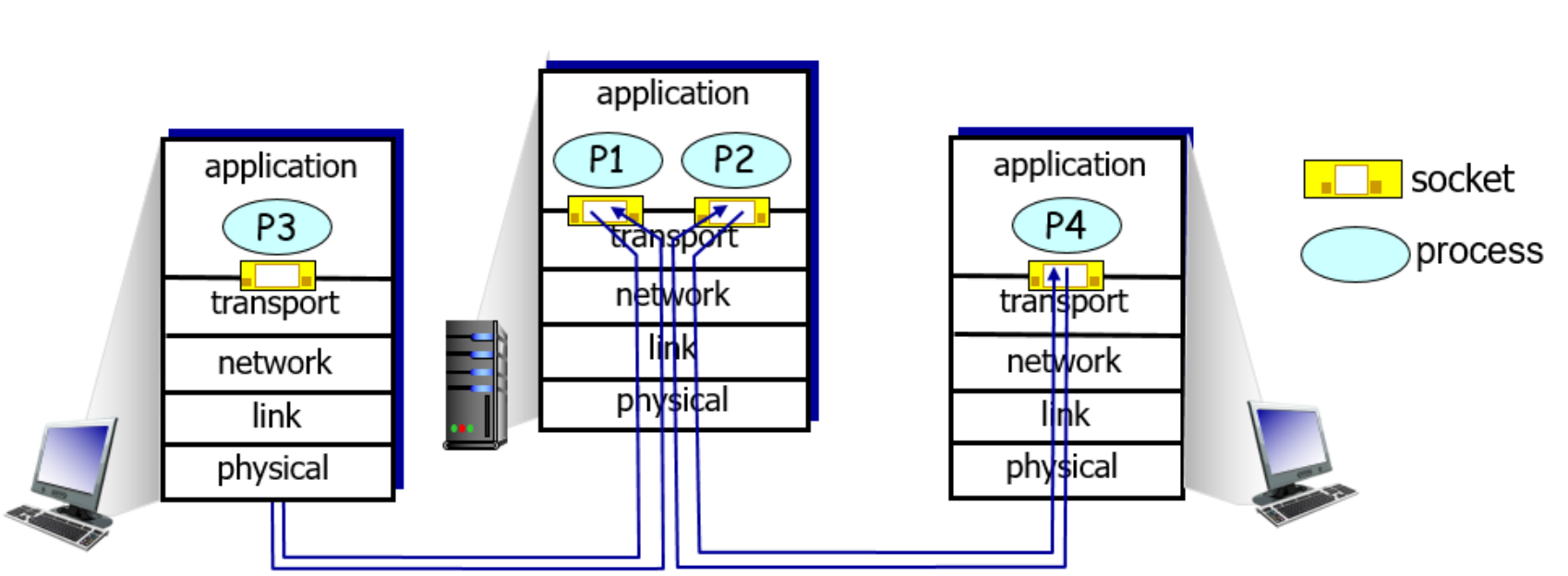
第七章 传输层

合肥工业大学
计算机与信息学院

网络层：不同主机之间的通信

传输层的功能：不同主机上的**应用进程**之间的通信



传输层的功能

1）不同进程之间的通信

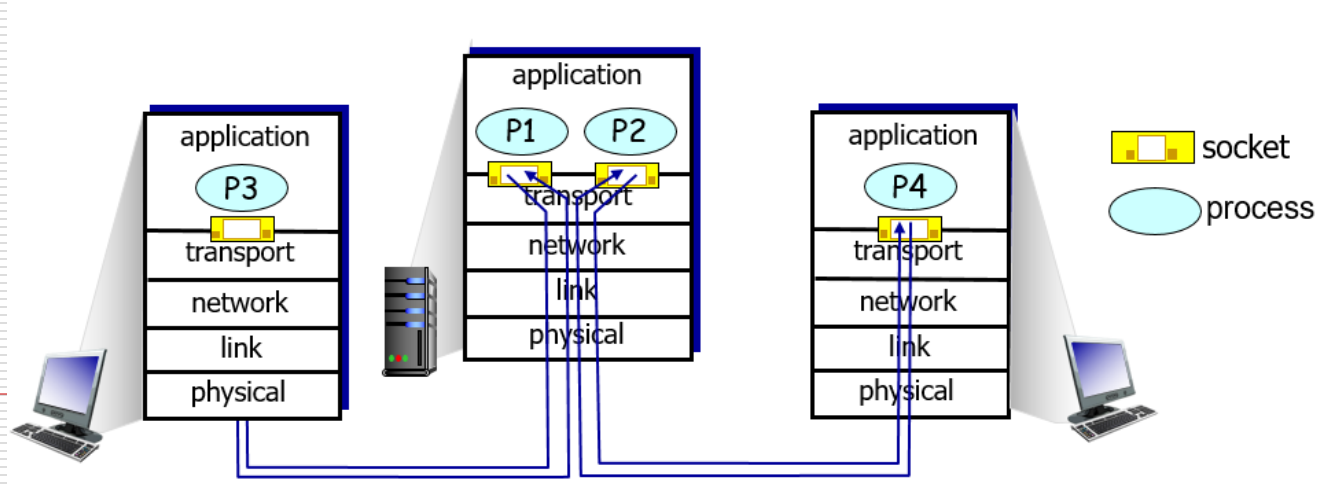
2）可靠传输

3）流量控制

4）拥塞控制

不同进程之间的通信

- 对进程进行标识
- 规定应用进程收发数据的地址空间，对其进行编号（端口号）



- 两个进程之间进行通信

<IP address，Port number>唯一标识了一台主机上的一个进程

端口号

- 16位（0~65535）
  - 服务端使用的端口号
    - 熟知端口号：（ 0~1023）
    - 登记端口号：（1024~49151）
  - 客户端使用的端口号
    - 临时端口号：(49152~65535)

传输层的功能

1）不同进程之间的通信

2）可靠传输

3）流量控制

4）拥塞控制

传输层的功能

1）不同进程之间的通信

2）可靠传输

3）流量控制

4）拥塞控制

TCP/IP体系结构

- 面向连接的服务

——TCP：传输控制协议

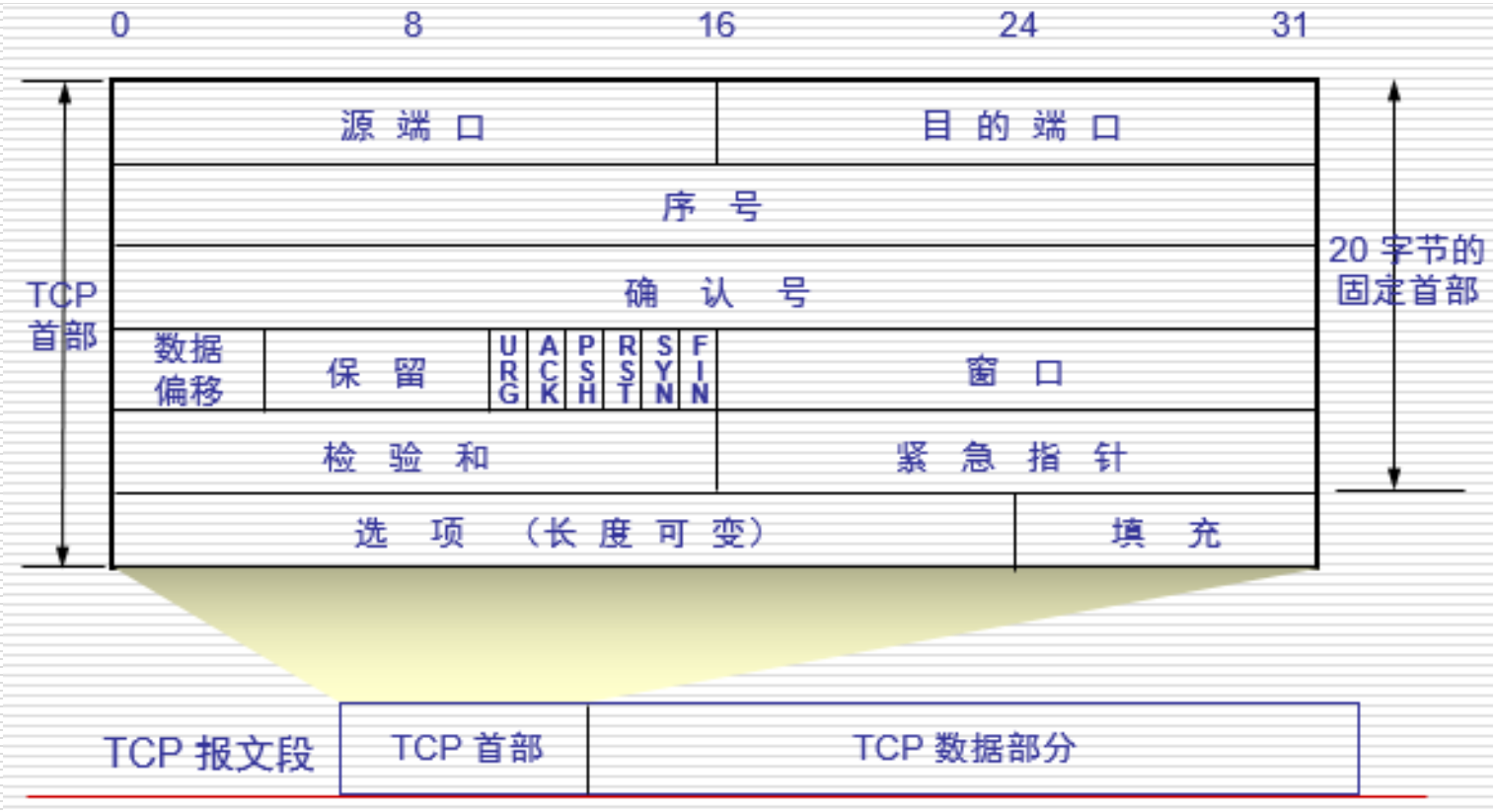
- 无连接的服务

——UDP：用户数据报协议

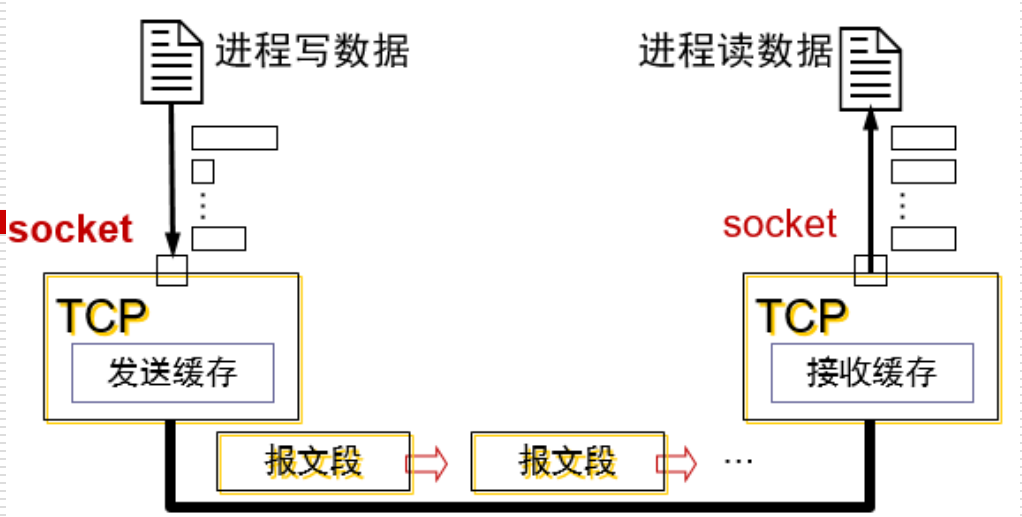
TCP

1. TCP报文格式
2. TCP可靠传输
3. TCP连接管理
4. TCP流量控制
5. TCP拥塞控制

1、TCP报文格式

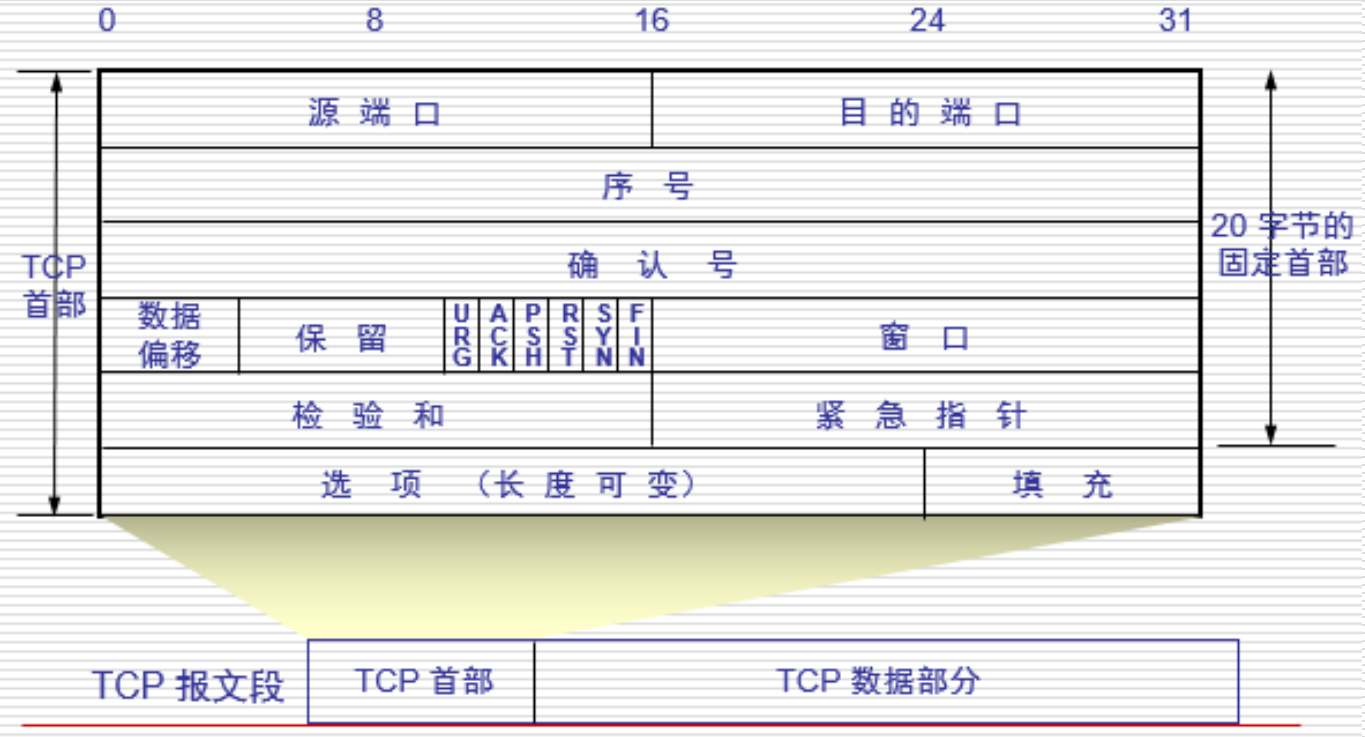

0
8
16
24
31
源 端 口
目 的 端 口
序 号
确 认 号
数据偏移
保 留
URG
ACK
PSH
RST
SYN
FIN
窗 口
检 验 和
紧 急 指 针
选 项 （长 度 可 变）
填 充
TCP 首部
20 字节的固定首部
TCP 报文段
TCP 首部
TCP 数据部分

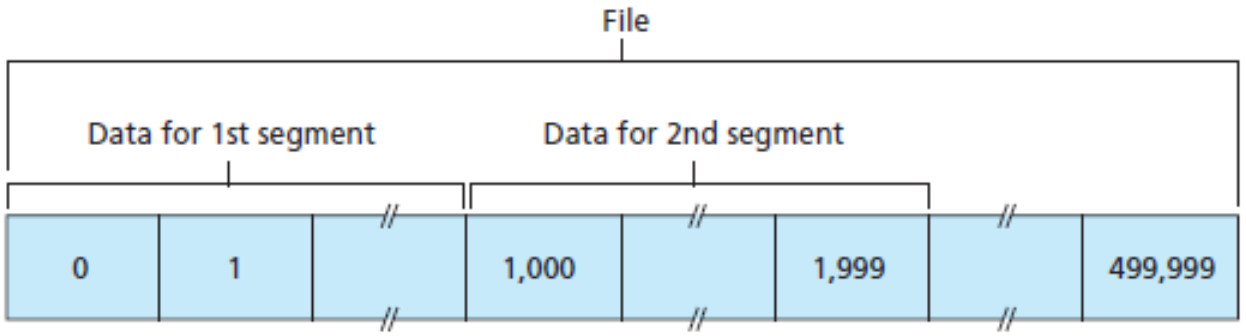
2、可靠传输



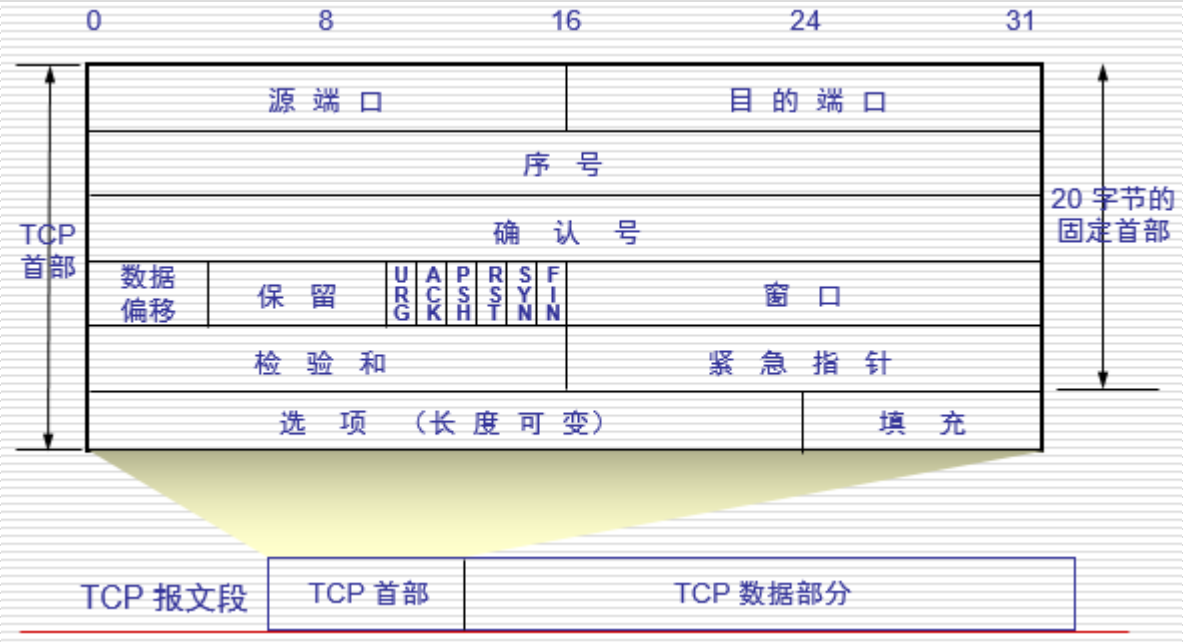
- TCP把上层交付的数据看成字节流，不保证数据块之间的对应关系
- 接收方收到的字节流：无差错、不重复、顺序一致

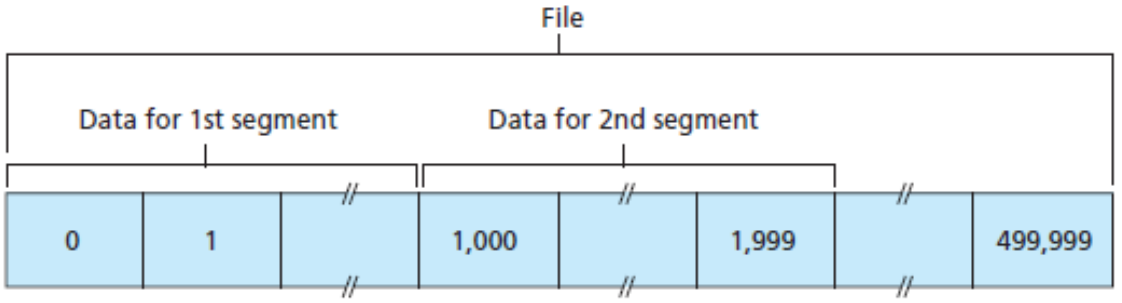
- 停等协议、回退N（GBN），选择重传（SR）和滑动窗口





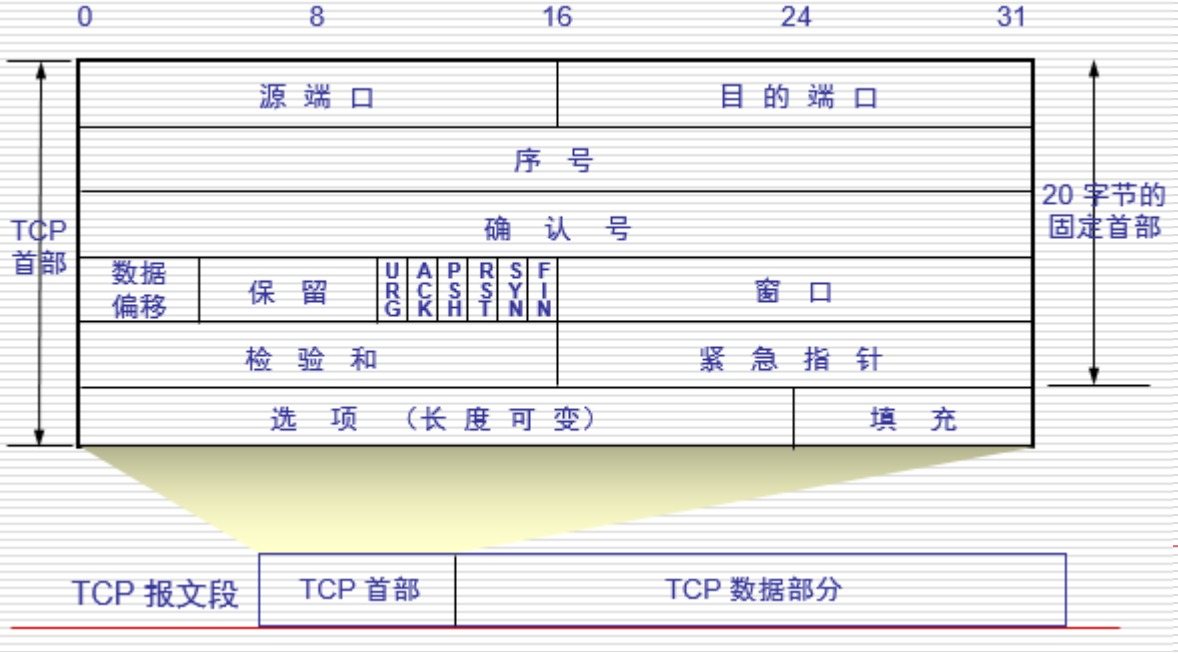
- 序号：每个报文段中首字节的编号





确认号

- 接收方期望收到的下一个字节的编号，同时对该编号之前数据的确认



单选题 2分

设置

A →B4个报文段，B连续收到1st报文段0~999，2nd报文段1000~1999，4th报文段3000~3999，采用累积确认，则B返回确认号

A 1000

B 2000

C 3000

D 4000

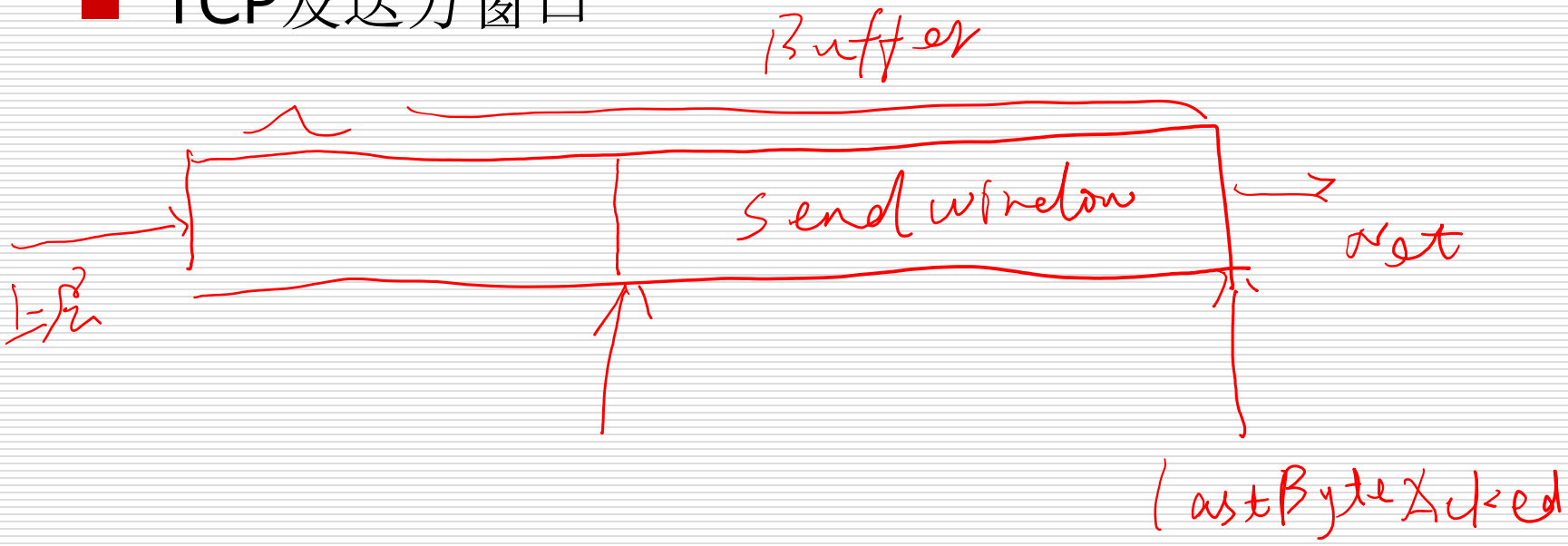
提交

TCP的可靠传输

- 按序接收，累积确认
- GBN？
- 许多TCP实现，将失序的报文段缓存起来（选择确认：SACK选项字段）

滑动窗口

- TCP发送方窗口

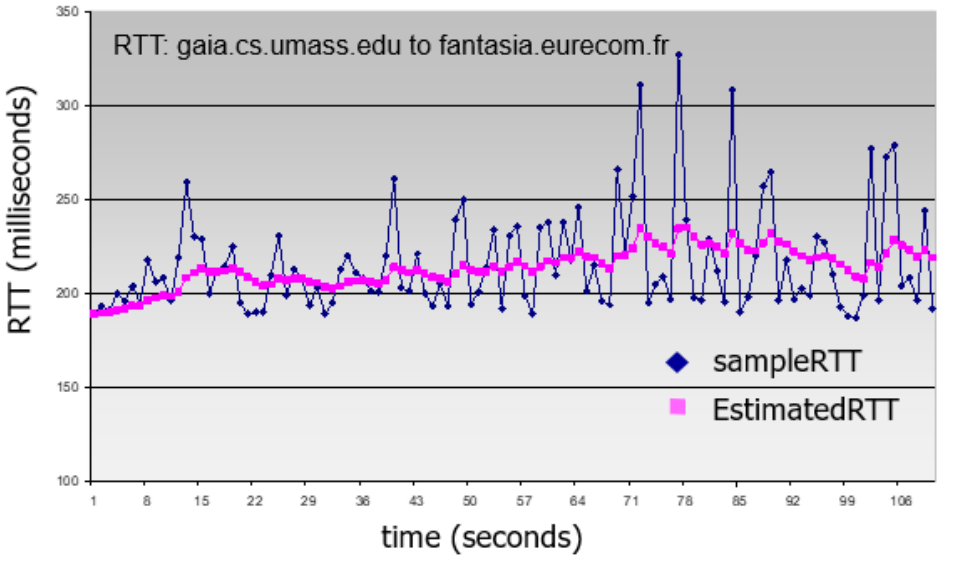


- 可靠传输
  - 接收方返回确认
  - 发送方在规定时间内没有收到对报文段的确认

——超时重传

- 超时时间的设置：比RTT稍微长一点，长多少？
- 设置多少个计时器？

估计RTT

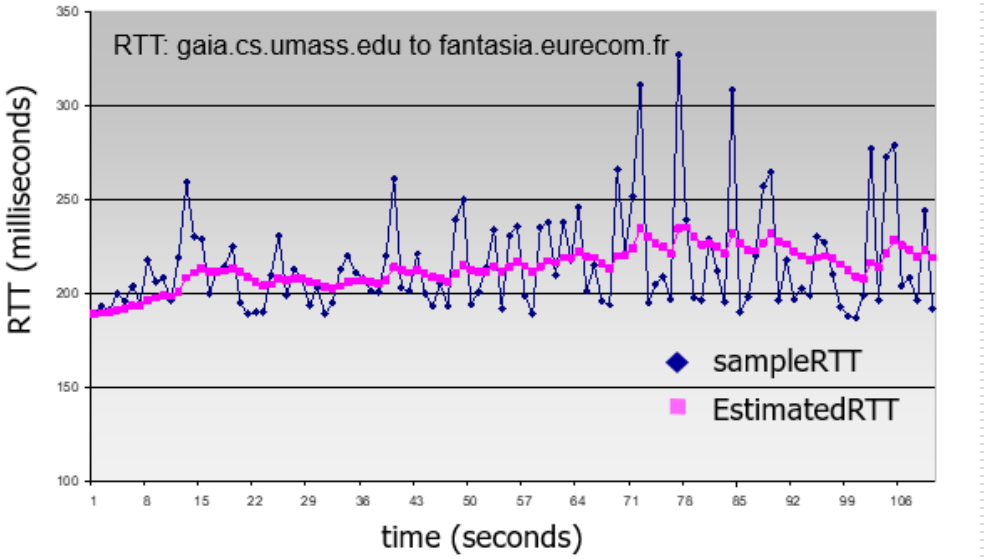


在任意时刻，采样一个报文段的RTT。作为sampleRTT

EstimatedRTT = (1- α)*EstimatedRTT + α*SampleRTT

- a的推荐值：0.125

超时时间



$$DevRTT = (1-\beta)*DevRTT + \beta*|SampleRTT-EstimatedRTT|$$

(typically, $\beta = 0.25$)

$$TimeoutInterval = EstimatedRTT + 4*DevRTT$$

estimated RTT

"safety margin"

传输层

- 实现主机进程之间的通信：端口

进程标识：<IP：port>

- 传输层两个协议
  - TCP：

    ——首部格式

    ——可靠传输（以字节为单位，序号、确认号、滑动窗口，超时重传）
  - UDP：简单

TCP

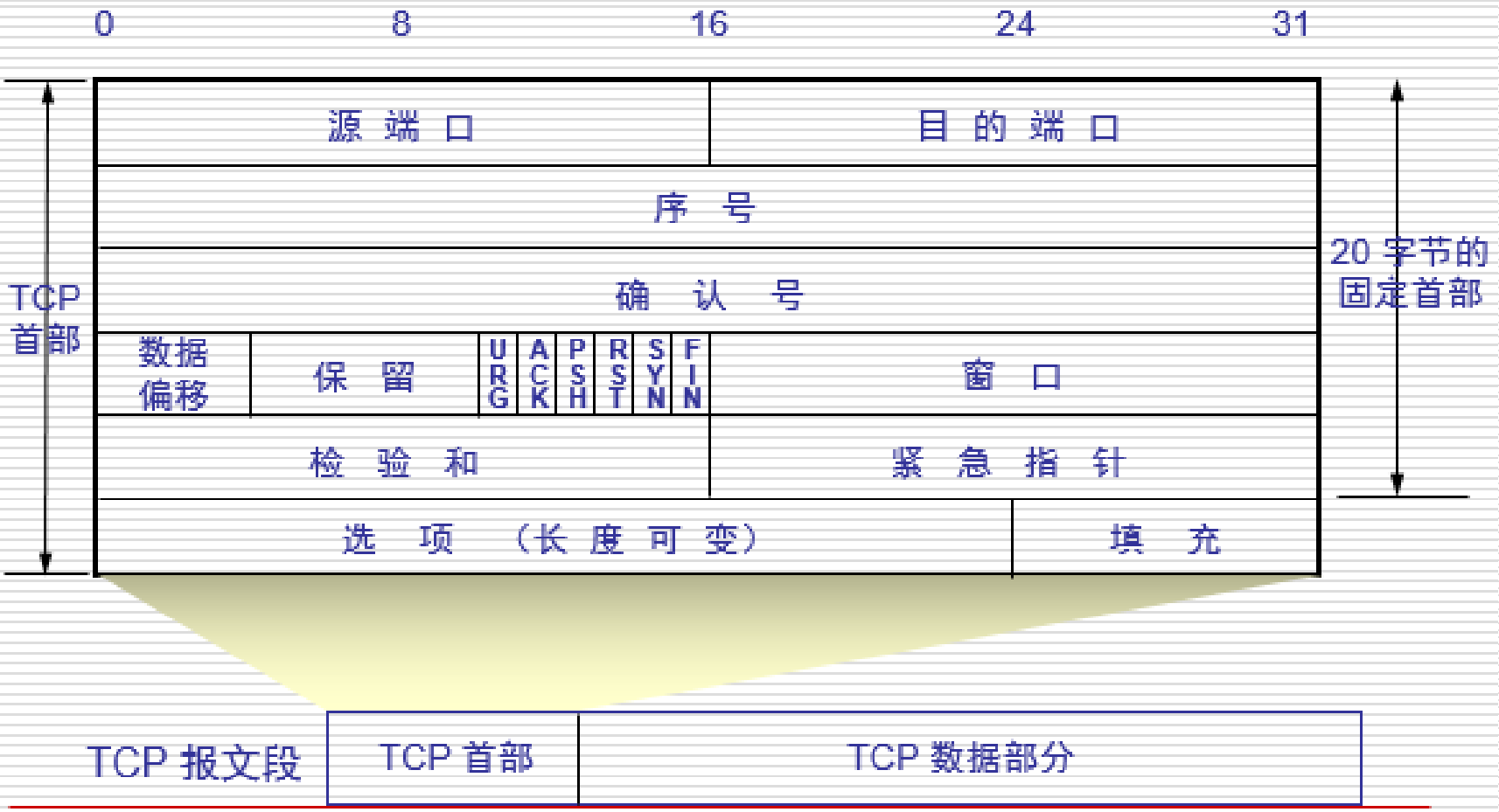
1. TCP报文格式
2. TCP可靠传输
3. TCP连接管理
4. TCP流量控制
5. TCP拥塞控制

3、TCP连接管理

- 采用TCP协议，传输数据前，两个进程必须先建立连接

1. 确定接收方的应用进程已经做好接收准备
2. 初始化与TCP连接相关的状态变量

TCP 连接 ： {(IP1: port1), (IP2: port2)}

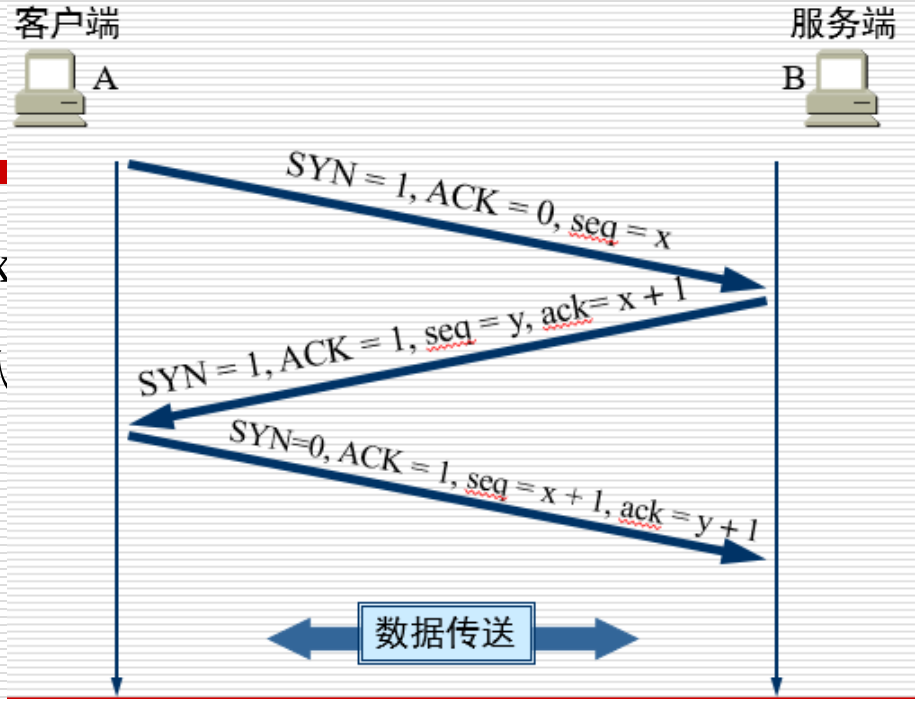

0
8
16
24
31
源 端 口
目 的 端 口
序 号
确 认 号
数据偏移
保 留
URG
ACK
PSH
RST
SYN
FIN
窗 口
检 验 和
紧 急 指 针
选 项 （长 度 可 变）
填 充
TCP 首部
20 字节的固定首部
TCP 报文段
TCP 首部
TCP 数据部分

建立连接

1）A—>B：SYN置1，随机选择序号x

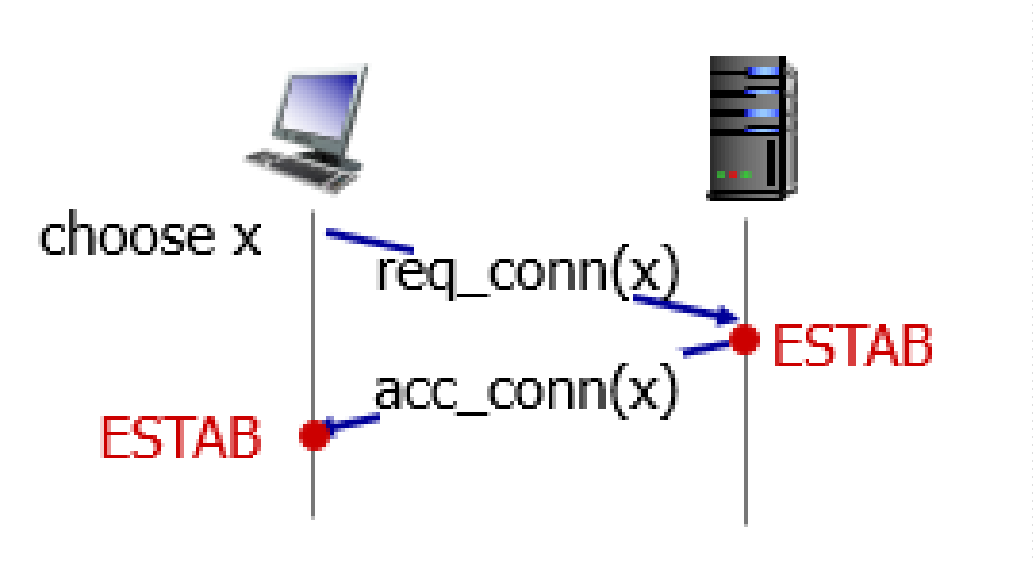
2）B—>A：SYN置1，ACK置1，确认号x+1，序号y

3）A—>B：SYN置0，ACK置1，序号x+1，确认号y+1



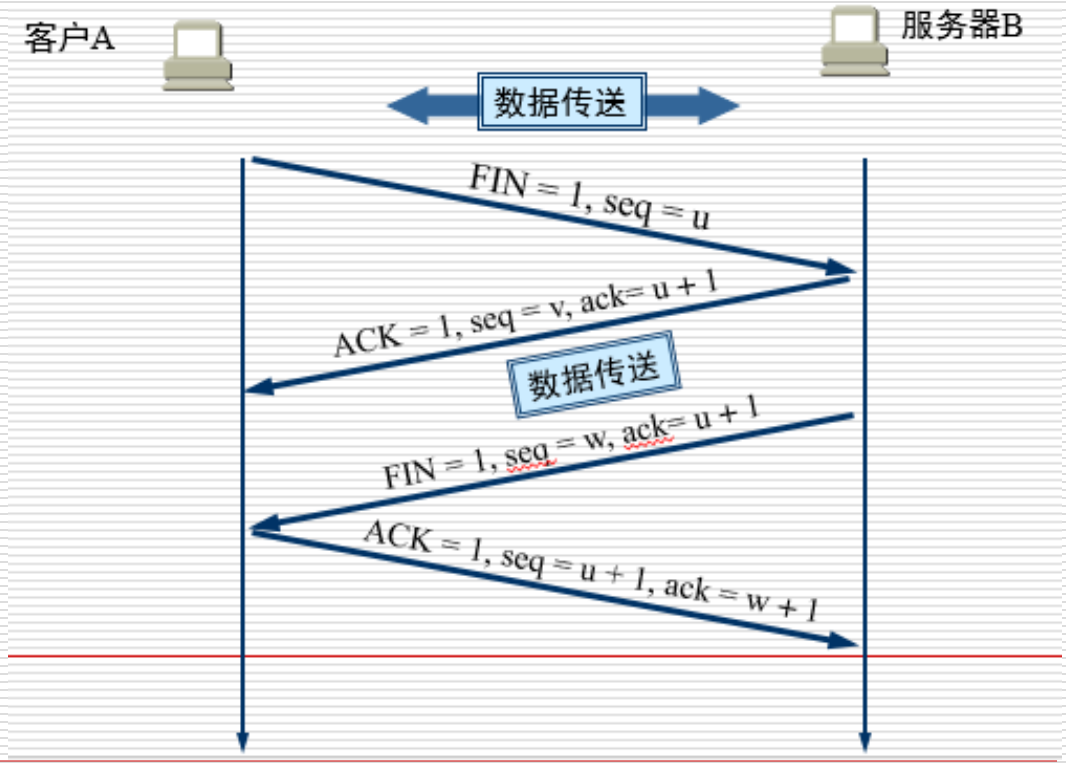
三次握手

- 两次握手？



释放连接

连接结束后，释放主机中的资源：四次握手



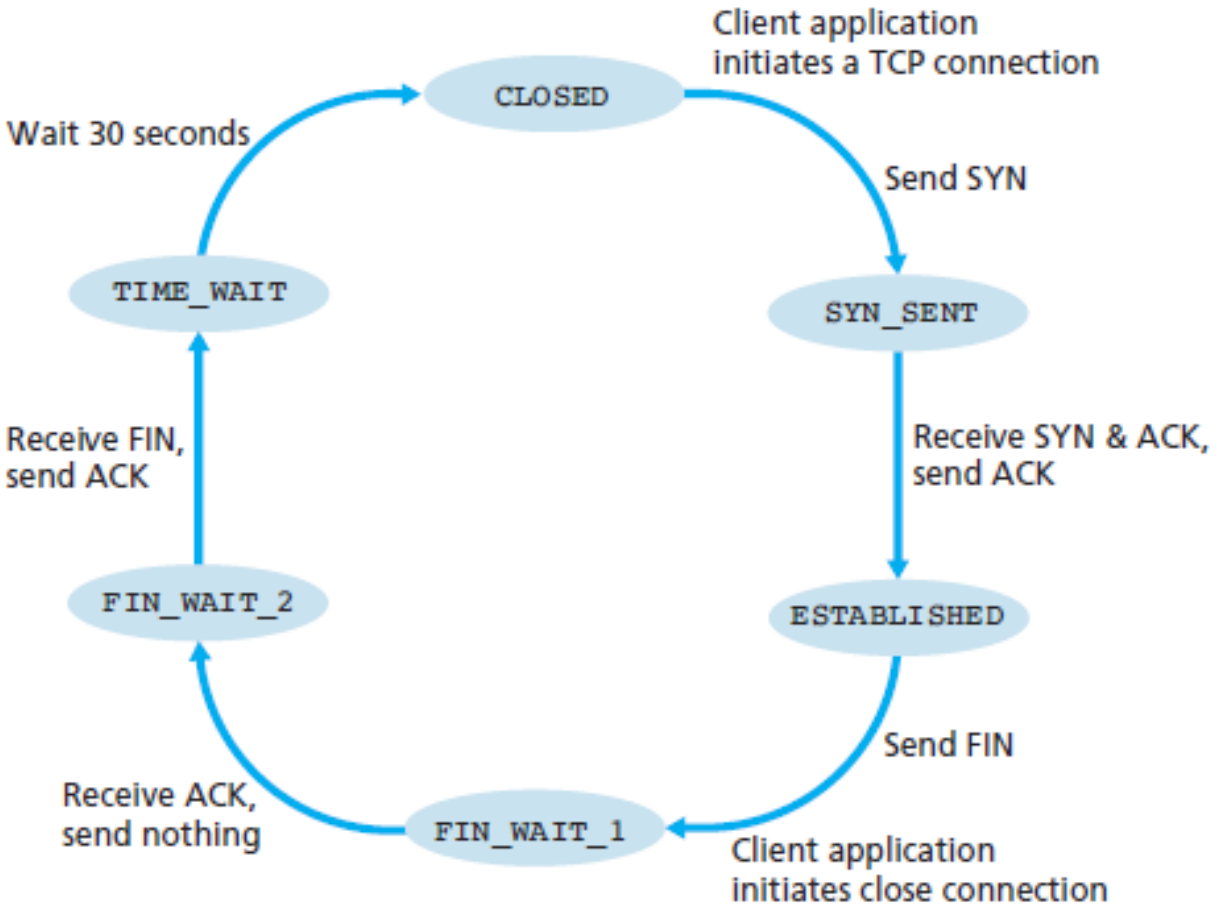


Figure 3.41 ♦ A typical sequence of TCP states visited by a client TCP

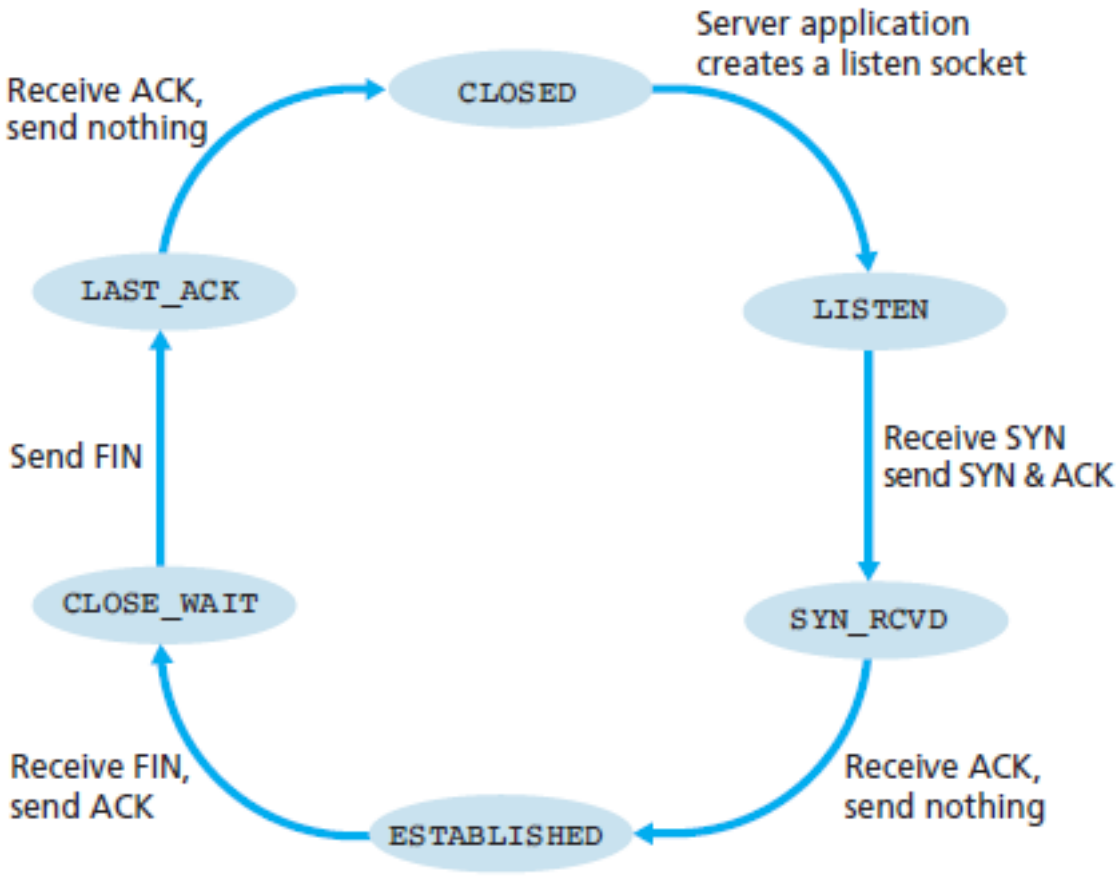
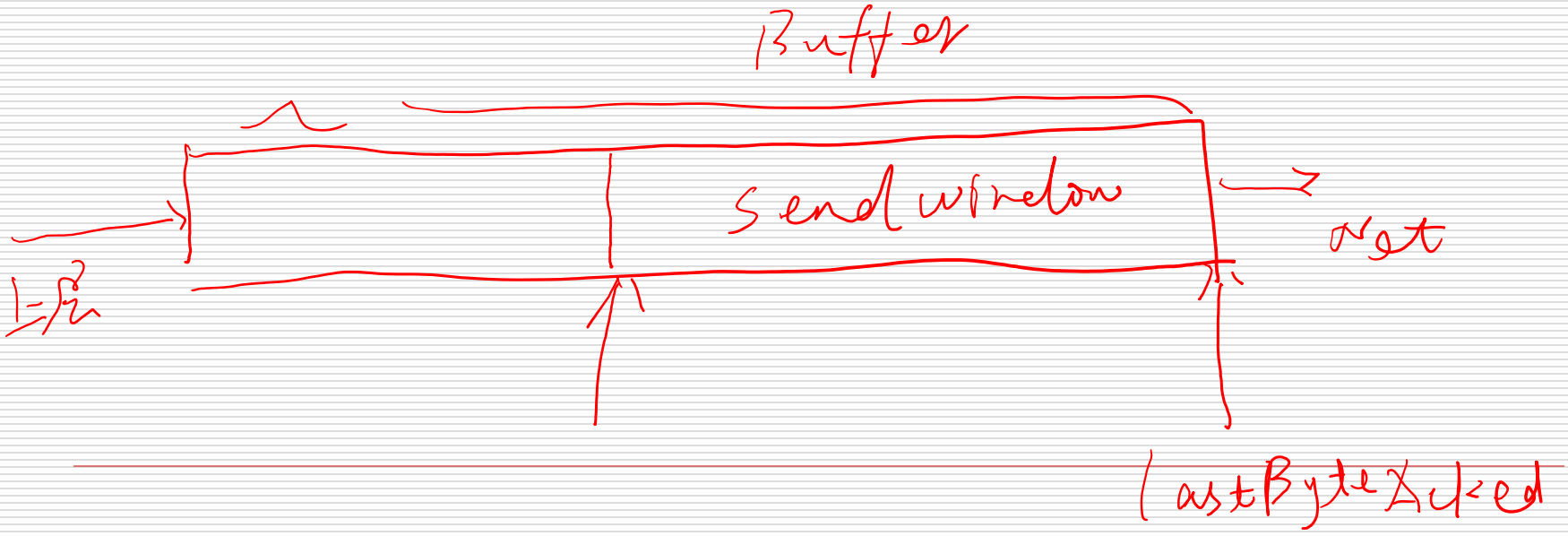


Figure 3.42 ♦ A typical sequence of TCP states visited by a server-side TCP

4、流量控制

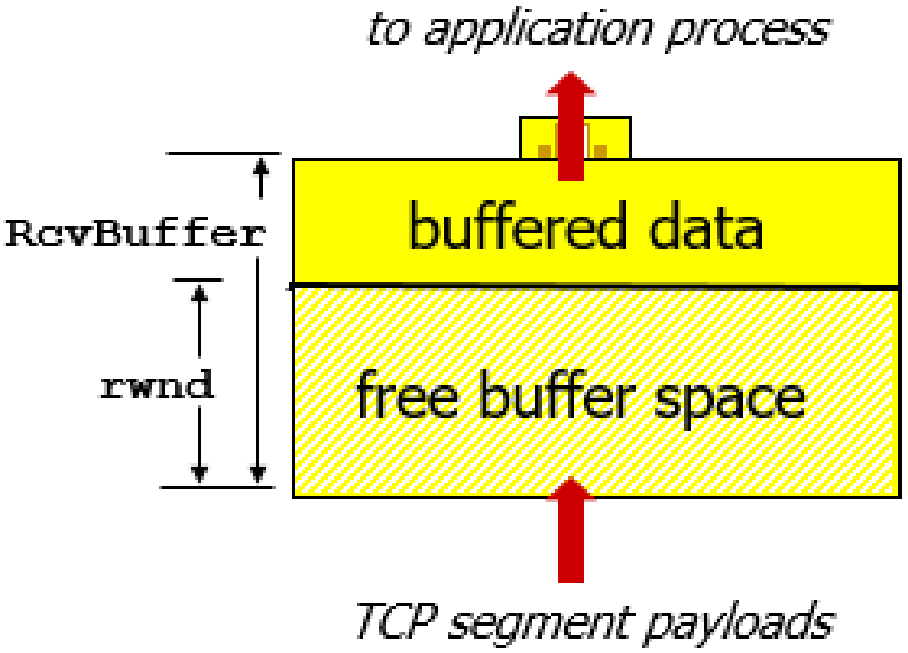
- 控制发送方的发送速度，避免接收方的缓冲区溢出

- 发送方的发送窗口



接收缓冲区

- 接收窗口RecvWnd



——避免缓冲区溢出：SendWind<=RecvWind

- 接收方：维护一个接收窗口，并通知发送方，控制发送方的窗口大小

5、 拥塞控制

- 网络拥塞
  - 丢包：路由器缓冲区溢出
  - 长延时：路由器队列排队

- 大量的源（主机）以过高的速率发送数据

1. 端到端的拥塞控制：发送方根据网络拥塞程度限制其发送速率
2. 网络辅助的拥塞控制：路由器向发送方反馈网络的拥塞状态（ICMP）

端到端的拥塞控制

- 发送方根据网络拥塞程度限制其发送速率

——没有拥塞，增大发送速率

——否则，降低发送速率

- 发送方根据网络拥塞程度限制其发送速率

——没有拥塞，增大发送速率，否则，降低发送速率

1. 发送方如何判断网络拥塞？

——发送方：超时重传，即判断网络出现了丢包，拥塞发生

- 发送方根据网络拥塞程度限制其发送速率

——没有拥塞，增大发送速率，否则，降低发送速率

2. 如何控制发送速率？

发送窗口

TCP发送方维持一个变量：拥塞窗口CongWnd

SendWind< = Min [RecvWnd, Congwnd]

- 发送方根据网络拥塞程度限制其发送速率

——没有拥塞，增大发送速率，否则，降低发送速率

3. 如何调整拥塞窗口？

TCP拥塞控制方法

1. 慢启动
2. 拥塞避免
3. 快恢复[推荐]

慢启动

- 拥塞窗口cwnd
  - ✓ 初始值：1个MSS

MSS： TCP数据部分的最大长度(Max Segment Size)

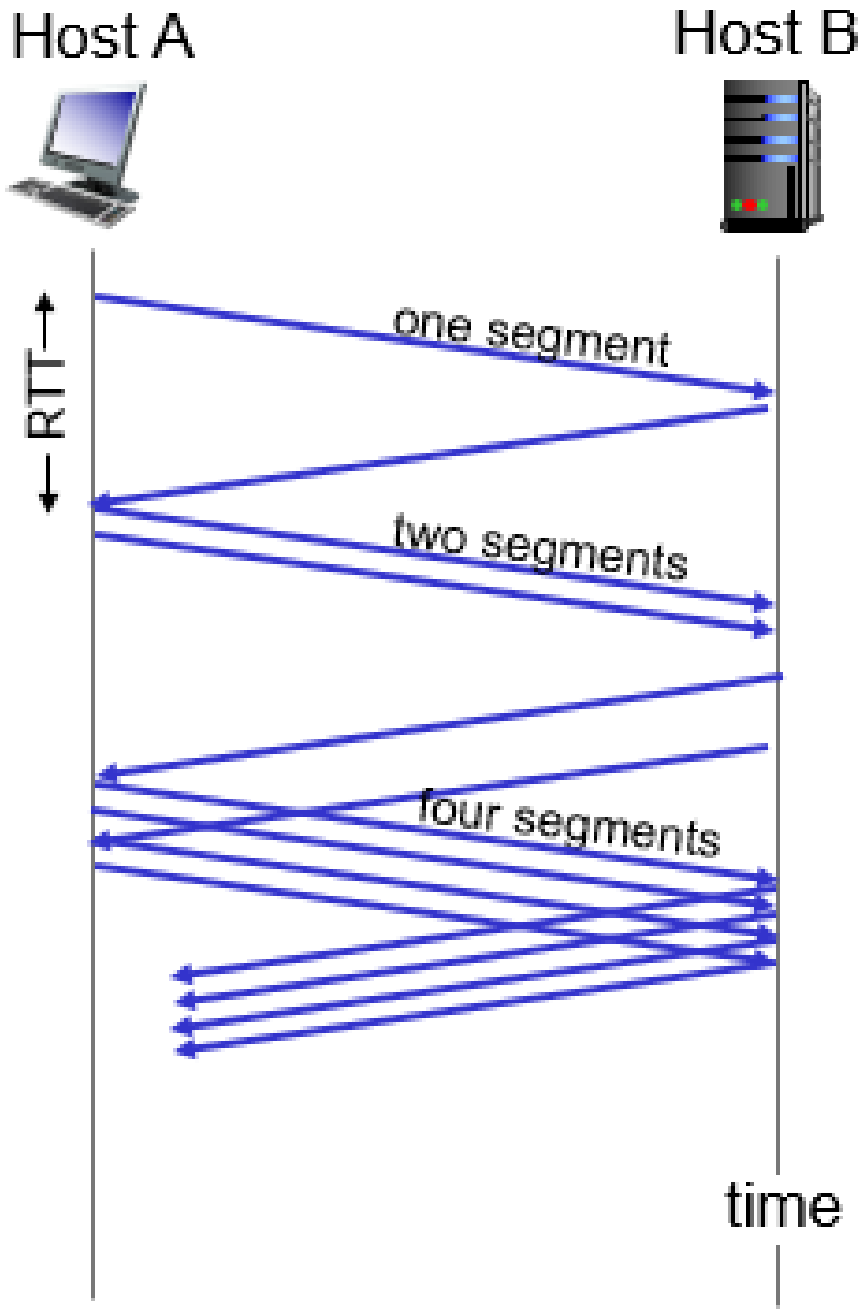
——IP分组的总长度限制（65535字节）

——链路层的最大传输单元（MTU）的限制

慢启动

- 拥塞窗口cwnd
  - ✓ 初始值：1个MSS
  - ✓ 每收到1个新的报文段的确认

    cwnd=cwnd+1个MSS


Host A
Host B
RTT
one segment
two segments
four segments
time

- 慢启动维持一个变量
  - ssthresh（慢启动阈值）

cwnd > ssthresh，进入拥塞避免阶段（cwnd线性增长）

拥塞避免

- 每经过一个RTT

  cwnd=cwnd+1个MSS

- 具体实现：每收到一个确认（假设每个报文段返回1个确认）

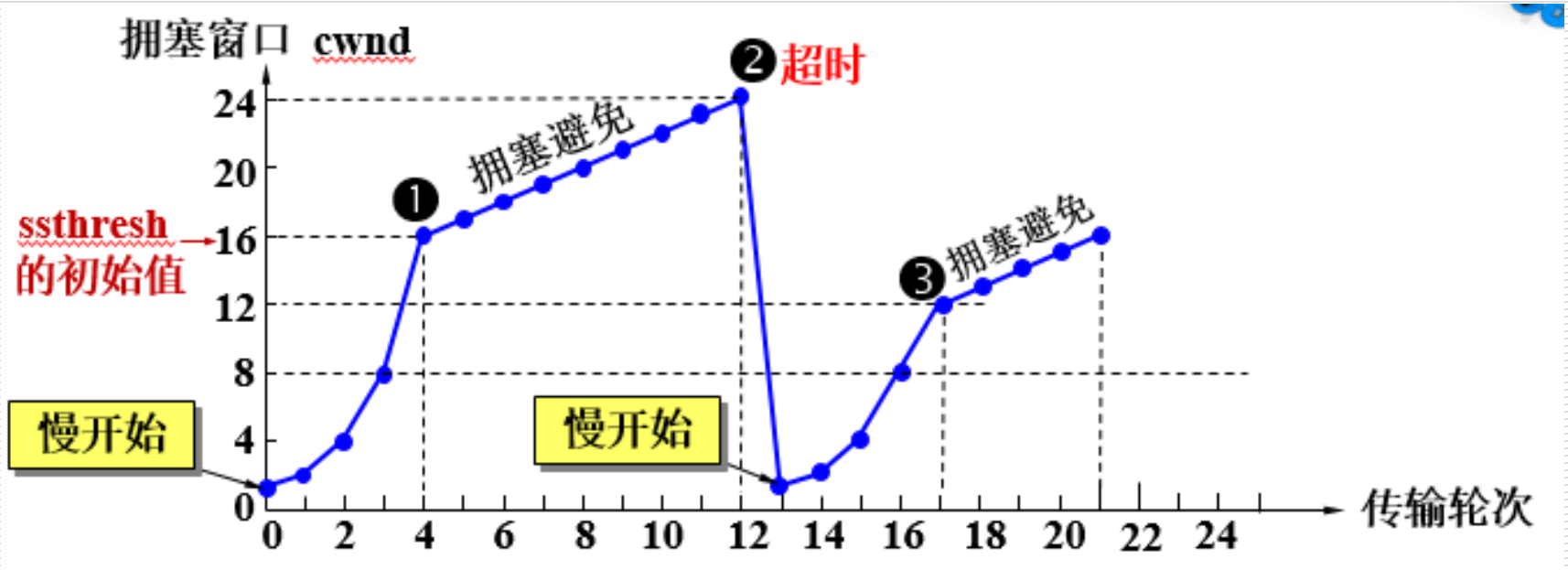
cwnd=cwnd+（1MSS/cwnd）*1MSS

- 慢启动/拥塞避免阶段，检测到拥塞（超时重传）

ssthresh =1/2（拥塞时的cwnd）

cwnd=1个MSS

进入慢启动阶段


拥塞窗口 cwnd
24
20
16
12
8
4
0
ssthresh
的初始值
慢开始
拥塞避免
超时
慢开始
拥塞避免
1
2
3
0 2 4 6 8 10 12 14 16 18 20 22 24
传输轮次

TCP拥塞控制方法

1. 慢启动
2. 拥塞避免
3. 快恢复[推荐]

快重传

- TCP Tahoe版本：
  - 在慢启动和拥塞避免的基础上，增加了快重传
  - 超时重传存在的问题？

    ——增加了端到端的时延

快重传

- TCP Tahoe版本：
  - 在慢启动和拥塞避免的基础上，增加了快重传
  - 超时重传存在的问题？

    ——增加了端到端的时延

快重传

——TCP接收方收到失序报文段，对已接收到最后一个字节数据重复确认（多个ACK）

——发送方收到多个ACK，重传定时器超时之前的报文段（快速重传）

1. 发送方如何判断网络拥塞？

——发送方：超时重传，即判断网络出现了丢包，拥塞发生

快重传（收到接收方多个冗余的ACK）

cwnd=1个MSS

ssthresh =1/2（拥塞时的cwnd）

进入慢启动

快恢复

- TCP Reno版本
  - 失序的发生，实际网络并没有拥塞

cwnd=1个MSS

ssthresh =1/2（拥塞时的cwnd）

进入慢启动

快重传后，进入快恢复阶段

cwnd=1个MSS

ssthresh =1/2（拥塞时的cwnd）

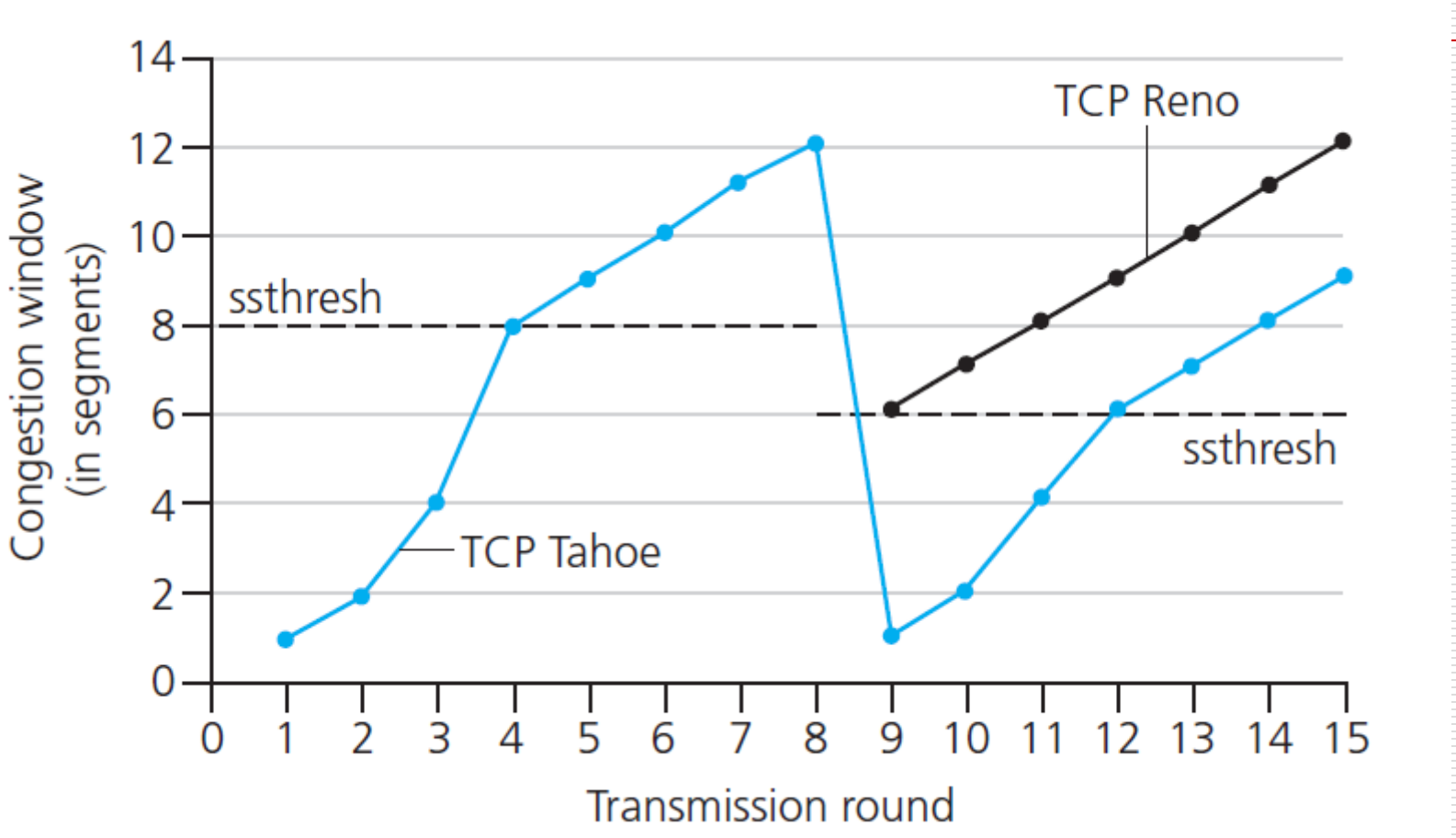
进入慢启动阶段

ssthresh =1/2（拥塞时的cwnd）

Cwnd= ssthresh / Cwnd=

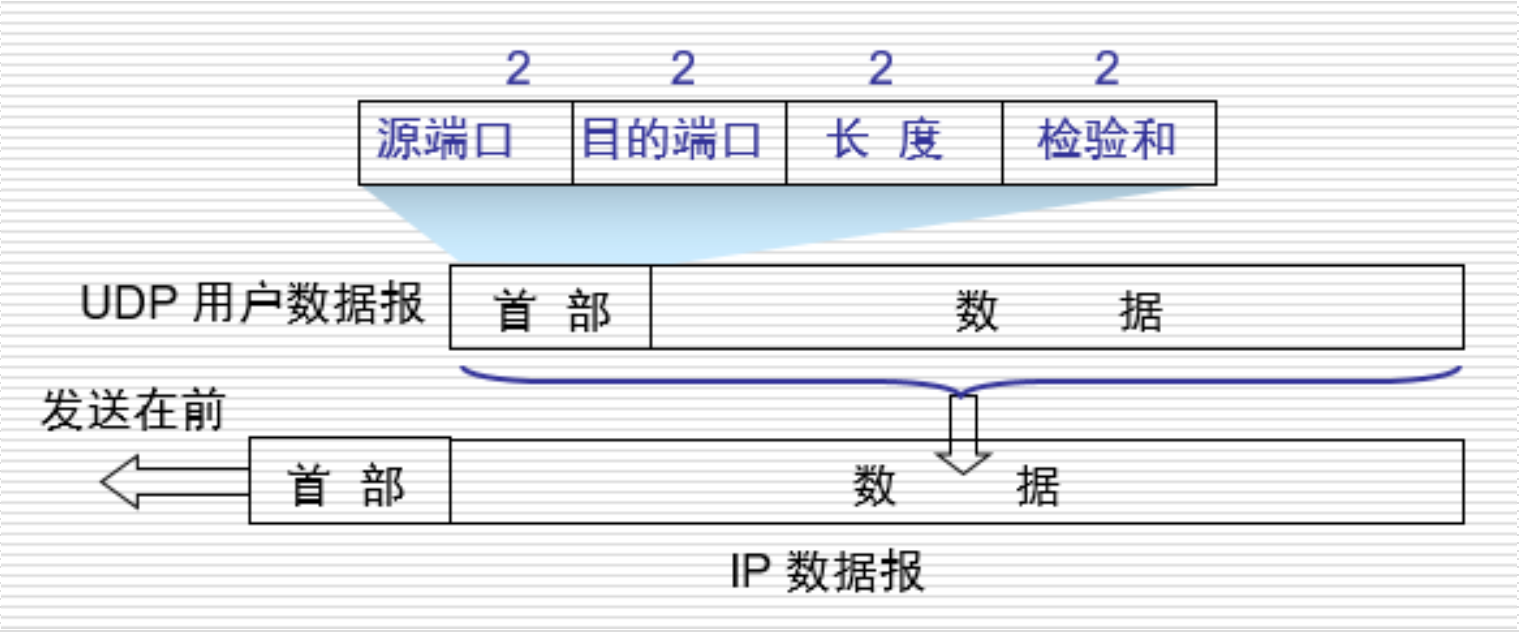
ssthresh+n*MSS

进入拥塞避免阶段


Congestion window (in segments)
14
12
10
8
6
4
2
0
ssthresh
TCP Tahoe
TCP Reno
ssthresh
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Transmission round

UDP

- 功能
  - 标识不同进程、差错检测



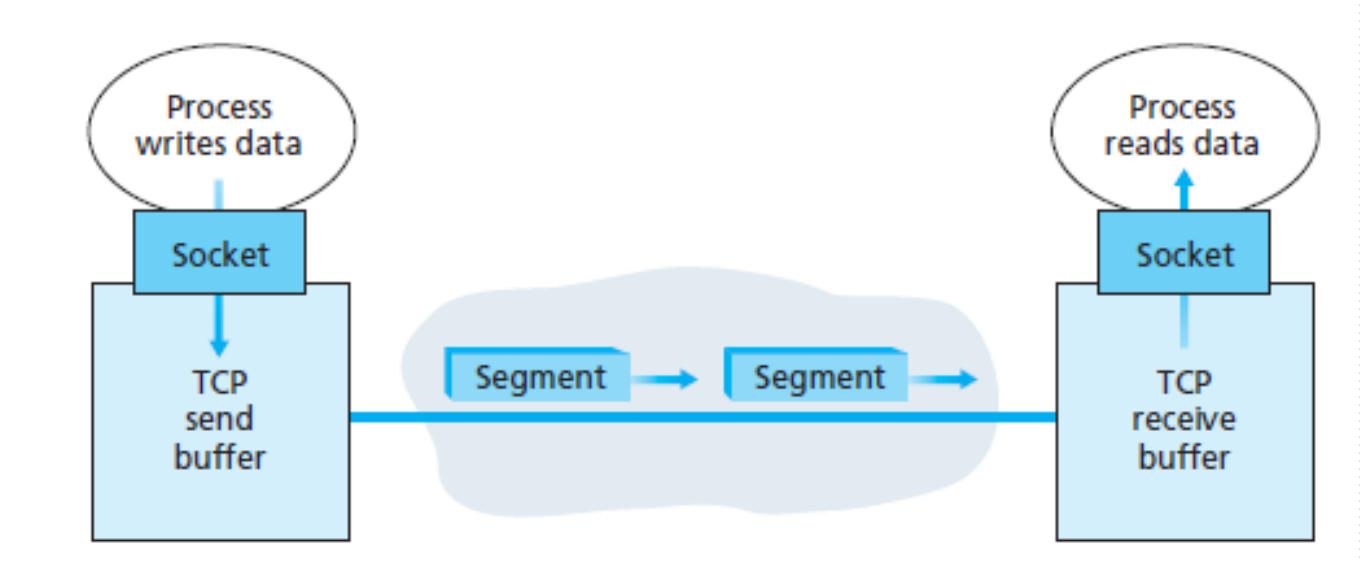
关于UDP

1. 无连接，支持广播和组播
2. UDP面向报文，一次发送一个报文
3. 不可靠传输

Why UDP?

1. 首部开销小（8个字节）
2. 无需建立连接，不会引入建立连接的时延
3. 不维护连接状态（参数及缓冲区），服务端应用进程采用UDP，可以支持更多的客户访问
4. 无拥塞控制和流量控制，适用于某些实时应用（延迟小，容忍一定的数据丢失）

- Socket（套接字）：应用进程和传输层之间的接口，开发网络应用程序的可编程接口



多选题 5分

设置

TCP/IP体系结构中，传输层的功能包括：

A 实现主机之间的通信

B 实现端到端的通信

C 可靠传输

D 路由和转发

提交

单选题 5分

设置

一个主机上的进程可以用（）标识

A 端口号

B MAC地址

C IP地址

D IP地址+端口号

提交

- 作业
  - 6.5、6.8
  - 6.14、6.25、6.27