

第8章

通信网和服务



传输层协议: UDP 和 TCP



大纲



1. UDP 协议
2. TCP 可靠的流服务
3. TCP 协议
4. TCP 连接管理
5. TCP 拥塞控制

1、 用户数据报协议 - UDP

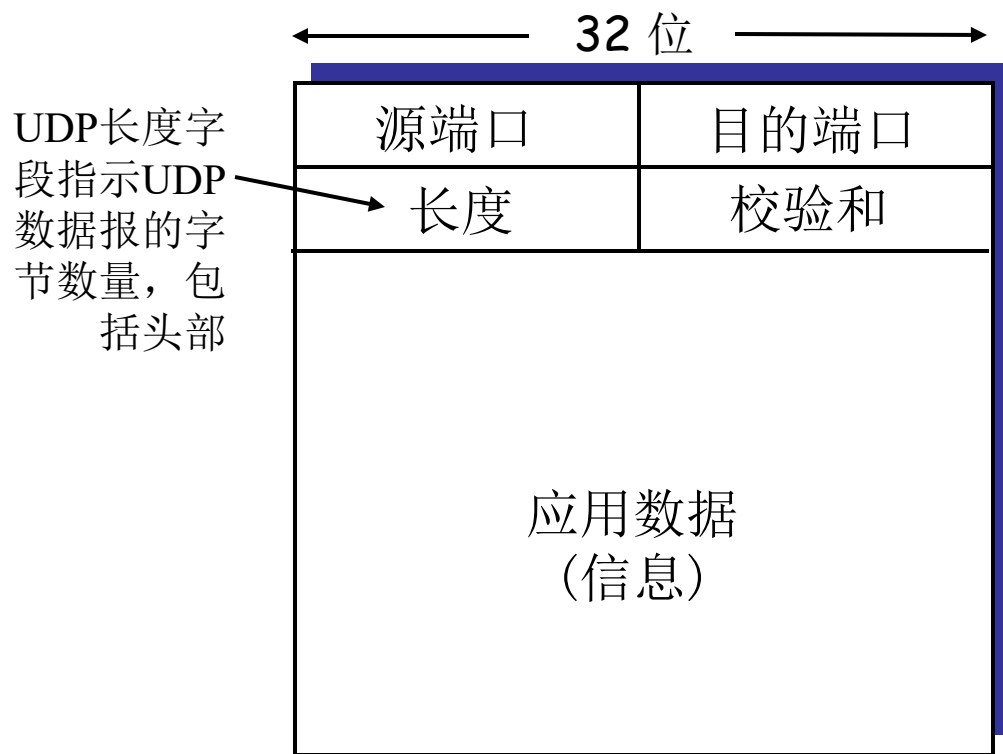


- 尽力而为数据报服务
- 多路复用支持共享IP数据报服务
- 简单的发送端和接收端
 - 无连接：没有握手和连接状态
 - 低头部开销
 - 没有流控制，没有差错控制，没有拥塞控制
 - UDP 数据报可能会丢失或乱序
- 应用
 - 多媒体 (e.g. RTP)
 - 网络服务 (e.g. DNS, RIP, SNMP)

用户数据报协议 (续)



UDP 数据报



UDP 段格式

- 源端口和目的端口号
 - 客户端使用临时性的端口号
 - 服务器使用大家熟知的端口号
 - 最大端口号为65,535
- UDP长度
 - 数据报的字节总数（包括头部）
 - $8 \leq \text{长度（字节）} \leq 65,535$
- UDP校验和
 - 可选地用来检查UDP数据报的错误

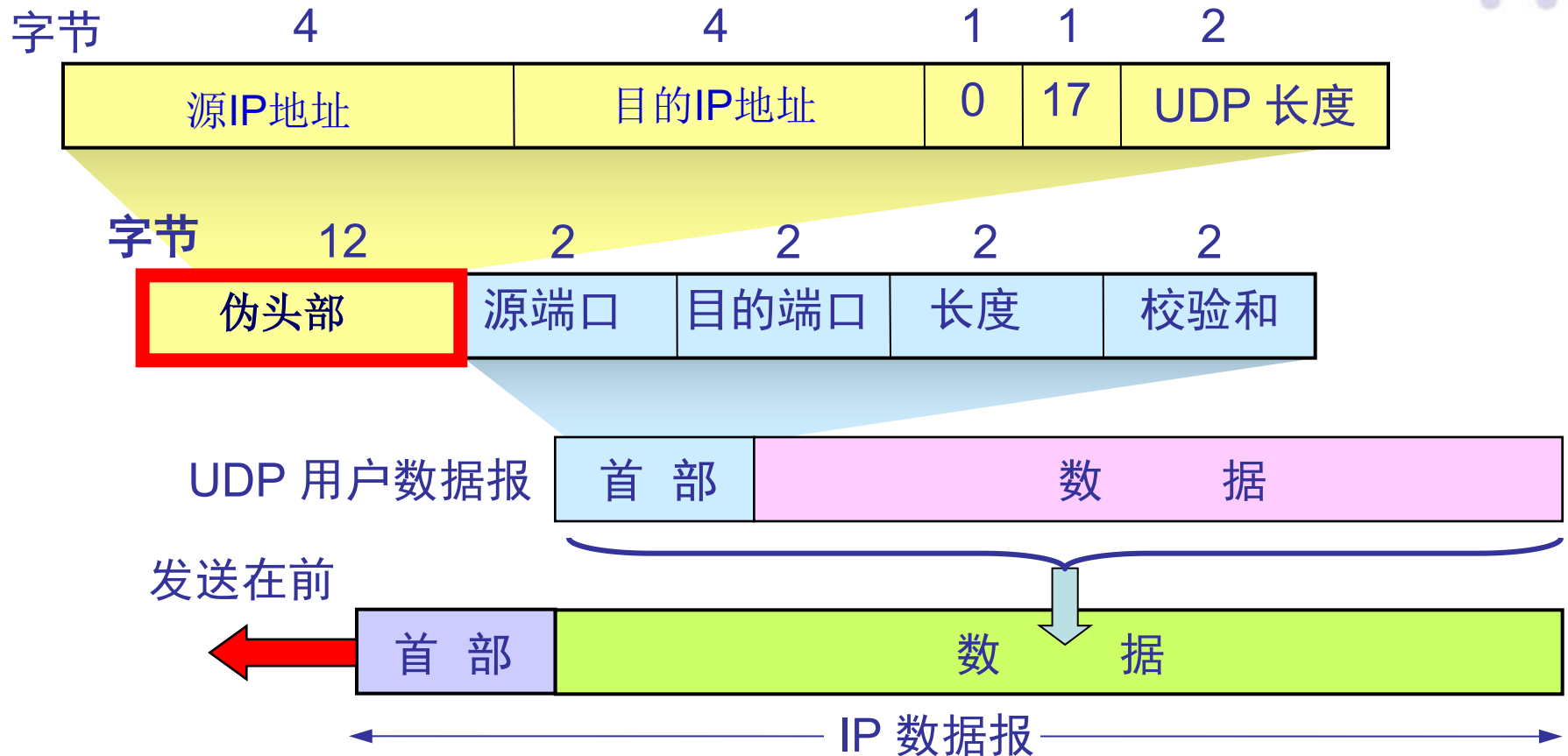


用户数据报协议 (续)

部分已分配的 **UDP** 端口

端口号	服务进程	说明
53	Name server	域名服务
67	Bootps	下载引导程序信息的服务器端口
68	Bootpc	下载引导程序信息的客户机端口
69	TFTP	简单文件传输协议
111	RPC	远程过程调用
123	NTP	网络时间协议
161	SNMP	简单网络管理协议

UDP校验和：伪头部



1. UDP 校验和检测端到端错误
2. 在伪头部后跟UDP数据报
3. 含有IP地址用于针对错误传递进行检测
4. IP & UDP校验和在计算过程中设置为0
5. 如果UDP长度为奇数，在后面填充0（变成16位的整数倍）



UDP 接收端校验和

- UDP 接收端重新计算校验和，然后悄悄丢弃有错误被检测出的数据报
 - “悄悄”意味着没有差错信息产生
- UDP校验和的使用是可选的
- 但主机被强制要求有校验和能够被使用



例 [UDP接收端校验和]

12 字节 伪头部	153.19.8.104			
	171.3.14.11			
	全 0	17	15	
8 字节 UDP 头部	1087		13	
	15		全 0	
7 字节 数据	数据	数据	数据	数据
	数据	数据	数据	全 0

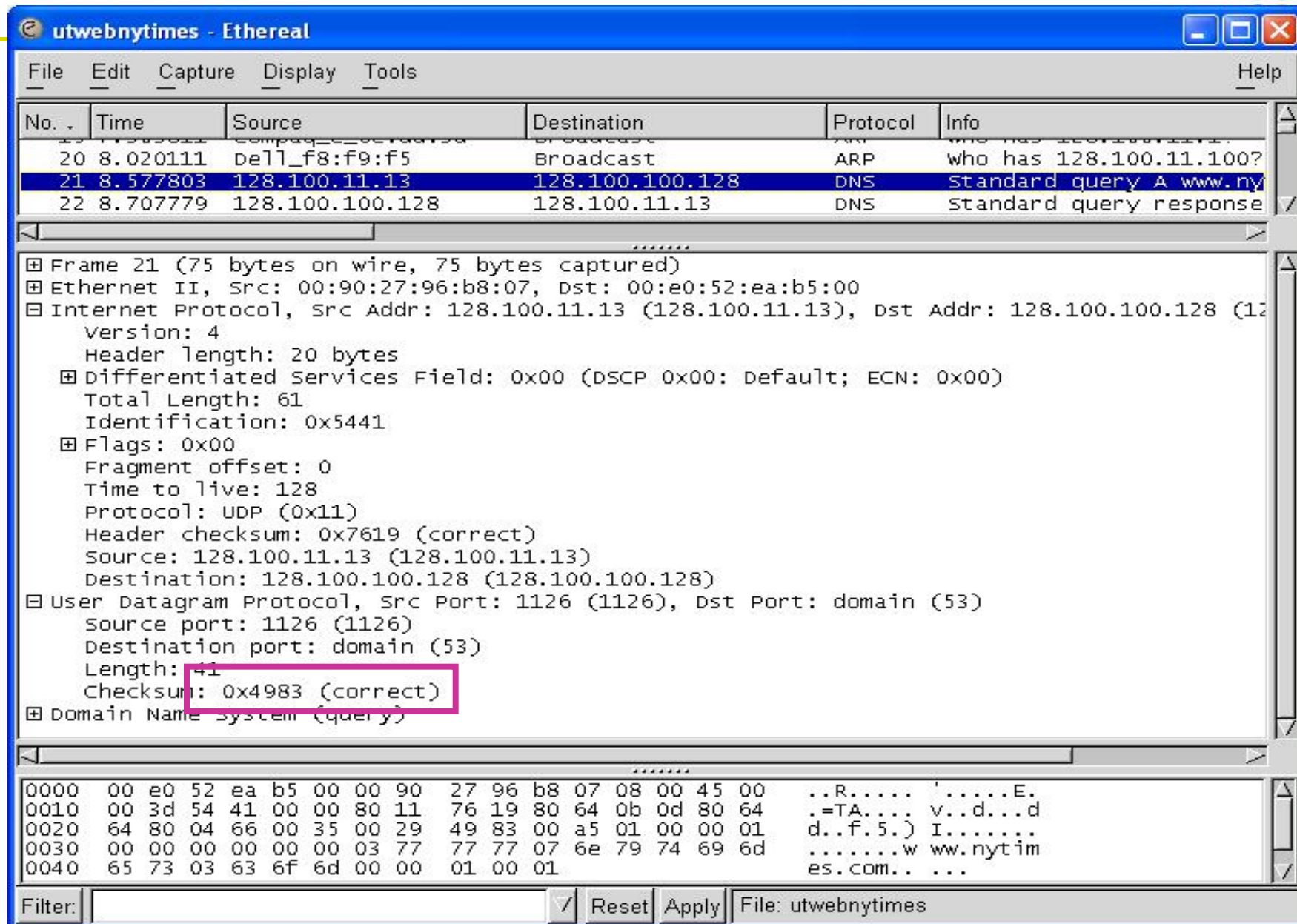
填充

10011001 00010011 → 153.19
00001000 01101000 → 8.104
10101011 00000011 → 171.3
00001110 00001011 → 14.11
00000000 00010001 → 0 和 17
00000000 00001111 → 15
00000100 00111111 → 1087
00000000 00001101 → 13
00000000 00001111 → 15
00000000 00000000 → 0 (检验和)
01010100 01000101 → 数据
01010011 01010100 → 数据
01001001 01001110 → 数据
01000111 00000000 → 数据和 0 (填充)

按二进制反码运算求和 10010110 11101101 → 求和得出的结果
将得出的结果求反码 01101001 00010010 → 检验和

请注意：进行反码运算求和时，最高位有进位 2，这个 2 应当加到最低位。

例



The image shows a screenshot of the 'utwebnytimes - Ethereal' window. The top menu bar includes File, Edit, Capture, Display, Tools, and Help. Below the menu is a table of captured packets. Packet 21 is selected, showing a DNS standard query from 128.100.11.13 to 128.100.100.128. The packet details pane shows the structure of the packet, including Ethernet II, Internet Protocol, User Datagram Protocol, and Domain Name System (query). The packet length is 75 bytes. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Info
20	8.020111	dell_f8:f9:f5	Broadcast	ARP	who has 128.100.11.100?
21	8.577803	128.100.11.13	128.100.100.128	DNS	Standard query A www.ny
22	8.707779	128.100.100.128	128.100.11.13	DNS	Standard query response

Frame 21 (75 bytes on wire, 75 bytes captured)

- Ethernet II, Src: 00:90:27:96:b8:07, Dst: 00:e0:52:ea:b5:00
- Internet Protocol, Src Addr: 128.100.11.13 (128.100.11.13), Dst Addr: 128.100.100.128 (128.100.100.128)
 - Version: 4
 - Header length: 20 bytes
 - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
 - Total Length: 61
 - Identification: 0x5441
 - Flags: 0x00
 - Fragment offset: 0
 - Time to live: 128
 - Protocol: UDP (0x11)
 - Header checksum: 0x7619 (correct)
 - Source: 128.100.11.13 (128.100.11.13)
 - Destination: 128.100.100.128 (128.100.100.128)
- User Datagram Protocol, Src Port: 1126 (1126), Dst Port: domain (53)
 - Source port: 1126 (1126)
 - Destination port: domain (53)
 - Length: 41
 - Checksum: 0x4983 (correct)
- Domain Name System (query)

0000 00 e0 52 ea b5 00 00 90 27 96 b8 07 08 00 45 00 ..R.....E.
0010 00 3d 54 41 00 00 80 11 76 19 80 64 0b 0d 80 64 .=TA....v..d...d
0020 64 80 04 66 00 35 00 29 49 83 00 a5 01 00 00 01 d..f.5.)I.....
0030 00 00 00 00 00 00 03 77 77 77 07 6e 79 74 69 6dw ww.nytim
0040 65 73 03 63 6f 6d 00 00 01 00 01 es.com.. ...

Filter: [] [Reset] [Apply] File: utwebnytimes

用户数据报协议 (续)



优点:

- **更低的开销** (没有连接建立)
- 更加高效 (不提供可靠交付)
- **实时** 应用 (没有差错检测或流控制)
 - E. g. 天气, 时间, 视频, 音频, 游戏
- 从多台机器接收数据

使用UDP的协议: SNMP、DNS、BOOTP、TFTP、SUNRPC、SNMPTRAP、NFS RIP、GTP、BIFF、WHO、SYSLOG、RTP

用户数据报协议 (续)



UDP显著特点:

1. 传输前不需要建立连接;
2. 接收端收到UDP报文不需要做出确认;
3. 不提供可靠交付,但在一些情况下效率更高!

注意!

伪头部仅为计算校验和时由源主机和目的主机创建,不占用实际传输开销!

大纲



1. UDP 协议
2. TCP 可靠的流服务
3. TCP 协议
4. TCP 连接管理
5. TCP 拥塞控制

2、TCP可靠的流服务



— 可靠的字节流服务

— 更加复杂的发送端和接收端

- 面向连接：客户端和服务端进程之间全双工单播连接
- 连接建立，连接状态，连接释放
- 更高的头部开销
- 差错控制，流控制和拥塞控制
- 相较于UDP更高的延迟

— 绝大多数应用使用TCP

- HTTP, SMTP, FTP, TELNET, POP3, ...

TCP可靠的流服务 (续)

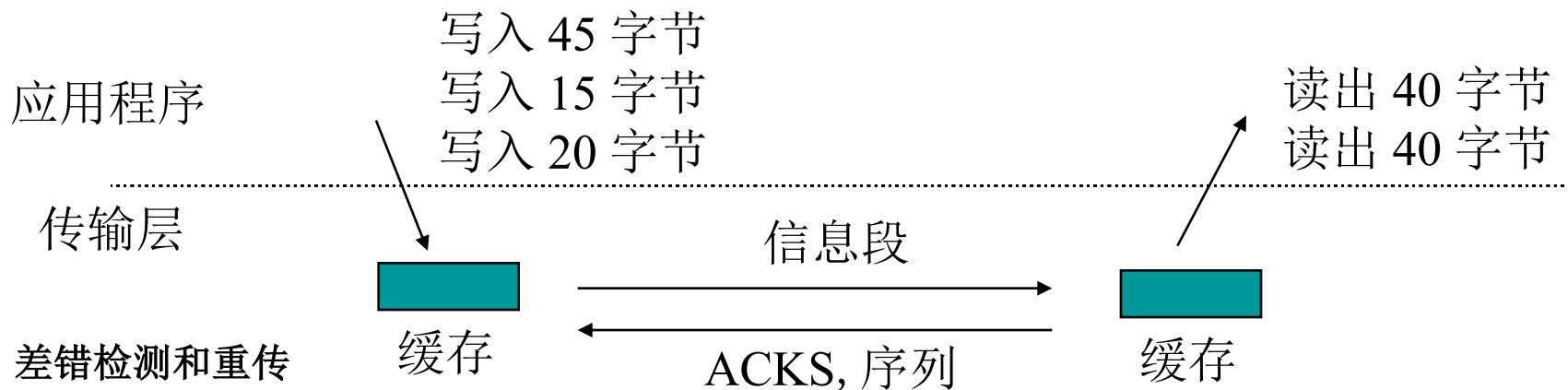


— 数据流传输

- 通过网络传输连续的字节流，不指示边界
- 将字节分组为段(信息段)
- 在方便时传输信息段 (定义了推送功能)

— 可靠性

- **错误控制机制** 应对IP传输错误

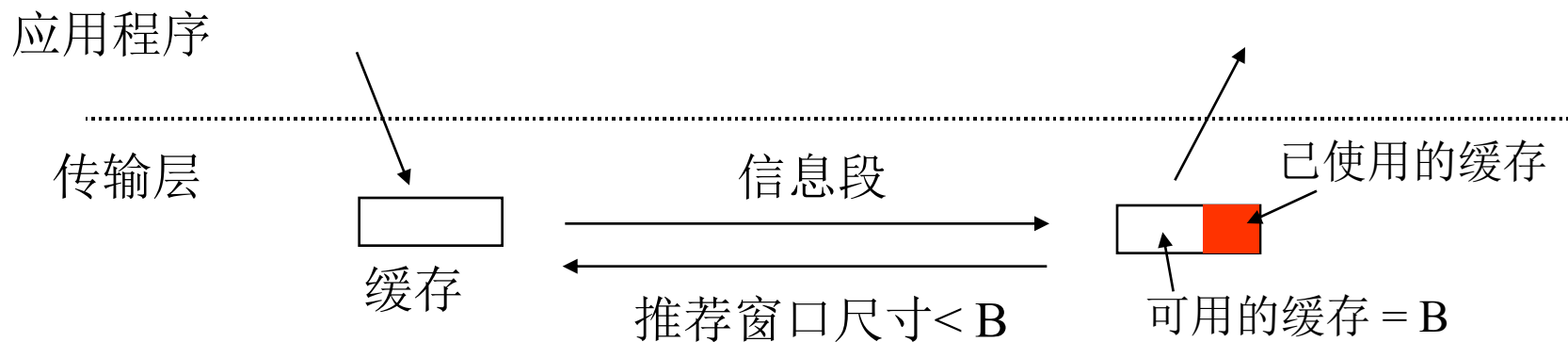


TCP可靠的流服务 (续)



流量控制:

- 缓存限制和速率不匹配会导致到达目的地的数据丢失
- 接收端控制发送端发送分组的速率来防止缓存溢出

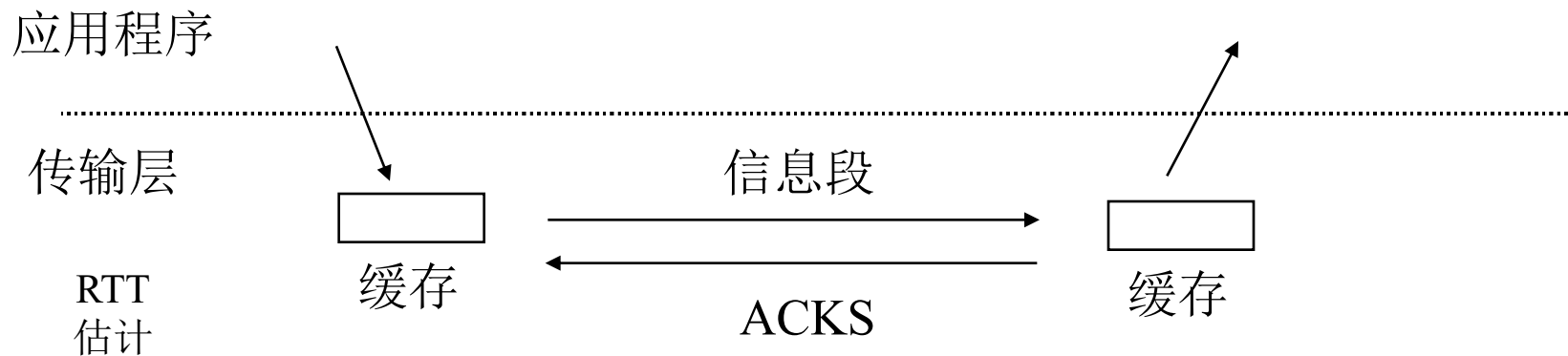


TCP可靠的流服务 (续)



拥塞控制:

- 到目的地的可用带宽随其他用户的活动而变
- 发送端根据RTT（往返时间）和ACKs指示的网络拥塞程度动态调整传输速率
- 网络带宽的弹性利用

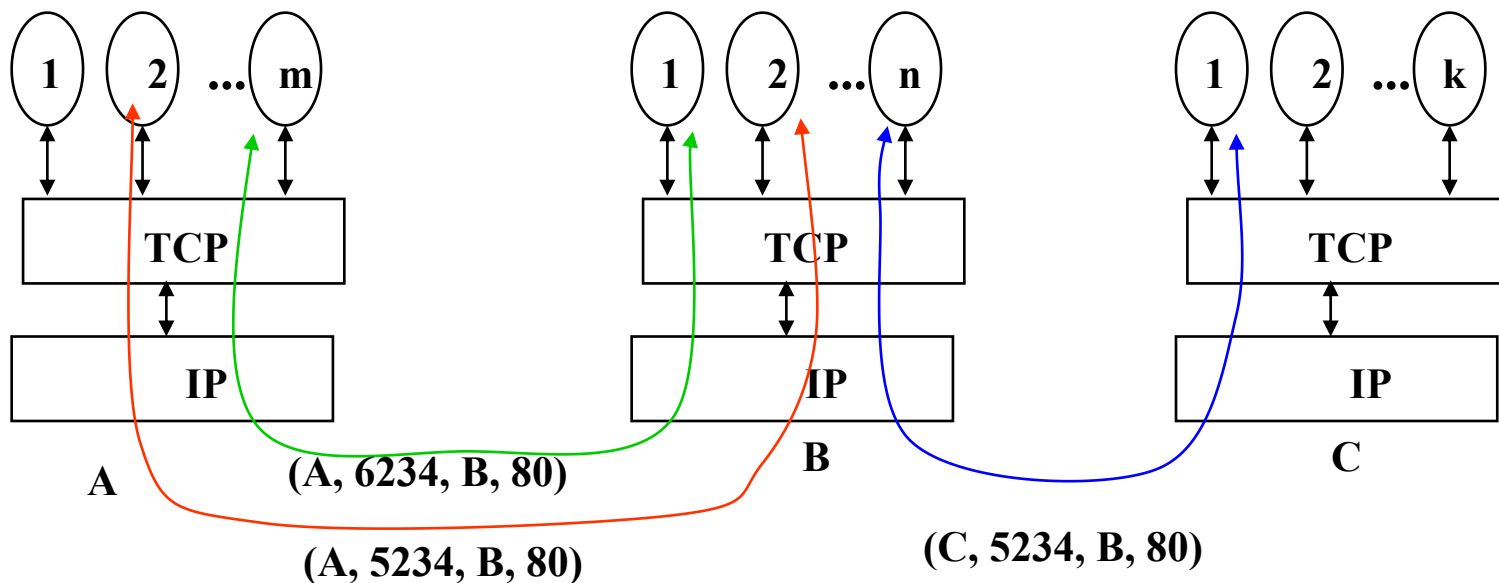




TCP多路复用

TCP多路复用:

- 一个TCP连接由一个四元组唯一标识
 - (源IP地址, 源端口, 目的IP地址, 目的端口)
- TCP允许在终端系统之间多路复用多个连接, 以同时支持多个应用
- 数据段由连接的四元组指引到达目的地



大纲

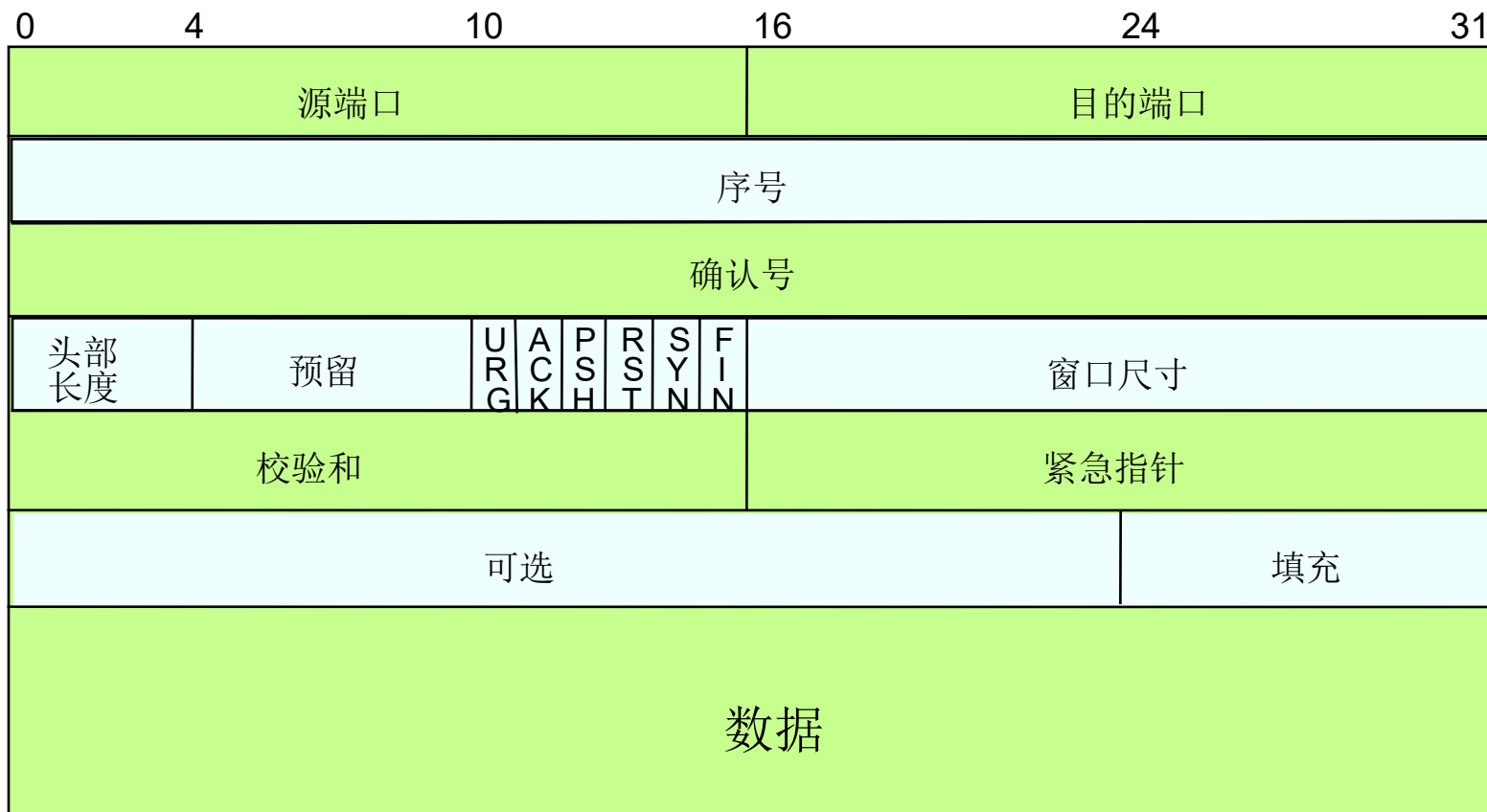


1. UDP 协议
2. TCP 可靠的流服务
3. TCP 协议
4. TCP 连接管理
5. TCP 拥塞控制



3、TCP协议

TCP段格式



- 每个TCP段头部都有1个20字节的固定部分加上一个大小可变的任选字段

TCP协议 (续)



TCP头部:

端口号

- 一个套接字标识连接的一端
 - **IP 地址 + 端口**
- 一个连接可以由一对套接字唯一标识
- 熟知的端口
 - **FTP** **20, 21**
 - **Telnet** **23**
 - **SMTP** **25**
 - **DNS** **53**
 - **TFTP** **69**
 - **HTTP** **80**
 - **POP3** **110**

TCP协议 (续)



头部长度

- 4 位
- 头部长度是32位字长的整数倍

- 头部

- 头

确认号

- 接

- 确

- 如

序号

-

-

- 长

- $0 \leq$

- 在连接建立时，选择初始序号

1. 面向字节：TCP 报文==字节组成的数据流+每个字节对应于一个序号

2. 连接建立时，双方商定初始序号。报文段首部中的序号字段数值表示负载部分第一个字节的序号

3. TCP 对接收到数据的最高序号表示确认，即确认号==已收到数据最高序号加 1，即接收端期望下次收到的数据中第一个字节的序号

TCP协议 (续)



预留字段

- 6 位

控制字段

- 6 位
- URG: 紧急指针标志
 - 紧急信号末端 = 序号 + 紧急指针
- ACK: 数据包确认标志
- PSH: 覆盖TCP缓存
- RST: 重置连接
 - 收到RST后, 连接中止, 应用层得到通知
- SYN: 建立连接
- FIN: 关闭连接

TCP协议 (续)



窗口尺寸

- 通知窗口尺寸为16位
- 用于流量控制
- 发送端将接受序号从ACK 到 $ACK + \text{窗口} - 1$ 的字节
- 最大窗口尺寸为 65535 字节

TCP校验和

- 网络检验和方法
- TCP伪头部 + TCP段

TCP协议 (续)



TCP校验和计算

- TCP差错检测使用和UDP相同的伪头部



TCP伪头部

大纲



1. UDP 协议
2. TCP 可靠的流服务
3. TCP 协议
4. TCP 连接管理
5. TCP 拥塞控制

4、TCP连接管理



初始序号 (ISN) :

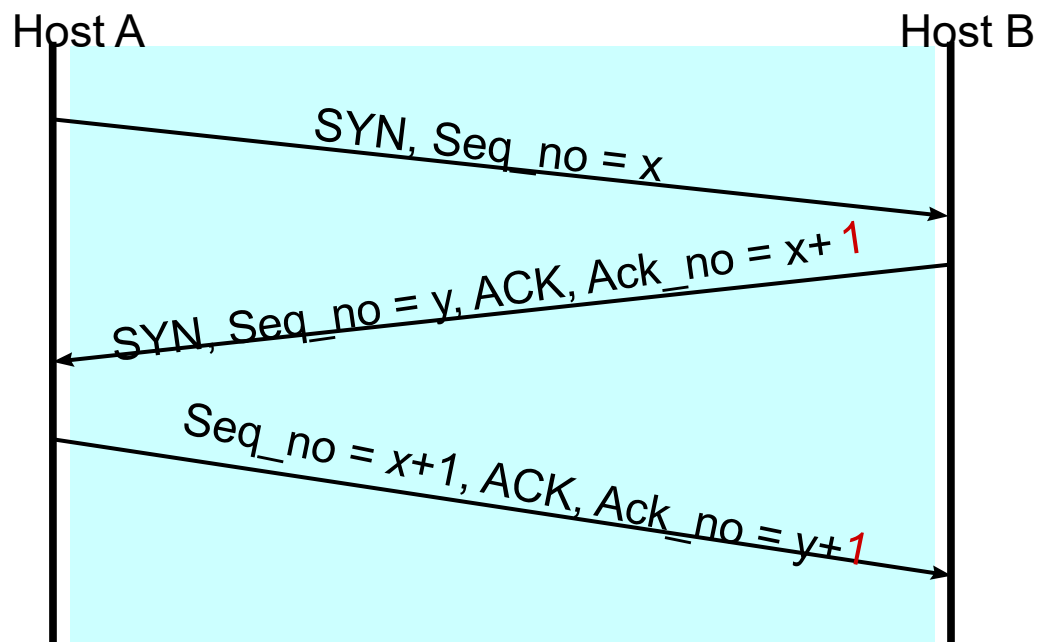
- 选择ISN来防止接收来自之前连接的信息段(可能会在网络中循环并在更晚的时间到达)
- 选择ISN来防止和之前连接的序号重复
- 使用本地时钟来选择ISN序号
- 时钟经过一个完整周期的时间应该大于一个段的最大生存期 (MSL) e.g. 通常 $MSL=120$ 秒
- 高带宽连接带来了问题

TCP连接建立 (续)



TCP连接建立：

- “三次握手”
- ISN防止接收来自之前连接的信息段

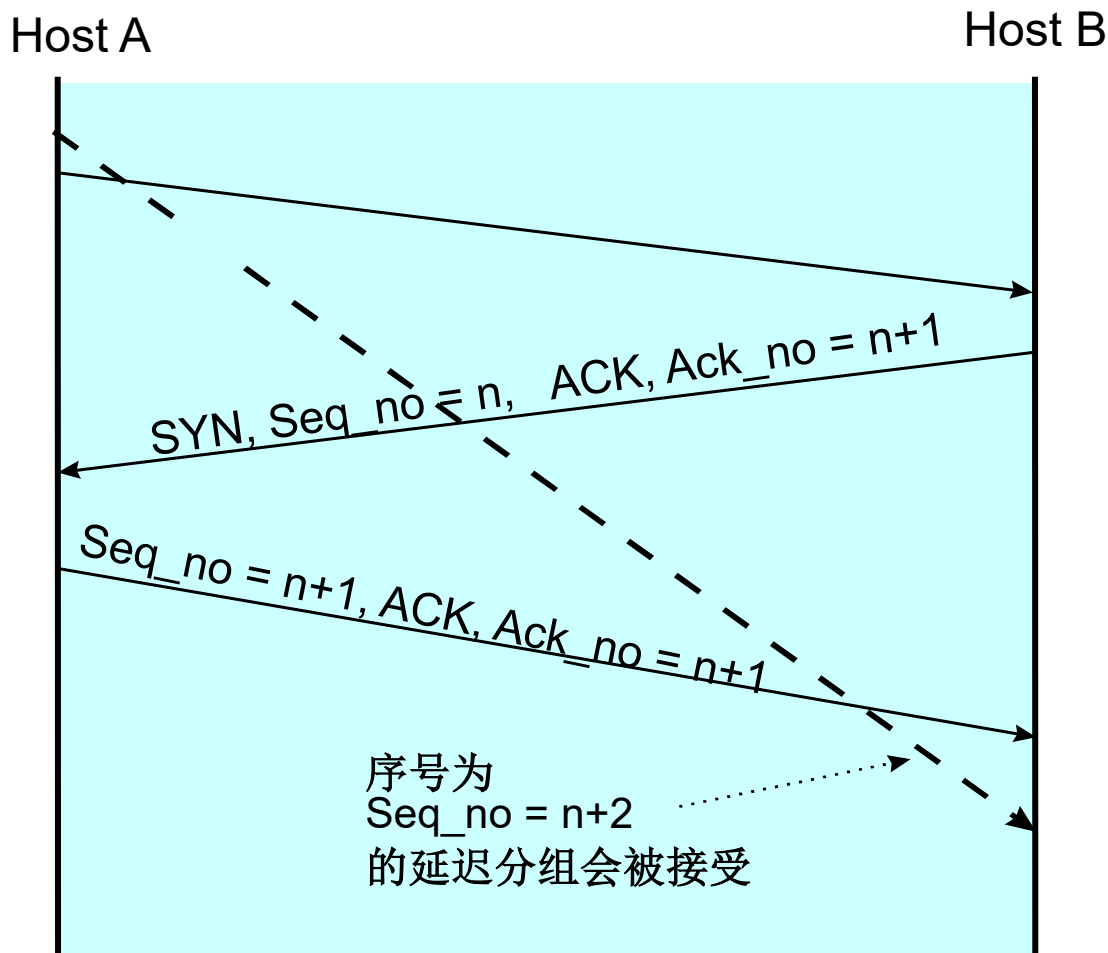


采用三次握手建立连接是为了防止已失效的连接请求报文突然又传到B，而产生错误。如：一段时间当A发送的第一个连接请求并未丢失，由于网络滞留影响A超时未受到B的确认，则发一个新的连接请求。在这次连接释放后前一个请求报文到达B。B以为A向它发出新的连接请求，于是向A发出确认报文，而主机A并未发出连接请求确认报文，因此不理睬B的确认报文。若把三次握手改成二次握手可能发生死锁。



TCP连接建立 (续)

例 [如果主机一直使用相同的 ISN]



TCP连接建立 (续)



最大段尺寸:

- 最大段尺寸
 - TCP发送到另一端的最大数据块
- 每个终端可以在连接建立阶段宣布自己的MSS
- 默认值是576字节，包括20字节用于IP报头和20字节用于TCP报头
- 以太网限制MSS为1460字节
- IEEE 802.3限制MSS为1452字节

近端：连接请求



utwebnytimes - Ethereal

File Edit Capture Display Tools Help

No.	Time	Source	Destination	Protocol	Info
22	8.707779	128.100.100.128	128.100.11.13	DNS	Standard query response A 64.15.247.200 A 64.15.247.200
23	8.709327	128.100.11.13	64.15.247.200	TCP	1127 > http [SYN] Seq=3638689752 Ack=0 win=0
24	8.746089	64.15.247.200	128.100.11.13	TCP	http > 1127 [SYN, ACK] Seq=1396200325 Ack=3638689752 win=0
25	8.746123	128.100.11.13	64.15.247.200	TCP	1127 > http [ACK] Seq=3638689753 Ack=1396200326 win=0
26	8.746491	128.100.11.13	64.15.247.200	HTTP	GET / HTTP/1.1
27	8.783242	64.15.247.200	128.100.11.13	TCP	http > 1127 [ACK] Seq=1396200326 Ack=3638690000 win=0
28	8.814479	64.15.247.200	128.100.11.13	HTTP	HTTP/1.1 200 OK
29	8.814526	64.15.247.200	128.100.11.13	HTTP	Continuation

Frame 23 (62 bytes on wire (42 bytes captured))

- Ethernet II, Src: 00:90:27:96:b8:07, Dst: 00:e0:52:ea:b5:00
- Internet Protocol, Src Addr: 128.100.11.13 (128.100.11.13), Dst Addr: 64.15.247.200 (64.15.247.200)
- Transmission Control Protocol, Src Port: 1127 (1127), Dst Port: http (80), Seq: 3638689752, Ack: 0, Len: 0
 - Source port: 1127 (1127)
 - Destination port: http (80)
 - Sequence number: 3638689752
 - Header length: 28 bytes
 - Flags: 0x0002 (SYN)
 - 0... = Congestion window Reduced (CWR): Not set
 - .0.. = ECN-Echo: Not set
 - ..0. = Urgent: Not set
 - ...0 = Acknowledgment: Not set
 - 0... = Push: Not set
 -0.. = Reset: Not set
 -1. = Syn: Set
 -0 = Fin: Not set
 - window size: 16384
 - checksum: 0xa2e4 (correct)
 - Options: (8 bytes)
 - Maximum segment size: 1332 bytes
 - NOP
 - NOP
 - SACK permitted

0000 00 e0 52 ea b5 00 00 90 27 96 b8 07 08 00 45 00 ..R.....'.....E.
0010 00 30 54 42 40 00 80 06 e3 3c 80 64 0b 0d 40 0f .0TB@...<.d..@.
0020 f7 c8 04 67 00 50 d8 e1 ff d8 00 00 00 00 70 02 ...g.P.....p.
0030 40 00 a2 e4 00 00 02 04 05 34 01 01 04 02 @.....4....

Filter: Reset Apply File: utwebnytimes

远端：确认和请求



utwebnnytimes - Ethereal

File Edit Capture Display Tools Help

No.	Time	Source	Destination	Protocol	Info
22	8.707779	128.100.100.128	128.100.11.13	DNS	Standard query response A 64.15.247.200 A 64.15.247.200
23	8.709327	128.100.11.13	64.15.247.200	TCP	1127 > http [SYN] Seq=3638689752 Ack=0 win=0
24	8.746089	64.15.247.200	128.100.11.13	TCP	http > 1127 [SYN, ACK] Seq=1396200325 Ack=3638689753
25	8.746123	128.100.11.13	64.15.247.200	TCP	1127 > http [ACK] Seq=3638689753 Ack=1396200325
26	8.746491	128.100.11.13	64.15.247.200	HTTP	GET / HTTP/1.1
27	8.783242	64.15.247.200	128.100.11.13	TCP	http > 1127 [ACK] Seq=1396200326 Ack=3638690000
28	8.814479	64.15.247.200	128.100.11.13	HTTP	HTTP/1.1 200 OK
29	8.814526	64.15.247.200	128.100.11.13	HTTP	Continuation

Frame 24 (62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0)

Ethernet II, Src: 00:e0:52:ea:b5:00, Dst: 00:90:27:96:b8:07

Internet Protocol, Src Addr: 64.15.247.200 (64.15.247.200), Dst Addr: 128.100.11.13 (128.100.11.13)

Transmission Control Protocol, Src Port: http (80), Dst Port: 1127 (1127), Seq: 1396200325, Ack: 3638689753, Source port: http (80), Destination port: 1127 (1127), Sequence number: 1396200325, Acknowledgement number: 3638689753, Header length: 28 bytes

Flags: 0x0012 (SYN, ACK)

- 0... = Congestion window Reduced (CWR): Not set
- .0... = ECN-Echo: Not set
- ..0. = Urgent: Not set
- ...1 = Acknowledgment: Set
- 0... = Push: Not set
-0.. = Reset: Not set
-1. = Syn: Set
-0 = Fin: Not set

Window size: 1460

Checksum: 0x35e2 (correct)

Options: (8 bytes)

- NOP
- NOP
- SACK permitted
- Maximum segment size: 1460 bytes

近端：确认

utwebnnytimes - Ethereal

File Edit Capture Display Tools Help

No.	Time	Source	Destination	Protocol	Info
22	8.707779	128.100.100.128	128.100.11.13	DNS	Standard query response A 64.15.247.200 A 64.15.247.200
23	8.709327	128.100.11.13	64.15.247.200	TCP	1127 > http [SYN] Seq=3638689752 Ack=0 win=0
24	8.746089	64.15.247.200	128.100.11.13	TCP	http > 1127 [SYN, ACK] Seq=1396200325 Ack=3638689753
25	8.746123	128.100.11.13	64.15.247.200	TCP	1127 > http [ACK] Seq=3638689753 Ack=1396200326
26	8.746491	128.100.11.13	64.15.247.200	HTTP	GET / HTTP/1.1
27	8.783242	64.15.247.200	128.100.11.13	TCP	http > 1127 [ACK] Seq=1396200326 Ack=363869
28	8.814479	64.15.247.200	128.100.11.13	HTTP	HTTP/1.1 200 OK
29	8.814526	64.15.247.200	128.100.11.13	HTTP	Continuation

Frame 25 (54 bytes on wire, 54 bytes captured)

Ethernet II, Src: 00:90:27:96:b8:07, Dst: 00:e0:52:ea:b5:00

Internet Protocol, Src Addr: 128.100.11.13 (128.100.11.13), Dst Addr: 64.15.247.200 (64.15.247.200)

Transmission Control Protocol, Src Port: 1127 (1127), Dst Port: http (80), Seq: 3638689753, Ack: 1396200326, Source port: 1127 (1127), Destination port: http (80), Sequence number: 3638689753, Acknowledgement number: 1396200326, Header length: 20 bytes

Flags: 0x0010 (ACK)

- 0... = Congestion window Reduced (CWR): Not set
- .0.. = ECN-Echo: Not set
- ..0. = Urgent: Not set
- ...1 = Acknowledgment: Set
- 0... = Push: Not set
-0.. = Reset: Not set
-0. = Syn: Not set
-0 = Fin: Not set

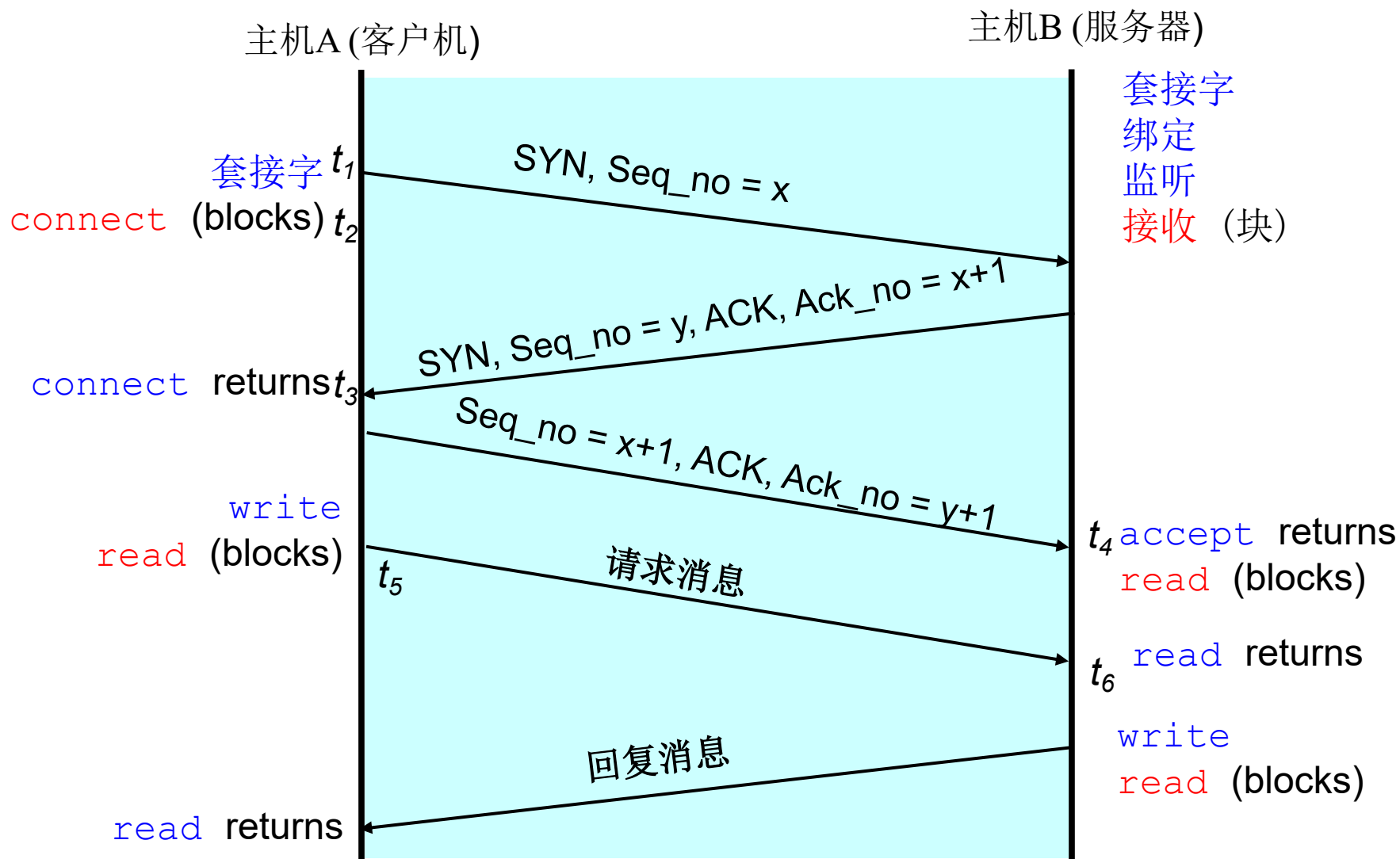
window size: 17316
checksum: 0x24b6 (correct)

0000 00 e0 52 ea b5 00 00 90 27 96 b8 07 08 00 45 00 ..R.....'.....E.
0010 00 28 54 44 40 00 80 06 e3 42 80 64 0b 0d 40 0f .(TD@... .B.d..@.
0020 f7 c8 04 67 00 50 d8 e1 ff d9 53 38 53 86 50 10 ...g.P.. ..S8S.P.
0030 43 a4 24 b6 00 00 C.\$...

TCP连接管理 (续)



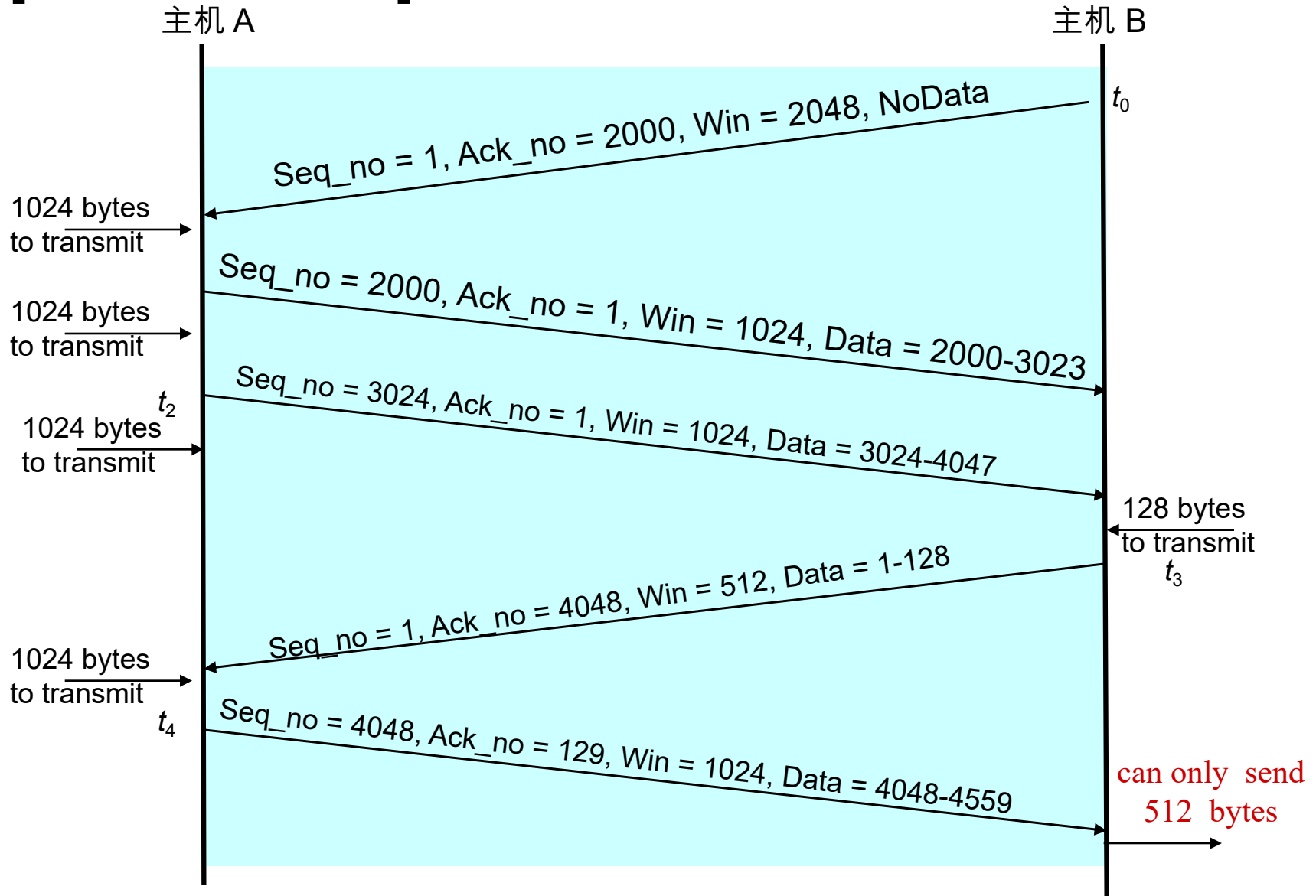
例 [客户端-服务器应用程序]



TCP连接管理 (续)



例 [TCP窗口流控制]



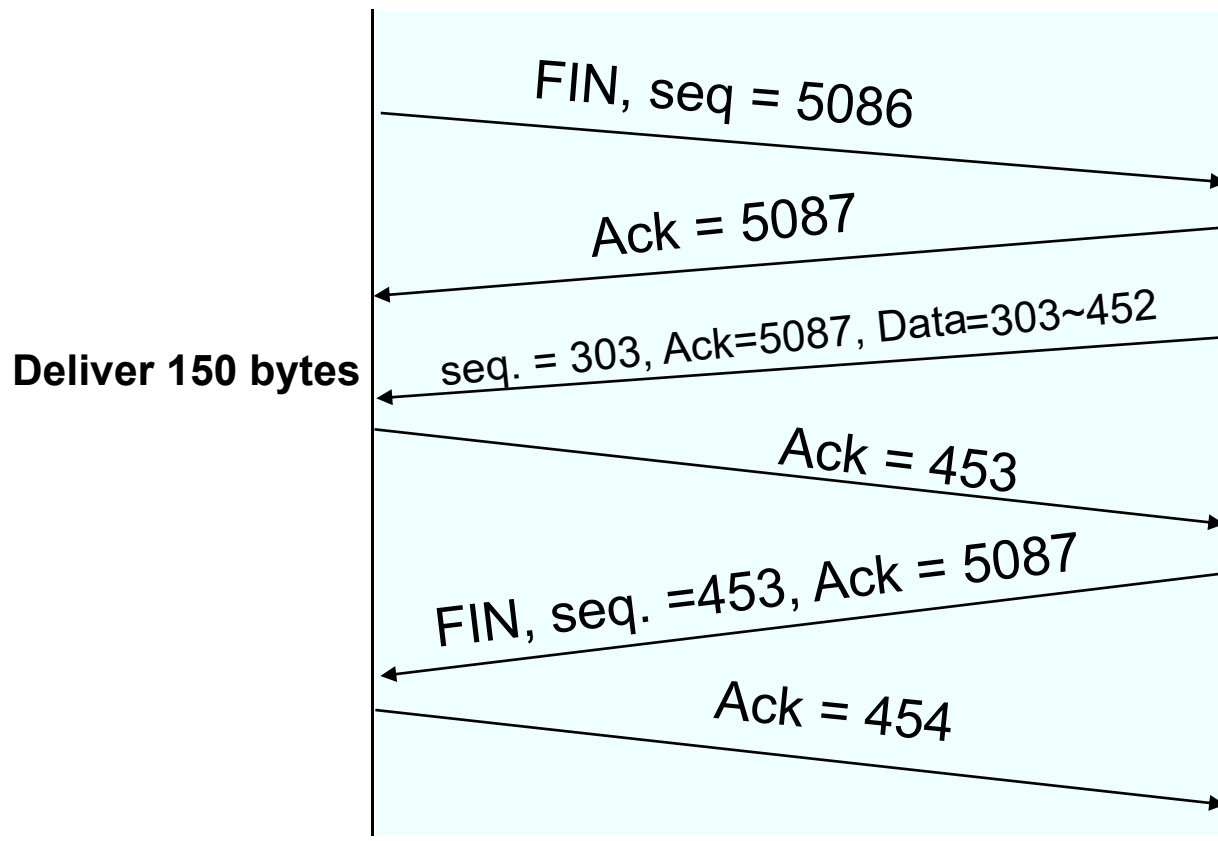
TCP连接管理 (续)



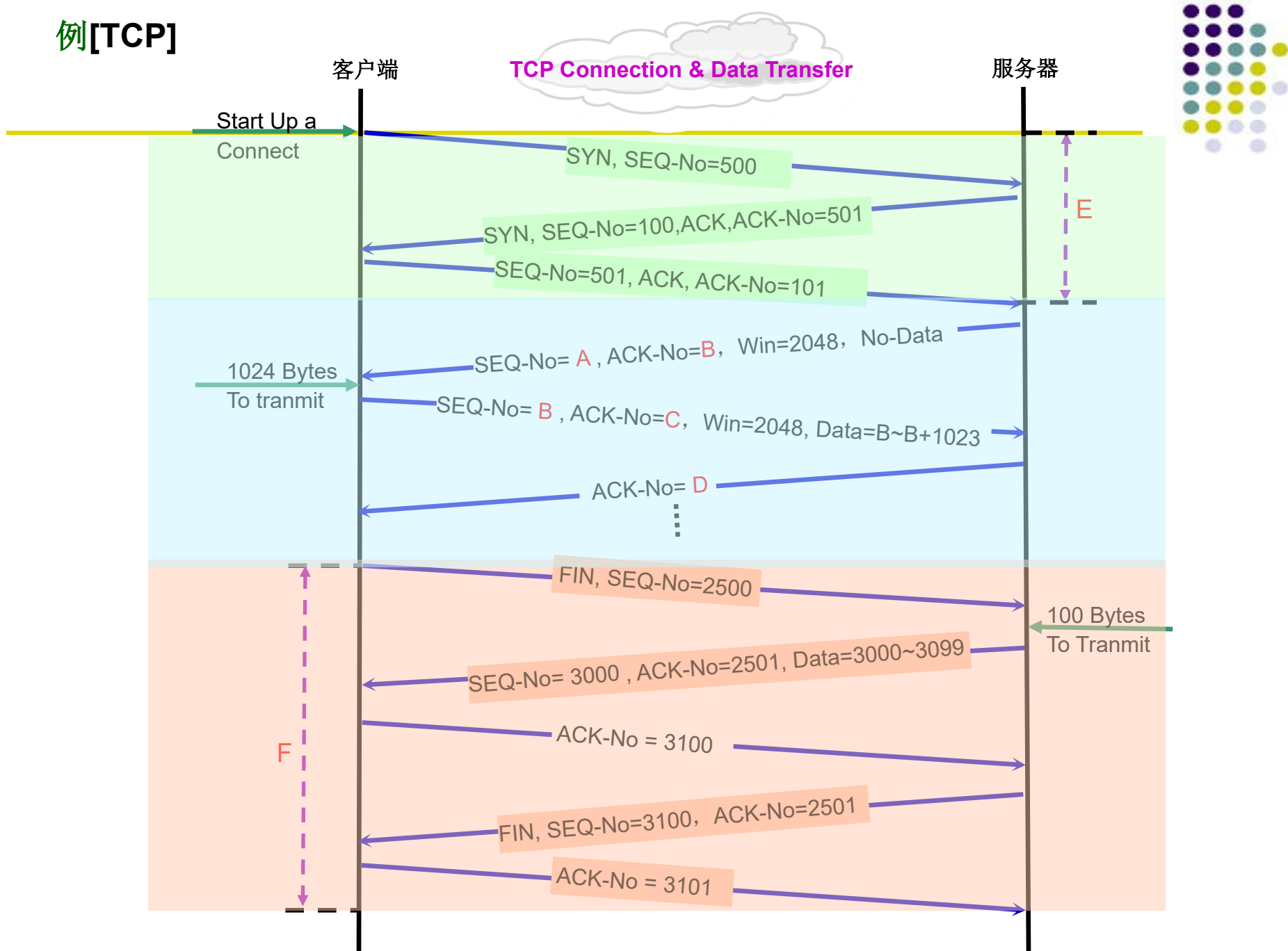
TCP连接关闭 -- “ 温和关闭 ”
-- “ 四次握手 ”

主机 A

主机 B



例[TCP]



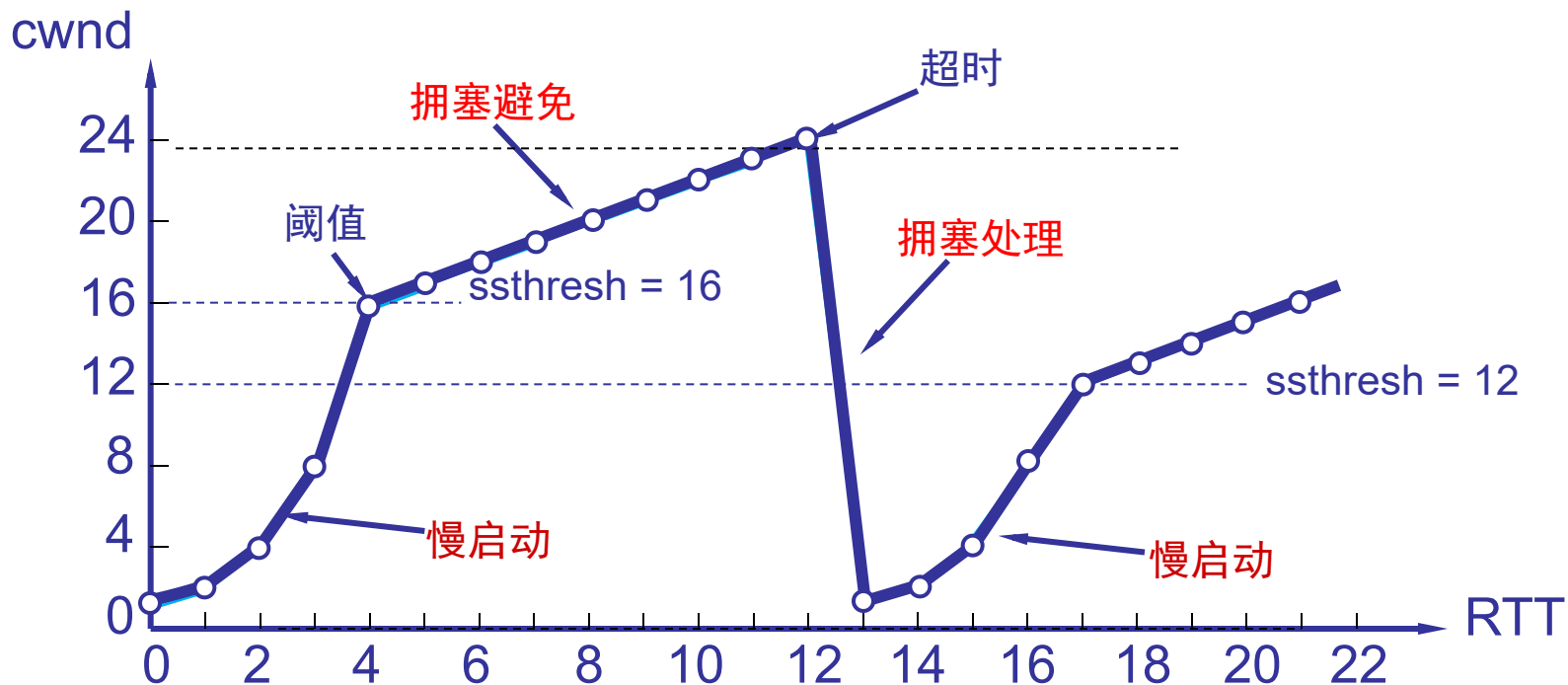
大纲



1. UDP 协议
2. TCP 可靠的流服务
3. TCP 协议
4. TCP 连接管理
5. TCP 拥塞控制



5、TCP拥塞控制



cwnd : 拥塞窗口

$$\text{SendWin} = \text{Min} [\text{rwnd}, \text{cwnd}]$$

TCP状态转移图

