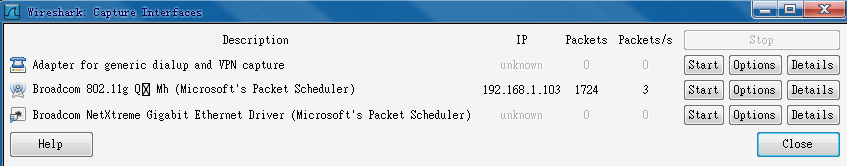
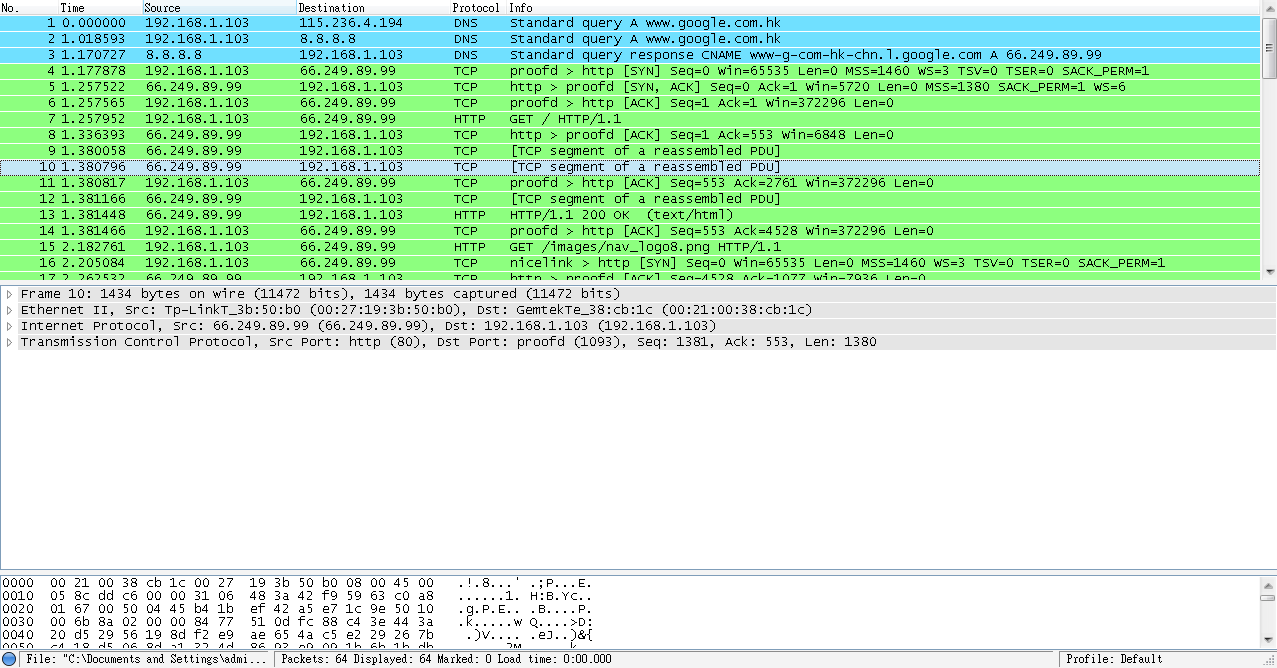
学习用wireshark进行抓包分析

姓名：罗小嘉 学号：2801305018

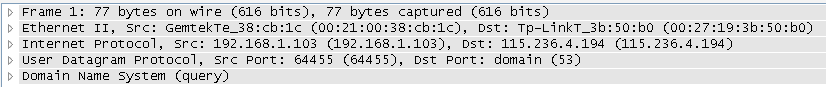
首先，运行wireshark，打开capture interface选择有数据的网卡，点击start便开始进行抓包。我们可以在options里面对包进行过滤。



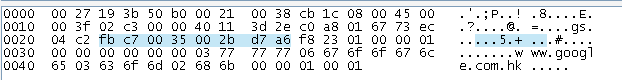
首先，在确保我个人电脑没有arp攻击的情况下。关闭所有可能会请求网络的文件。在点击start后在IE浏览器里面访问[www.google.com.hk](http://www.google.com.hk)后抓到如下数据包。



现在我们开始对抓到的包进行分析。



为所选取的包的结构。结构的显示是完全按照OSI的七层模型来显示的。从上至下分别是物理层，以太网层，IP层，第3层对应的内容，应用层。



为包结构的2进制表示。

现在，我们对抓到的包进行具体分析。



起始的3个DNS包即对[www.google.com.hk进行翻译。把它译为域名所对应的IP](http://www.google.com.hk进行翻译。把它译为域名所对应的IP)。这里，我电脑由于连接了路由器。地址为192.168.1.103。由这个地址向DNS服务器发送请求以返回[www.google.com.hk的地址66.249.89.99](http://www.google.com.hk的地址66.249.89.99)。

然后 在TCP/IP协议中，TCP协议提供可靠的连接服务，采用三次握手建立一个连接。

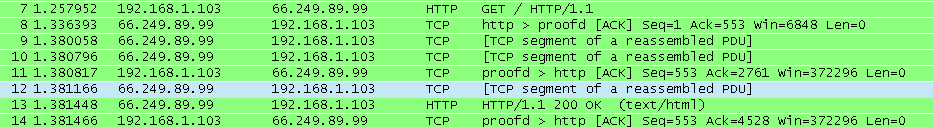


第一次握手：建立连接时，客户端A发送SYN包(SYN=j)到服务器B，并进入SYN\_SEND状态，等待服务器B确认。

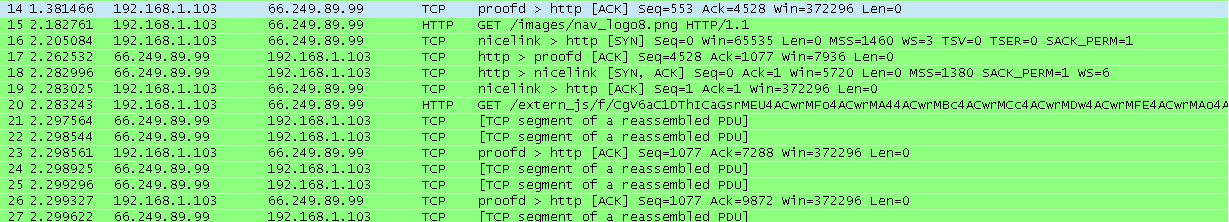
 第二次握手：服务器B收到SYN包，必须确认客户A的SYN(ACK=j+1)，同时自己也发送一个SYN包(SYN=k)，即SYN+ACK包，此时服务器B进入SYN\_RECV状态。

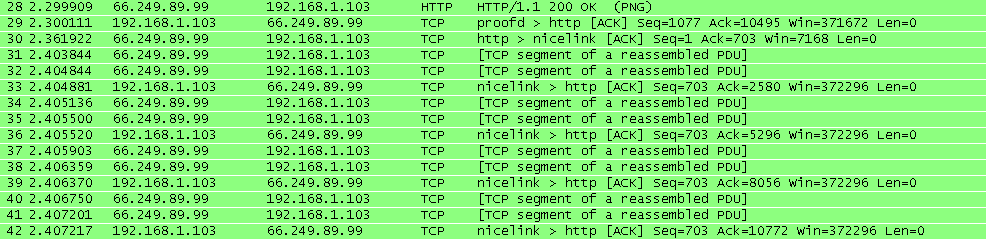
 第三次握手：客户端A收到服务器B的SYN＋ACK包，向服务器B发送确认包ACK(ACK=k+1)，此包发送完毕，客户端A和服务器B进入ESTABLISHED状态，完成三次握手。

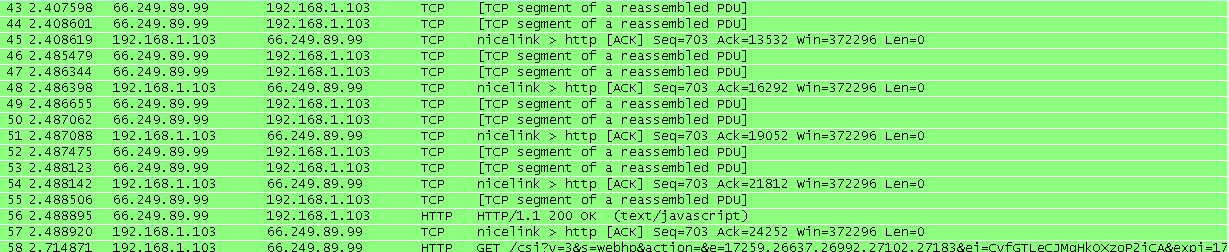
完成三次握手，客户端与服务器开始传送数据。



由我向服务器发送请求，服务器分析请求后，返回确认收到，我确认服务器的返回信息后，服务器开始发送我请求的数据。如下图



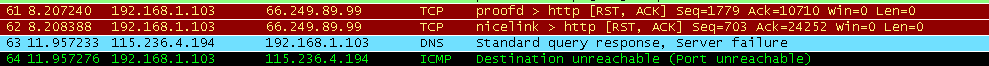






这些包都是服务器在向我传送数据，包括PNG,TEXT等文件。

其中的[TCP segment of a reassembled PDU]的意义是：主机响应一个查询或者命令时如果要回应很多数据（信息）而这些数据超出了TCP的最大MSS时，主机会通过发送多个数据包来传送这些数据（注意：这些包并未被分片）。对wireshark来说这些对相应同一个查询命令的数据包被标记了“TCP segment of a reassembled PDU”。wireshark根据sequence number识别多个数据包是对同一个查询数据包的响应，这些数据包ACK number是相同的，当然number的数值与查询数据包中的next sequence number也是一样的。



上图为抓到的最后几个包。

TCP的终止通过双方的四次握手实现。发起终止的一方执行主动关闭，响应的另一方执行被动关闭。

1. 发起方(client)更改状态为FIN\_WAIT\_1，关闭应用程序进程，发出一个TCP的FIN段；  
2. 接收方收到FIN段，返回一个带确认序号的ACK，同时向自己对应的进程发送一个文件结束符EOF，同时更改状态为CLOSE\_WAIT(server)，发起方接到ACK后状态更改为FIN\_WAIT\_2(client)；  
3. 接收方关闭应用程序进程，更改状态为LAST\_ACK(server)，并向对方发出一个TCP的FIN段；  
4. 发起方接到FIN后状态更改为TIME\_WAIT(client)，并发出这个FIN的ACK确认。ACK发送成功后(2MSL内)双方TCP状态变为CLOSED。

我们不难看出上面的显示的结果的意思。根据TCP协议，主动发起关闭的一方，会进入TIME\_WAIT状态(TCP实现必须可靠地终止连接的两个方向(全双工关闭))，持续2\*MSL(Max Segment Lifetime)，缺省为240秒.

第一次挥手：客户端A发送一个FIN，用来关闭客户A到服务器B的数据传送；  
第二次挥手：服务器B收到这个FIN，它发回一个ACK，确认序号为收到的序号加1，和SYN一样，一个FIN将占用一个序号；  
第三次挥手：服务器B关闭与客户端A的连接，发送一个FIN给客户端A；  
第四次挥手：客户端A发回ACK报文确认，并将确认序号设置为收到序号加1。

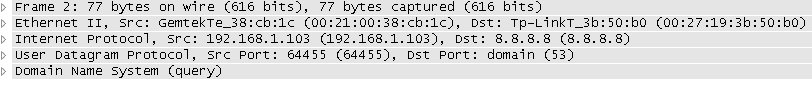
但由收到的包发现并没有产生4次挥手。而出现的是如下的包



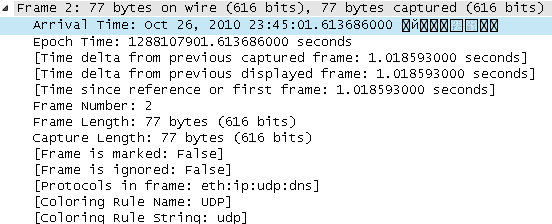
这可能是由于在发送完数据前我过早的关闭了浏览器。

现在任意的以一个包的结构为例对包结构进行分析。在这儿我选了如下一个DNS包。



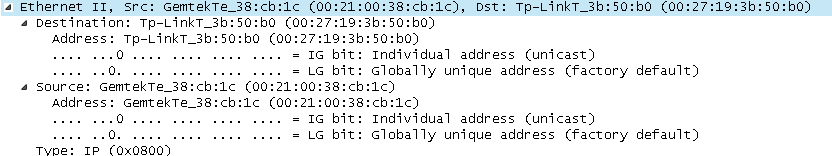


现在对包的结构的各个部分分别分析。

第一行。结构2，发送77字节的报文，抓到77字节的报文。具体内容如下

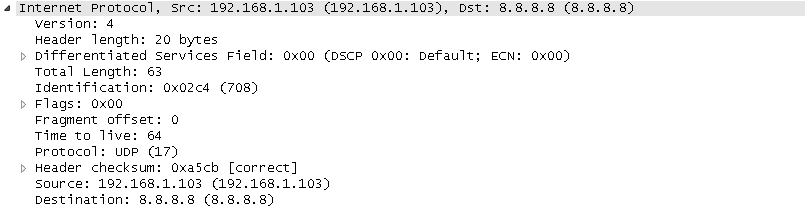
分别说明了，包的到达时间，抓取间隔时间，包的标号，包的大小，是否标记、忽略等。

接下来



以太网层。分别告知目的地和来源。这儿可以看出，destination为路由器分配的地址，source对应的address则为gemtekte\_38:cb:1c

对于IP层

vision：4表示结构为4

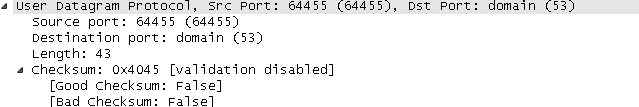
报头长为20字节

无网络分区服务

标志位为0

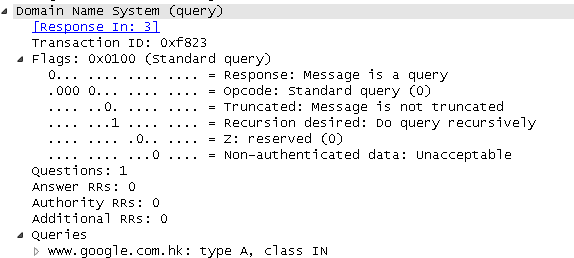
采用UDP协议

报头检查正确。



用户的数据协议，从端口64455到区域端口53

Checksum：0x4045表示确认失败



区域名系统（疑问）

Response in：3表示在frame 3中响应。交换地址的16进制表示为0xf823

对于标志flags，对应英文理解就是了，这儿不做衍叙。

其它的包可以按照相同方法进行分析。