

实验 1.1 应用层

09021227 金桥

2023 年 11 月 21 日

1 实验内容

1. 学会使用 Wireshark 抓包软件，会使用过滤器。
2. 学习 Wireshark 基本操作：重点掌握捕获过滤器和显示过滤器。分析 HTTP 和 DNS 协议。
3. 测试 curl 命令，访问一个 Web 页面。（选做）
4. 利用 telnet 命令测试 GET 命令，访问 www.baidu.com。（选做）
5. 利用 telnet 命令测试 SMTP 服务，解析其过程。（选做）
6. 测试 tracert 命令，并解析其过程。
7. 使用 nslookup 查询域名信息，简要分析。

2 实验过程

2.1 Wireshark 基本操作

2.1.1 捕获过滤器

在启动 Wireshark 之后可以在捕获过滤器的输入框中输入过滤器，例如过滤 ARP 包。如图 1 所示：

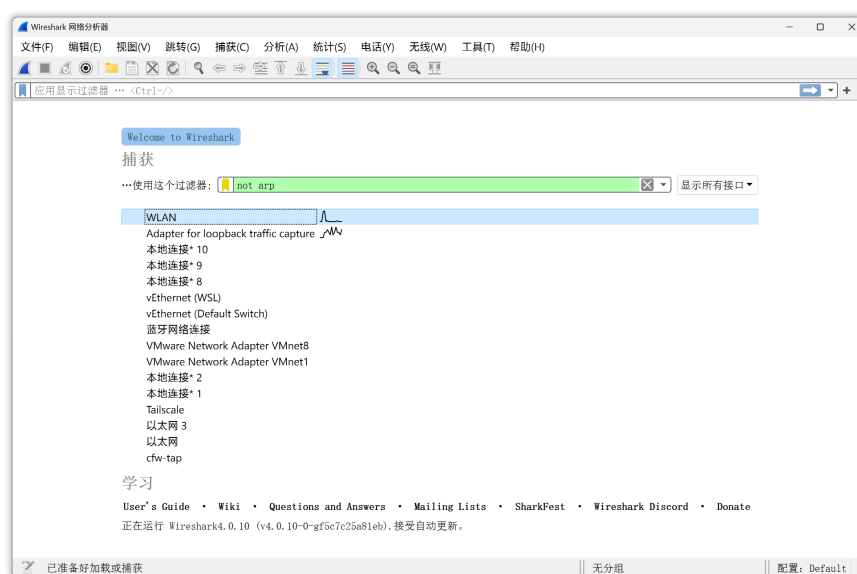


图 1: Wireshark 捕获过滤器过滤 ARP 包。

2.1.2 显示过滤器

在开始捕获之后可以在显示过滤器的输入框中输入过滤器，例如过滤 ARP 包。如图 2 所示：

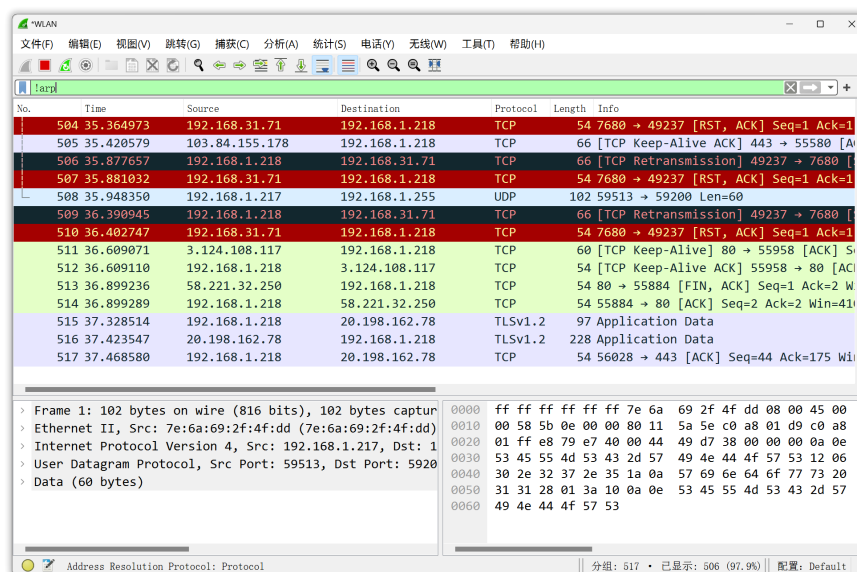


图 2: Wireshark 显示过滤器过滤 ARP 包。

2.1.3 分析 HTTP 协议

随机查看一个 HTTP 分组的内容，如图 3 所示：

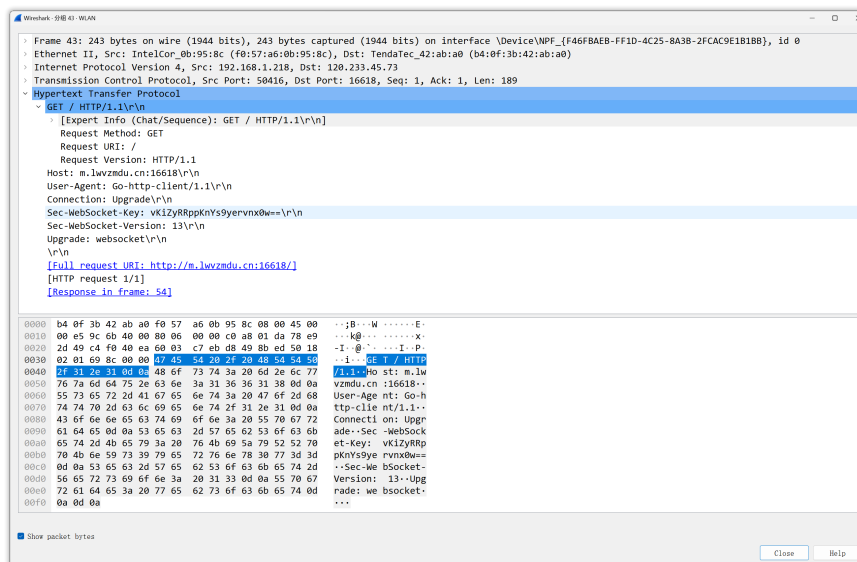


图 3: Wireshark 查看 HTTP 分组内容

查阅相关资料，分析如下：这是一个用于建立 WebSocket 连接的 HTTP 请求。客户端发送这个请求到服务器，请求升级当前的 HTTP 连接到 WebSocket 连接。具体细节如下：

- GET / HTTP/1.1\r\n: 这是请求行，包含 GET 方法、请求的资源 URI，以及 HTTP 版本。
- Host: m.lwvzmdy.cn:16618\r\n: 这是 Host 头部，指定了请求的目标主机和端口。

- **User-Agent:** Go-http-client/1.1\r\n: 这是 User-Agent 头部, 提供了发起请求的客户端软件的信息。
- **Connection:** Upgrade\r\n: 这是 Connection 头部, 指示这个 HTTP 连接应该被升级。
- **Sec-WebSocket-Key:** vKiZyRRppKnYs9yervnx0w==\r\n: 这是 Sec-WebSocket-Key 头部, 包含了一个 Base64 编码的随机值, 用于 WebSocket 握手过程。
- **Sec-WebSocket-Version:** 13\r\n: 这是 Sec-WebSocket-Version 头部, 表示客户端支持的 WebSocket 协议版本。
- **Upgrade:** websocket\r\n: 这是 Upgrade 头部, 指定了要升级到的协议 (在这个例子中是 WebSocket)。
- **\r\n:** 这是一个空行, 表示头部字段的结束和消息体的开始。这里没有消息体。

2.1.4 分析 DNS 协议

随机查看一个 DNS 响应分组的内容, 如图 4 所示:

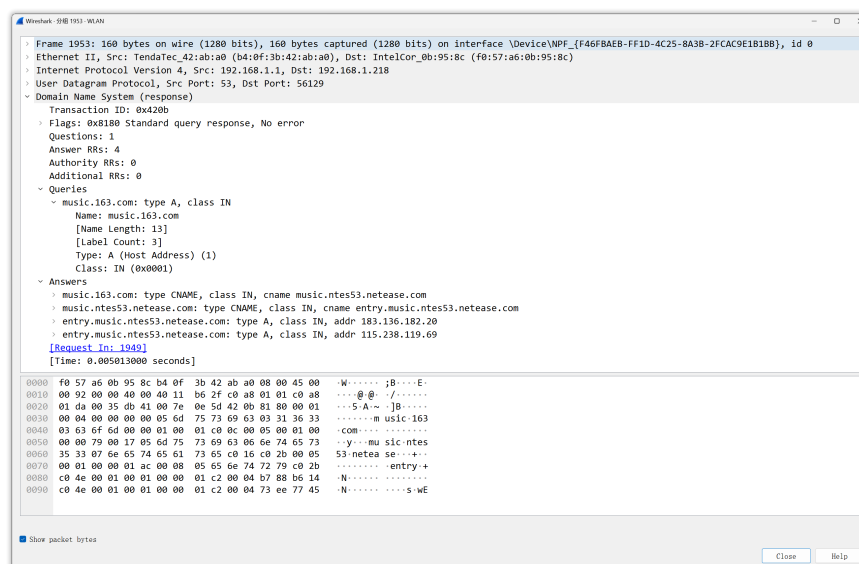


图 4: Wireshark 查看 DNS 分组内容

查阅相关资料, 分析如下: 这是一个 DNS 查询的响应, 它包含了对 music.163.com 的查询结果。具体细节如下:

- **Transaction ID: 0x420b:** 这是事务 ID, 用于匹配请求和响应。
- **Flags: 0x8180 Standard query response, No error:** 这是响应的标志字段, 表示这是一个标准查询响应, 并且没有错误。
- **Questions: 1:** 这是问题部分的数量, 表示响应对应的请求中包含了一个问题。
- **Answer RRs: 4:** 这是回答资源记录的数量, 表示响应中包含了四个资源记录。
- **Authority RRs: 0 和 Additional RRs: 0:** 这两个字段表示响应中没有权威资源记录和附加资源记录。
- **Queries 部分** 包含了请求中的查询信息, 包括查询的域名以及查询的类型和类别。
- **Answers 部分** 包含了查询的回答。这里有四个回答。

2.2 curl 命令

打开终端，输入 `curl www.google.com` 命令，得到以下输出：

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows

PS C:\Users\26354> curl www.google.com

StatusCode      : 200
StatusDescription : OK
Content          : <!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="zh-CN"><head><meta con
                  tent="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1
                  x/ ...
RawContent       : HTTP/1.1 200 OK
                  Connection: close
                  Content-Security-Policy-Report-Only: object-src 'none';base-uri 'self';script-src 'nonce-OVIdMit00s
                  HysK4IeKMSnw' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'uns ...
Forms            : {}
Headers          : {[Connection, close], [Content-Security-Policy-Report-Only, object-src 'none';base-uri 'self';scrip
                  t-src 'nonce-OVIdMit00sHysK4IeKMSnw' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline'
                  https: http;report-uri https://csp.withgoogle.com/csp/gws/other-hp], [Cache-Control, private, max
                  -age=0], [Content-Type, text/html; charset=UTF-8] ... }
Images           : {@{innerHTML=; innerText=; outerHTML=<IMG id=hplogo style="PADDING-BOTTOM: 14px; PADDING-TOP: 28px;
                  PADDING-LEFT: 0px; PADDING-RIGHT: 0px" alt=Google src="/images/branding/googlelogo/1x/googlelogo_w
                  hite_background_color_272x92dp.png" width=272 height=92>; outerText=; tagName=IMG; id=hplogo; style
                  =PADDING-BOTTOM: 14px; PADDING-TOP: 28px; PADDING-LEFT: 0px; PADDING-RIGHT: 0px; alt=Google; src=/i
                  mages/branding/googlelogo/1x/googlelogo_white_background_color_272x92dp.png; width=272; height=92}}
InputFields      : {@{innerHTML=; innerText=; outerHTML=<INPUT type=hidden value=zh-CN name=hl>; outerText=; tagName=I
                  NPUT; type=hidden; value=zh-CN; name=hl}, @{{innerHTML=; innerText=; outerHTML=<INPUT type=hidden va
                  lue=hp name=source>; outerText=; tagName=INPUT; type=hidden; value=hp; name=source}, @{{innerHTML=;
                  innerText=; outerHTML=<INPUT type=hidden name=biw>; outerText=; tagName=INPUT; type=hidden; name=bi
                  w}, @{{innerHTML=; innerText=; outerHTML=<INPUT type=hidden name=bih>; outerText=; tagName=INPUT; ty
                  pe=hidden; name=bih} ... }
Links            : {@{innerHTML=<SPAN class=gbtb2></SPAN><SPAN class=gbts>搜索</SPAN>; innerText=搜索; outerHTML=<A id
                  =gb_1 class="gbzt gbz0l gbp1" href="https://www.google.com.hk/webhp?tab=ww"><SPAN class=gbtb2></SPA
                  N><SPAN class=gbts>搜索</SPAN></A>; outerText=搜索; tagName=A; id=gb_1; class=gbzt gbz0l gbp1; href
                  =https://www.google.com.hk/webhp?tab=ww}, @{{innerHTML=<SPAN class=gbtb2></SPAN><SPAN class=gbts>图
                  片</SPAN>; innerText=图片; outerHTML=<A id=gb_2 class=gbzt href="https://www.google.com.hk/imghp?hl
                  =zh-CN&tab=wi"><SPAN class=gbtb2></SPAN><SPAN class=gbts>图片</SPAN></A>; outerText=图片; tagNa
                  me=A; id=gb_2; class=gbzt; href=https://www.google.com.hk/imghp?hl=zh-CN&tab=wi}, @{{innerHTML=<
                  SPAN class=gbtb2></SPAN><SPAN class=gbts>地图</SPAN>; innerText=地图; outerHTML=<A id=gb_8 class=gb
                  zt href="http://ditu.google.cn/maps?hl=zh-CN&tab=wl"><SPAN class=gbtb2></SPAN><SPAN class=gbts>
                  地图</SPAN></A>; outerText=地图; tagName=A; id=gb_8; class=gbzt; href=http://ditu.google.cn/maps?hl
                  =zh-CN&tab=wl}, @{{innerHTML=<SPAN class=gbtb2></SPAN><SPAN class=gbts>Play</SPAN>; innerText=Pl
                  ay; outerHTML=<A id=gb_78 class=gbzt href="https://play.google.com/?hl=zh-CN&tab=w8"><SPAN clas
                  s=gbtb2></SPAN><SPAN class=gbts>Play</SPAN></A>; outerText=Play; tagName=A; id=gb_78; class=gbzt; h
                  ref=https://play.google.com/?hl=zh-CN&tab=w8} ... }
ParsedHtml       : mshtml.HTMLDocumentClass
RawContentLength : 51059
```

图 5: curl 命令访问 `www.google.com` 输出

2.3 telnet 命令

2.3.1 测试 GET 命令

telnet 命令使用 GET 访问 `www.baidu.com` 步骤如下：

1. 打开终端输入 `telnet www.baidu.com 80`
2. 按下 `Ctrl+]` 之后按下回车，打开回显
3. 输入 `GET / HTTP/1.1` 并按下回车
4. 输入 `Host: www.baidu.com` 并按三次回车，如图 6 所示
5. 屏幕上显示出 GET 返回的内容

```
Windows PowerShell
com/" target="_blank" class="mnav c-font-normal c-color-t">学术</a><div class="mnav s-top-more-btn"><a href="//w
ww.baidu.com/more/" name="tj_briicon" class="s-bri c-font-normal c-color-t" target="_blank">更多</a></div></div>
<div id="u1" class="s-top-right s-isindex-wrap"><a class="s-top-login-btn c-btn c-btn-primary c-btn-mini lb" style
="position:relative;overflow:visible" name="tj_login" href="//www.baidu.com/bdorz/login.gif?login&tpl=mn&u
=http%3A%2F%2Fwww.baidu.com%2F%3Fbdorz_come%3D1">登录</a></div><div id="head_wrapper" class="head_wrapper s-isin
dex-wrap s-ps-islite"><div class="s_form"><div class="s_form_wrapper"><div id="lg" class="s-p-top"><map name="mp"><area style="outline:0" hidefocus="true" shape="rect" coords="0,0,270,129" hre
f="//www.baidu.com/s?wd=%E7%99%B%E5%BA%A6%E7%83%AD%E6%90%9C&sa=ire_dl_gh_logo_texing&rsv_dl=igh_logo_pcs"
form id="form" name="f" action="//www.baidu.com/s" class="fm"><input type="hidden" name="ie" value="utf-8"> <input
type="hidden" name="f" value="8"> <input type="hidden" name="rsv_bp" value="1"> <input type="hidden" name="rsv_id
x" value="1"> <input type="hidden" name="ch" value=""> <input type="hidden" name="tn" value="baidu"> <input type="
" maxlength="255" autocomplete="off"> </span><span class="s_btn_wr"><input type="submit" id="su" value="百度一
下" class="bg s_btn"> </span><input type="hidden" name="rn" value=""> <input type="hidden" name="fenlei" value="2
56"> <input type="hidden" name="oq" value=""> <input type="hidden" name="rsv_pq" value="b9ff093e0000e419"> <input
type="hidden" name="rsv_t" value="3635FYbdbC8tlWmudZmYaUnaucNe+RzTzNEGqg/JuniQU10WL5mtMQehIrU"> <input type="hidde
n" name="rqlang" value="cn"> <input type="hidden" name="rsv_enter" value="1"> <input type="hidden" name="rsv_dl" v
alue="ib"></form></div></div></div><div id="bottom_layer" class="s-bottom-layer s-isindex-wrap"><div class="s-bott
om-layer-content"><p class="lh"><a class="text-color" href="//home.baidu.com/" target="_blank">关于百度</a></p>
<p class="lh"><a class="text-color" href="//ir.baidu.com/" target="_blank">About Baidu</a></p><p class="lh"><a cl
ass="text-color" href="//www.baidu.com/duty" target="_blank">使用百度前必读</a></p><p class="lh"><a class="
text-color" href="//help.baidu.com/" target="_blank">帮助中心</a></p><p class="lh"><a class="text-color" href=
="//www.beian.gov.cn/portal/registerSystemInfo?recordcode=11000002000001" target="_blank">京公网安备1100000200
0001号</a></p><p class="lh"><a class="text-color" href="//beian.miit.gov.cn/" target="_blank">京ICP证030173号<
/a></p><p class="lh"><span id="year" class="text-color"></span></p><p class="lh"><span class="text-color">互联
◆药品信息服务资格证书 (京)-经营性-2017-0020</span></p><p class="lh"><a class="text-color" href="//ww
w.baidu.com/licence/" target="_blank">信息网络传播视听节目许可证 0110516</a></p></div></div></div></div>
<script type="text/javascript">var date=new Date,date.getFullYear();document.getElementById("year").innerT
ext="@"+year+" Baidu "</script></body></html>
```

图 6: telnet 命令使用 GET 访问 www.baidu.com 输出

2.3.2 测试 SMTP 服务

以 QQ 邮箱为例，使用 telnet 发送邮件的步骤如下：

1. 登录 QQ 邮箱网页端，获取授权码
2. 将邮箱地址与授权码转为 Base64 编码保存
3. 打开终端，执行 telnet smtp.qq.com 25，依次键入以下内容，每次输入后按回车：
 - helo qq.com
 - auth login
 - 邮箱地址的 Base64 编码
 - 邮箱授权码的 Base64 编码
4. 此时成功登录进邮箱，依次键入以下内容以发送邮件，每次输入后按回车：
 - mail from: <sender@mail.address>
 - rcpt to: <receiver@mail.address>
 - data
 - from: <sender@mail.address>
 - to: <receiver@mail.address>
 - subject: the subject of mail
 - the content of mail
 - .
5. 成功发送邮件，终端以及邮箱截图如图 7 所示



图 7: telnet 命令测试 QQ 邮箱 SMTP 服务，右图为邮箱截图。

2.4 tracert 命令

以 `www.github.com` 为例测试 `tracert` 命令，输出如图 8 所示。



图 8: tracert 命令测试 `www.github.com`

在地图上标记出 IP 对应的位置，如图 9 所示。

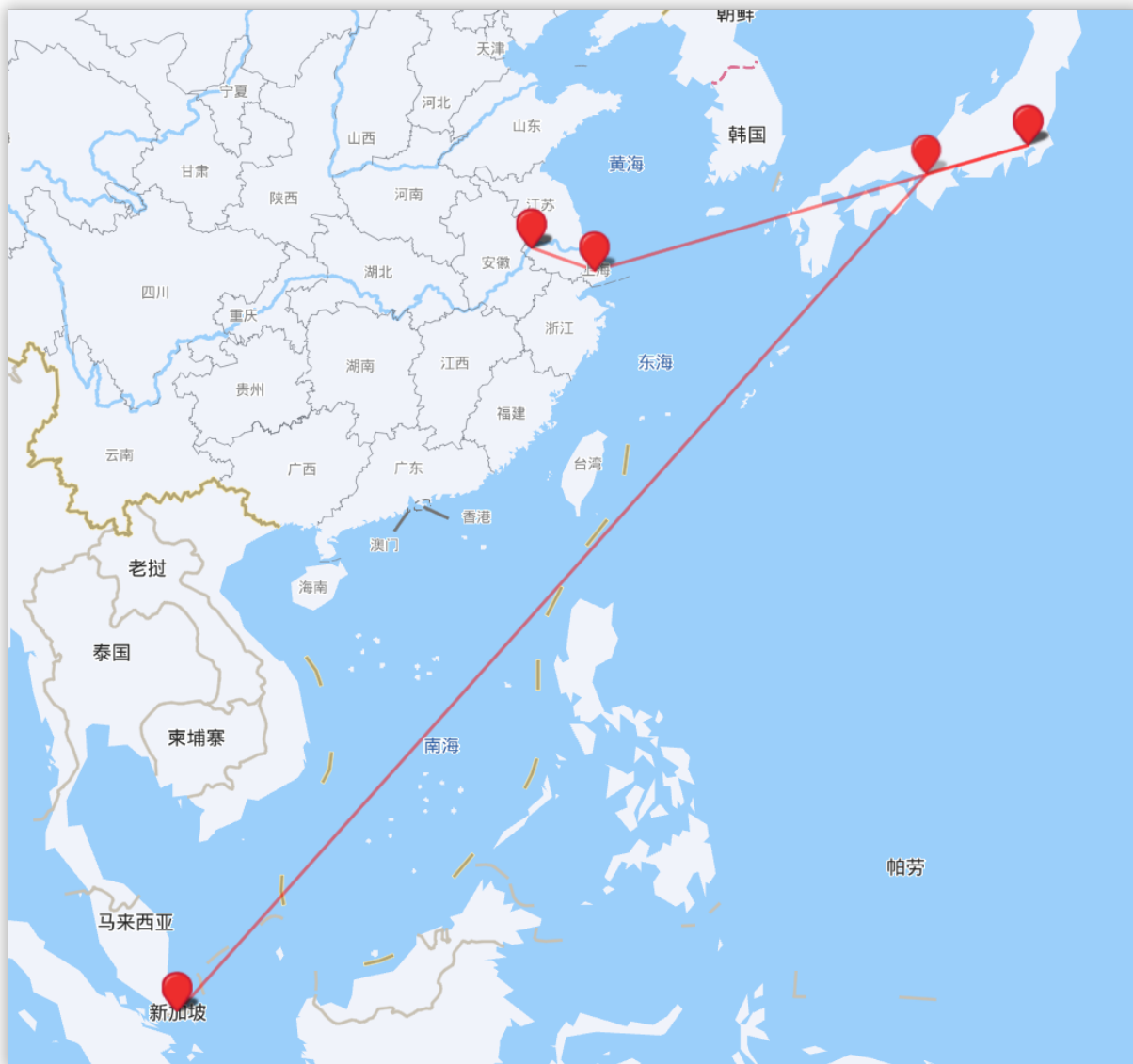


图 9: 在地图上标记出 `tracert` 输出 IP 对应的位置

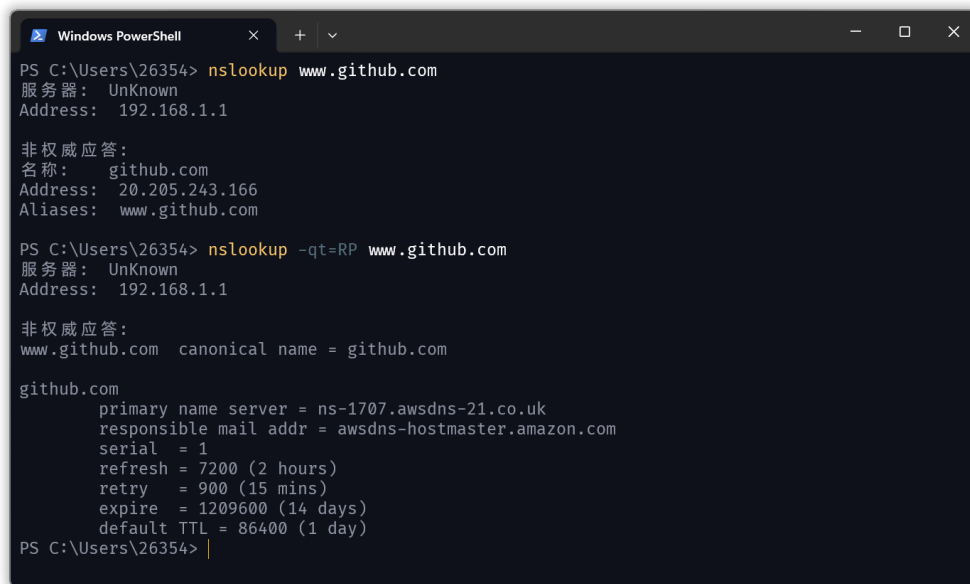
首先数据包经过路由器（第 1 行），进入校园网（第 2-5 行），之后经过多个中转节点（第 6-23 行），最终到达 GitHub 的服务器（第 24 行）。

数据包地理位置的变化：南京 → 上海 → 日本东京 → 日本大阪 → 新加坡。其中境外的服务器的域名均以 `ntwk.msn.net` 结尾，推测为 GitHub 的数据中心。

值得注意的是，有几个跃点没有响应，显示为“请求超时”。这可能是因为这些路由器被配置为不响应 ICMP Echo Request 消息。

2.5 nslookup 命令

以 `www.github.com` 为例测试 `nslookup` 命令，输出如图 8 所示，包括默认参数以及指定 `-qt=RP` 参数的输出。



```
PS C:\Users\26354> nslookup www.github.com
服务器:  UnKnown
Address:  192.168.1.1

非权威应答:
名称:     github.com
Address:  20.205.243.166
Aliases:  www.github.com

PS C:\Users\26354> nslookup -qt=RP www.github.com
服务器:  UnKnown
Address:  192.168.1.1

非权威应答:
www.github.com canonical name = github.com

github.com
primary name server = ns-1707.awsdns-21.co.uk
responsible mail addr = awsdns-hostmaster.amazon.com
serial = 1
refresh = 7200 (2 hours)
retry = 900 (15 mins)
expire = 1209600 (14 days)
default TTL = 86400 (1 day)
PS C:\Users\26354> |
```

图 10: nslookup 命令测试 `www.github.com`

分析如下:

- 输出 `www.github.com canonical name = github.com` 表示 `www.github.com` 是一个别名, 它的规范名称 (即实际的域名) 是 `github.com`。
- 在 `github.com` 下面的部分是关于 `github.com` 的 RP 记录的信息, 但是这个信息看起来更像是 SOA (Start of Authority) 记录的内容, 而非 RP 记录。猜测可能是因为 `github.com` 没有设置 RP 记录, 所以 `nslookup` 命令返回了 SOA 记录。SOA 记录包含了关于该域的权威 DNS 服务器和其他元数据的信息。
 - `primary name server = ns-1707.awsdns-21.co.uk`: 这是 `github.com` 的主 DNS 服务器的域名。
 - `responsible mail addr = awsdns-hostmaster.amazon.com`: 这是负责管理这个域的人员的电子邮件地址。这里邮件地址的 `@` 被替换为了 `.`
 - `serial = 1`: 这是区域文件的序列号。每当区域文件有更改时, 这个数字就会增加。
 - `refresh = 7200 (2 hours)`: 从属服务器多久检查一次更新的时间。
 - `retry = 900 (15 mins)`: 如果从属服务器尝试联系主服务器失败, 它应该多久后重试。
 - `expire = 1209600 (14 days)`: 从属服务器多久没有联系到主服务器后, 应该停止回答关于这个区域的查询。
 - `default TTL = 86400 (1 day)`: 其他服务器和应用应该将这个区域的信息缓存多久。

3 实验体会

通过这次实验, 我学习了如何使用 Wireshark 抓包软件以及 `curl` 等命令的使用方法。对于网络有了更深刻的认识。