《计算机网络协议开发》实验报告

第1次实验

Wireshark 数据包嗅探实验

姓名：陈越琦

学号：121160005

计算机系13级

邮箱：Yueqichen.0x0@gmail.com

时间: 02/25/2017

实验目的

熟悉Wireshark工具和数据包嗅探

实验内容

1. 捕获的对话文件为附件trace.pcapng
2. 在Filter中输入“http”过滤得到所有与http协议相关的数据包。选择结果中的某个使用HTTP 协议的数据包，右击选择Follow TCP Stream 可以得到浏览器与web服务器之间交换的数据包。

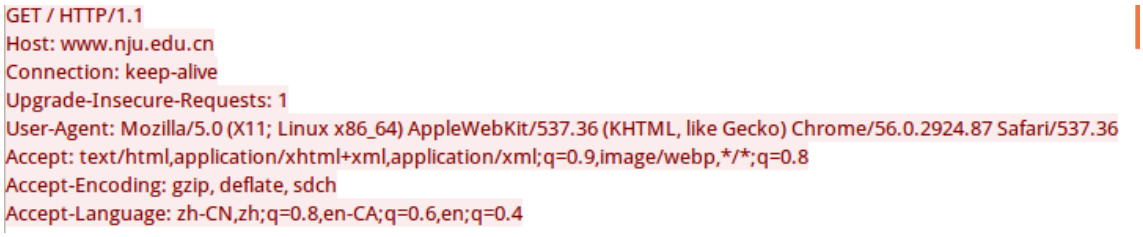

图 1: HTTP 报文

通过阅读HTTP报文的用户代理信息（图1），发现使用的浏览器是Chrome，包含在这个报文中还有其他的信息，比如使用的操作系统是Linux ，均符合事实。在下载的web页面源代码中不存在这样的信息，根据网络分层体系结构，传输内容本身是协议无关的，换句话说，从web服务器向我的浏览器传输的网页超文本，搭载在HTTP报文中，不需要考虑对等进程的细节。

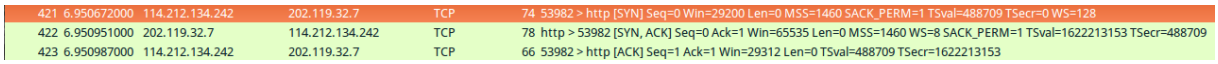

图 2: TCP 三次握手

3. 根据2.中过滤得到的结果可以看出（图2），我的浏览器与web服务器之间使用的传输层协议是TCP。截图中给出了网页访问前，TCP协议进行三次握手的过程。根据网络分层体系结构，TCP作为传输层协议为HTTP(应用层)提供服务，包括拥塞控制等。要想实现为浏览器应用提供web服务的目标，仅仅依靠传输层的协议是不够的，还需要HTTP作为应用层协议提供更多更具体化的支持。

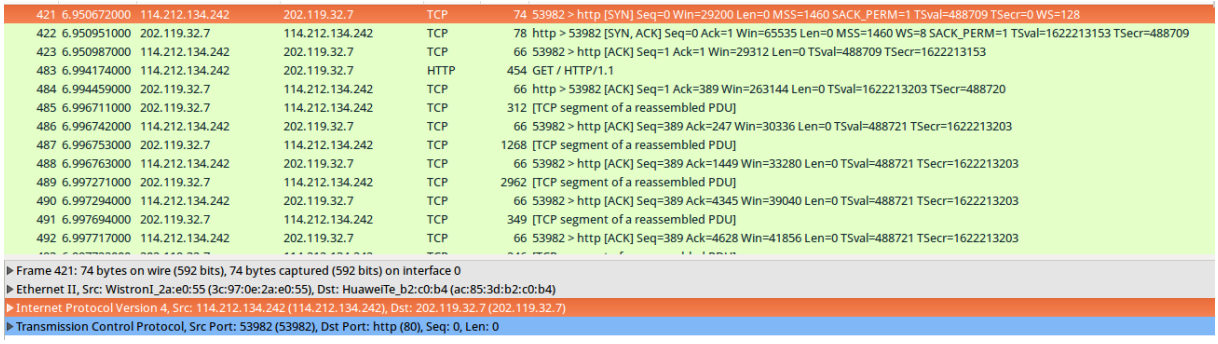

图 3: TCP 报文

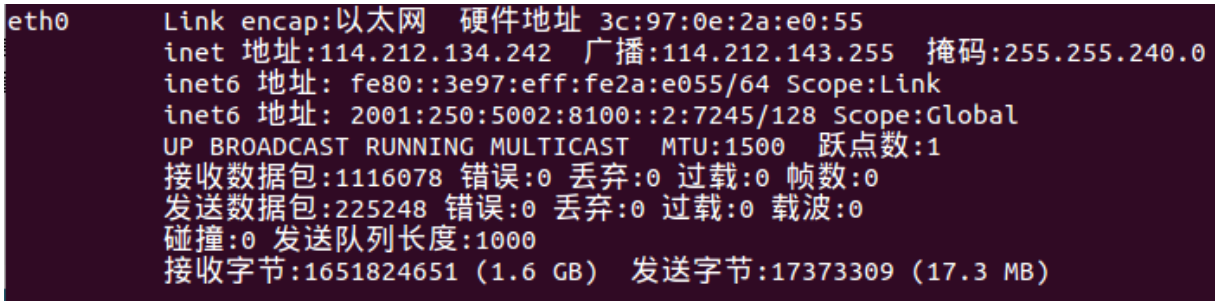

图 4: 链路情况

4. 图3中橘黄色的报文汇总信息是浏览器向web服务器发出的握手信息的内容汇总，从报文首部详细信息窗口中可以看出我的机器IP地址为114.212.134.242，通过ifconfig验证正确（图4）。该数据包的作用是向web服务器发出握手请求，要求建立TCP链接。

5. 在五层网络体系结构中，与IP地址相关的是网络层。网络层负责为网络中的不同主机提供通信服务，承担了将数据报文从源主机通过网络送到目的主机的任务。在报文到达目的主机之后，还需要对应到不同的端口，再从端口对应到进程，才能为应用层提供，而不是网络层直接与应用层相关联。

6. 依然是图3中的报文，可以看出我的有线网卡的MAC地址是3c:97:0e:2a:e0:55（通过图4可以得到验证）。总体上说，分层网络体系结构要求每一个层次都能够独立构建独立的模块，不同层次需要有各自的寻址方案。具体的说，假设适配器（网卡）中的地址是IP地址而不是MAC地址的话，适配器就不能够支持其他的网络层协议，比如说IPX。同时每次开机关机，适配器中的地址都要重新得配置，非常麻烦。所以除了IP地址以外，还需要MAC地址。

7. Wireshark抓到的包中，还有使用UDP和DNS协议的报文。这是两个非常常见的协议。协议阅读笔记在下一小节给出。

8. 协议是规则，保证通信双方的对等体对所需处理的报文的含义达成一致。为了达成一致，协议本身显然要标准化。同时为了提高通信效率和可靠性，协议也需要经过特殊

的设计。但是协议的具体实现不需要标准化，比如说可以使用不同的编程语言实现传输层的协议。TCP/IP协议栈分层实现的主要目标是方便设计和调优，每一层协议只需要考虑需要为上层协议提供哪些服务，以及下一层能够为本层提供哪些服务。假设没有分层，那么每个有通信需求的开发者都需要按照自己的具体需求开发一套完整的通信系统，既要满足应用需求，同时还要考虑底层的硬件实现，费时费力，而且容易出错。有了分层之后，就能够直接使用底层的设计成果，提高效率。在同一层上进行调优也成为可能，比如在传输层可以使用不同的拥塞控制算法应对不同吞吐量和带宽的网络。事实上，分层的思想（或者说逐步抽象）在计算机系统中处处都有体现：比如从CPU的流水实现到ISA的设计再到搭载在ISA之上的操作系统，虚拟机，再比如编译器中，从前端的语法分析和文法分析生成中间代码，再到中间代码优化，最后生成针对不同系统结构的指令序列。

9. 论文阅读感想

协议阅读笔记

UDP

UDP在RFC 768 [1] 中定义。

UDP适用于互连网络，是一个定义在IP之上的协议。UDP面向事务处理，不保证传达和冗余保护。

UDP报文首部信息包括源端口，目的端口，报文长度（以字节为单位，包括了首部和数据）以及检验和。报文首部每个组成部分都是2个字节，一共8个字节。

用户接口包括新建接收端口，接收报文操作（需要向用户返回收到的报文信息（八位组）以及报文的发送端地址和端口）和发送报文操作（需要给定待发送的数据，源地址，目的地址，源端口和目的端口）。

UDP在拿到IP给定的报文之后，要能够知道报文的源地址和目的地址以及协议域组成（即UDP报文首部信息）。一个可行的UDP/IP接口是返回完整的包括了网络层首部的数据包。这样一个接口也使得UDP能够向网络层提供一个完整的网络层数据包，而网络层只要计算检验和就可以了。

UDP适用于Internet Name Server和无关紧要的文件传输，比如我之所有抓到这个包是因为我当时正在使用网易云音乐听歌。

网络层记录UDP的协议编号为17(D)

DNS

DNS是一个复杂的系统，在RFC 1034 [2] 和RFC 1035 [3]中定义，并且在几个附加的RFC中进行了更新。为了避免过多得陷入细节而忽视了DNS本身，我直接阅读了[4]中内容并做笔记如下：

DNS的功能是将主机名转换为IP地址。所有的DNS请求和回答都使用UDP数据报经过端口53发送。

DNS服务器的主要功能是将对用户主机发送的查询请求做出响应。为了缓解单点故障，通信容量的限制，以及降低维护成本，DNS采取了分布式的设计方案。

DNS服务器以层次结构组织起来，自顶向下分别是根服务器，顶级域名服务器（com, org等）和权威服务器（大学，大公司或者其他机构维护的可公共访问的DNS记录）。此外还有本地DNS服务器。当主机发出DNS请求后，请求会先访问本地DNS服务器（相当于本地代理），本地服务器无法解决会向根服务器求助，根服务器返回响应的结果之后，本地服务器再根据响应结果逐步得访问顶级域名服务器，权威服务器，再到对方主机。因此，从请求主机到本地DNS服务器的查询时递归的，其余的查询都是迭代的（本地服务器以自己的名义查询）。

为了提高查询效率，同时降低根服务器的查询压力，DNS服务器普遍缓存查询结果。

论文阅读感想

我想从系统设计的世界观的角度，而不是更为具体的方法论上谈谈我的感想：

1. 系统设计要考虑到现有的系统和资源，不是一切从头开始。TCP/IP的设计的一大目标就是融合当时已有的各种不同的网络，而不是从头开始设计一个完整的统一的网络系统。所以，TCP/IP的设计上也使用了当时已有的存储转发技术和packet switching技术。

2. 系统设计首先要搞清楚系统使用的环境或者说具体的需求。TCP/IP一开始是出于军事的目的设计的，所有系统的容错性是首要目标，商业系统中常见的低价等需求则为其次。

3. 系统的设计有的时候要考虑到实现的方便。TCP/IP在端点保存会话信息，中间节点都是无状态的，而不是将信息保存在数据包中，这样的设计一方面提高了系统的容灾能力（中间节点退出服务不会影响到会话）。另外一方面也考虑到了工程实现上的便捷。

4. 设计出来的系统不可能支持所有的服务，尤其是可能出现而设计之初并不存在的服务。在这种情况下，应该对系统进行分层处理。TCP/IP在实践中就将TCP和IP分层，从而为实现实时数据传输的需求（UDP）提供了可能。

5. 设计出来的系统不可能在所有的评价标准下都是完美的，就像有的同学成绩好，有的同学跑步跑得快。因此要确定首要的设计目标和次要得设计目标，对于次要的设计目标可以放到今后解决甚至直接放弃。TCP/IP在一开始就没有考虑到资源分布式管理。

6. 系统中的某个设计可能出于某个特性的考虑，但是这个设计有的时候反而会损伤这个特性。TCP/IP将会话信息保存在端点以提高容灾能力，相应的，会话的管理也在端点进行。但是如果端点设计有误或者不成熟，反倒会损伤整个Internet。

7. 在系统设计之初，某些特性可能不是目标，但是随着系统投入运行和应用环境的变化，系统也需要加入这些特性。TCP/IP在设计之初，因为是国防项目没有考虑到计费的问题。但是随着使用环境的商业化，计费问题也需要得到解决。

8. 系统中的某些在设计之初并不成熟，在使用过程中还需要不断得修改以满足需求。比特流原先使用EOL标志将数据流分成不同的记录，但是这与合并数据报实现重传矛盾，所以采用了更加弱化的PSH标志来区别。

实验启示

本次实验中，我尝试使用Wireshark工具分析了web服务数据流。通过实验，我掌握了Wireshark使用的基本技巧，加深了对TCP，IP，UDP和DNS协议的理解。更重要的是，我对网络分层设计有了更加深入的认识,这为我在今后的学习与工作中设计大型复杂系统提供了经验指导。

实验时间统计

- 1-8（不包括协议阅读）：30 min
- 协议阅读：20 min
- 9 (论文阅读)：3 hour （论文发表时间过于久远，无论是版面清晰度和语言表达上都拖累阅读速度）
- 实验报告撰写：1 hour

参考文献

[1] https://www.rfc-editor.org/rfc/pdfrfc/rfc768.txt.pdf

[2] https://www.rfc-editor.org/rfc/pdfrfc/rfc1034.txt.pdf

[3] https://www.rfc-editor.org/rfc/pdfrfc/rfc1035.txt.pdf

[4] Computer Networking A Top-Down Approach by James F. Kurose, Keith W. Ross, section 2.5.2 Overview of How DNS works page 133-139 (sixth edition)