分片传输

一个包没有固定长度，以太网限制在46－1500字节，1500就是以太网的MTU，超过这个量，TCP会为IP数据报设置偏移量进行分片传输

明确一点，TCP作为流，发包是不会整包到达的，而是源源不断的到，那接收方就必须组包。而UDP作为消息或数据报，它一定是整包到达接收方。

***int send( SOCKET s,const char\* buf,int len,int flags);***

这里只描述同步Socket的send函数的执行流程。当调用该函数时，

send先比较待发送数据的长度len和套接字s的发送缓冲的 长度，如果len大于s的发送缓冲区的长度，该函数返回SOCKET\_ERROR；

如果len小于或者等于s的发送缓冲区的长度，那么send先检查协议 是否正在发送s的发送缓冲中的数据，

如果是就等待协议把数据发送完，

如果协议还没有开始发送s的发送缓冲中的数据或者s的发送缓冲中没有数据，

那么 send就比较s的发送缓冲区的剩余空间和len，

如果len大于剩余空间大小send就一直等待协议把s的发送缓冲中的数据发送完，

如果len小于剩余 空间大小send就仅仅把buf中的数据copy到剩余空间里（注意并不是send把s的发送缓冲中的数据传到连接的另一端的，而是协议传的，send仅仅是把buf中的数据copy到s的发送缓冲区的剩余空间里）。

如果send函数copy数据成功，就返回实际copy的字节数，

如果send在copy数据时出现错误，那么send就返回SOCKET\_ERROR；

如果send在等待协议传送数据时网络断开的话，那么send函数也返回SOCKET\_ERROR。  
  
要注意send函数把buf中的数据成功copy到s的发送缓冲的剩余空间里后它就返回了，但是此时这些数据并不一定马上被传到连接的另一端。如 果协议在后续的传送过程中出现网络错误的话，那么下一个Socket函数就会返回SOCKET\_ERROR。（每一个除send外的Socket函数在执 行的最开始总要先等待套接字的发送缓冲中的数据被协议传送完毕才能继续，如果在等待时出现网络错误，那么该Socket函数就返回 SOCKET\_ERROR）  
  
注意：在Unix系统下，如果send在等待协议传送数据时网络断开的话，调用send的进程会接收到一个SIGPIPE信号，进程对该信号的默认处理是进程终止。

***int recv( SOCKET s,char\* buf,int len,int flags);***

这里只描述同步Socket的recv函数的执行流程。当应用程序调用recv函数时，

recv先等待s的发送缓冲 中的数据被协议传送完毕，如果协议在传送s的发送缓冲中的数据时出现网络错误，那么recv函数返回SOCKET\_ERROR，

如果s的发送缓冲中没有数 据或者数据被协议成功发送完毕后，recv先检查套接字s的接收缓冲区，如果s接收缓冲区中没有数据或者协议正在接收数据，那么recv就一直等待，只到 协议把数据接收完毕。当协议把数据接收完毕，recv函数就把s的接收缓冲中的数据copy到buf中（注意协议接收到的数据可能大于buf的长度，所以 在这种情况下要调用几次recv函数才能把s的接收缓冲中的数据copy完。recv函数仅仅是copy数据，真正的接收数据是协议来完成的），recv函数返回其实际copy的字节数。如果recv在copy时出错，那么它返回SOCKET\_ERROR；如果recv函数在等待协议接收数据时网络中断了，那么它返回0。  
  
注意：在Unix系统下，如果recv函数在等待协议接收数据时网络断开了，那么调用recv的进程会接收到一个SIGPIPE信号，进程对该信号的默认处理是进程终止。

**send 会出现发送数据不全的情况，为什么？**

socket发送缓冲区大小，cat /proc/sys/net/ipv4/tcp\_wmem

TCP之ACK发送情景

TCP之1460MSS和1448负载

MSS在以太网上通常大小是1460字节，而我们在后续发送过程中的单个TCP包的最大数据承载量是1448字节

send（）只是负责拷贝，拷贝完立即返回，不会等待发送和发送之后的ACK。如果socket出现问题，RST包被反馈回来。在RST包返回之时，如果send（）还没有把数据全部放入内核或者发送出去，那么send（）返回-1，errno被置错误值；如果RST包返回之时，send（）已经返回，那么RST导致的错误会在下一次send（）或者recv（）调用的时候被立即返回。  
场景3完全说明send（）只要完成拷贝就成功返回，如果发送数据的过程中出现各种错误，下一次send（）或者recv（）调用的时候被立即返回。

单个TCP包的运载能力是1448 B

以太网Ethernet最大的数据帧是1518字节

**1500字节的MTU**

最大数据帧是1518个字节，帧头14个字节，帧尾4个字节 1518-18=1500 字节的数据运载能力

MSS TCP单包最大传输量 1460

IP 数据包 20个字节

TCP数据包包头 20个字节

1500 – 20 -20 = 1460

实际场景下 TCP包头会带有12个字节的选项 – 时间戳

1460 – 12 =1448 Len = 1448

TCP 层实际上可以打包更大的数据，但是会降低传输效率。TCP数据包传递到IP层，IP层会发生分片操作，如果发生分片就会降低效率，所有以1448作为TCP的数据包大小

问题1：int listen(int fd,int backlog)

TCP socket

传输socket

pending socket:未决套接字，处于socket SYN\_RECV状态

established socket：收到到ACK的套接字

监听socket

<http://chuansong.me/n/895177>