Open Access AP(未加密AP)

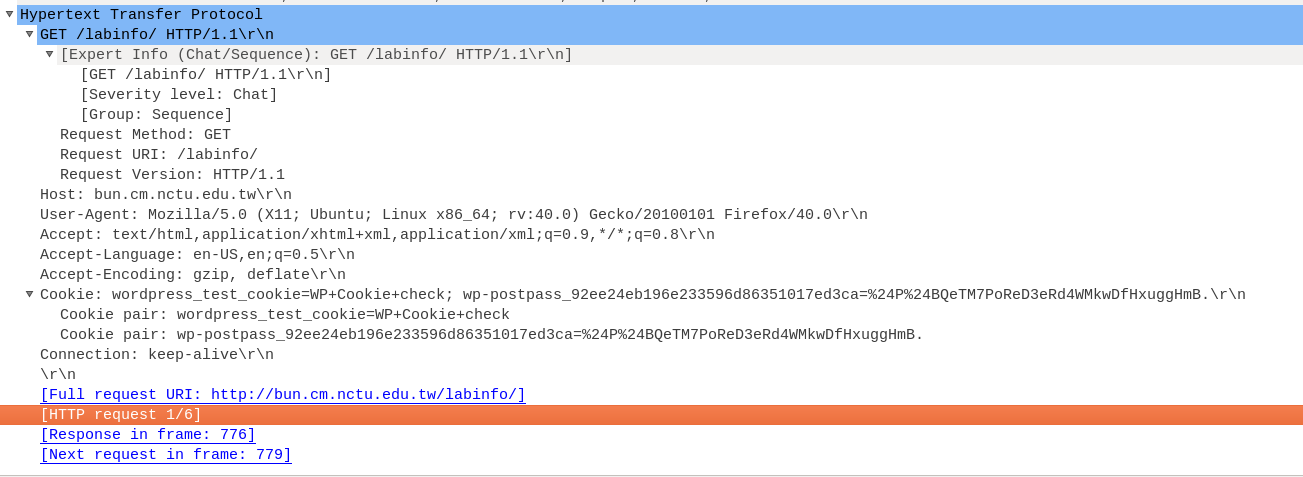
通过天线接收到的未加密讯息可以正常接收到HTTP, FTP, Telnet 和SSH的封包，并且可以在wireshark的protocol一栏里区分出不同协定的packet。

而且在封包讯息里都可看见Frame大小、number、到达的时间、interface的id、name，然后还可以看到Header的相关讯息：header的长度、data rate、以及实验中所选的channel8的frequency等等。Tx端和Rx端的IP、MAC address，header的checksum。由于packet是通过TCP协定传输的，所以还有TCP相关的讯息，eg：source、destination的port（其中HTTP：80，FTP：21，Telnet：23，SSH：22），以及ACK的相关讯息

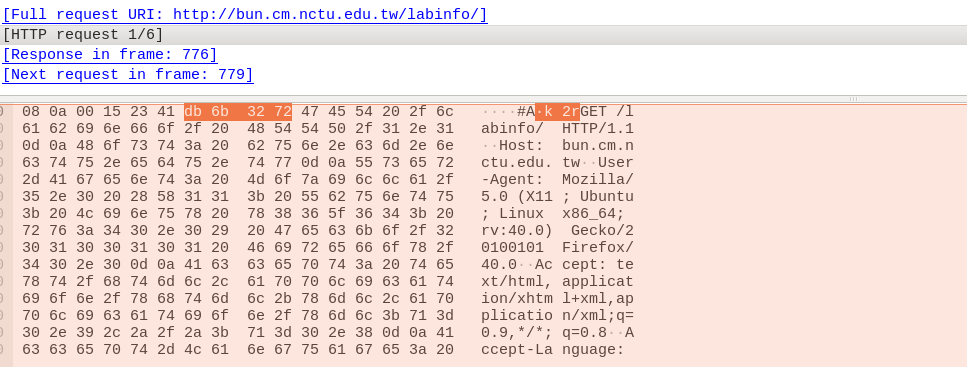
除了以上packet传输的基本讯息，不同协定的data一栏有不同的讯息。

HTTP：

在wireshark的Hypertext Transfer Protocol一栏中可以看到



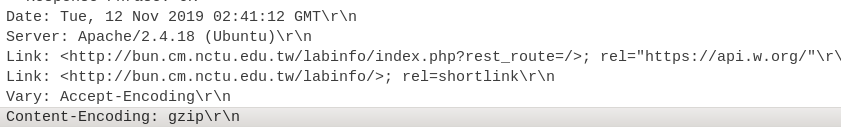
user向server端请求，可以看到request的版本为HTTP1.1，请求访问的完整URL为



以及接收端的语言为en-US，编码方式为gzip。

Server回应user的封包里：

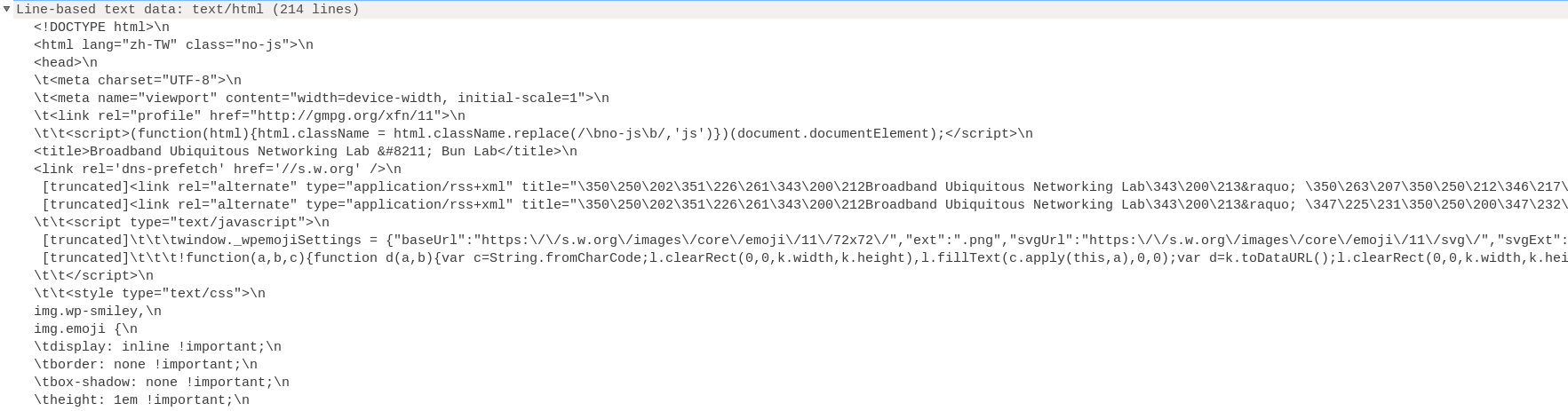
包含了目前的时间，server版本，也使用gzip编码，使用HTTP v1.1回应



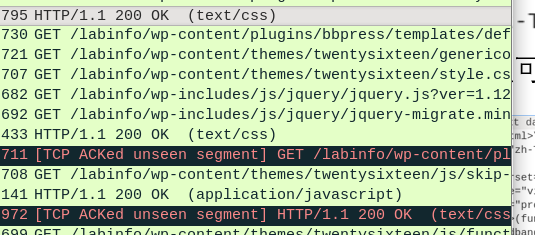
内容格式为html，字符集使用UTF-8编码：



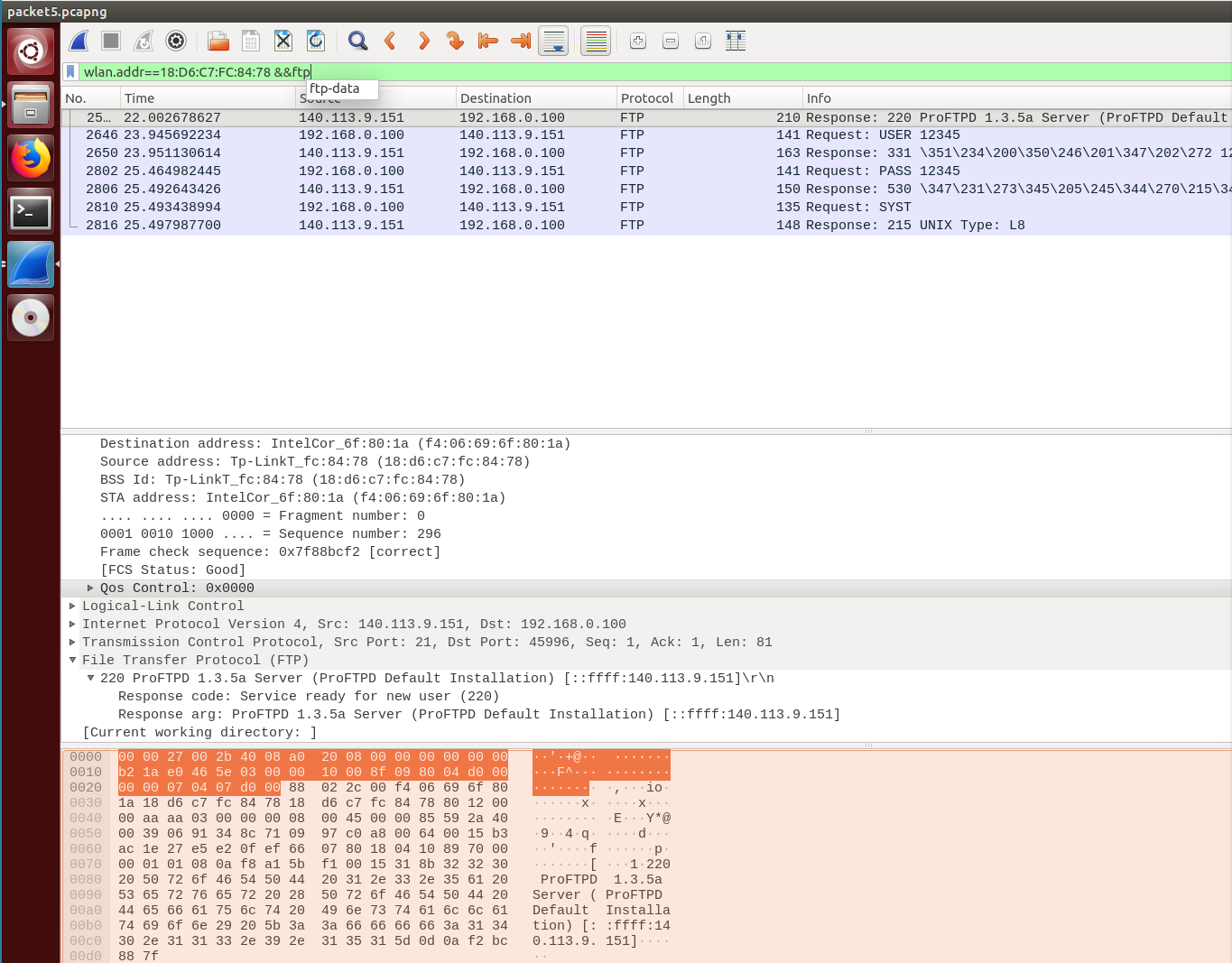
Data里可以看到的完整html代码：



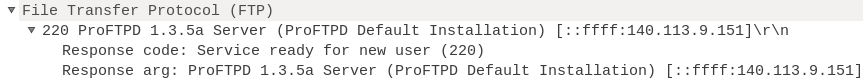
后续的封包data里可以看到陆续向server请求css和javascript的代码



FTP：



Server端使用unix平台上的FTP服务器程序ProFTPD （1.3.5）



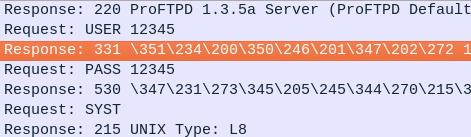
请求建立一个user，用户名12345



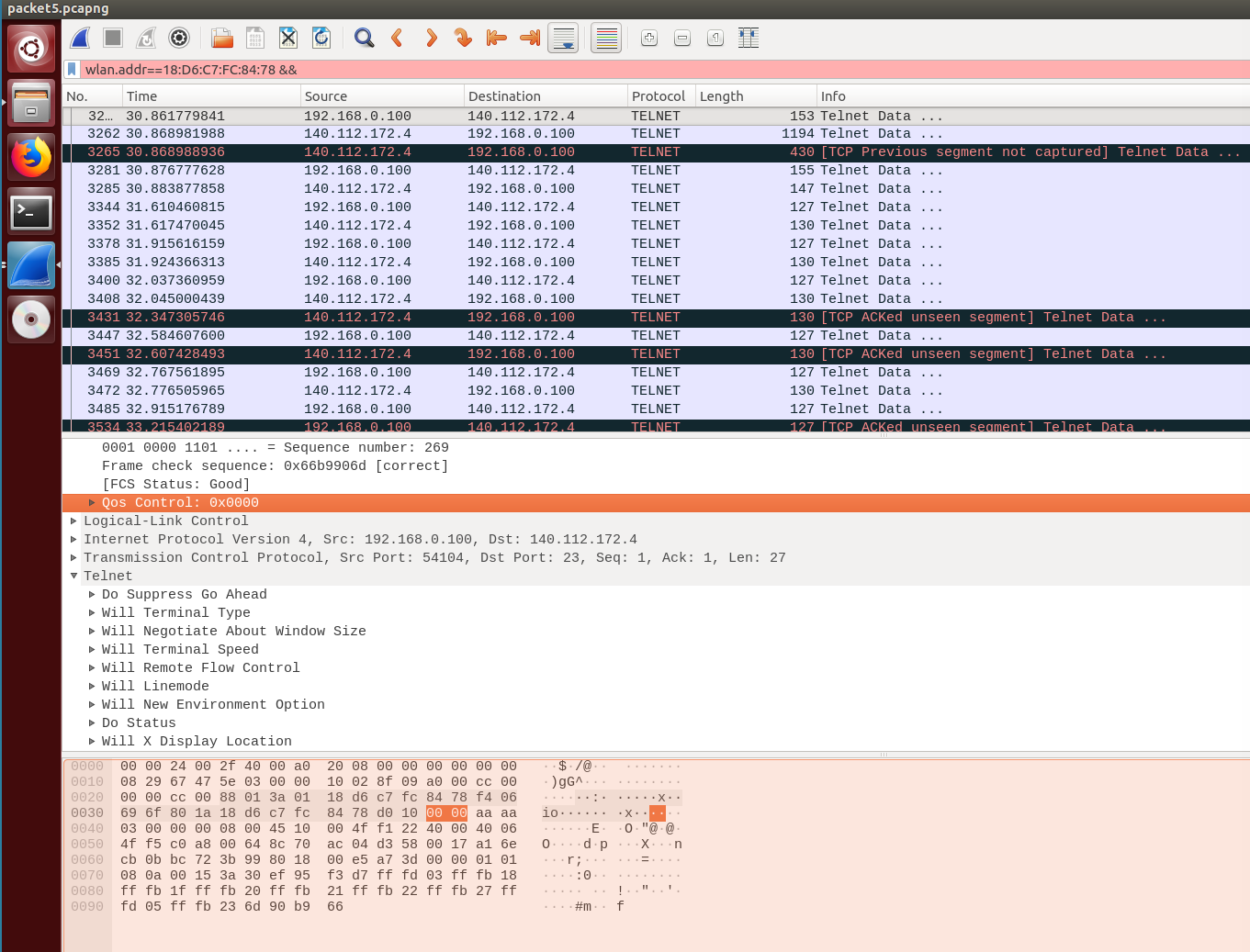
回应：用户名合法，需要密码



密码为：12345，成功建立FTP连线

  
Telnet：

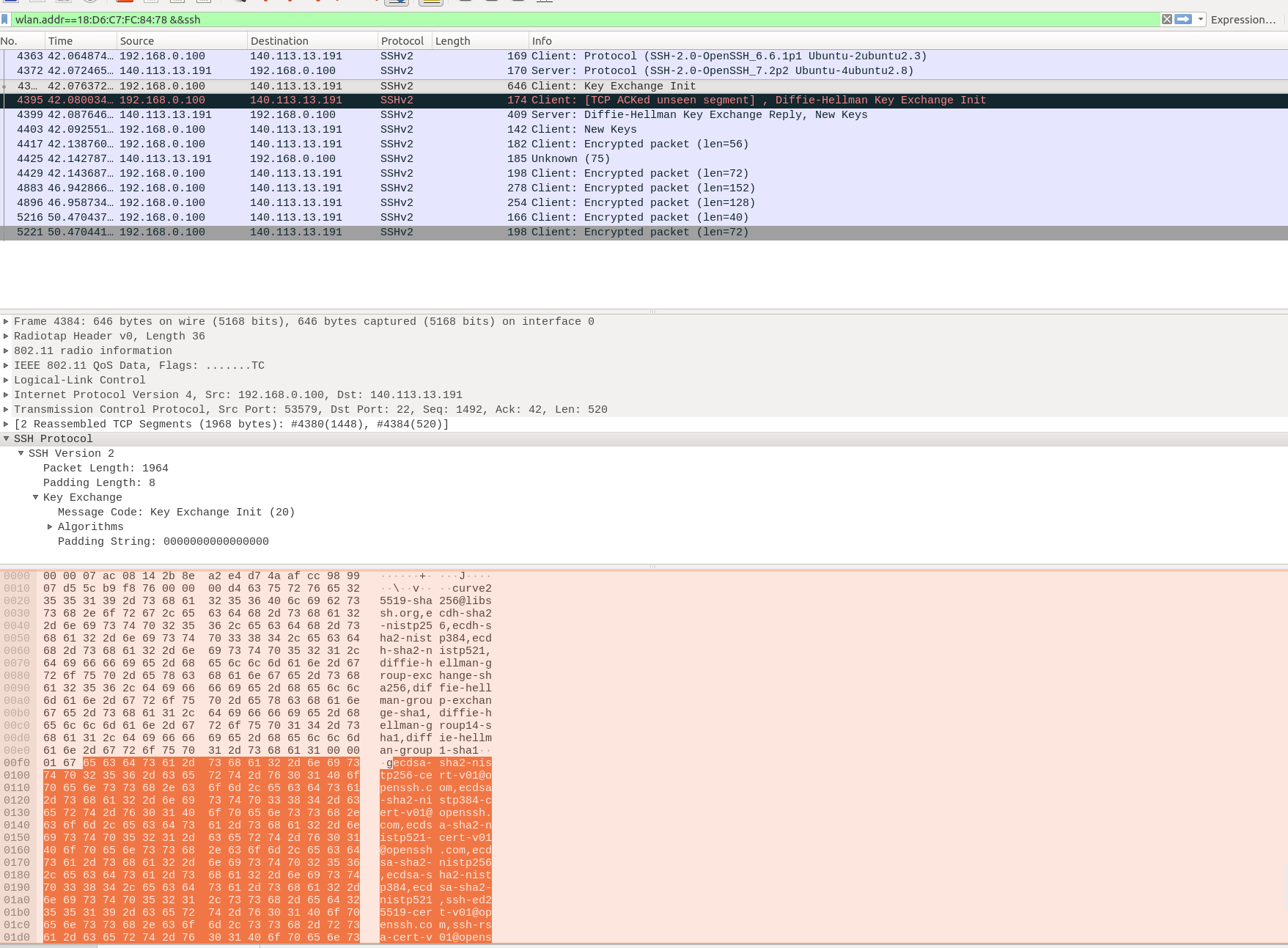
telnet是远程操作服务协定，前几个封包用于建立链接。



建立链接以后终端使用者输入的指令都会在server端执行，终端每输入一个字母就把这个字母传给server，所以在后面的封包里可以看见每个封包都只包含一个字母

SSH：



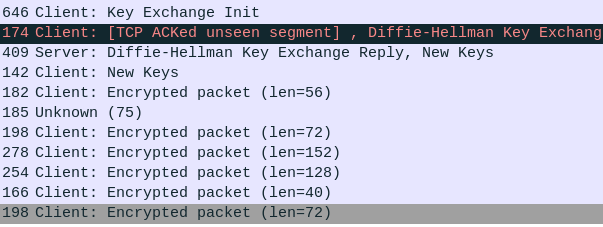
用户端的协定版本为SSH-2.0-openssh——6.6.1p1

Server端的协定版本SSH-2.0-openssh——7.2p2

都运行在ubuntu的系统上



可以看到，key的初始化。但第一次key的初始化失败

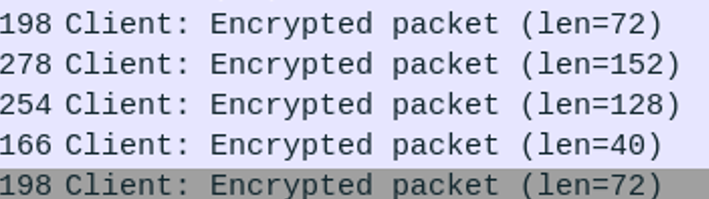




以及最终产生的new key：



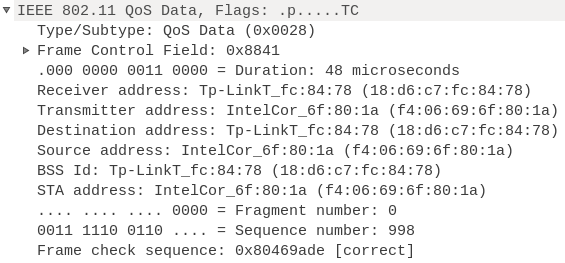
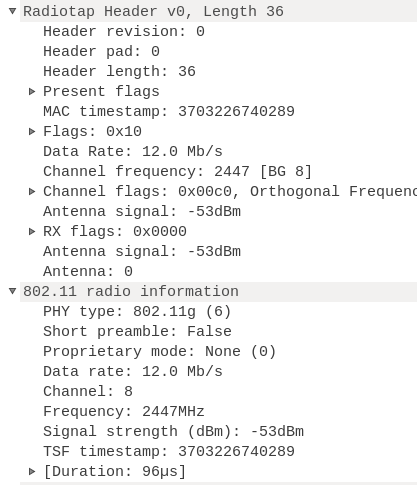
后续收到的封包，data都被加密过只能看见乱码。

Encrypted AP（加密AP）

此时通过天线接收到的讯息在protocol只会显示802.11，没有HTTP, FTP, Telnet 和SSH的封包出现，通过wlan.addr==18:D6:C7:FC:84:78来筛选与发送加密讯息AP有关的packet，发现仍然可以接收到这些packet，但是无法再区分不同协定的packet。

Rx、Tx端的MAC address仍然可以看见，但是IP address、port不可见。Sequence number，data rates，channel，frequency仍然可见。



所有封包的data栏位均为乱码，无法解析出data的具体含义。

