1. OSI 模型与 TCP/IP 模型的结构功能，都有那些协议。

   OSI（7 层）：物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。
   TCP/IP（4 层）：网络接口层、网际层、传输层、应用层。
   五层协议：物理层、数据链路层、网络层、传输层、应用层。

   应用层：通过进程间通信来实现特定的网络应用。应用层协议有：http、ftp、DNS..
   传输层：为两进程间通信提供数据传输服务。常见传输协议：TCP 面向链接、UDP..
   网络层：负责处理网络上面不同主机间的通信服务。传输单位 IP 数据报常见协议 IP。
   数据链路层：将 IP 数据报封装成帧，在相邻节点间的链路上传输帧。
   物理层：以二进制形式在物理介质上传输数据。

2. TCP 和 UDP 的区别

   TCP：可靠的、面向连接的传输协议，传输速度比较慢、适合一些对数据有完整性、安全性需求的传输。
   UDP：不可靠、面向无连接的传输协议，传输速度比较快、适合一些

2.1 tcp 如何保证可靠的，丢包如何处理？

Tcp 是面向连接的，只有在双方建立了连接后才能够进行数据的交互。
而且每一次数据接收都会进行数据包确认，超时或数据未接收都会导致重传。

3. TCP 报文结构

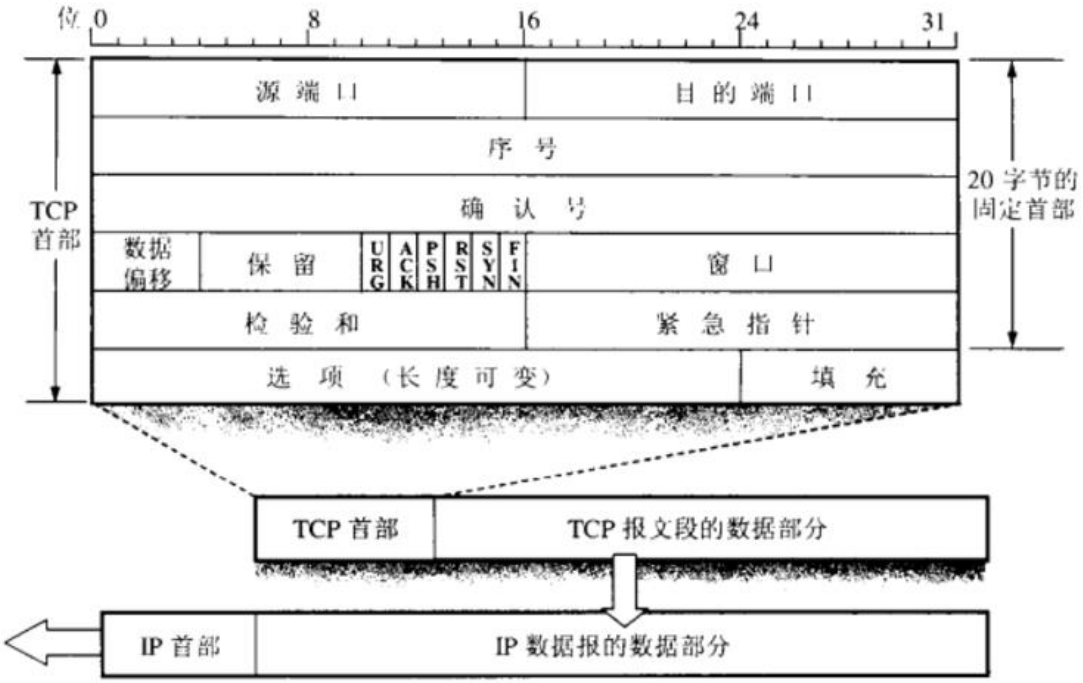


图 5-14　TCP 报文段的首部格式

4. TCP 的三次握手和四次挥手过程，各状态名称和含义，TIME-WAIT 作用

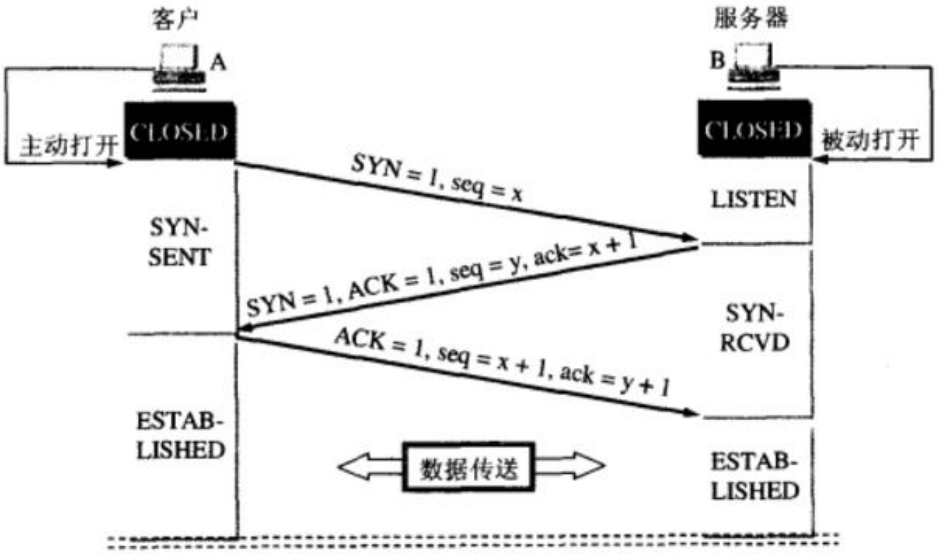


图 5-31　用三次握手建立 TCP 连接

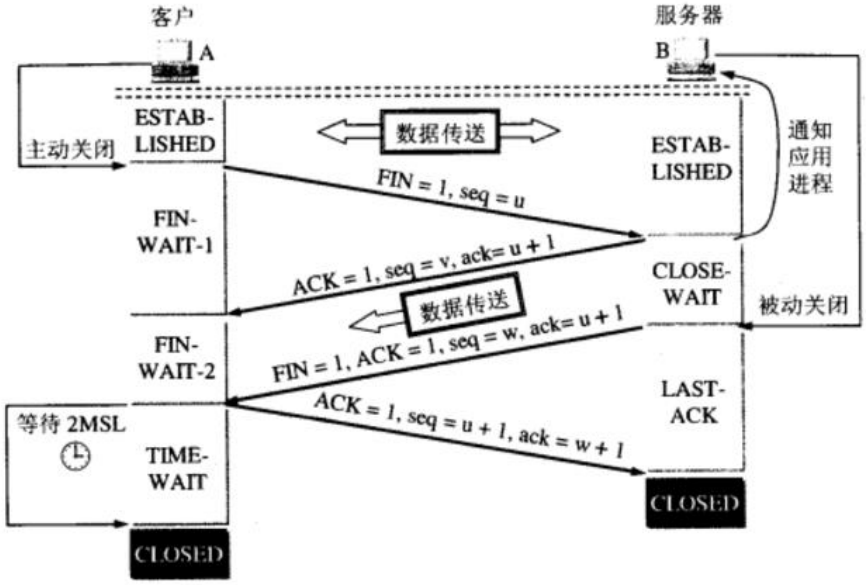


图 5-32　TCP 连接释放的过程

三次握手：CLOSE > SYN_SENT > SYN_RCVD > ESTABLASH(c) > ESTABLEASH(S)..

四次挥手：ESTABLASH > FIN WAIT-1 > CLOSE-WAIT > FIN WAIT-2

> CLOSE(s) > CLOSE(c)

TIME-WAIT 作用：确保最后 client 发送的确认信息 ACK 能够到达 service

为什么 3 次握手？

--- 为了防止失效的请求报文在错误的时间送到服务器端。

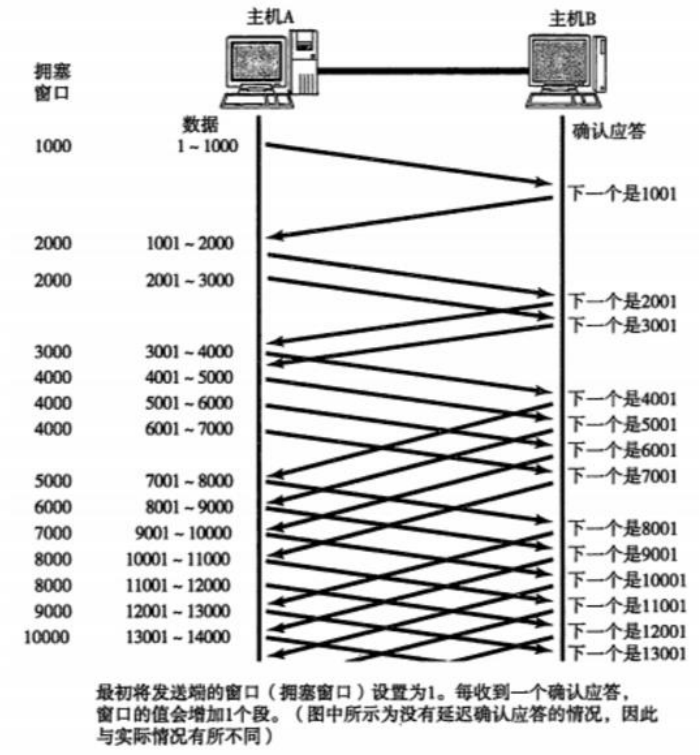
为什么 4 次挥手？

Tcp 是全双工模式的，头两次挥手目的是确认发送端无数据要发送。而接收端此时不确定无数据返回。　因此需要把数据都返回通过 2 次挥手确认关闭链接。

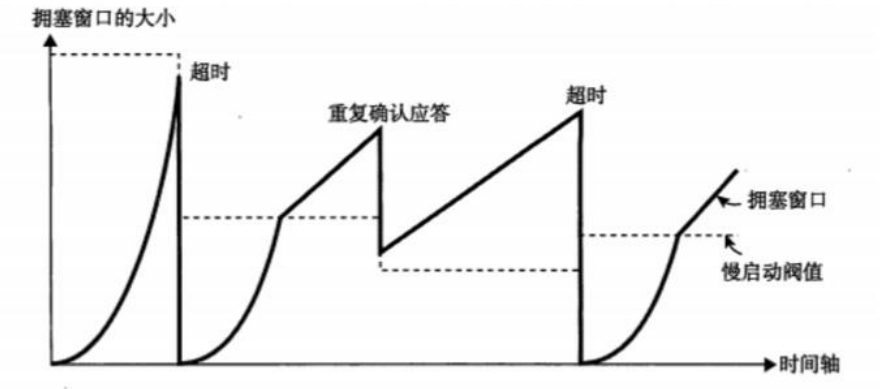
5. TCP 拥塞控制

慢开始 2^k.. -> 拥塞避免 +1 >(三个重复确认)> 快开始

- 拥塞控制



最初将发送端的窗口（拥塞窗口）设置为1。每收到一个确认应答，窗口的值会增加1个段。（图中所示为没有延迟确认应答的情况，因此与实际情况有所不同）

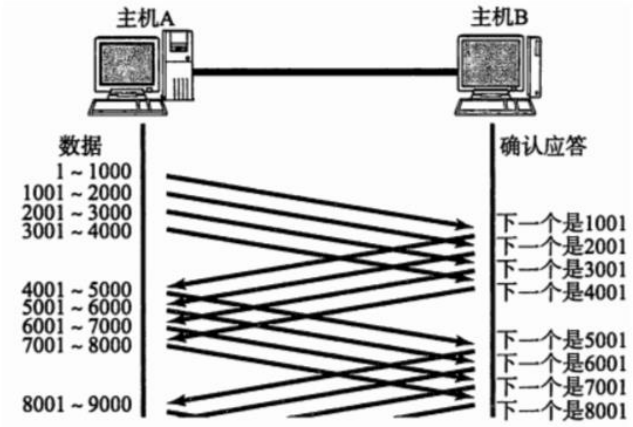


拥塞窗口越大，确认应答的数目也会增加。不过随着每收到一个确认应答，其涨幅也会逐渐减少，甚至小过比一个数据段还要小的字节数。因此，拥塞窗口的大小会呈直线上升的趋势。

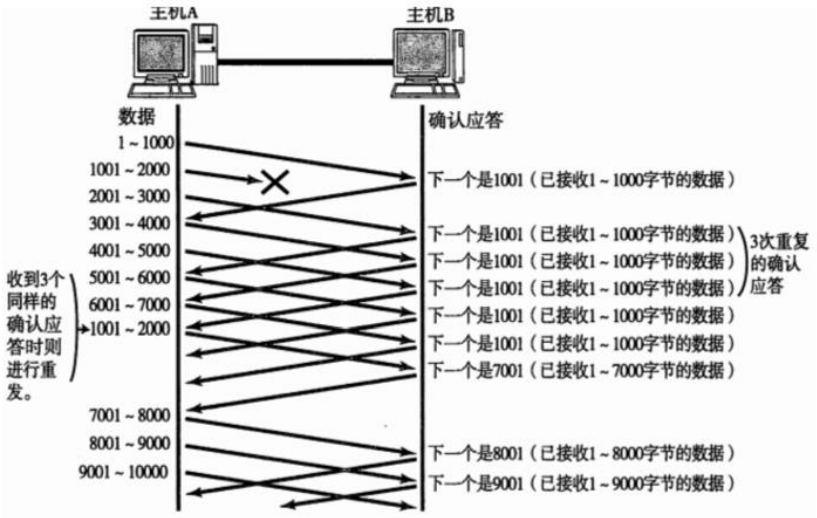
TCP 的通信开始时，并没有设置相应的慢启动阀值。而是在超时重发时，才会设置为当时拥塞窗口一半的大小。

6. TCP 滑动窗口与回退 N 针协议

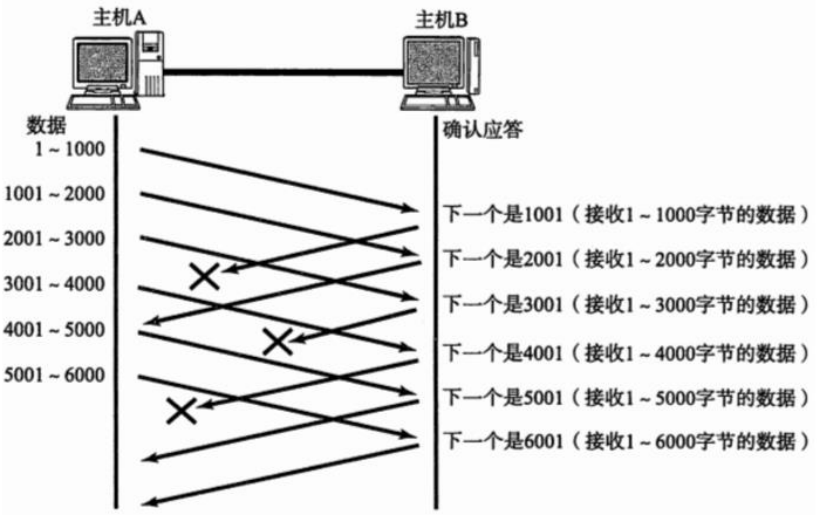
- 滑窗



· 根据窗口为4000字节时返回的确认应答，下一步就发送比这个值还要大4000个序列号为止的数据。这跟前面每个段接收确认应答以后再发送另一个新段的情况相比，即使往返时间变长也不会影响网络的吞吐量。



接收端在没有收到自己所期望序号的数据时，会对之前收到的数据进行确认应答。发送端则一旦收到某个确认应答后，又连续3次收到同样的确认应答，则认为数据段已经丢失，需要进行重发。这种机制比起超时机制可以提供更为快速的重发服务。



窗口在一定程度上较大时，即使有少部分的确认应答丢失也不会进行数据重发。可以通过下一个确认应答进行确认。

回退 N 帧：一次性发送窗口大小的数据帧，若某一帧出现错误，出现超时重传，就要回退到该帧，并重新发送该帧后面的所有数据帧。

选择重传：是回退 N 帧的改良版，使用缓冲技术，在某一帧出现错误重新回传时，只需回传该帧。而后面的数据帧都保留在接收端的缓冲区中。

7. Http 的报文结构

请求报文：

请求行

请求头 --请求方法、请求 uri、协议版本..

请求体 --内容实体

相应报文：

相应行

响应头 --状态码、协议版本..字段

响应体 --响应内容

8. HTTP 状态码含义

1xx：请求正在处理

2xx：请求成功

200:ok

204:请求成功，无回复

206:部分资源请求成功

3xx：资源重定向

4xx：客户端请求错误，服务器无法处理请求。

404：服务器无法找到资源。

5xx：服务器端处理请求出错。

500：服务器内部资源故障

503：服务器超负荷，宕机。

9. Http request 的几种类型

GET：向服务器请求资源

POST：向服务器发送数据

PUT：向服务器上传文件至指定路径 uri

DELETE：在服务器指定路径 uri 删除文件

X. GET 和 POST 区别

GET：从服务器上获取数据，GET 的参数信息会写在 URL 上，可见的。2KB 限制

POST：向服务器发送数据，POST 将数据信息写在请求体上，不可见的。大小不限。

了解：PUT 向指定 URL 发送文件，DELETE 向指定路径删除文件。

10. Http1.0 和 Http1.1 区别

Http1.0：不支持长连接、每次请求都重新创建新连接、不支持断点续传。

Http1.1：支持长连接（默认，即同一个 tcp 链接可以传送多个 http 请求和响应）、支持断点续传。

http2.0：支持头部压缩、流量控制。

11. Http 如何处理长连接

在 http1.0 中，需要添加请求头：connection：keep-alive

在 http1.1 中，默认是保持长连接的。

12. Cookie 和 Session 的作用原理

Cookie 是基于客户浏览器端的技术，Session 是基于服务器端的技术。

Cookie 和 Session 的作用都是为了或临时保存一些用户信息。

Session 是服务器端的，安全性系数相对 Cookie 会高一些。（用户敏感信息）

都能够设置失效时间。

13. 电脑上访问 www.baidu.com 过程是怎样的？

1 首先浏览器通过域名服务器 DNS 解析出该域名对应的 IP 地址。

2 浏览器针对 ip 地址发起一个 Http 的请求（请求头、请求体..）

3 通过 TCP 进行封装，将 http 请求分成报文段，添加源端口、目的端口。

4 将数据包传至网络层，由网路层进行路由寻址。

5 数据到达服务器由服务器（一层层解析）处理请求并返回响应（一层层封装）。

6 浏览器获取资源数据，进行数据解析并显示。

14. Ping 的整个过程，ICMP 报文是什么？

1 使用 ping 命令

2 向目标主机发送 ICMP 包

3 目标主机接收到 ICMP 包

4 目标主机回复 ICMP

ICMP 报文是基于网络层 ip 的协议，用来诊断网络信息。

15. C/S 模式下使用 socket 通信，几个关键函数。

- 服务端

创建 ServerSocket，监听服务器端指定端口。

监听并接收客户端跟服务器端的链接：serversocket.accept();

– 客户端

创建一个 Socket，指定对应的 ip 地址跟端口号。

通过 socket 发送字节/字符数据~

核心方法：

Accept() --侦听并接受到此套接字的连接。

Connect() --将此套接字连接到服务器。

Bind() --将套接字绑定到本地地址。

GetInputerStream --输入流

GetOutputStream --输出流

16. IP 地址分类

32 位二进制数表示，分为 4 字节，每个字节大小用 10 进制表示。

A 类：以 0 开头的 ip 地址，前 8 位代表网络地址，后 24 位表示主机，0.0.0.0 ~ 127.0.0.0

B 类 :以 10 开头的 ip 地址, , 前 16 位代表网络地址, 后 16 位主机, 128.0.0.0 ~ 191.255.0.0

C 类：以 110 开头的 ip 地址，前 24 位表示网络地址，后 8 为表示主机。
192.0.0.0 ~ 223.255.255.0
D 类：以 1110 开头，224.0.0.0 ~ 239.255.255.255
E 类：240.0.0.0 ~ 255.255.255.255

17. 路由器和交换机的区别

路由器：首先路由器是基于网路层的产品，作用是共享一个 ip 地址，让多台设备上网。
交换机：基于数据链路层的产品，简单来说，就是共享一条网线，接入不同的设备上网。
宿舍例子。

网关：连接两个不同的网络设备。不同协议网络的转换。