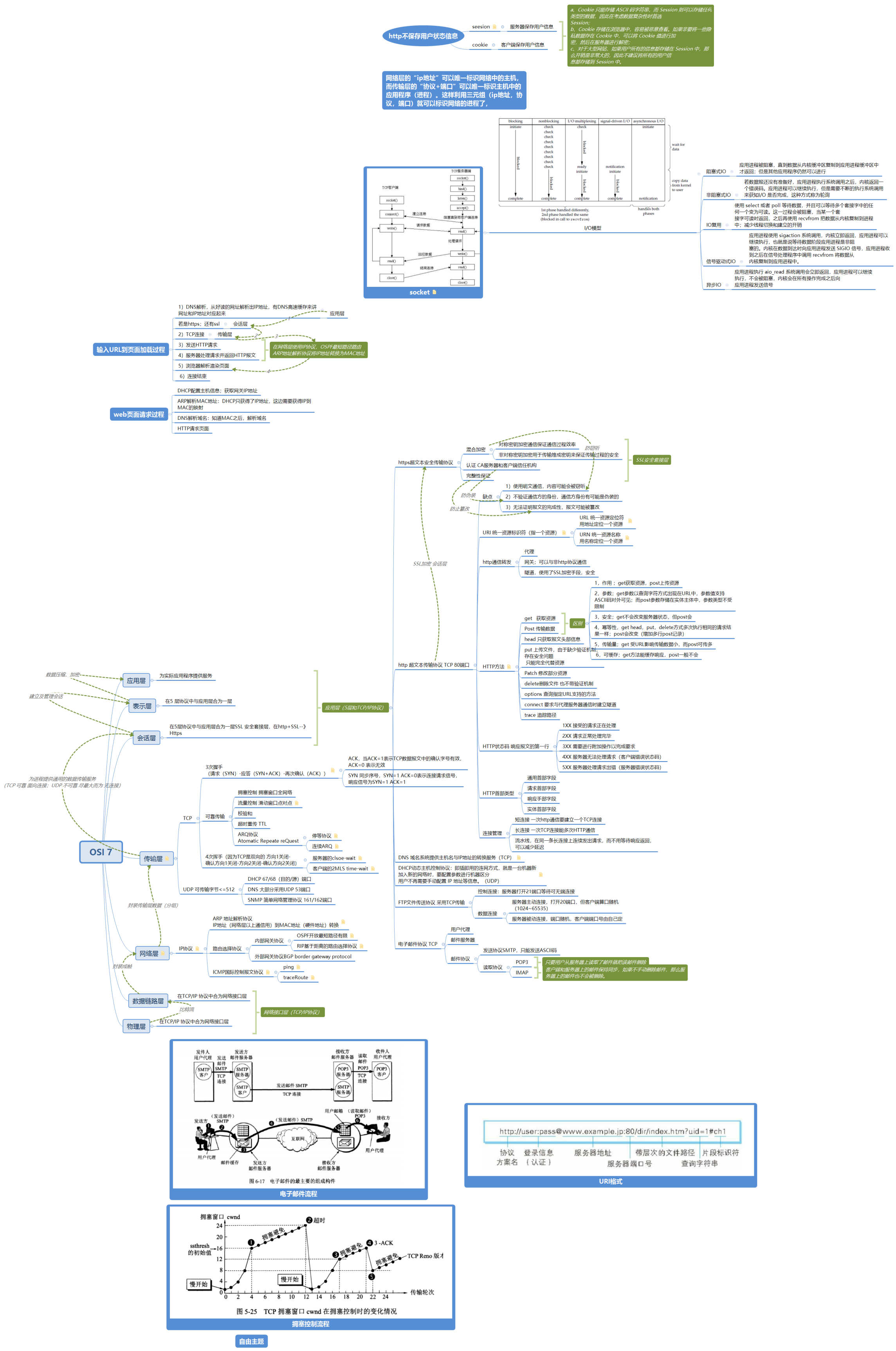
OSI 7


http不保存用户状态信息
seesion
cookie
socket
I/O模型
输入URL到页面加载过程
web页面请求过程
OSI 7
应用层
表示层
会话层
传输层
网络层
数据链路层
物理层
电子邮件流程
URI格式
拥塞控制流程
自由主题

1. 应用层

参见: 表示层 (数据压缩, 加密)

1.1. 为实际应用程序提供服务

2. 表示层

参见: 应用层 (数据压缩, 加密), 会话层 (建立及管理会话)

2.1. 在5 层协议中与应用层合为一层

3. 会话层

参见: 表示层 (建立及管理会话), 传输层 (为进程提供通用的数据传输服务 (TCP 可靠 面向连接;UDP 不可靠 尽最大而为 无连接))

3.1. 在5层协议中与应用层合为一层SSL 安全套接层，在http+SSL--》Https

4. 传输层

参见: 会话层 (为进程提供通用的数据传输服务 (TCP 可靠 面向连接;UDP 不可靠 尽最大而为 无连接)), 网络层 (封装传输层数据(分组))

网络层只把分组发送到目的主机，但是真正通信的不是主机，而是主机中的进程，传输层提供了进程间的逻辑通信。循环数层向高层用户屏蔽了下面网络层的核心细节，是的应用程序看起来像是在两个传输成实体之间的一条端到端的逻辑通信信道。

4.1. TCP

4.1.1. 3次握手

（请求（SYN）-应答（SYN+ACK）-再次确认（ACK））

执行第三次握手是为了***避免打开失效***的连接;

比如连接请求的滞留带来的问题:

客户端发送的连接请求如果在网络中滞留，那么就会隔很长一段时间才能收到服务器端发回的连接确认。客户端等待
一个超时重传时间之后，就会重新请求连接。但是这个滞留的连接请求最后还是会到达服务器，如果不进行三次握
手，那么服务器就会打开两个连接。如果有第三次握手，客户端会忽略服务器之后发送的对滞留连接请求的连接确
认，不进行第三次握手，因此就不会再次打开连接。

ACK，当ACK=1表示TCP数据报文中的确认字号有效，

ACK=0 表示无效

SYN 同步序号，SYN=1 ACK=0表示连接请求信号，

响应信号为SYN=1 ACK=1

4.1.2. 可靠传输

拥塞控制 拥塞窗口全网络

流量控制 滑动窗口点对点

控制发送方的发送速率，
1）流量控制保证接收方来得及接受。接收方发送信息控制发送方的窗口大小
2）拥塞控制是为了保证整个网络，因为出现拥塞数据丢失之后，发送方会重传
恶心循环。（发送方维护拥塞窗口--一个状态变量，实际发送数据还是由发送方窗口决定）

校验和

超时重传 TTL

ARQ协议

Atomatic Repeate reQuest

停等协议

每发完一个分组就停止发送，等待对方确认（回复ACK）。如果过了一段时间（超时时间后），还是没有收到 ACK 确认，说明没有发送成功，需要重新发送，直到收到确认后再发下一个分组；

连续ARQ

提高信道利用率。发送方维持一个发送窗口，凡位于发送窗口内的分组可以连续发送出去，而不需要等待对方确认。接收方一般采用累计确认，对按序到达的最后一个分组发送确认，表明到这个分组为止的所有分组都已经正确收到了。

4.1.3. 4次挥手（因为TCP是双向的 方向1关闭-确认方向1关闭-方向2关闭-确认方向2关闭）

服务器的clsoe-wait

客户端发送了 FIN 连接释放报文之后，服务器收到了这个报文，就进入了 CLOSE-WAIT 状态。这个状态是为了让服务器端发送还未传送完毕的数据，传送完毕之后，服务器会发送 FIN 连接释放

客户端的2MLS time-wait

1）确保最后一个确认报文能够到达。如果 B 没收到 A 发送来的确认报文，那么就会重新发送连接释放请求报文，
2）等待一段时间就是为了处理这种情况的发生。
等待一段时间是为了让本连接持续时间内所产生的所有报文都从网络中消失，使得下一个新的连接不会出现旧
的连接请求报文。

4.2. UDP 可传输字节<=512

4.2.1. DHCP 67/68（目的/源）端口

4.2.2. DNS 大部分采用UDP 53端口

4.2.3. SNMP 简单网络管理协议 161/162端口

5. 网络层

参见: 传输层 (封装传输层数据（分组）), 数据链路层 (封装成帧)

网络层到传输层：就是将原来网络层的IP数据包作为网络层的数据部分，然后再前段加入对应的TCP/UDP首部信息； 再反转的时候 就是拆下对应的首部信息 然后

5.1. IP协议

在通信过程中IP数据包的原地址和目的地址
始终不变, 而MAC地址随着链路改变而改变。
ARP实现了IP地址到MAC地址的映射

5.1.1. ARP 地址解析协议

IP地址（网络层以上通信用）到MAC地址（硬件地址）转换

A主机可以通过ARP请求获取别的主机B的MAC地址

5.1.2. 路由选择协议

内部网关协议

OSPF开放最短路径有限

有显示有Dijkstra最短路劲算法

RIP基于距离的路由选择协议

距离指跳数，最多15跳，超过15跳表示不可达。按照固定的时间间隔仅和相邻的路由器交换自己的路由表

外部网关协议BGP border gateway protocol

5.1.3. ICMP国际控制报文协议

主要是为了更有效的转发IP数据报和提高交付成功
ICMP报文分为差错报告报文和询问报文

ping

测试两台主机之间的连通性

traceRoute

用来跟踪一个分组从源点到终点的路径

6. 数据链路层

参见: 网络层 (封装成帧), 物理层 (比特流)

6.1. 在TCP/IP 协议中合为网络接口层

7. 物理层

参见: 数据链路层 (比特流)

7.1. 在TCP/IP 协议中合为网络接口层

应用层（5层和TCP/IP协议）(应用层, 表示层, 会话层)

7.2. https超文本安全传输协议

参见: http 超文本传输协议 TCP 80端口 (SSL加密 会话层)

7.2.1. 混合加密

参见: 1）使用明文通信，内容可能会被窃听 (防窃听)

对称密钥加密通信保证通信过程效率

非对称密钥加密用于传输堆成密钥来保证传输过程的安全

7.2.2. 认证 CA服务器和客户端信任机构

参见: 2）不验证通信方的身份，通信方身份有可能是伪装的 (防伪装)

7.2.3. 完整性保证

参见: 3）无法证明报文的完成性，报文可能被篡改 (防止篡改)

7.3. http 超文本传输协议 TCP 80端口

参见: https超文本安全传输协议 (SSL加密 会话层)

7.3.1. 缺点

1）使用明文通信，内容可能会被窃听

参见: 混合加密 (防窃听)

2）不验证通信方的身份，通信方身份有可能是伪装的

参见: 认证 CA服务器和客户端信任机构 (防伪装)

3）无法证明报文的完成性，报文可能被篡改

参见: 完整性保证 (防止篡改)

7.3.2. URI 统一资源标识符（指一个资源）

URI 就是由某个协议方案表示的资源的定位标识符。协议方案是指访问资源所使用的协议类型名称。如HTTP协议

URL 统一资源定位符

用地址定位一个资源

一种具体的 URI，即 URL 可以用来标识一个资源，而且还指名了如何 locate（定位）这个资源。通俗的讲，URL 是 Internet 上用来描述资源的字符串，标识了一个互联网资源，并指定了对其进行操作或者获取资源的方法。目前最大的缺点是当信息资源的存放地点发生变化时，必须对 URL 作出相应的改变。

URN 统一资源名称

用名称定位一个资源

即通过名称来标识资源，不依赖于位置，并且有可能减少失效链接个个数。用于标识唯一书目的 ISBN 系统是一个典型的 URN 使用范例。所以不限定互联网使用

7.3.3. http通信转发

代理

网关；可以与非http协议通信

隧道，使用了SSL加密手段，安全

7.3.4. HTTP方法

在请求报文的第一行显示了方法字段

get 获取资源

Post 传输数据

head 只获取报文头部信息

put 上传文件，由于缺少验证机制，
存在安全问题
只能完全代替资源

Patch 修改部分资源

delete删除文件 也不带验证机制

options 查询指定URL支持的方法

connect 要求与代理服务器通信时建立隧道

trace 追踪路径

区别(get 获取资源, Post 传输数据)

1，作用 ；get获取资源，post上传资源

2，参数；get参数以查询字符方式出现在URL中，参数值支持ASCII码对外可见；而post参数存储在实体主体中，参数类型不受限制

3，安全；get不会改变服务器状态，但post会

4，幂等性，get
head，put，delete方式多次执行相同的请求结果一样；post会改变（增加多行post记录）

5，传输量；get 受URL影响传输数据小，而post可传多

6，可缓存；get方法能缓存响应，post一般不会

7.3.5. HTTP状态码 响应报文的第一行

1XX 接受的请求正在处理

2XX 请求正常处理完毕

3XX 需要进行附加操作以完成要求

4XX 服务器无法处理请求（客户端错误状态码）

5XX 服务器处理请求出错（服务器错误状态码）

7.3.6. HTTP首部类型

通用首部字段

请求首部字段

响应手部字段

实体首部字段

7.3.7. 连接管理

短连接 一次http通信要建立一个TCP连接

长连接 一次TCP连接能多次HTTP通信

流水线，在同一条长连接上连续发出请求，而不用等待响应返回，可以减少延迟

7.4. DNS 域名系统提供主机名与IP地址的转换服务（TCP）

分为（根域名 顶级域名和二级域名）3层
一般只用UDP(域名系统自己处理超时和重传保证可靠性)
但是在响应字节大于512, 以及区域传输时采用TCP传输

7.5. DHCP动态主机控制协议：即插即用的连网方式，就是一台机器新加入新的网络时，要配置参数进行机器区分
用户不再需要手动配置 IP 地址等信息。（UDP）

客服端DIscover--1个或多个服务端offer--客户端request（选择合适的DPCH服务器）--服务端ack

7.6. FTP文件传送协议 采用TCP传输

7.6.1. 控制连接：服务器打开21端口等待可无端连接

7.6.2. 数据连接

服务器主动连接，打开20端口，但客户端算口随机（1024~65535）

服务器被动连接，端口随机，客户端端口号由自己定

7.7. 电子邮件协议 TCP

7.7.1. 用户代理

7.7.2. 邮件服务器

7.7.3. 邮件协议

发送协议**SMTP**，只能发送**ASCII**码

读取协议

POP3

IMAP

只要用户从服务器上读取了邮件就把该邮件删除(**POP3**)

客户端和服务器上的邮件保持同步，如果不手动删除邮件，那么服务器上的邮件也不会被删除。(**IMAP**)

SSL安全套接层(**https**超文本安全传输协议)

网络接口层（**TCP/IP**协议）(数据链路层, 物理层)

电子邮件流程

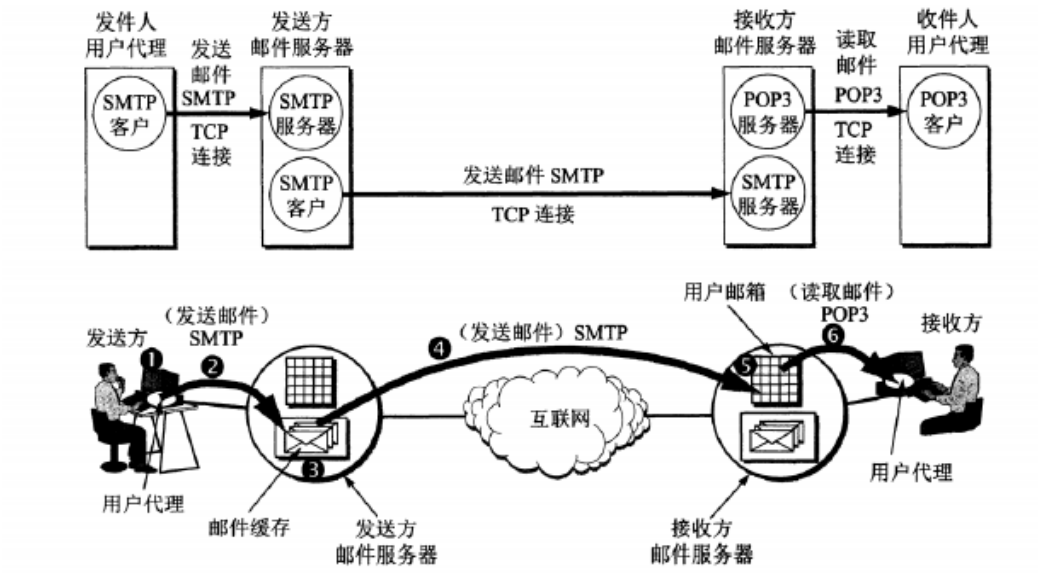


图 6-17　电子邮件的最主要的组成构件

拥塞控制流程

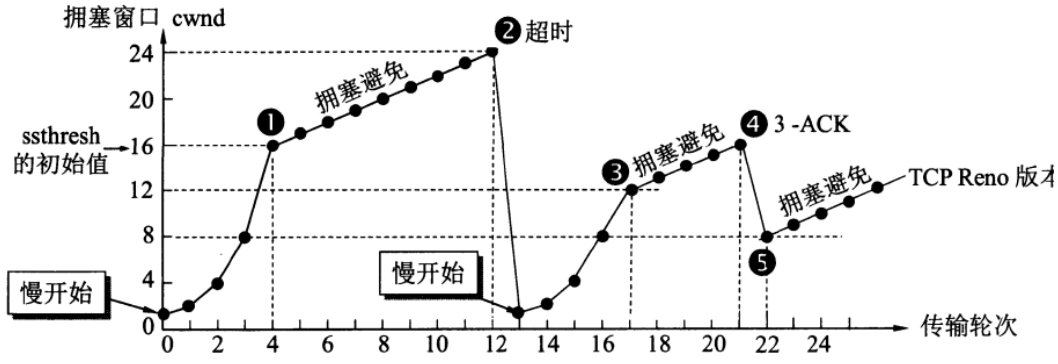


图 5-25　TCP 拥塞窗口 cwnd 在拥塞控制时的变化情况

自由主题

web页面请求过程

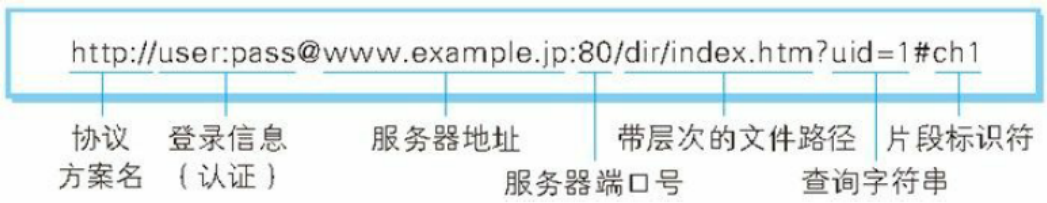
DHCP配置主机信息；获取网关IP地址

ARP解析MAC地址：DHCP只获得了IP地址，这边需要获得IP到MAC的映射

DNS解析域名：知道MAC之后，解析域名

HTTP请求页面

URI格式



输入URL到页面加载过程

1）DNS解析，从好读的网址解析出IP地址，有DNS高速缓存来讲网址和IP地址对应起来

应用层

参见: 会话层 (1)

若是https；还有ssl

会话层

参见: 应用层 (1), 传输层 (2)

2）TCP连接

传输层

参见: 会话层 (2), 在网络层使用IP协议，OSPF最短路径路由

ARP地址解析协议将IP地址转换为MAC地址 (3)

3）发送HTTP请求

4）服务器处理请求并返回HTTP报文

5）浏览器解析渲染页面

参见: 在网络层使用IP协议，OSPF最短路径路由

ARP地址解析协议将IP地址转换为MAC地址 (4)

6）连接结束

在网络层使用IP协议，OSPF最短路径路由

ARP地址解析协议将IP地址转换为MAC地址(3）发送HTTP请求,

4）服务器处理请求并返回HTTP报文)

参见: 传输层 (3), 5）浏览器解析渲染页面 (4)

http不保存用户状态信息

seesion

1、客户端存储服务器发来的seesionID；用户进行登录时，用户提交包含用户名和密码的表单，放入HTTP 请求报文中；

服务器验证该用户名和密码，如果正确则把用户信息存储到 Redis 中，它在 Redis 中的 Key 称为Session ID；

服务器返回的响应报文的 Set-Cookie 首部字段包含了这个 Session ID，客户端收到响应报文之后将该Cookie

值存入浏览器中；

客户端之后对同一个服务器进行请求时会包含该 Cookie 值，服务器收到之后提取出 Session ID，从Redis 中取

出用户信息，继续之前的业务操作。

2；客服端存储服务器发来的加密的数据

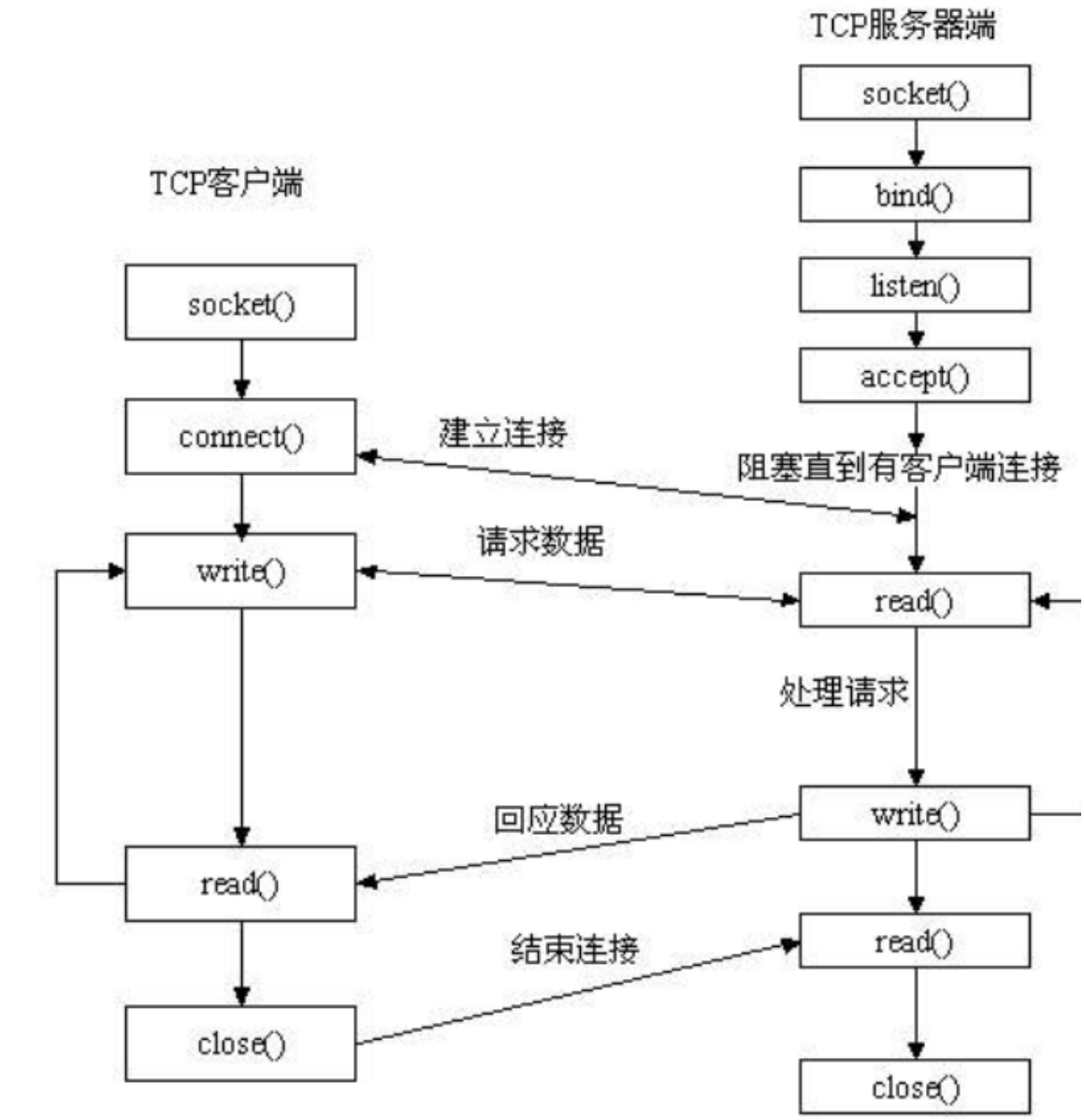
服务器保存用户信息

cookie

客户端保存用户信息

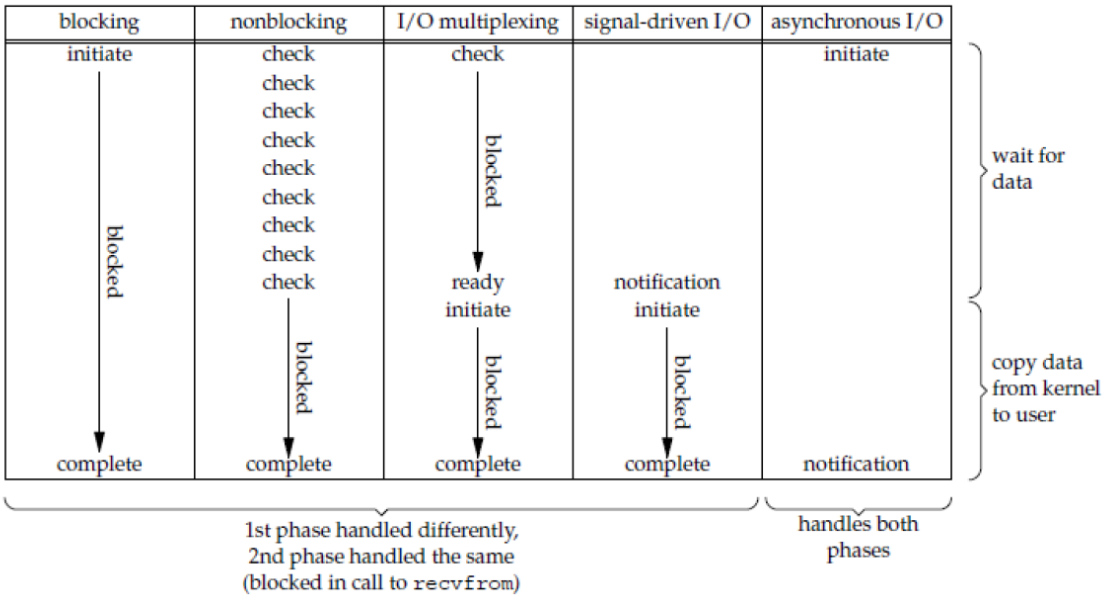
a、Cookie 只能存储 ASCII 码字符串，而 Session
则可以存储任何类型的数据，因此在考虑数据复杂性时首选
Session；
b、Cookie 存储在浏览器中，容易被恶意查看。如果非要将一些隐私数据存在
Cookie 中，可以将 Cookie 值进行加
密，然后在服务器进行解密；
c、对于大型网站，如果用户所有的信息都存储在 Session
中，那么开销是非常大的，因此不建议将所有的用户信
息都存储到 Session 中。(seesion, cookie)

socket



socket本质是**编程接口**(API)，**它把**TCP/IP层**复**杂**的操作抽象**为**几个**简单**的接口**供**应用层调**用以**实现进**程在网**络**中通信**。TCP/IP**只是一个**协议栈，**必**须**要具体**实现，**同**时还**要提供**对**外的操作接口**（API），这**就是**Socket**接口**。**通**过Socket,**我**们**才能使用**TCP/IP协议。

I/O模型



阻塞式IO

应用进程被阻塞，直到数据从内核缓冲区复制到应用进程缓冲区中才返回；但是其他应用程序仍然可以进行

非阻塞式IO

若数据报还没有准备好，应用进程执行系统调用之后，内核返回一个错误码。应用进程可以继续执行，但是需要不断的执行系统调用来获知I/O是否完成，这种方式称为轮询

IO复用

使用 select 或者 poll

等待数据，并且可以等待多个套接字中的任何一个变为可读。这一过程会被阻塞，当某一个套

接字可读时返回，之后再使用 recvfrom

把数据从内核复制到进程中；减少线程切换和建立的开销

信号驱动式IO

应用进程使用 sigaction

系统调用，内核立即返回，应用进程可以继续执行，也就是说等待数据阶段应用进程是非阻

塞的。内核在数据到达时向应用进程发送 SIGIO

信号，应用进程收到之后在信号处理程序中调用 recvfrom 将数据从

内核复制到应用进程中。

异步IO

应用进程执行 aio_read

系统调用会立即返回，应用进程可以继续执行，不会被阻塞，内核会在所有操作完成之后向

应用进程发送信号

网络层的“ip地址”可以唯一标识网络中的主机，而传输层的“协议+端口”可以唯一标识主机中的应用程序（进程）。这样利用三元组（ip地址，协议，端口）就可以标识网络的进程了，