第四章基于TCP的服务端/客户端(1)

TCP（Transmission Control Protocol（传输控制协议））套接字是面向连接的，因此又称基于流（stream）的套接字。

TCP/IP协议栈

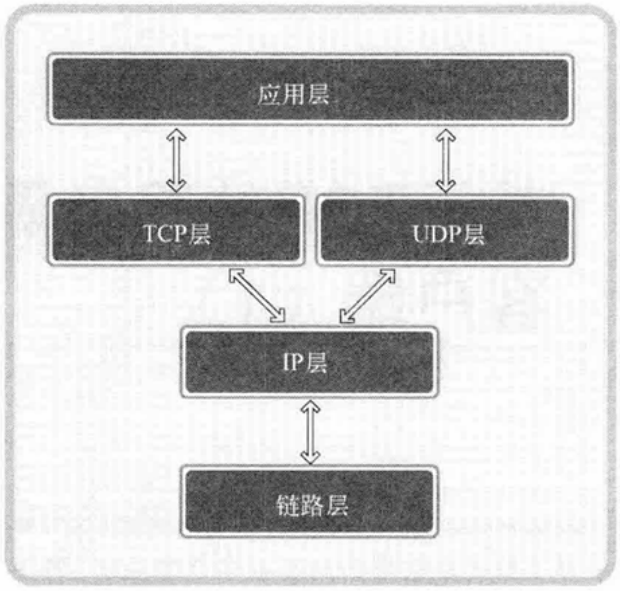

图4-1　TCP/IP协议栈

如上图所述，TCP/IP协议栈分为四层，每层都对应着协议，只有遵循所有的协议才能实现一个在互联网上基于TCP/UDP套接字的数据传输功能。

链路层

是物理链接领域标准化的结果，专门定义LAN、WAN、MAN等网络标准，在物理层面上进行连接，实现网络进行数据交换的第一步。

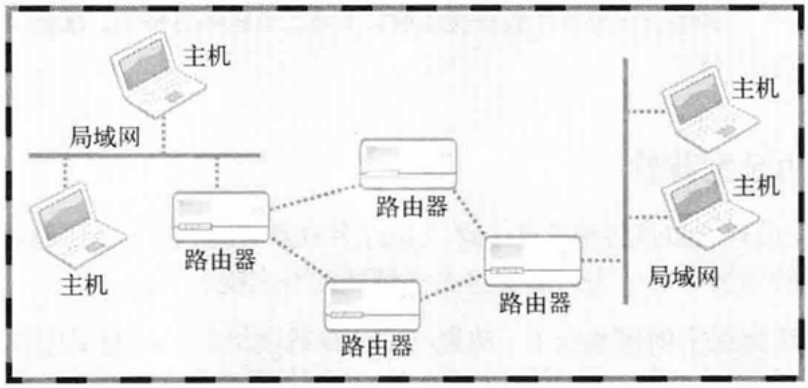

图4-4　网络连接结构

IP层

IP层在网络数据传输过程中起到了路径选择的作用，每次路径的选择是不一致的，但会确保能够找到一条路径用来传输数据；IP层不会保证数据完整性，如果发生数据丢失或者错误，则无法解决。

TCP/UDP层

TCP/UDP层以IP层提供的路径信息为基础完成实际的数据传输，所以称为传输层（Transport）。

TCP层

IP层是以一个个包（数据传输的基本单位）来进行传输，但不确保传输包的顺序和包是否会丢失。所以TCP层
用作保证可靠的数据传输。

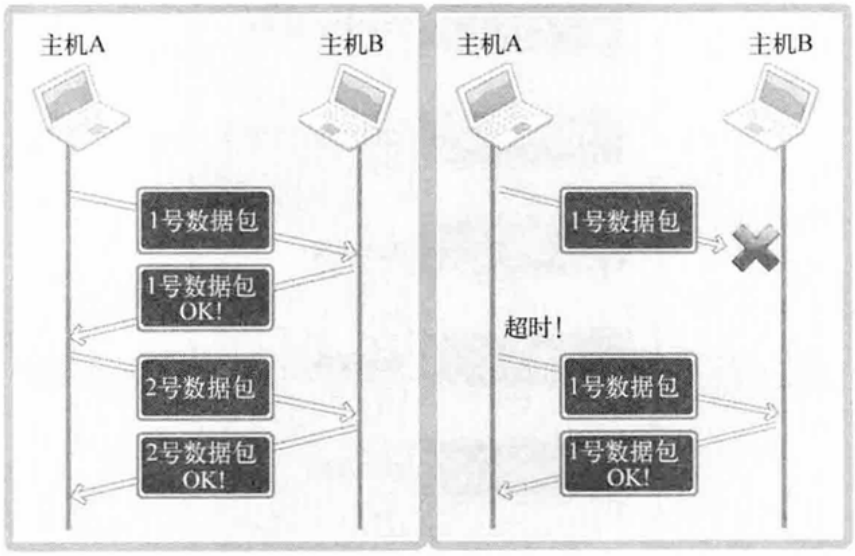

图4-5　传输控制协议

应用层

编写软件的过程中，需要根据程序特点决定服务器端和客户端之间的数据传输规则，这便是应用层协议。网络编程的大部分内容就是设计并实现应用层协议。

实现基于TCP的服务器端/客户端

TCP服务器端默认的函数调用顺序

图4-6　TCP服务器端函数调用顺序

进入等待连接请求状态

```
//只有调用了listen函数，客户端才能进入可发出连接请求的状态。换言之，这时客户端才能调用connect函数
#include <sys/socket.h>
int listen(int sock, int backlog);//sock希望进入等待连接请求的套接字文件描述符，传递的描述符套接字参数称为服务器端套接字(监听套接字)；backlog(连接请求等待队列的长度，若为5，则队列长度为5，表示最多使5个连接请求进入队列)。
//成功时返回0，失败时返回-1
```

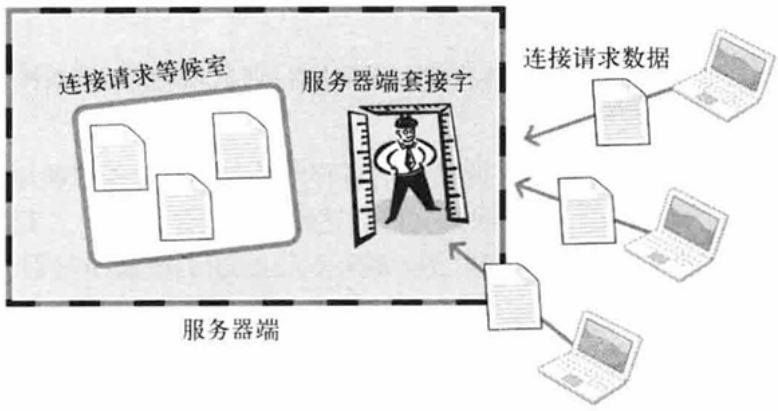

图4-7　等待连接请求状态

受理客户端连接请求

由于服务端的套接字是用来“接待”客户端的连接请求的，所以与客户端进行数据传输需要另外有个数据套接字，无需主动创建，在accept函数成功接收客户端的请求后，会自动返回一个数据套接字，用来与某个客户端进行数据传输。

```
#include <sys/socket.h>
int accept(int sock, struct sockaddr* addr, socklen_t* addrlen);//sock是连接套接字的文件描述符，addr是保存发起连接请求的客户端地址信息的变量地址值，调用函数后向传递来的地址变量参数填充客户端地址信息；addrlen是addr结构体的长度，但是存有长度的变量地址。函数调用完成后，该变量即被填入客户端地址长度。
//成功时返回一个数据套接字句柄（文件描述符），失败时返回-1
```

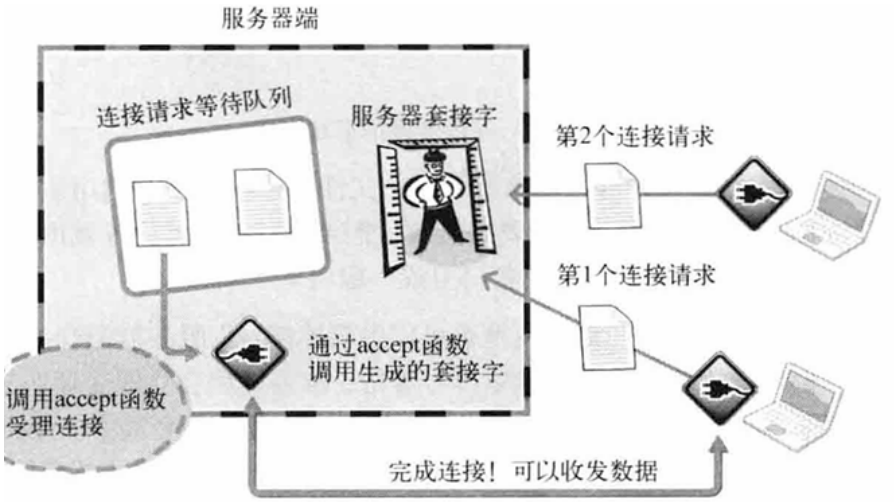

图4-8　受理连接请求状态

上图展示了"从等待队列中取出一个连接请求，创建套接字并完成连接请求"的过程。服务端单独创建的套接字与客户端建立连接后进行数据交换。

TCP客户端的默认函数调用顺序

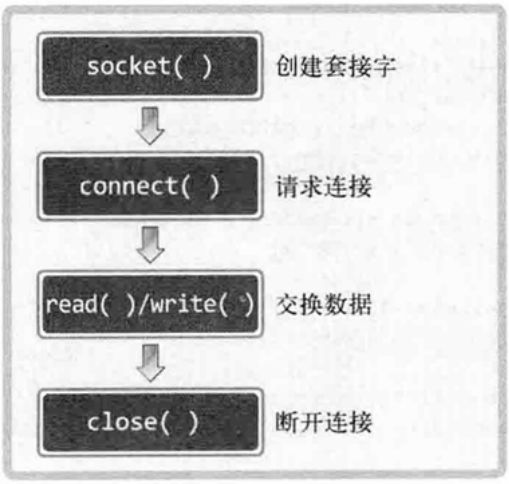

图4-9　TCP客户端函数调用顺序

服务器端调用listen函数后创建连接请求等待队列，之后客户端即可请求连接

```
#include <sys/socket.h>
int connect(int sock, struct sockaddr* servaddr, socklen_t addrlen);//sock是客户端的数据套接字，servaddr用来保存服务端的地址信息的变量地址值；addrlen以字节为单位传递已传递给第二个结构体参数servaddr的地址变量长度。
//成功时返回0，失败时返回-1
```

客户端的IP地址和端口号在connect函数调用时自动分配，那么在客户端就无需调用标记的bind函数进行分配。

基于TCP的服务端/客户端函数调用关系

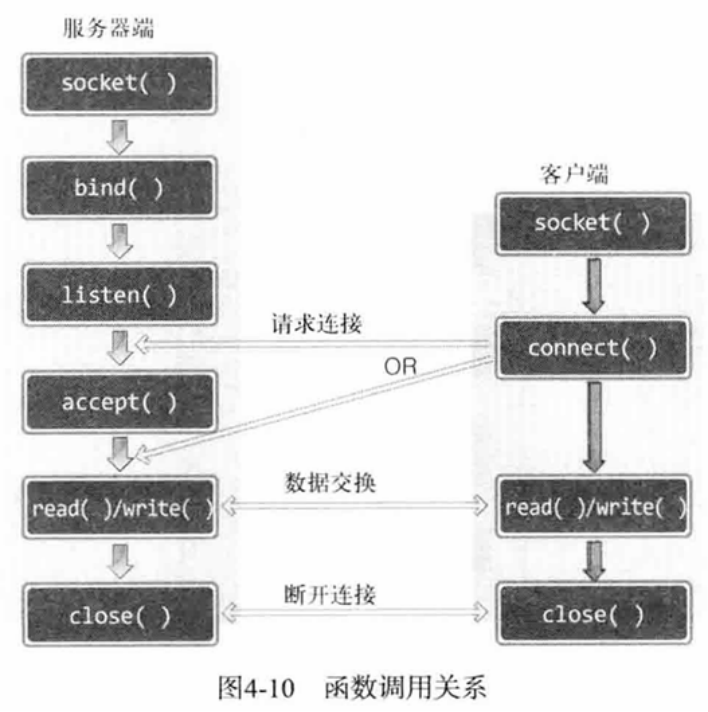

图4-10　函数调用关系

实现迭代服务器端/客户端

迭代服务器端：顾名思义，服务器端将客户端传输的字符串数据原封不动地传回给客户端，就像回声一样。

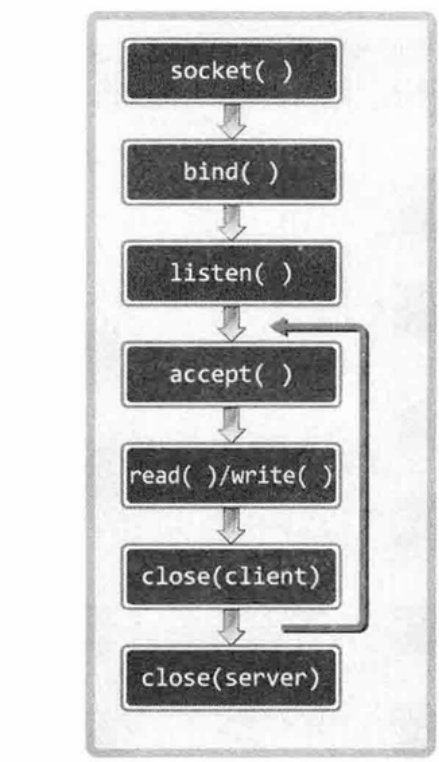

图4-11　迭代服务器端的函数调用顺序

第四章习题

1、请说出TCP/IP的4层协议栈，并说明TCP和UDP套接字经过的层级及结构差异。

答：

- 应用层：包括HTTP、FTP、SMTP、DNS等应用协议
- 传输层：包括TCP、UDP两种主要的传输协议
- 网络层：主要是IP协议
- 网络结构层：负责数据链路层和物理层的功能，如以太网协议

关于TCP和UDP套接字的层级及结构差异：TCP套接字工作在传输层、是一种面向连接的通信并且需要先建立三次握手，从而提供一种可靠的、有序的数据传输。UDP套接字也是工作在传输层，但是是一种无连接的通信，它不需要建立连接就可以发送数据，这使得它提供了一种不可靠的且无序的数据传输。

2、请说出TCP/IP协议栈中链路层和ip层的作用，并给出二者关系。

答：（1）链路层的作用：①在物理网络上传输数据帧、②提供点对点数据传输、③使用mac地址进行本地寻址。

（2）IP层的作用：①负责数据包的路由和转发、②提供端到端的数据传输、③使用IP地址进行全局寻址

二者之间的关系：IP层位于链路层之上，通过链路层传输数据；IP层负责路径选择，链路层负责实际的传输；IP地址通过ARP协议转换为MAC地址后才能由链路层传输。

3、为何需要把TCP/IP协议栈分成四层（或7层）？结合开放式系统回答。

答： 将TCP/IP 协议栈分为四层（或 OSI 模型分为七层）的主要目的是通过分层设计简化复杂的网络通信系统，使得网络开发、实现和维护更加高效和灵活。分层设计符合开放系统互联的思想，通过规范化接口定义简化了系统开发，使开发者可以专注于某一层的实现，而无需关心其他层的细节。这种模块化的设计同时也使不同厂商的设备和软件可以通过标准接口互操作，促进网络的开放性和兼容性。

4、客户端调用connect函数向服务器端发送连接请求。服务器端调用哪个函数后，客户端可以调用connect函数?

答：服务端调用listen函数后，客户端才能调用connect函数。因为listen函数将套接字转为可接受连接状态，使服务端准备好接收客户端的连接请求。此时服务器端的套接字变为被动套接字（监听套接字）。

5、什么时候创建连接请求等待队列？它有何作用？与accept有什么关系?

答：服务端在listen函数调用后便创建了连接请求等待队列，服务端通过调用accept函数逐次地从等待队列里面取出连接请求，完成客户端和服务端的连接；同时服务端建立了一个数据流套接字，用于与客户端进行数据的传输；当队列为空时accept会阻塞等待新请求。

6、客户端中为何不需要调用bind函数分配地址？如果不调用bind函数，那何时、如何向套接字分配IP地址和端口号?

答：客户端在connect函数调用后，系统自动的为客户端分配IP和端口号；客户端ip使用主机ip，端口号由系统随机分配；客户端一般不需要固定端口号，临时端口号即可。