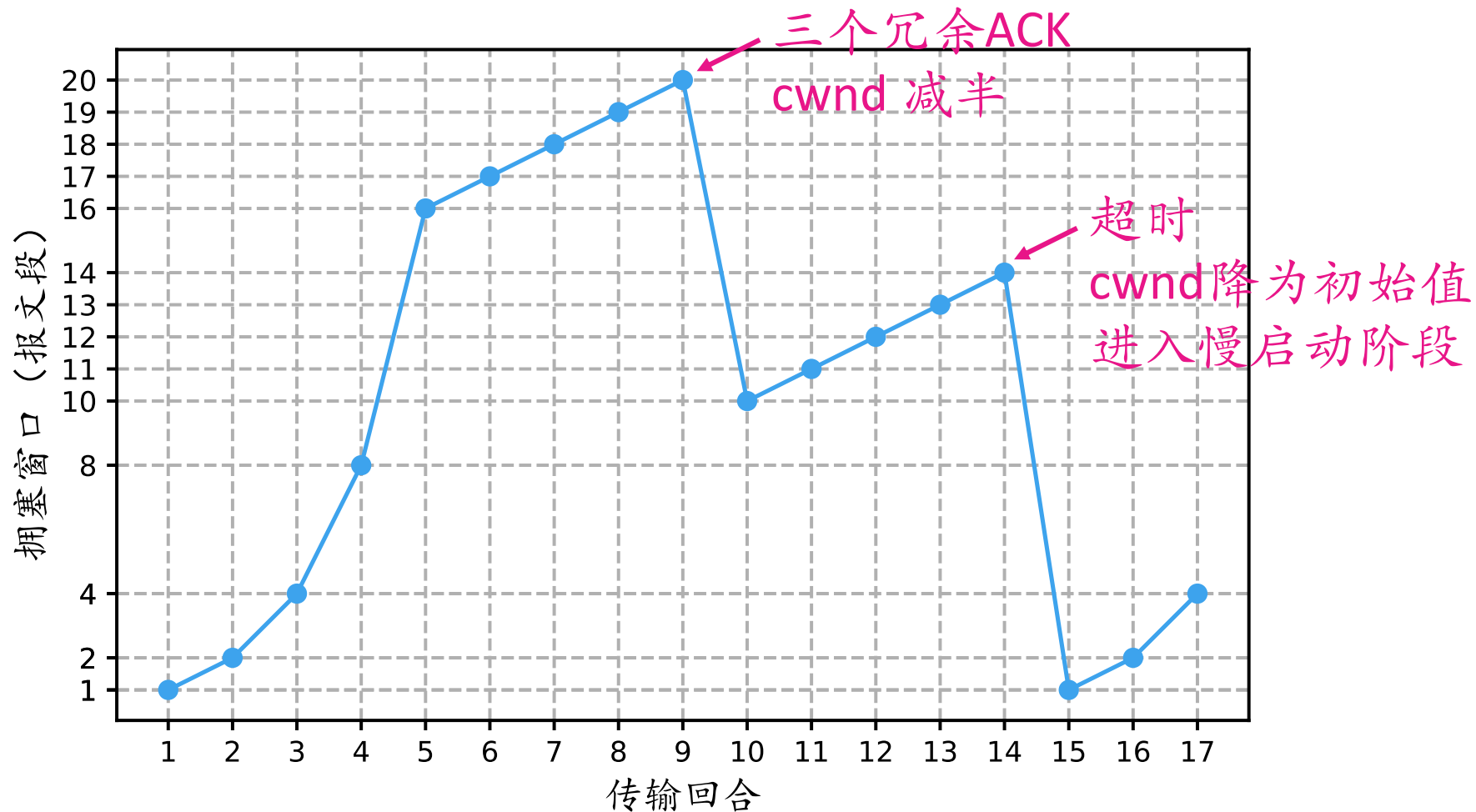


# TCP 拥塞控制

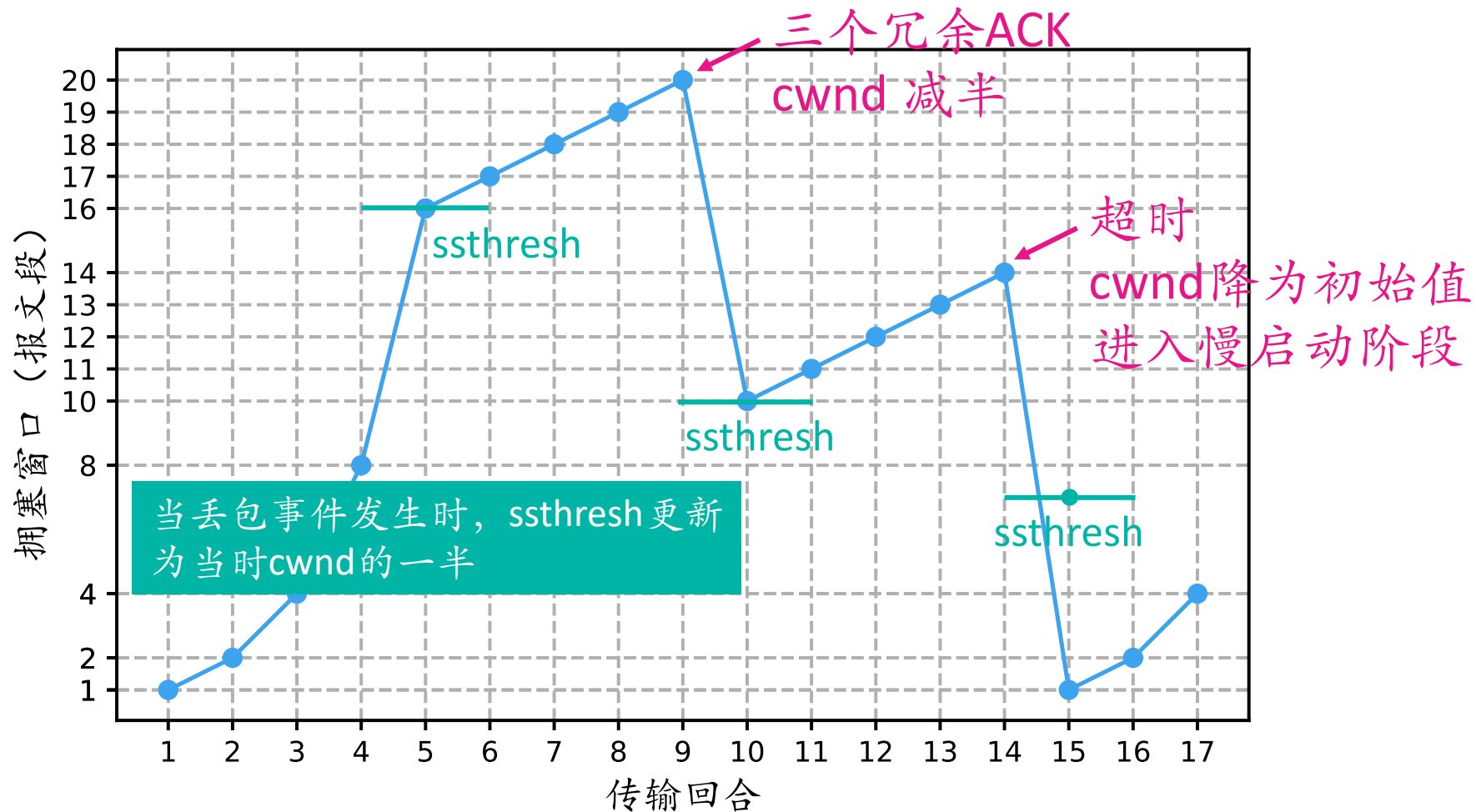
- 问题三： TCP发送方如何动态调节发送速率(cwnd)?
- 子问题3： 丢包时，如何降低cwnd?
- 定时器超时
  - 说明拥塞
  - 将cwnd降为初始值，进入慢启动（指数增长）阶段
- 收到3个冗余ACK
  - 虽然某个分组丢失了，但有些分组被收到了，说明拥塞不是很严重
  - 将cwnd减半，进入拥塞避免（线性增长）阶段



# TCP 拥塞控制

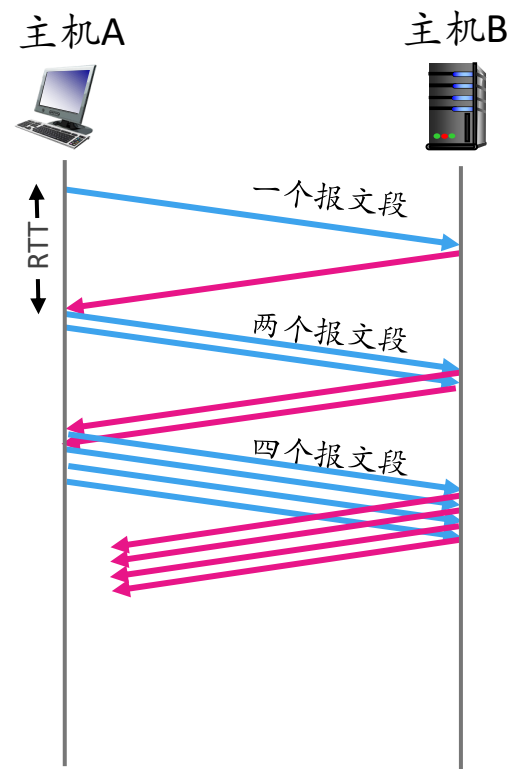


# TCP 拥塞控制



# TCP 拥塞控制

- 如何算法实现cwnd增加?
- 慢启动中的指数增加:  
每个RTT增加一倍MSS
- 每收到一个ACK, cwnd  
增加一个报文段





# TCP 拥塞控制

---

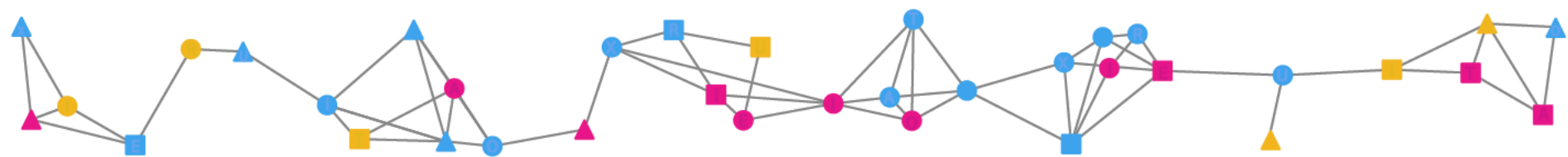
- 如何算法实现cwnd增加?
- 拥塞避免中的线性增加: 每个RTT增加一个MSS
- 每收到一个ACK, cwnd增加多少字节?
- $MSS * (MSS / cwnd)$



# 第三章知识点汇总

---

- 理解TCP(Reno)拥塞控制算法的原理
- 理解TCP(Reno)拥塞控制算法的设计逻辑
- 理解实现cwnd增加的算法



# 计算机网络

## 第四章 网络层

---



# 第四章讲解内容

---

1. 网络层概述与数据平面
2. 控制平面



# 数据平面讲解内容

---

- 网络层概述与功能
- 路由器体系结构
- IP数据报格式与分片
- IP地址
- 转发
- 划分地址块与地址聚合
- DHCP协议
- 网络地址转换NAT
- IPv6



# IP 数据报分片

- 举例：数据报4000字节，MTU=1500字节

	数据报长度 = 4000	16位标识 = x	0	DF = 0	MF = 0	13位片偏移 = 0	
	首部+数据	由源主机添加，标识相同的分片	DF: 0 = May Fragment, 1 = Don't Fragment.  MF: 0 = Last Fragment, 1 = More Fragments.			数据部分的偏移，8字节为单位	



# IP数据报分片

## ■ 举例：数据报4000字节，MTU=1500字节

20字节首部 +  
3980字节数据

	数据报长度 = 4000	16位标识 = x	0	DF = 0	MF = 0	13位片偏移 = 0	
--	-----------------	--------------	---	-----------	-----------	---------------	--

20字节首部 +  
1480字节数据

	数据报长度 = <b>1500</b>	16位标识 = x	0	DF = 0	MF = <b>1</b>	13位片偏移 = <b>0</b>	
--	------------------------	--------------	---	-----------	------------------	----------------------	--

20字节首部 +  
1480字节数据

	数据报长度 = <b>1500</b>	16位标识 = x	0	DF = 0	MF = <b>1</b>	13位片偏移 = <b>1480/8 = 185</b>	
--	------------------------	--------------	---	-----------	------------------	---------------------------------	--

20字节首部 +  
1020字节数据

	数据报长度 = <b>1040</b>	16位标识 = x	0	DF = 0	MF = <b>0</b>	13位片偏移 = <b>370</b>	
--	------------------------	--------------	---	-----------	------------------	------------------------	--



## 第四章知识点汇总

---

- 理解IP数据报分片的原因
- 掌握IP数据报分片的方法



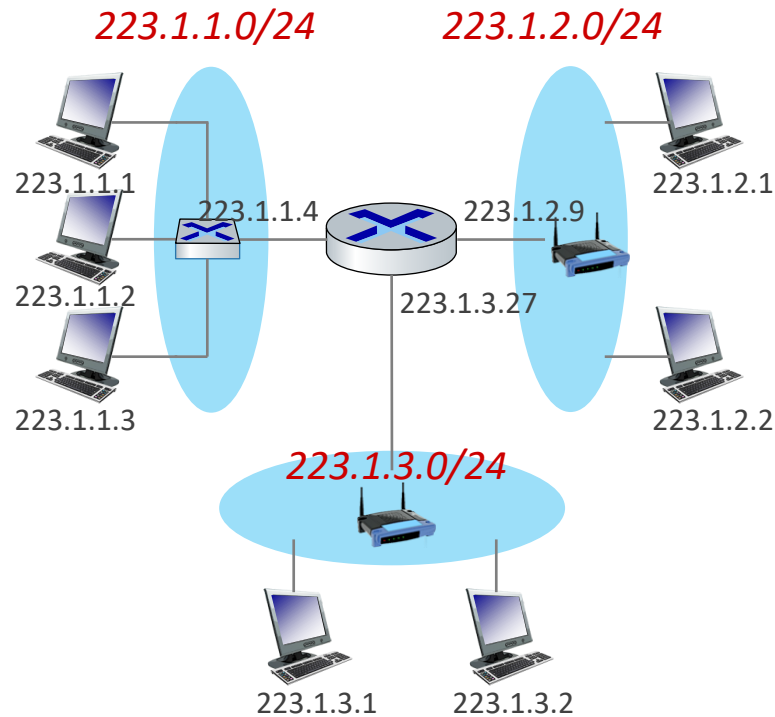
# 数据平面讲解内容

---

- 网络层概述与功能
- 路由器体系结构
- IP数据报格式与分片
- IP地址
- 转发
- 划分地址块与地址聚合
- DHCP协议
- 网络地址转换NAT
- IPv6

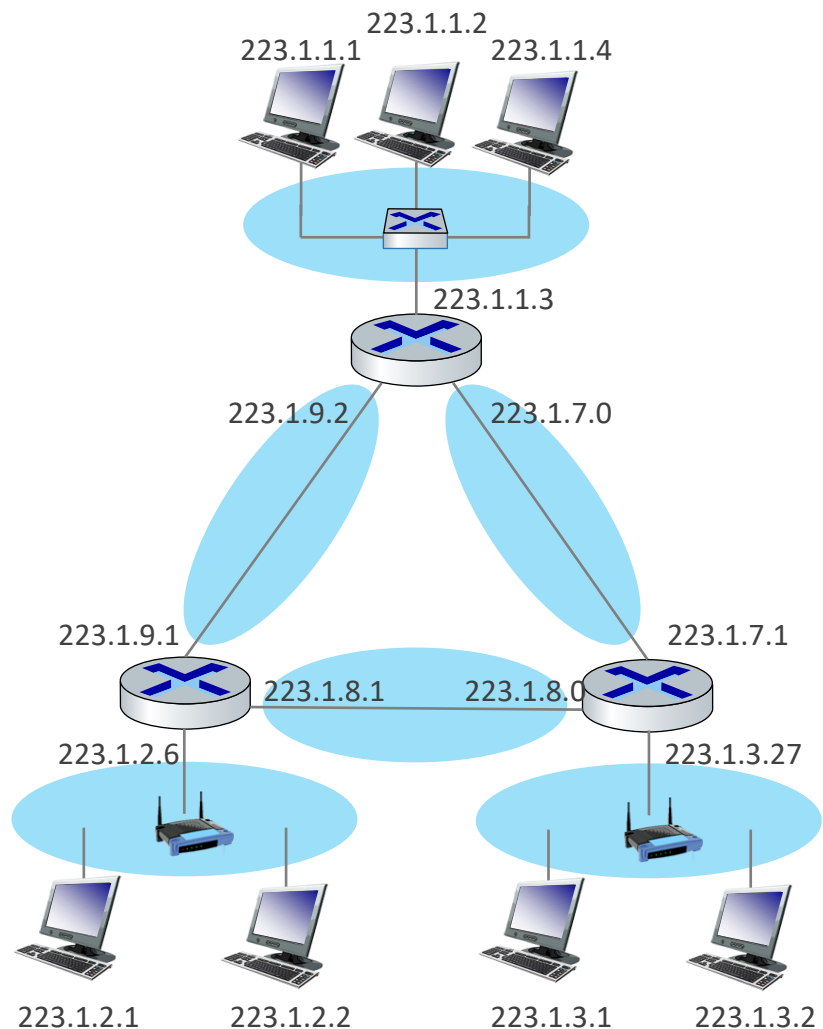
# IP地址与子网

- 该网络中含有多少个子网？
- 物理上不通过三层设备连接的接口形成子网
- 每个子网合理的网络地址分别是什么？



# 子网

- 该网络中含有多少个子网？
- 物理上不通过三层设备连接的接口形成子网





## 第四章知识点汇总

---

- 了解IP地址和网络地址
- 了解子网
- 了解CIDR
- 了解以前的A/B/C分类编址方法，以及其缺点



# 数据平面讲解内容

---

- 网络层概述与功能
- 路由器体系结构
- IP数据报格式与分片
- IP地址
- 转发
- 划分地址块与地址聚合
- DHCP协议
- 网络地址转换NAT
- IPv6





# 转发

- 查询转发表，用目的IP地址匹配条目
  - 11001000 00010111 00010110 10100001 转发接口：0
  - 11001000 00010111 00011000 10101010 转发接口：1
- 最长前缀匹配
  - 转发的时候，利用最长前缀匹配目的地址

网络地址	链路接口
11001000 00010111 00010000 00000000/21	0
11001000 00010111 00011000 00000000/24	1
11001000 00010111 00011000 00000000/21	2
其他	3



# 转发

- 查询转发表，用目的IP地址匹配条目
  - 200.23.22.161 转发接口？
  - 200.23.24.170 转发接口？

网络地址	链路接口
200.23.16.0/21	0
200.23.24.0/24	1
200.23.24.0/21	2
其他	3



# 第四章知识点汇总

---

- 理解转发的原理
- 理解最长前缀匹配的原理



# 数据平面讲解内容

---

- 网络层概述与功能
- 路由器体系结构
- IP数据报格式与分片
- IP地址
- 转发
- 划分地址块与地址聚合
- DHCP协议
- 网络地址转换NAT
- IPv6



# 划分地址块

- ISP有一个地址块
- 11001000 00010111 00010000 00000000 200.23.16.0/20
- 有8个机构申请地址块，每个机构需要512个地址
- ISP该如何分配地址块呢？



# 划分地址块

- ISP有一个地址块

- 11001000 00010111 00010000 00000000 200.23.16.0/20

- 机构地址块：

- 11001000 00010111 00010000 00000000 200.23.16.0/23

- 11001000 00010111 00010001 11111111

- 11001000 00010111 00010010 00000000 200.23.18.0/23

- 11001000 00010111 00010011 11111111

- 11001000 00010111 00010100 00000000 200.23.20.0/23

- .....

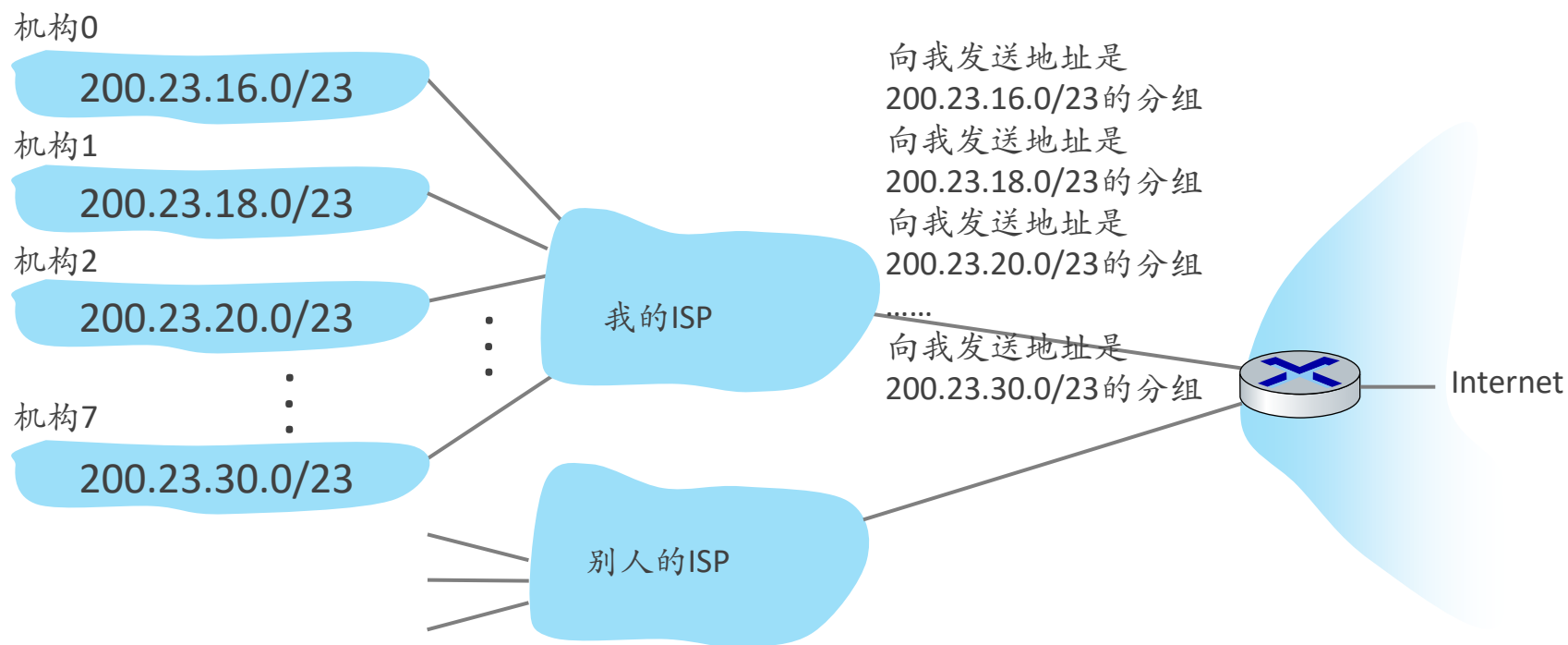
- 11001000 00010111 00011110 00000000 200.23.30.0/23



# 划分地址块

- ISP有一个地址块
- 11001000 00010111 00010000 00000000 200.23.16.0/20
- 机构地址块：
- 11001000 00010111 00010000 00000000 200.23.16.0/23
- 11001000 00010111 00010010 00000000 200.23.18.0/23
- 11001000 00010111 00010100 00000000 200.23.20.0/23
- .....
- 11001000 00010111 00011110 00000000 200.23.30.0/23

# 路由广播





# 路由广播

机构0

200.23.16.0/23

机构1

200.23.18.0/23

机构2

200.23.20.0/23

⋮

机构7

200.23.30.0/23

我的ISP

别人的ISP

网络地址	接口
200.23.16.0/23	0
200.23.18.0/23	0
200.23.20.0/23	0
200.23.22.0/23	0
200.23.24.0/23	0
200.23.26.0/23	0
200.23.28.0/23	0
200.23.30.0/23	0
其他	3



Internet

# 地址聚合

## ■ 聚合路由信息

- 缩小路由表
- 减少路由信息的广播代价

网络地址	接口
200.23.16.0/20	0
其他	3

机构0

200.23.16.0/23

机构1

200.23.18.0/23

机构2

200.23.20.0/23

⋮

机构7

200.23.30.0/23

我的ISP

向我发送地址是  
200.23.16.0/20的分组

别人的ISP

Internet

# 地址聚合

## ■ 聚合路由信息

网络地址	接口
200.23.16.0/20	0
199.31.0.0/16	2
其他	3

机构0

200.23.16.0/23

机构1

200.23.18.0/23

机构2

200.23.20.0/23

⋮

机构7

200.23.30.0/23

我的ISP

向我发送地址是  
200.23.16.0/20的分组

别人的ISP

向我发送地址是  
199.31.0.0/16的分组



Internet

# 地址聚合

- 聚合路由信息
- 需要更改转发表吗?

网络地址	接口
200.23.16.0/20	0
199.31.0.0/16	2
其他	3

机构0

200.23.16.0/23

机构2

200.23.20.0/23

⋮

机构7

200.23.30.0/23

机构1

200.23.18.0/23

我的ISP

向我发送地址是  
200.23.16.0/20的分组

别人的ISP

向我发送地址是  
199.31.0.0/16的分组

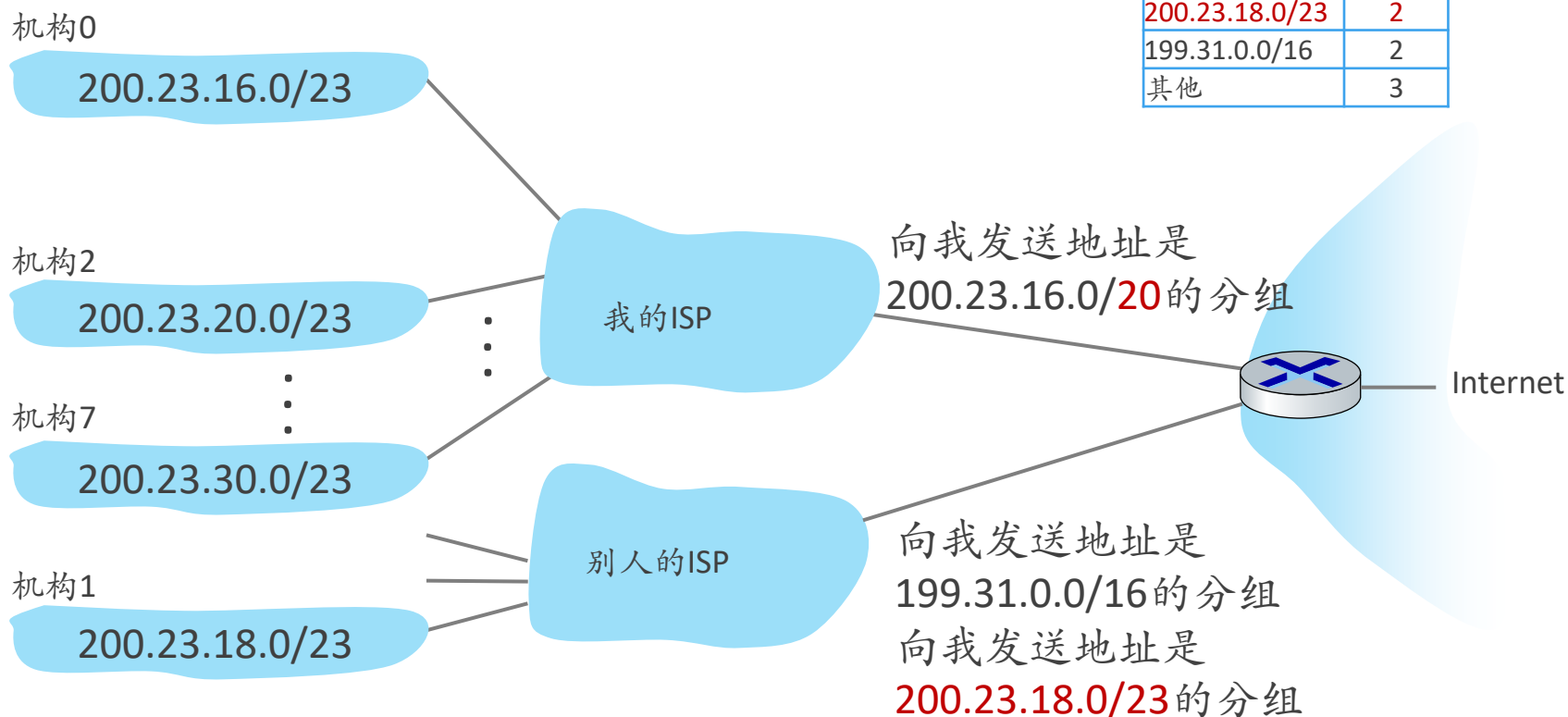


Internet

# 地址聚合

- 聚合路由信息
- 如何更改转发表?

网络地址	接口
200.23.16.0/20	0
200.23.18.0/23	2
199.31.0.0/16	2
其他	3





# 第四章知识点汇总

---

- 掌握划分地址块的方法
- 理解地址聚合



# 数据平面讲解内容

---

- 网络层概述与功能
- 路由器体系结构
- IP数据报格式与分片
- IP地址
- 转发
- 划分地址块与地址聚合
- DHCP协议
- 网络地址转换NAT
- IPv6



# 网络地址转换NAT

## ■ Network Address Translation NAT路由器实现

NAT转换表	
WAN端地址	LAN端地址
138.76.29.7, 5001	10.0.0.1, 3345

### 1. 替换源地址：用端口号区分私有网络内部的设备

(LAN IP地址, 端口号) → (WAN IP地址, 新端口号)

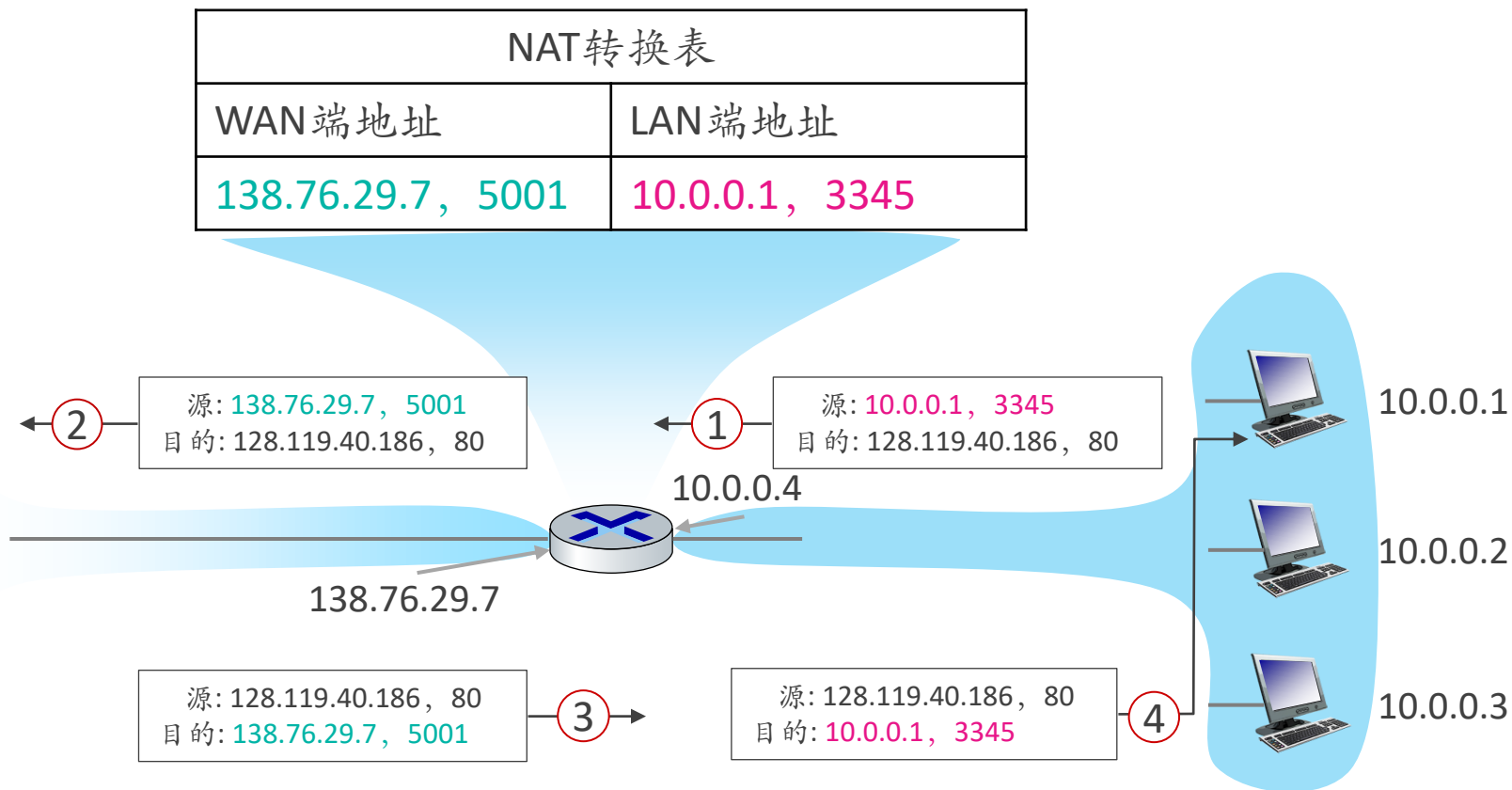
### 2. 在NAT转换表中添加条目

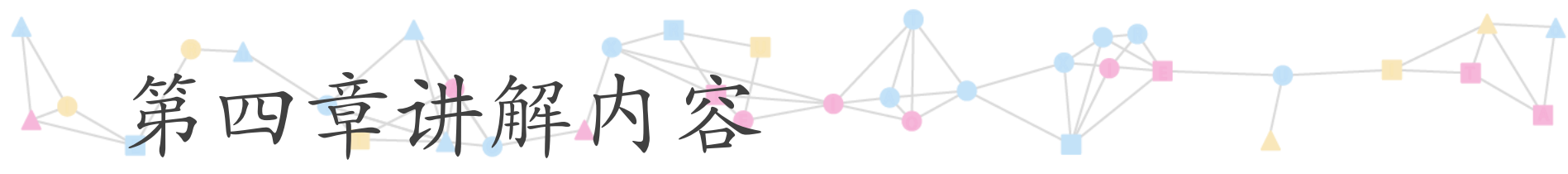
### 3. 替换目的地址

(LAN IP地址, 端口号) ← (WAN IP地址, 新端口号)



# 网络地址转换NAT





1. 网络层概述与数据平面
2. 控制平面



# 控制平面讲解内容

---

- 路由协议的目的与算法分类
- 距离向量路由选择算法
- 可扩展的路由选择
- AS内路由协议：OSPF
- AS间路由协议：BGP
- ICMP协议



# 距离向量路由选择算法

每个节点独立运行以下程序

等待本地链路变化或收到邻居的DV估计

利用B-F公式重新计算DV估计  
$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\}, \forall y \in N$$

如果DV发生了变化, 通知邻居("传八卦")



# 距离向量路由选择算法

- 分布式的
  - 每个节点独立运行自己的DV算法，并与邻居节点交换DV信息
- 迭代的
  - 每个节点从邻居节点接收信息，执行计算，然后再将结果分发给邻居，这一过程重复执行很多次
- 异步的
  - 不需要所有节点步调一致



# 第四章知识点汇总

---

- 理解DV算法的运行原理
- 理解DV算法的特点



# 控制平面讲解内容

---

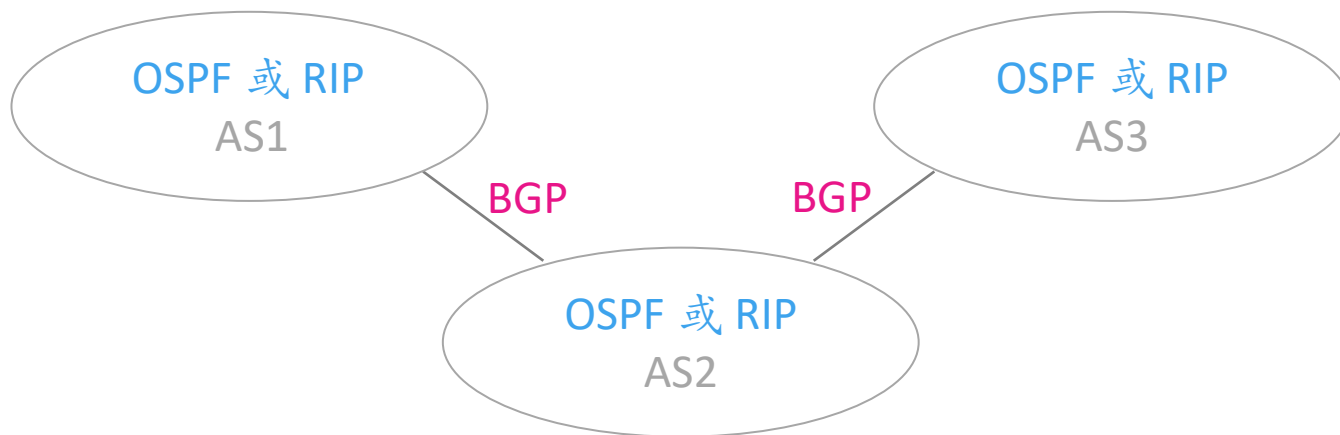
- 路由协议的目的与算法分类
- 距离向量路由选择算法
- 可扩展的路由选择
- AS内路由协议：OSPF
- AS间路由协议：BGP
- ICMP协议



# 可扩展的路由

## ■ AS内路由选择协议

- 不同的AS可以选择不同的域内路由协议
- 例如：OSPF，RIP
  - OSPF：Open Shortest Path First (链路状态路由算法)
  - RIP：Routing Information Protocol (距离向量路由算法)



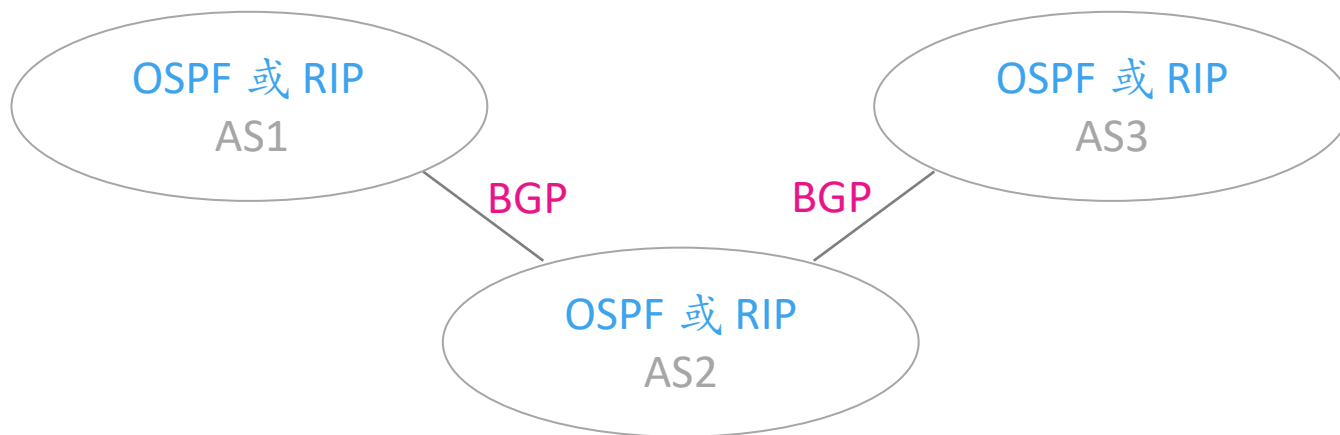




# 可扩展的路由

## ■ AS间路由选择协议

- 当源和主机位于不同的AS时，需要AS间路由协议
- Internet中，所有的AS运行相同的AS间路由协议：边界网关协议（Border Gateway Protocol，BGP）

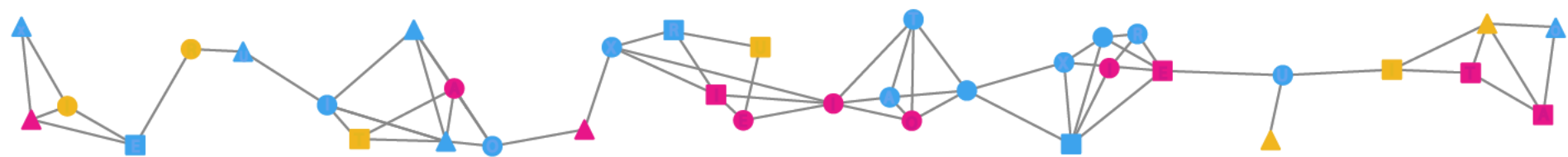




# 第四章知识点汇总

---

- 了解AS内路由选择协议
- 了解AS间路由选择协议



# 计算机网络

## 第五章 链路层

---

谢瑞桃

[xie@szu.edu.cn](mailto:xie@szu.edu.cn)

[rtxie.github.io](https://github.com/rtxie)

计算机与软件学院

深圳大学



# 第五章讲解内容

---

## 1. 链路层概述/服务/实现

## 2. 差错检测

- 奇偶校验、循环冗余校验

## 3. 多路访问

- 信道划分、CSMA/CD、二进制指数退避

## 4. MAC地址与地址解析协议

## 5. 局域网技术

- 交换机以太网、VLAN



# 循环冗余校验

- Cyclic Redundancy Check CRC / 多项式编码
- 给定D
- 给定 $r$ ，和一个 $r+1$ 位的生成多项式 $G$
- 求一个 $r$ 位的校验码 $R$ ，使得加上校验位以后的数据  $D * 2^r \oplus R$  能被 $G$ 整除



- 在接收方，如果收到的数据不能被 $G$ 整除，即检测出了差错；否则，没有检测出差错



# 循环冗余校验

- 如何获得R呢?
- $D \cdot 2^r \oplus R = nG$
- $D \cdot 2^r \oplus R \oplus R = nG \oplus R$
- $D \cdot 2^r = nG \oplus R$
- $R = \text{remainder} \frac{D \cdot 2^r}{G}$

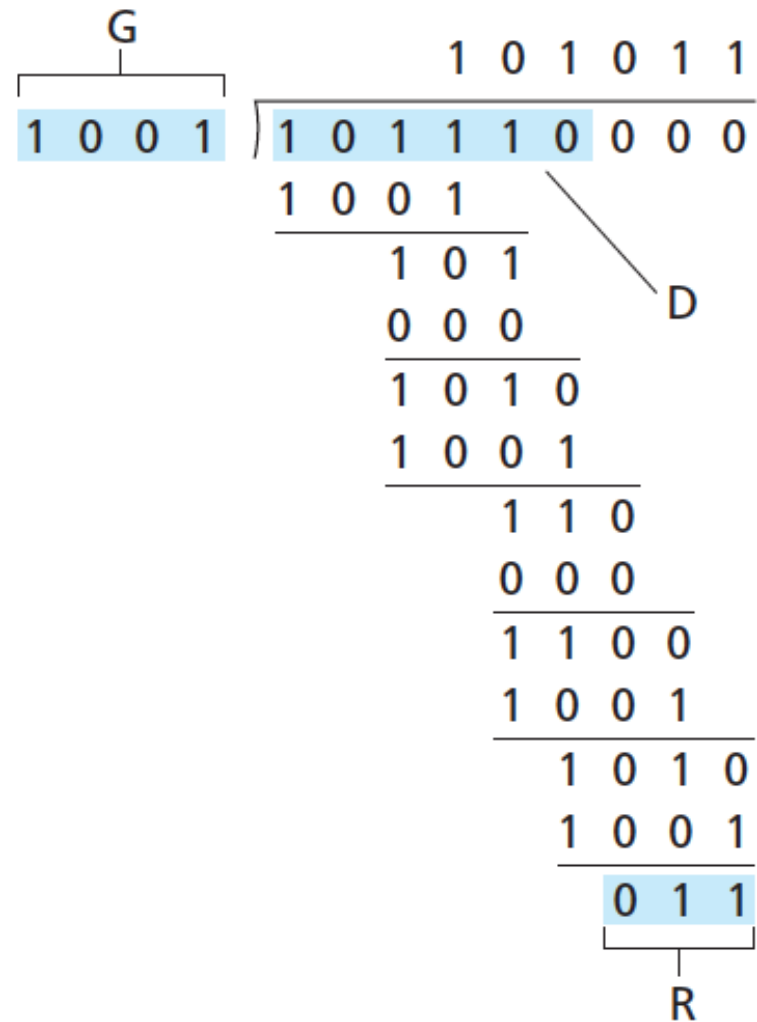
# 循环冗余校验

- $D = 101110$

- $r = 3$

- $G = 1001$

- $R = \text{remainder} \frac{D \cdot 2^r}{G}$





# 第五章知识点汇总

---

- 理解差错检测的原理
- 理解差错检测存在检测不出差错的情况
- 掌握奇偶校验码的计算方法
- 掌握CRC校验码的计算方法





# 第五章讲解内容

---

## 1. 链路层概述/服务/实现

## 2. 差错检测

- 奇偶校验、循环冗余校验

## 3. 多路访问

- 信道划分、CSMA/CD、二进制指数退避

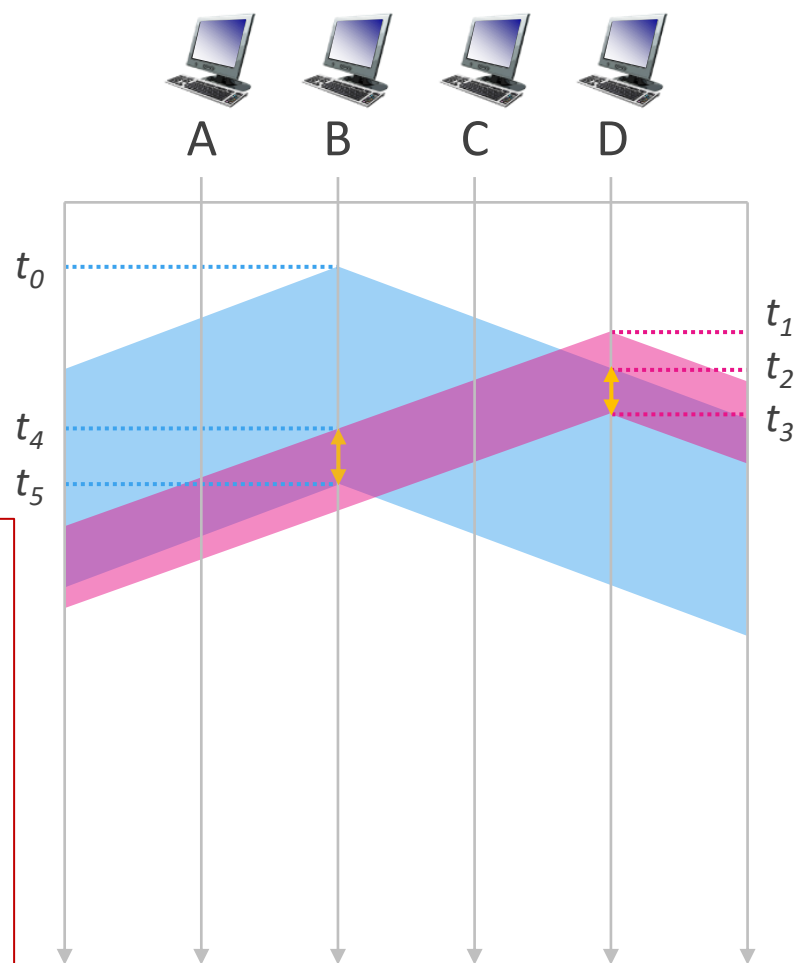
## 4. MAC地址与地址解析协议

## 5. 局域网技术

- 交换机以太网、VLAN

# 碰撞检测 Collision Detection

- 检测到碰撞时，停止帧传输，减小信道的浪费
- 碰撞检测
  - 测量信号强度发现





# CSMA/CD

- 碰撞中止传输以后，节点进入二进制指数退避过程
- 在该帧经历了一连串 $m$ 次碰撞后，节点随机地从 $\{0, 1, 2, \dots, 2^m - 1\}$ 中选择一个 $K$ 值，节点等待 $K \cdot 512$ 比特时间之后，重新尝试传输
- 设计的原理是：碰撞少，间隔时间短；碰撞多，间隔时间长



# CSMA/CD

1. NIC收到来自网络层的数据报，封装成帧
2. 如果NIC检测到信道空闲，则发送帧；否则，等待直到信道空闲，然后发送（载波监听CS）
3. 如果NIC在发送过程中没有检测到冲突，则传输成功；否则，停止传输（冲突检测CD）
4. 中止以后，NIC进入二进制指数退避过程
  - 在该帧经历了一连串 $m$ 次碰撞后，节点随机地从 $\{0, 1, 2, \dots, 2^m - 1\}$ 中选择一个 $K$ 值，NIC等待 $K \cdot 512$ 比特时间之后，回到步骤2



# 第五章知识点汇总

---

- 理解载波监听多路访问/冲突检测的原理
- 理解二进制指数退避方法的原理



# 第五章讲解内容

---

1. 链路层概述/服务/实现
2. 差错检测
  - 奇偶校验、循环冗余校验
3. 多路访问
  - 信道划分、CSMA/CD、二进制指数退避
4. MAC地址与地址解析协议
5. 局域网技术
  - 交换机以太网、VLAN



# 链路层地址

## ■ IP地址

- 接口的网络层地址，用于网络层转发分组
- 自动配置或软件获取
- 32位，4字节
- 192.168.2.36
- 类比工号/学号

## ■ MAC地址/物理地址

- 接口的链路层地址，用于链路层寻址
- NIC出厂时写在只读存储器ROM里，通常无法更改
- 48位，6字节
- 1A-2F-BB-76-09-AD
- 类比身份证号



# 地址解析协议

- 在一个局域网内，源主机只知道目的主机的IP地址，但是如何知道它的MAC地址呢？
- Address Resolution Protocol (ARP[RFC 826])
- 每个IP节点都有ARP表
  - 保存所属局域网内的部分节点的IP/MAC地址映射
  - < IP 地址                      MAC 地址                      TTL >
  - TTL (Time To Live): 删除某个映射的时间





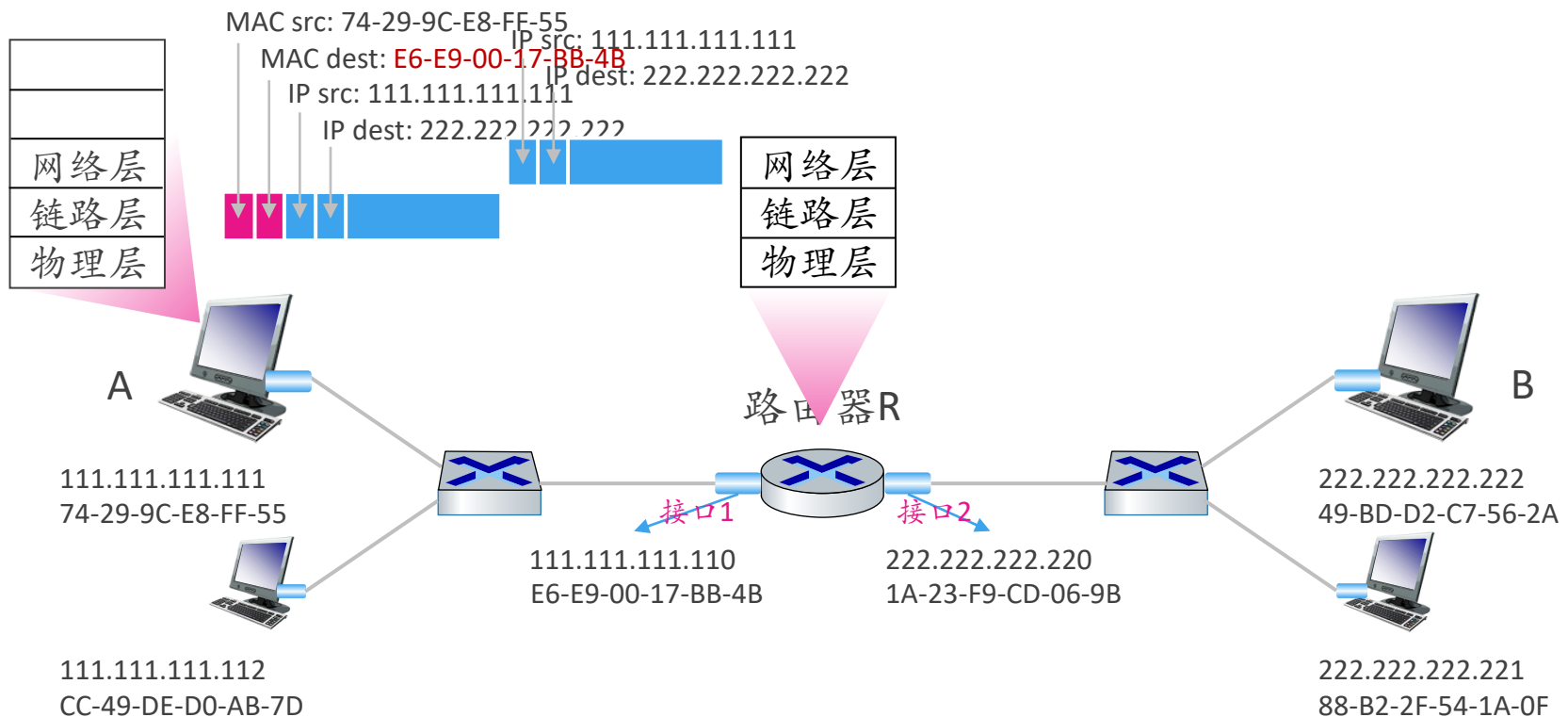
# 第五章知识点汇总

---

- 了解MAC地址
- 了解MAC地址与IP地址间的区别
- 理解地址解析协议

# 发送数据报到子网之外

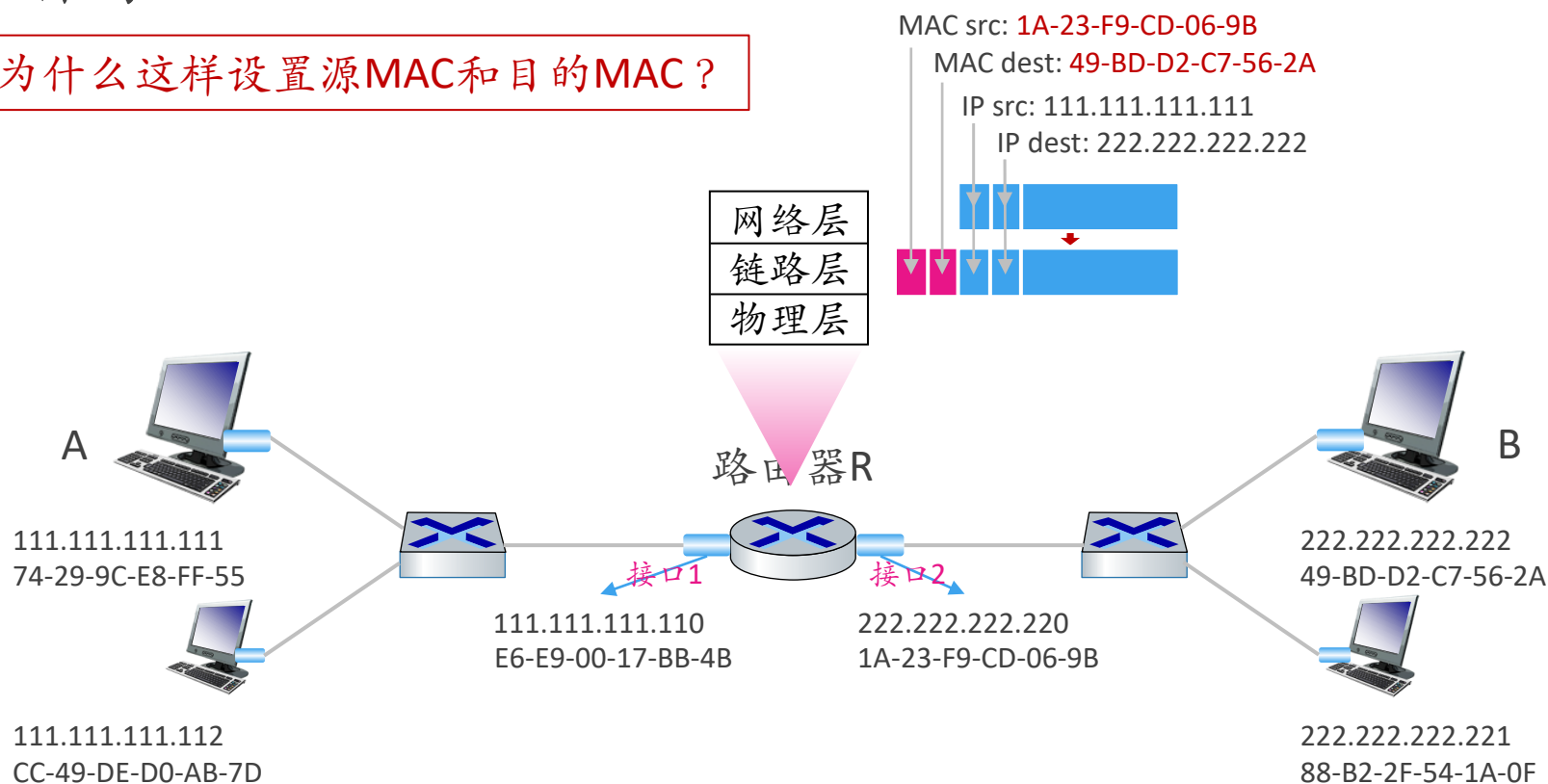
- 帧由A发往R
- R的接口1收到帧后，解封装，交付给网络层



# 发送数据报到子网之外

- R将该IP数据报转发到接口2
- 在接口2的NIC处，重新封装新的链路层帧，注意源MAC和目的MAC

为什么这样设置源MAC和目的MAC？





# 第五章知识点汇总

---

- 理解MAC地址的用途
- 理解网络之间发送IP数据报时，分组封装/解封装的过程
- 理解DHCP所起的作用
- 理解ARP所起的作用



# 第五章讲解内容

---

## 1. 链路层概述/服务/实现

## 2. 差错检测

- 奇偶校验、循环冗余校验

## 3. 多路访问

- 信道划分、CSMA/CD、二进制指数退避

## 4. MAC地址与地址解析协议

## 5. 局域网技术

- 交换机以太网、VLAN

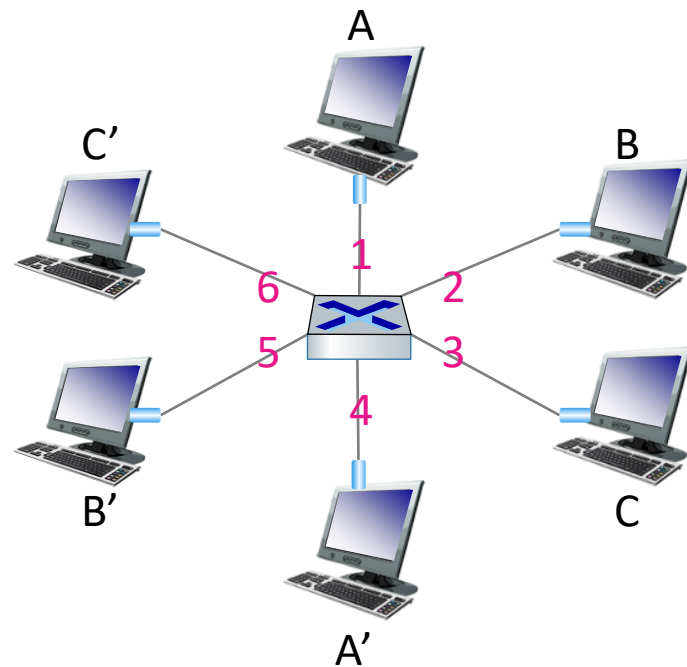


# 交换机如何工作?

- 链路层设备
  - 存储转发以太网帧
  - 检查帧的目的MAC地址，以转发帧
- 在主机和路由器眼里是透明的
  - 主机/路由器接口向另一个主机/路由器接口发送帧
- 即插即用
  - 不需要配置

# 交换机

- 主机直接与交换机接口相连
- 交换机的每一个接口都运行以太网协议，没有碰撞，全双工
- 交换：A向A'，B向B'可以同时传输，不引起冲突
- 交换机接口缓存帧

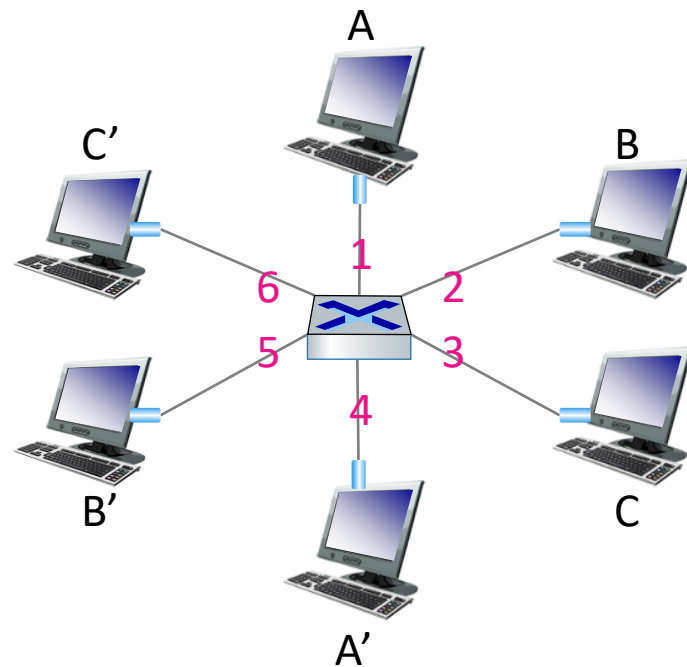


六接口交换机

# 交换机

- 交换机如何知道每个接口分别对应**哪些**MAC地址呢?
- 自学一个交换表(类似于路由转发表)

<MAC 地址, 接口, 时间戳>

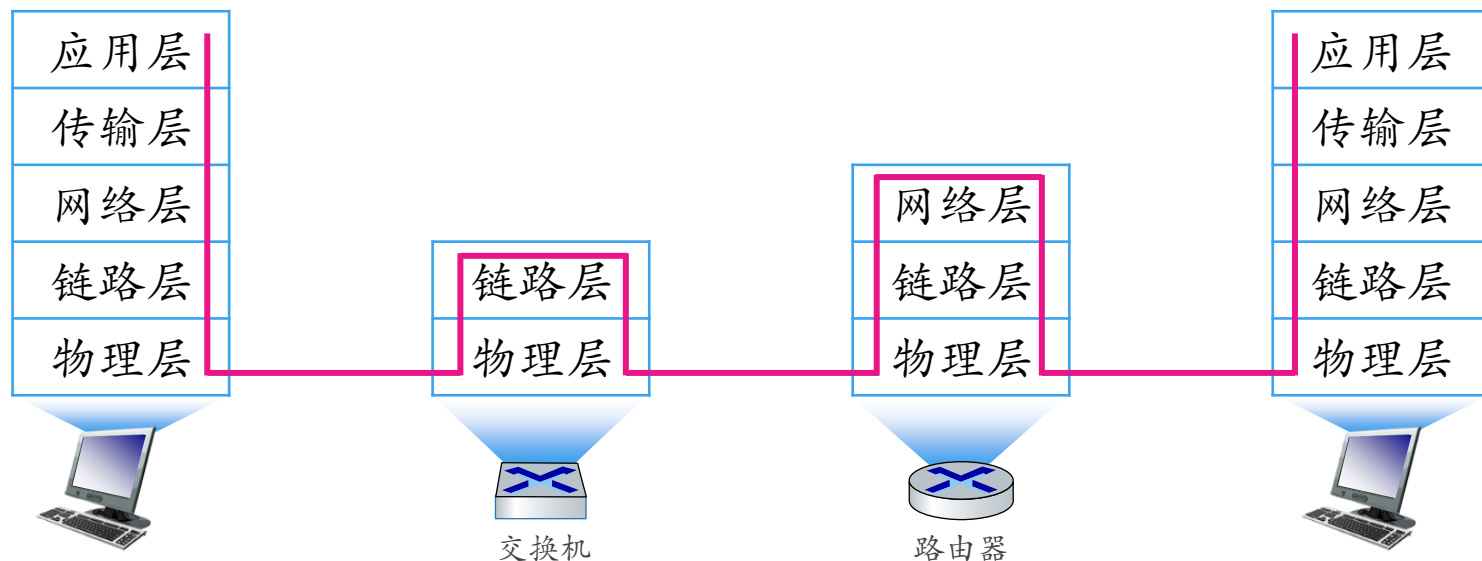


六接口交换机



# 交换机和路由器

- 都是存储转发设备，但有区别
  - **路由器**：网络层设备，查看网络层首部
  - **交换机**：链路层设备，查看链路层首部
- 都有转发表，但有区别
  - **路由器**：使用路由选择算法计算
  - **交换机**：使用自学、广播得出





# 第五章知识点汇总

---

- 理解交换机转发帧的原理
- 理解交换机自学转发表的原理
- 理解交换机与路由器的区别

感谢聆听，后会有期