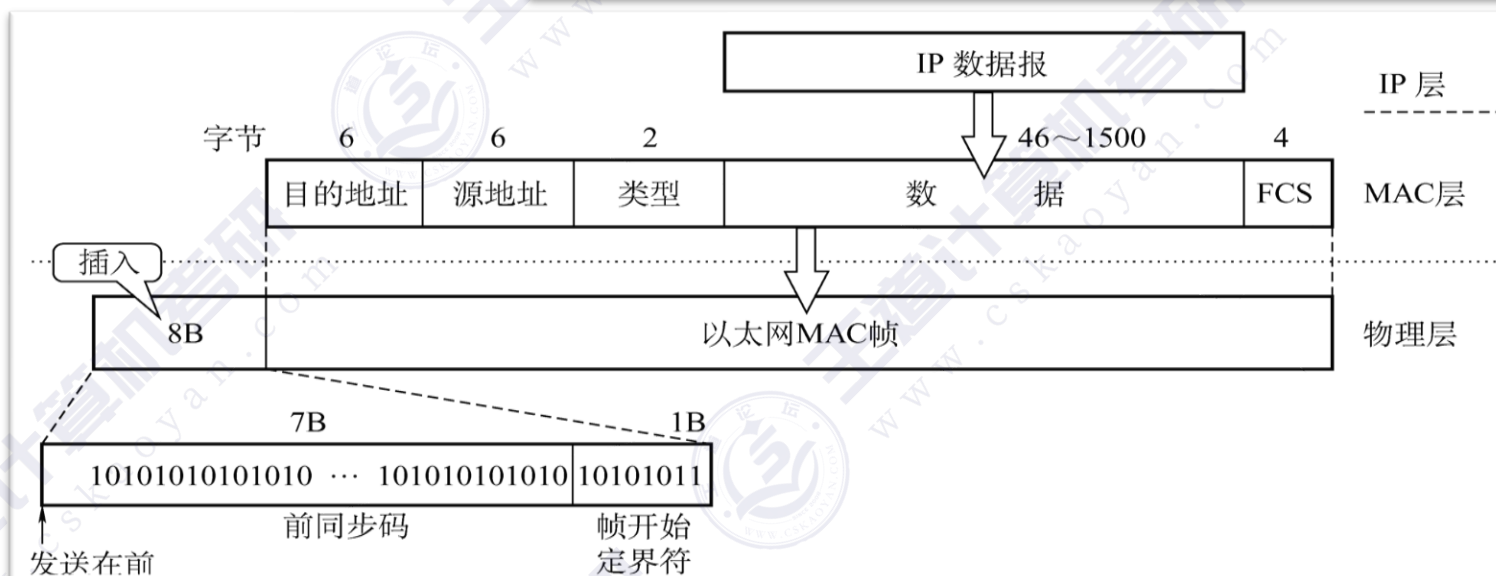
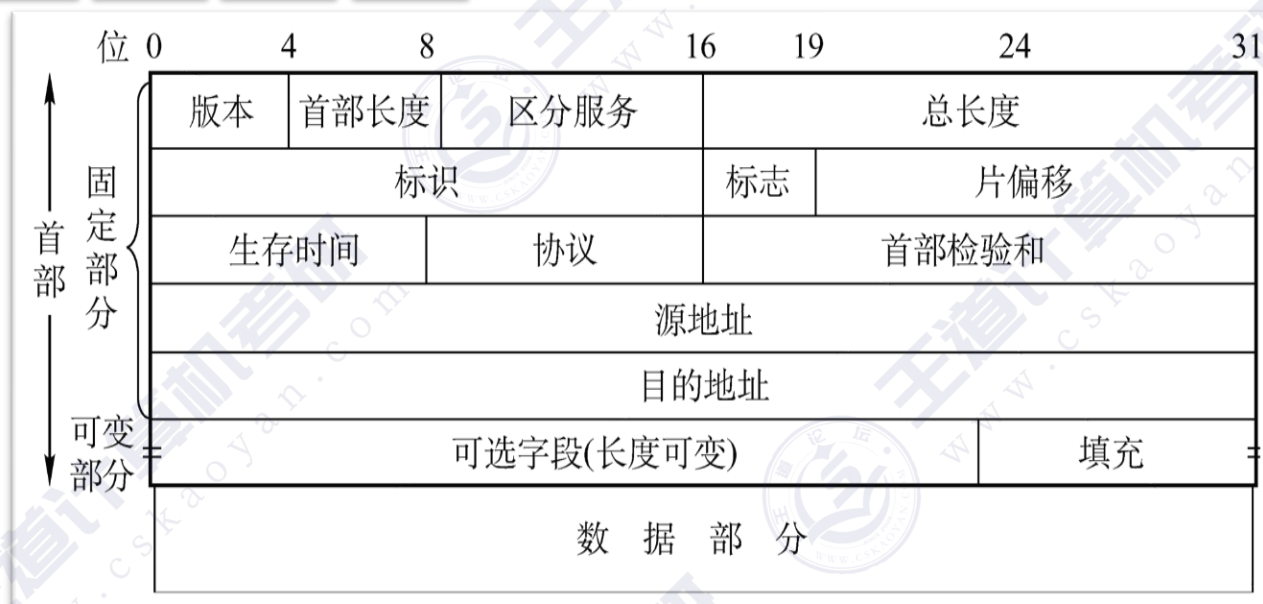
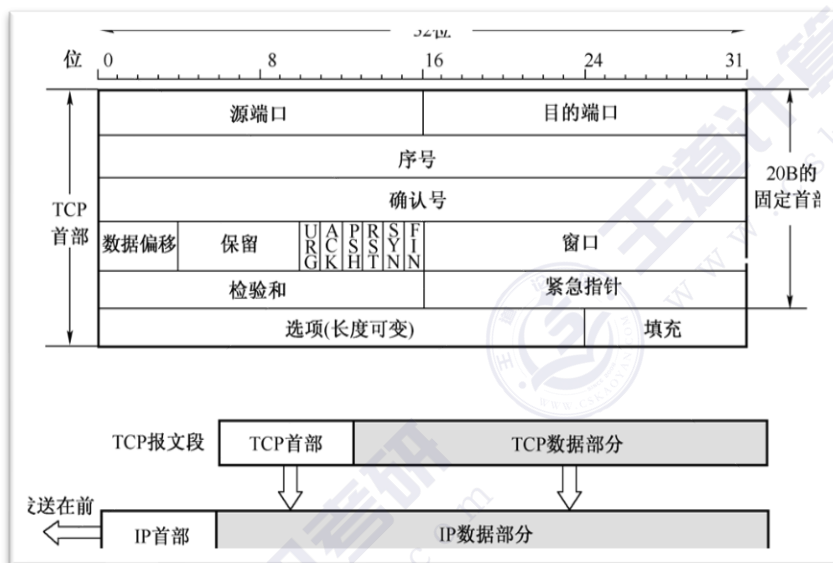


# TCP报文段、IP分组、MAC帧

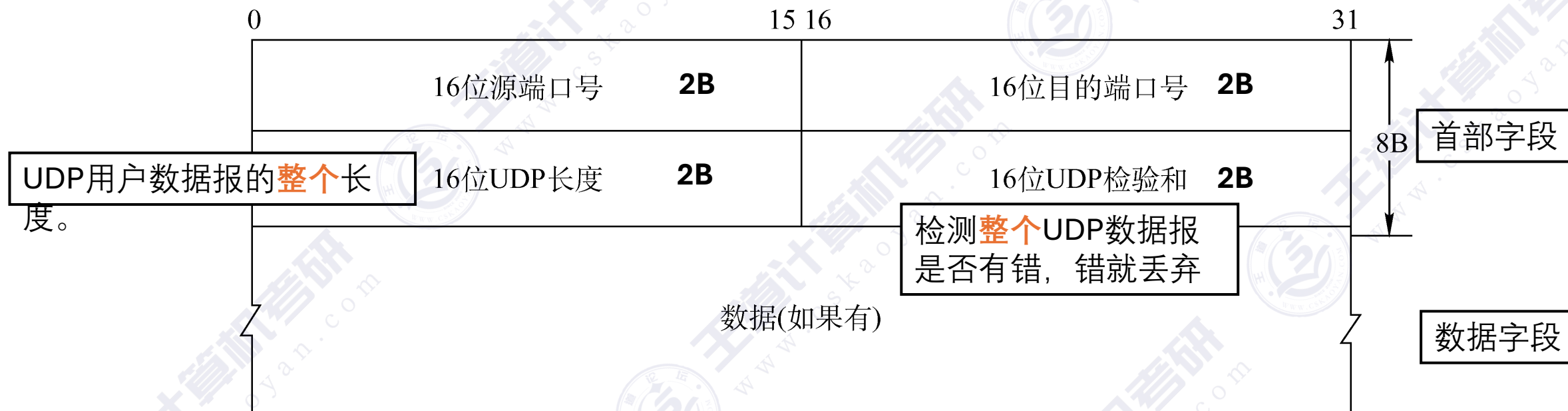


MAC帧首部+尾部=18B，数据部分为46B ~ 1500B

# TCP报文段、IP分组、MAC帧

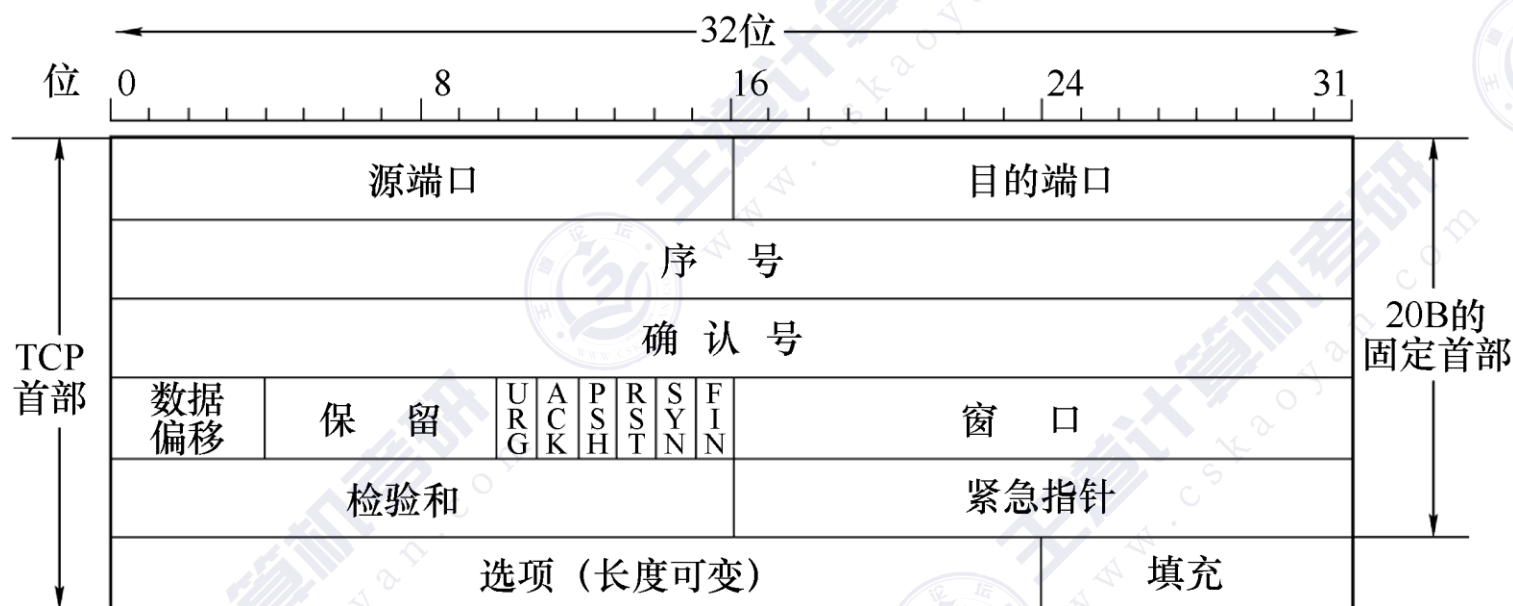


## UDP首部格式



分用时，找不到对应的目的端口号，就丢弃报文，并给发送方发送ICMP“端口不可达”差错报告报文。

## TCP报文段首部格式



### 6个控制位

**紧急位URG**：URG=1时，表明此报文段中有紧急数据，是高优先级的数据，应尽快传送，不用在缓存里排队，配合紧急指针字段使用。

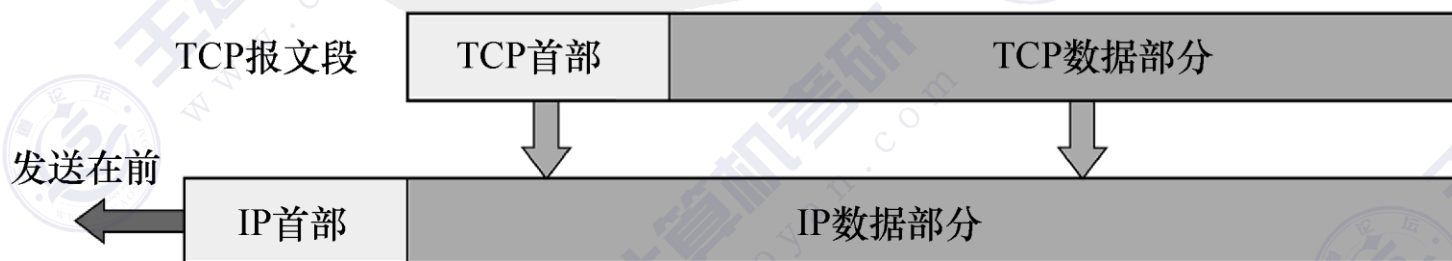
**确认位ACK**：ACK=1时确认号有效，在连接建立后所有传送的报文段都必须把ACK置为1。

**推送位PSH**：PSH=1时，接收方尽快交付接收应用进程，不再等到缓存填满再向上交付。

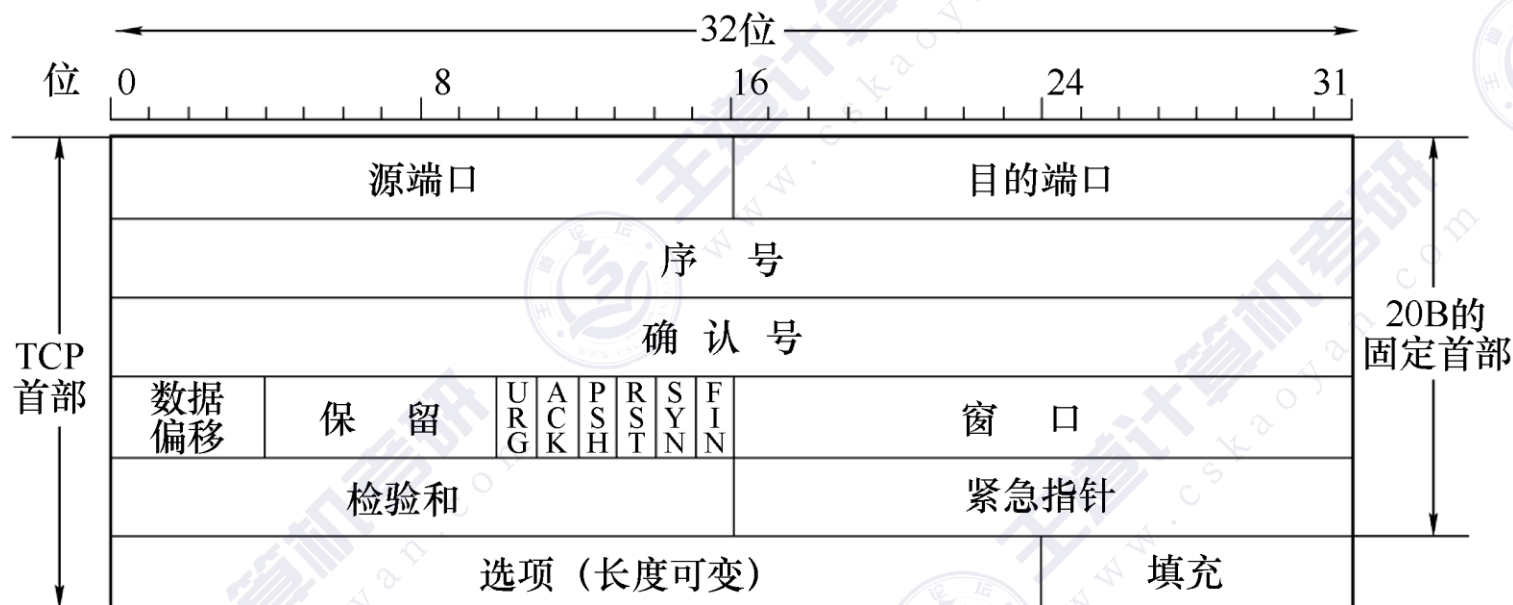
**复位RST**：RST=1时，表明TCP连接中出现严重差错，必须释放连接，然后再重新建立传输链接。

**同步位SYN**：SYN=1时，表明是一个连接请求/连接接受报文。

**终止位FIN**：FIN=1时，表明此报文段发送方数据已发完，要求释放连接。



## TCP报文段首部格式

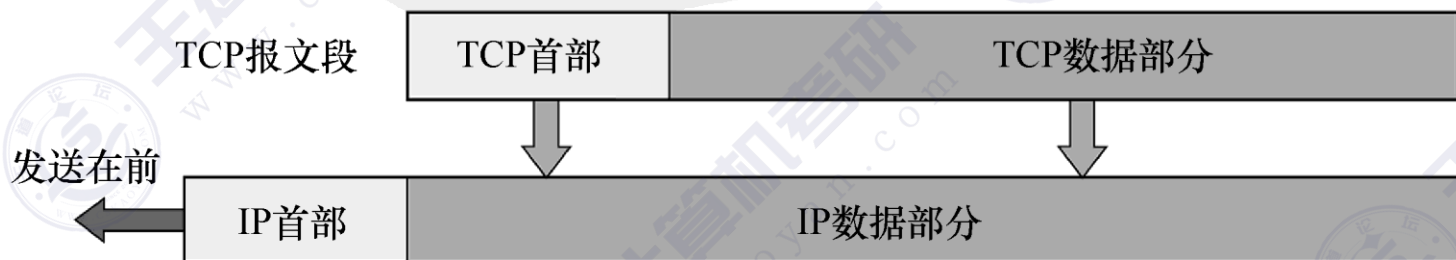


**窗口：**指的是发送本报文段的一方的接收窗口，即现在允许对方发送的数据量。

**检验和：**检验首部+数据，检验时要加上12B伪首部，第四个字段为6。

**紧急指针：**URG=1时才有意义，指出本报文段中紧急数据的字节数。

**选项：**最大报文段长度MSS、窗口扩大、时间戳、选择确认...





## 附：各报文段需要记忆的内容

★	HTTP报文	HTTP报文分为请求报文&响应报文 请求报文： 1.请求行：请求方法（常用get/post）、请求URL、HTTP协议版本 2.首部行 3.请求体/实体主体 响应报文： 1.状态行 2.响应头部 3.响应体
★★	UDP数据报	1.首部8B，由4个字段组成（都是2B） 2.长度字段包括首部+数据部分 3.检验和检验首部+数据部分（可选）
★★★	TCP报文段	1.首部固定部分为20B，最大值为60B（和IP分组一样） 2.源端口和目的端口各占2B 3.序号（本报文段第一个字节的序号）和确认号（期望收到下一个的序号）各占4B 4.数据偏移=首部长度的4B整数倍 5.确认位ACK、同步位SYN、终止位FIN什么时候为0/1 6.窗口字段表示允许对方发送的数据量（流量控制用）
★★★★	IP分组	1.首部固定部分为20B，首部最大值为60B 2.总长度（1）+片偏移的单位（8）+首部长度的4（“一种八片首饰”） 3.标志位MF和DF在分片时的取值 4.生存时间TTL，经过一个路由器减去1，直到为0 5.首部校验和字段只校验首部 6.源地址和目的地址字段长度都为4B
★★★★	MAC帧	1.前同步码8B 2.MAC地址长度6B 3.数据长度为46-1500B，首部和尾部是18B，因此最短帧长64B。

## 特殊IP地址

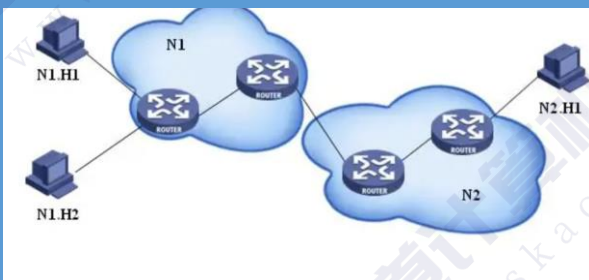
NetID 网络号	HostID主 机号	作为IP分组 源地址	作为IP分组目 的地址	用途
全0	全0	可以	不可以	本网范围内表示主机，路由表中用于表示默认路由 (表示整个Internet网络)
全0	特定值	可以	不可以	表示本网内某个特定主机
全1	全1	不可以	可以	本网广播地址 (路由器不转发)
特定值	全0	不可以	不可以	网络地址，表示一个网络
特定值	全1	不可以	可以	直接广播地址，对特定网络上的所有主机进行广播
127	任何数 (非全0/1)	可以	可以	用于本地软件环回测试，称为环回地址

## 网络层重要协议

### 路由选择协议

- 内部网关协议IGP:  
**RIP、OSPF**
- 外部网关协议EGP:  
**BGP**

路由协议主要作用在路由器上，创建了路由表，描述了网络拓扑结构；路由协议与路由器协同工作，执行路由选择和数据包转发功能。



### IP (Internet Protocol) 协议

- **IPv4**
- **IPv6**

IP协议主要包含三方面内容：IP编址方案、分组封装格式及分组转发规则。

①虽说借助子网化、无类寻址和NAT技术可以提高IP地址使用效率，因特网中IP地址的耗尽仍然是一个没有彻底解决的问题；②IPv4没有提供对实时音频和视频传输这种要求传输最小时延的策略和预留资源支持；③IPv4不能对某些有数据加密和鉴别要求的应用提供支持。为了克服这些缺点，IPv6 (Internet working Protocol version 6) 应运而生。



### 其他重要协议

- 地址解析协议：**ARP**
- 动态主机设置协议：**DHCP**
- 互联网控制消息协议：**ICMP**
  - ICMP差错报文
  - ICMP报告报文
- 组播协议：**IGMP**

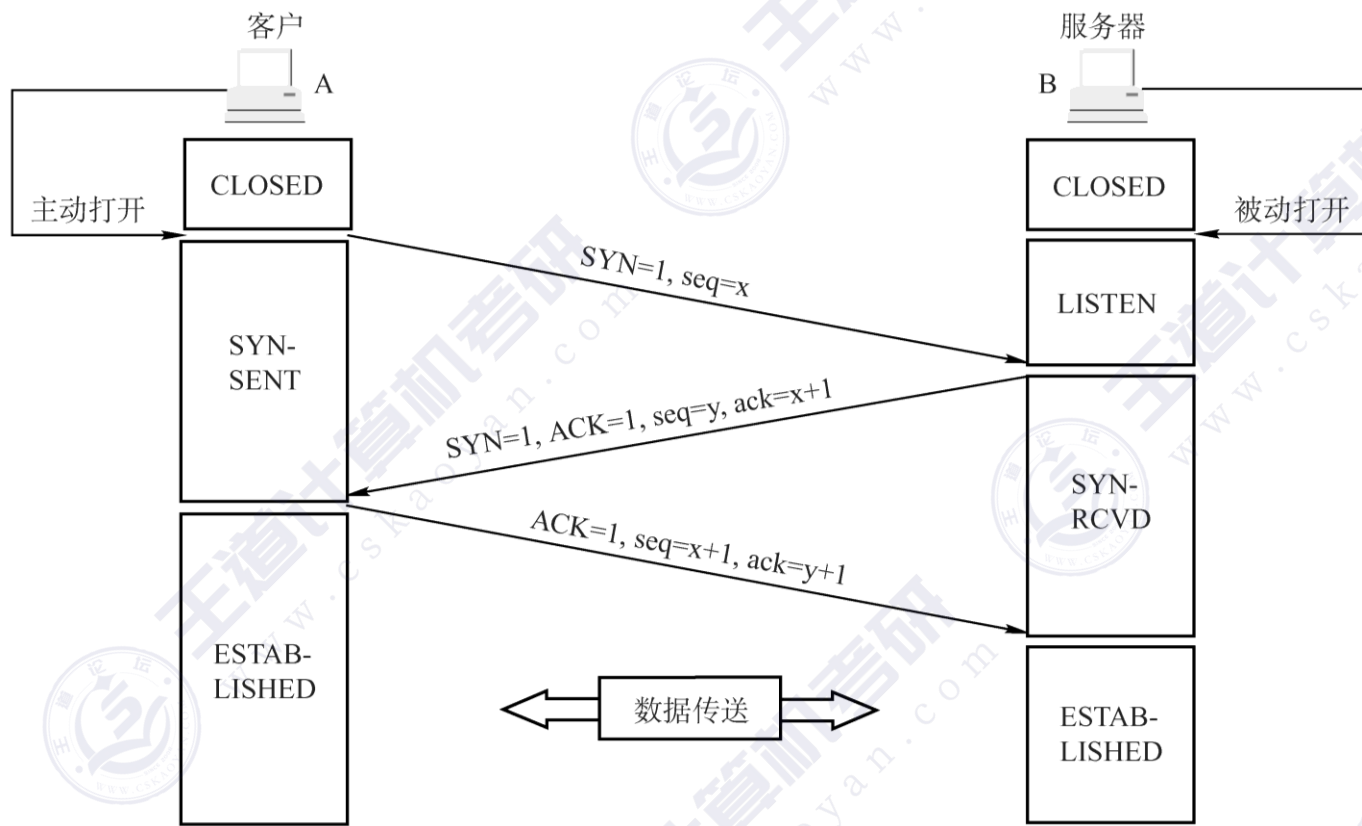
请问你是？





## TCP的连接建立

假设运行在一台主机（客户）上的一个进程想与另一台主机（服务器）上的一个进程建立一条连接，客户应用进程首先通知客户TCP，他想建立一个与服务器上某个进程之间的连接，客户中的TCP会用以下步骤与服务器中的TCP建立一条TCP连接：



### ROUND 1:

客户端发送连接请求报文段，无应用层数据。

$SYN=1, seq=x(\text{随机})$

### ROUND 2:

服务器端为该TCP连接分配缓存和变量，并向客户端返回确认报文段，允许连接，无应用层数据。

$SYN=1, ACK=1, seq=y(\text{随机}), ack=x+1$

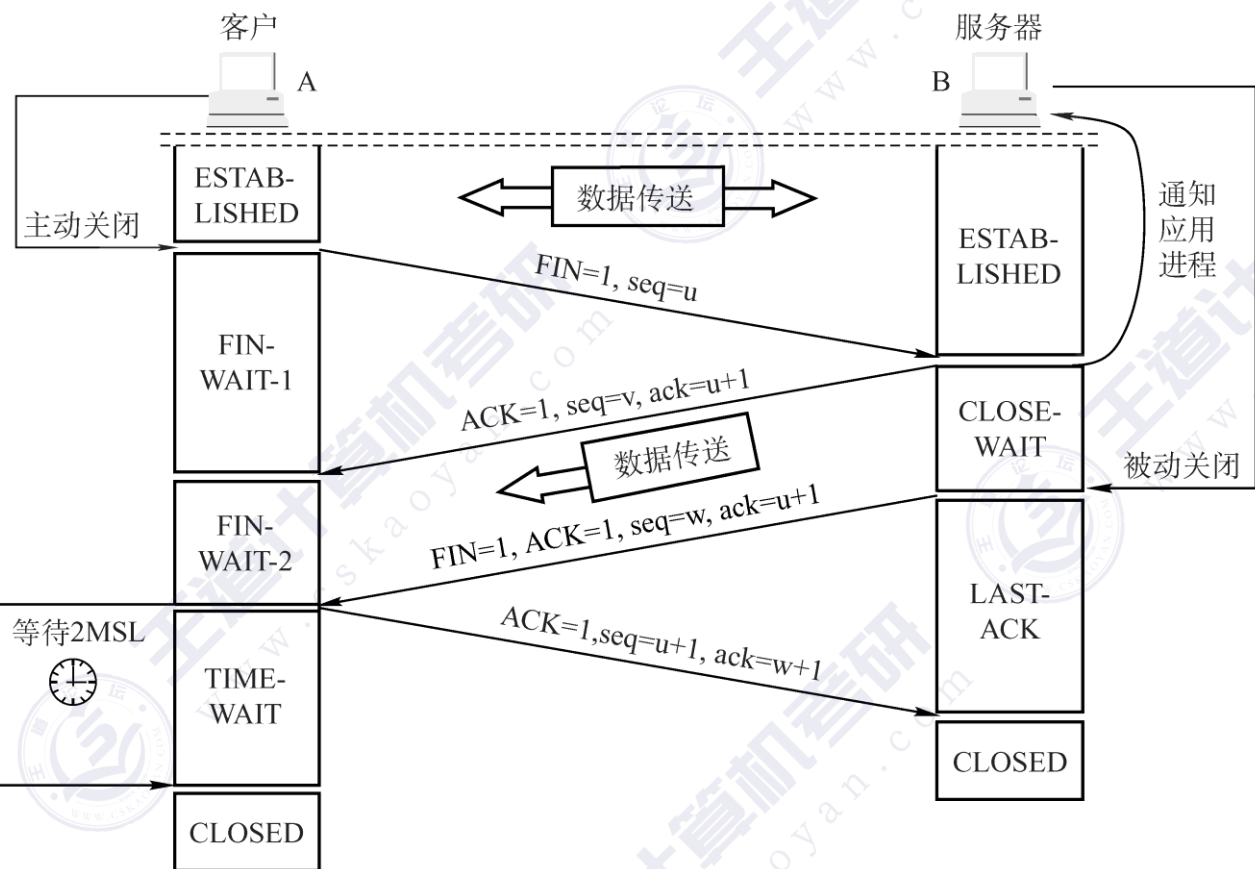
### ROUND 3:

客户端为该TCP连接分配缓存和变量，并向服务器端返回确认的确认，可以携带数据。

$SYN=0, ACK=1, seq=x+1, ack=y+1$

## TCP的连接释放

参与一条TCP连接的两个进程中的任何一个都能终止该连接，连接结束后，主机中的“资源”（缓存和变量）将被释放。



### ROUND 1:

客户端发送**连接释放报文段**，停止发送数据，主动关闭TCP连接。

$\text{FIN}=1, \text{seq}=\text{u}$

### ROUND 2:

服务器端回送一个确认报文段，客户到服务器这个方向的连接就释放了——半关闭状态。

$\text{ACK}=1, \text{seq}=\text{v}, \text{ack}=\text{u}+1$

### ROUND 3:

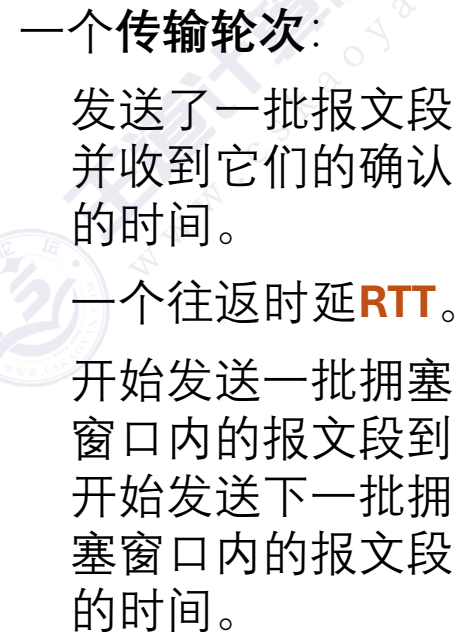
服务器端发完数据，就发出连接释放报文段，主动关闭TCP连接。

$\text{FIN}=1, \text{ACK}=1, \text{seq}=\text{w}, \text{ack}=\text{u}+1$

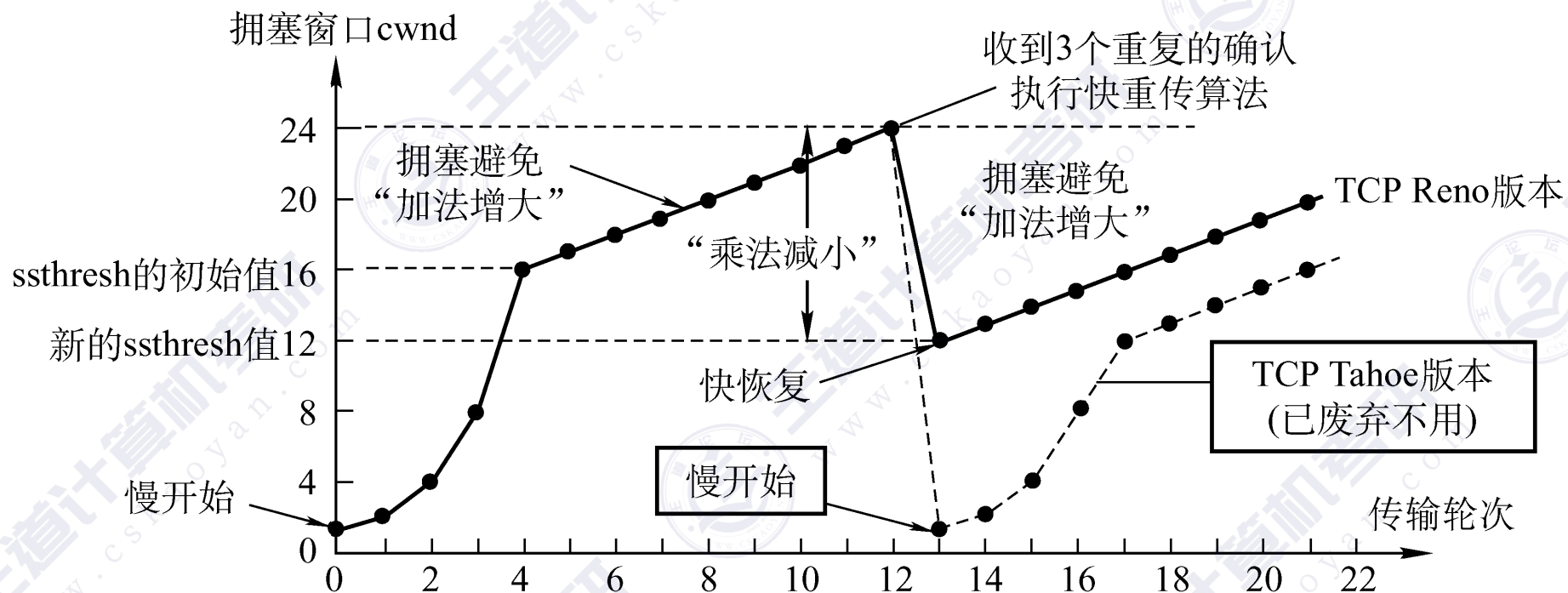
### ROUND 4:

客户端回送一个确认报文段，再等到时间等待计时器设置的2MSL（最长报文段寿命）后，连接彻底关闭。

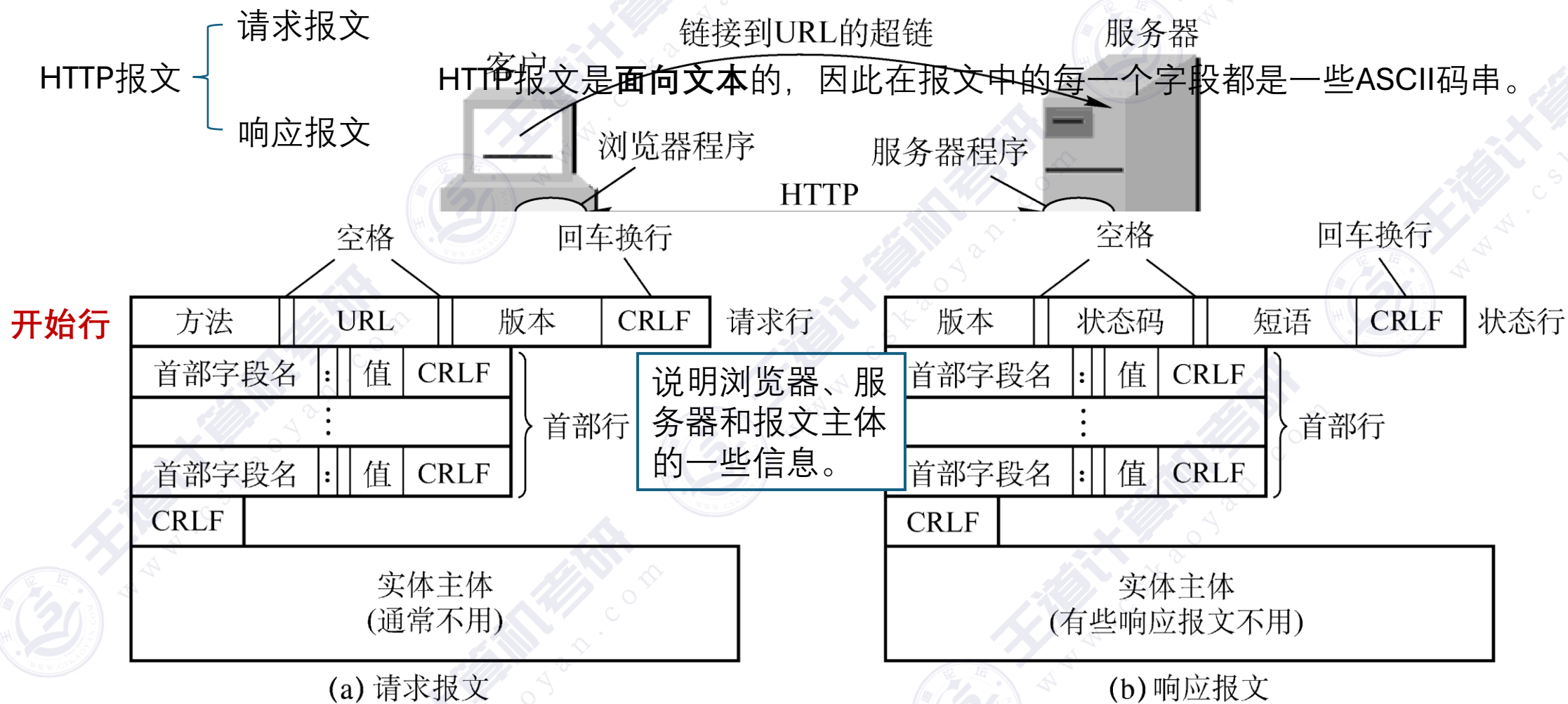
$\text{ACK}=1, \text{seq}=\text{u}+1, \text{ack}=\text{w}+1$



## 快重传和快恢复



# 超文本传输协议HTTP—报文结构





## 超文本传输协议HTTP—报文结构

某浏览器发出的请求报文

```
GET /index.html HTTP/1.1
Host: www.test.edu.cn
Connection: Close
Cookie: 123456
```

状态码:

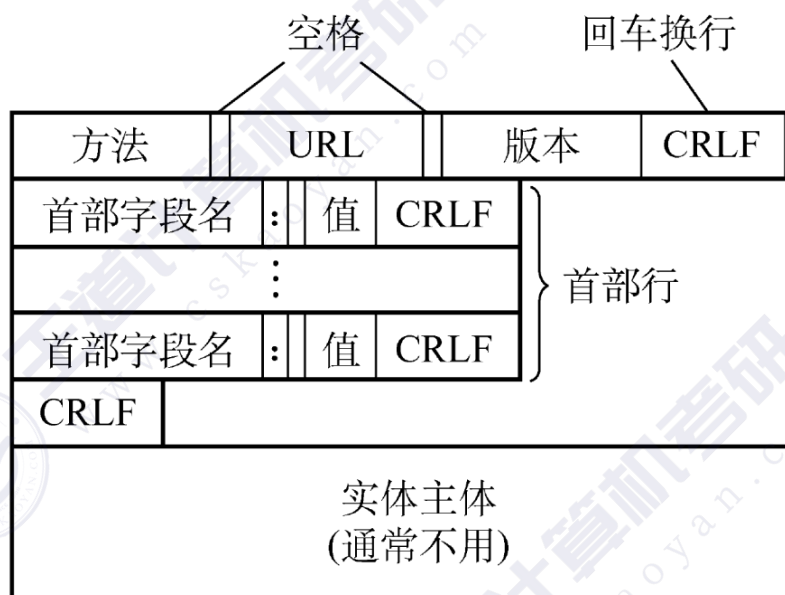
1xx表示通知信息的, 如请求收到了或正在处理。

2xx表示成功, 如接受或知道了。**202 Accepted**

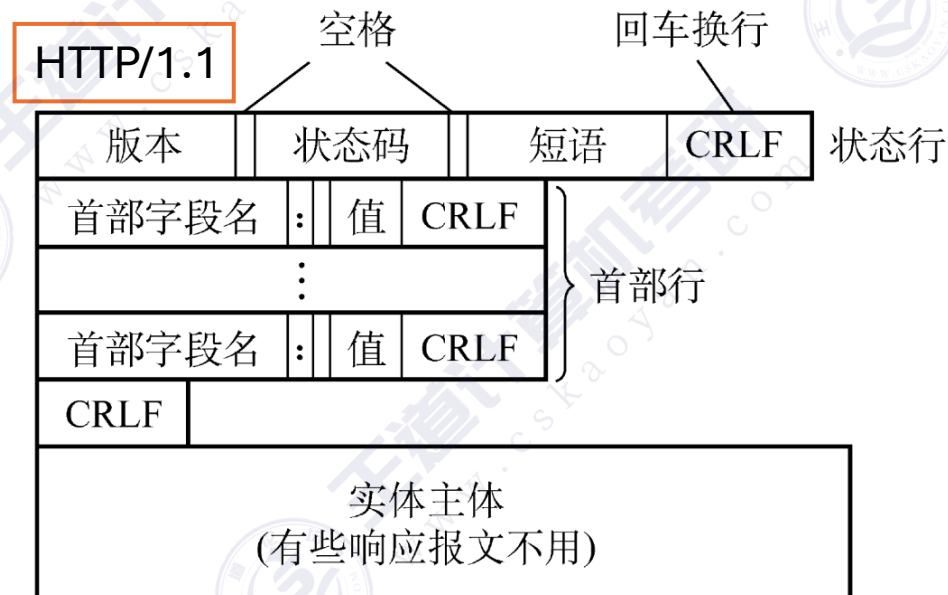
3xx表示重定向, 如要完成请求还必须采取进一步的行动。**301 Moved Permanently**

4xx表示客户的差错, 如请求中有错误的语法或不能完成。**404 Not Found**

5xx表示服务器的差错, 如服务器失效无法完成请求。



(a) 请求报文



(b) 响应报文

# 熟知端口号

应用程序	FTP	TELNET	SMTP	DNS	TFTP	HTTP	SNMP
熟知端口号	21	23	25	53	69	80	161