实验报告

姓名： 袁峥 学号： 2015K8009929008

一、实验题目：网络传输机制实验二

二、实验内容

1、运行给定网络拓扑(tcp\_topo.py)

2、在节点h1上执行TCP程序

执行脚本(disable\_arp.sh, disable\_icmp.sh, disable\_ip\_forward.sh , disable\_tcp\_rst.sh)，禁止协议栈的相应功能

在h1上运行TCP协议栈的服务器模式 (./tcp\_stack server 10001)

3、在节点h2上执行TCP程序

执行脚本(disable\_arp.sh, disable\_icmp.sh, disable\_ip\_forward.sh, disable\_tcp\_rst.sh)，禁止协议栈的相应功能

在h2上运行TCP协议栈的客户端模式，连接h1并正确收发数据 (./tcp\_stack client 10.0.0.1 10001)

client向server发送数据，server将数据echo给client

三、实验流程

1、数据接收和缓存

TCP协议栈收到数据包后，使用接收缓存来存储相应数据，供应用程序读取

①使用环形缓存(ring buffer)来实现

②接收缓存大小为recv\_window

③使用锁（pthread\_mutex\_t）来防止读写冲突

2、数据发送流程

发送包含数据的数据包

①待发送数据全部存储于上层应用buffer中

②如果对端recv\_window允许，则发送数据

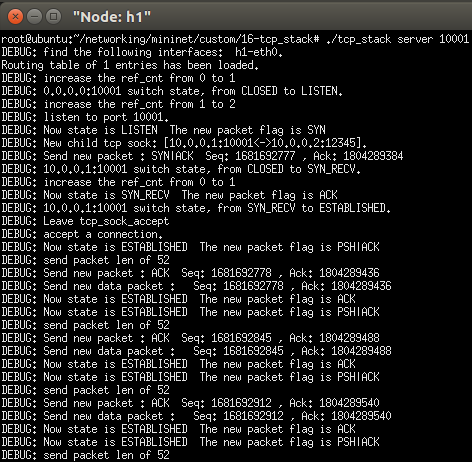
③每次读取1个数据包大小的数据

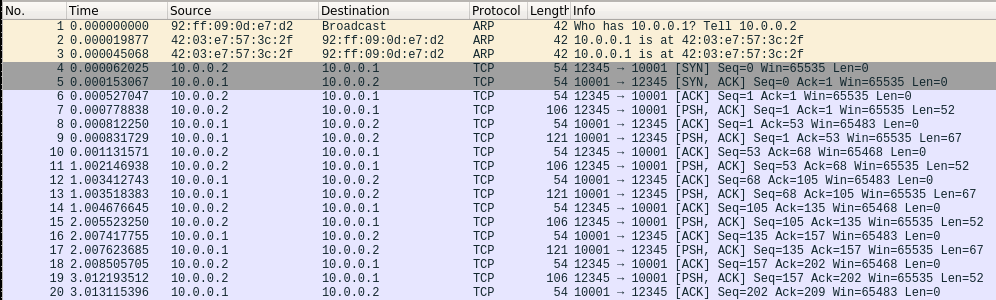
min(data\_len, 1500 - ETHER\_HDR\_SIZE - IP\_HDR\_SIZE - TCP\_HDR\_SIZE)

④封装数据包，通过IP层发送函数，将数据包发出去

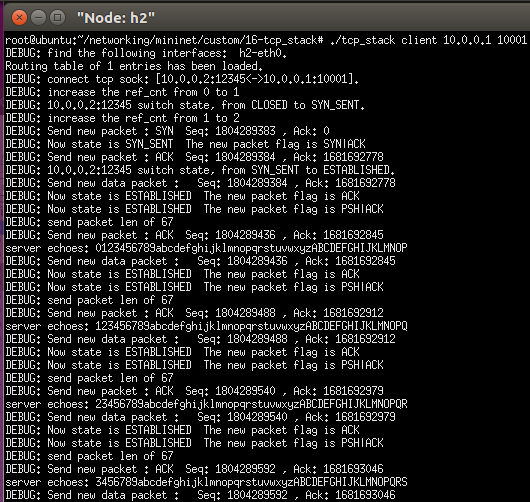
四、实验结果

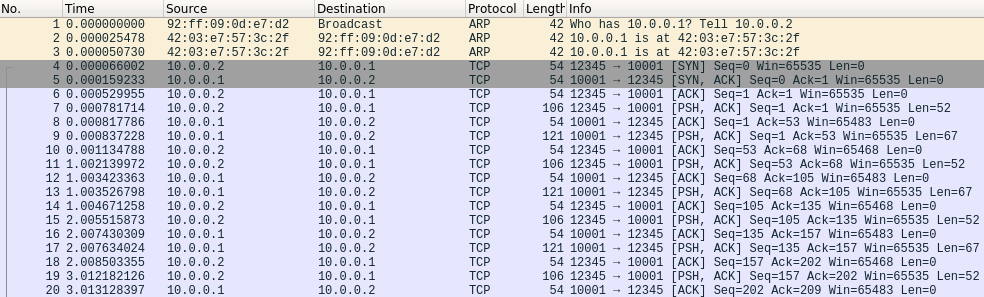
1、h1节点的连接过程





2、h2节点的连接过程





五、结果分析

socket的连接部分与上次实验一致，这次实验考察的重点在于数据传输部分。

从上述截图中可以看到，根据此次实验的tcp\_apps.c，首先由h2向h1发出PSH|ACK包，其中包含数据0-P，长度为52，h1收到后向h2回复ACK包，同时h2将收到的52字节数据包再加上一个头“server echoes:”后共计67字节发送给h1，作为一个PSH|ACK包的数据段，h1收到该包后向h2回复ACK包，至此一个循环结束。

再考察传输的数据包中Seq、Ack、Win等变量，首先在h1与h2之间的socket建立完成后，Seq、Ack号均为1，Win为65535，h2先向h1发送PSH|ACK包，此时Seq和Ack号为1，由于h2端buf没有存储数据，因此Win为65535，h1收到该数据包后，回复ACK包，由于收到的数据包数据段长度为52，因此回复的ACK包中Ack号为53，同时由于收到的数据包存储在buf中，因此接收窗口减少52，变为65483，而Seq仍为socket建立完成时的1。此后h1将先前收到的长度为52的数据包从buf中取出，再与一个长度为15字节的头拼接后，形成长度为67字节的新的数据包，作为PSH|ACK包的数据段发送给h2，此时由于buf中没有数据，因此接受窗口又变为65535，而由于ACK包不消耗序号，因此Seq和Ack与发送ACK包时一致。h2收到来自h1的数据包后，回复ACK包，由于需要将收到的数据段存放在buf中，因此ACK包中的Win变为65468，相应的，Seq更新为53，Ack为67。至此h1与h2的一个循环的收发数据包完成，后续情况类似。

六、实验总结

此次实验在上次实验的基础上添加了最简单的数据传输，需要完成的工作量很少，由于上次实验已经基本熟悉了代码结构和之间的逻辑结构，因此此次实验的代码编写过程十分顺利，也没有碰到什么大的问题。

此次实验一共完成了3个函数，分别为tcp\_sock.c中的tcp\_sock\_read、tcp\_sock\_write和tcp\_in.c中的tcp\_recv\_data。

①tcp\_sock\_read函数完成TCP协议栈中的数据接收，首先等待ring\_buffer中有数据，然后从中读取出相应长度的数据，同时更新接收窗口大小。

②tcp\_sock\_write函数完成TCP协议栈中的数据发送，首先等待对端接收窗口的大小大于需要发送数据包的长度，然后将需要发送的数据包按需切成较小的数据包，并分别封装为TCP包的数据段后发送。

③tcp\_recv\_data函数完成接收到数据包后的操作，由于此次实验只实现最简单的数据传输，此时要求新收到的数据包的Seq号与socket中的rcv\_nxt一致，否则直接丢弃。满足该要求后，首先更新对端接收窗口大小，然后将收到的数据段写入ring\_buffer，同时更新本端口的接收窗口大小，并修改rcv\_nxt和snd\_una信号。