这个例子可以说是比较完美了，通过TCPConnection（负责发送接收数据），TCPService（负责绑定端口，发布服务），NetUtilities（负责检查网络是否可用,可借鉴），TCPServer（负责整合TCPConnection和TCPService，并提供一系列委托通知用户），Networking\_Internal(为检查委托方法提供了一系列的封装，值的借鉴)。

但是其实我运行后是不成功的。

原因：

//服务端向客户端写东西，data是图片数据。私有方法

- (BOOL) \_writeData:(NSData\*)data

{

CFIndex length = [data length],

result;

Header header;

header.magic = NSSwapHostIntToBig(kMagic); //主机字节序转换成网络大字节序，应该是区分东西

header.length = NSSwapHostIntToBig(length);

result = CFWriteStreamWrite(\_outputStream, (const UInt8\*)&header, sizeof(Header));//打包成结构体，发送到对方那里去

if(result != sizeof(Header)) {

REPORT\_ERROR(@"Wrote only %i bytes out of %i bytes in header", (int)result, (int)sizeof(Header));

return NO;

}

while(length > 0) {

result = CFWriteStreamWrite(\_outputStream, (UInt8\*)[data bytes] + [data length] - length, length);

if(result <= 0) {

REPORT\_ERROR(@"Wrote only %i bytes out of %i (%i) bytes in data", (int)result, (int)length, [data length]);

return NO;

}

length -= result;

}

return YES;

}

- (NSData\*) \_readData

{

NSMutableData\* data;

CFIndex result,

length;

Header header;

result = CFReadStreamRead(\_inputStream, (UInt8\*)&header, sizeof(Header)); //读到，说明已经结束了

if(result == 0)

return (id)kCFNull;

if(result != sizeof(Header)) {

REPORT\_ERROR(@"Read only %i bytes out of %i bytes in header", (int)result, (int)sizeof(Header));

return nil;

}

if(NSSwapBigIntToHost(header.magic) != kMagic) {

REPORT\_ERROR(@"Invalid header", NULL);

return nil;

}

length = NSSwapBigIntToHost(header.length);

data = [NSMutableData dataWithCapacity:length];

[data setLength:length];

while(length > 0) {

result = CFReadStreamRead(\_inputStream, (UInt8\*)[data mutableBytes] + [data length] - length, length);

if(result <= 0) { //resulet=0,读完了

REPORT\_ERROR(@"Read only %i bytes out of %i (%i) bytes in data", (int)result, (int)length, [data length]);

return nil;

}

length -= result;

}

return data;

}

为了辨识是发过来的数据，这里有一个头部验证

typedef struct {

NSUInteger magic;

NSUInteger length;

} Header; //NOTE: This header is in big-endian 我感觉应该是个字节

发送和接收的时候都需要验证一下这个header，并且用lengh保存了NSData的字节数，但是在发送的时候

Header的大小是（sizeof）8，而在mac中是16，结果导致读入错误，我一直没改正过来，但是如果去掉头部验证的话是可以写入读取了，

这个问题先留着吧

其他的倒是没有什么了，解释写的很详细，如果忘了的话可以在调试一下，用一台机器就行，挺不错的