# 暨南大学本科实验报告专用纸

课程名称 计算机网络实验 成绩评定

实验项目名称 数据链路层和网络层协议分析 指导教师 潘冰

实验项目编号 07 实验项目类型 验证 实验地点 b402

学生姓名 邓芷灵 学号 2019051115

学院 智能科学与工程学院 系 专业 信息安全

实验时间2021 年11月2日 上 午～11月9日上 午

## 实验目的

1. 理解链路层、网络层主要协议格式，以及协议的工作原理
2. 理解网关和子网掩码概念
3. 学会利用网络嗅探器（如**Wireshark**）分析协议格式和协议的工作过程
4. 学会使用ping、tracert、arp等命令并使用嗅探器分析其工作过程。

## 实验内容

1. 用嗅探器捕获数据包。
2. 分析以太网帧、ARP协议、IP协议、ICMP协议格式
3. 分析PING、TRACERT、ARP命令的工作过程
4. 通过修改主机的网关为指定默认网关、本机IP地址或不设置网关，观察ping的结果，用嗅探器捕获数据包并分析。

## 实验设备

计算机一台（主机A），手机一部（主机B）。

|  |  |  |
| --- | --- | --- |
|  | **IP地址** | **MAC地址** |
| **主机A** | 192.168.2.144 | DC-F5-05-50-EB-13 |
| **主机B** | 192.168.2.241 | 7c:2a:db:03:0a:1d |

## 实验环境

局域网

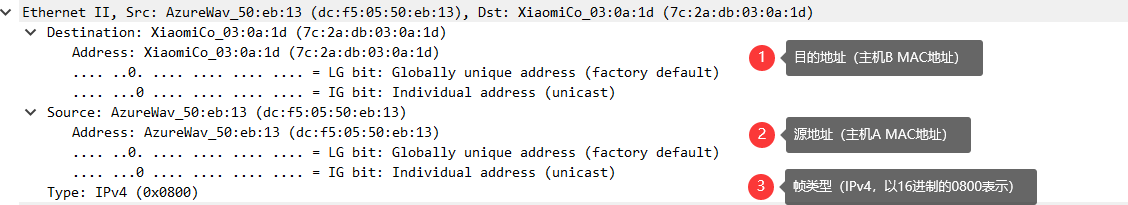
## 实验步骤

### 安装Wireshark

### 以太网协议分析

从主机A上向主机B发PING检测报文，捕获以太数据帧，记录并分析MAC帧各字段的含义。

主机A（192.168.2.144）Ping主机B（192.168.2.241），Wireshark捕获ICMP数据包，对其中一个Ping请求数据包进行分析：



如图所示，以太网首部由三部分构成：目的地址（Dst, Destination）、源地址（Src, Source）、帧类型（Type）。其中目的地址和源地址即主机B和主机A的MAC地址，帧类型代号0x0800表明这是个IPv4数据报（ICMP依靠IP协议完成任务），此外还有代表IPv6的0x86DD、代表ARP的0x0806等。

其中：



等号左边表示的是以6个二进制数表示的MAC地址。

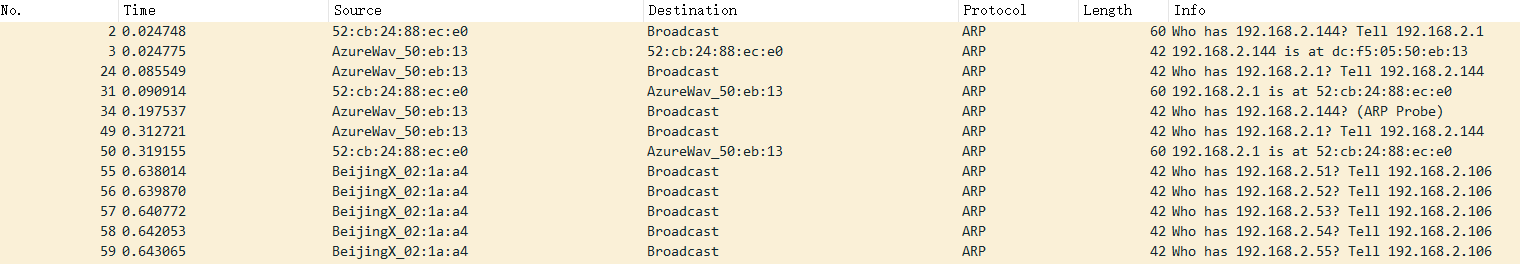
LG bit字段（在IEEE 802标准中又名为U/L bit）突出显示了MAC地址中的一个标志位，图中该标志位的值为0。该标志位用于指示MAC地址是由本地管理员（1）还是由通用管理员（0）分配的，当该位为0时，MAC地址必须是全局唯一的；为1时，MAC仅要求是本地唯一的。

IG bit字段突出显示了MAC地址中的另一个标志位，图中该标志位的值为0。该标志位用于指示MAC地址是一个单独的地址（0）还是一组地址（1）。如代表广播的全1的MAC地址就是一个特殊的组MAC地址。

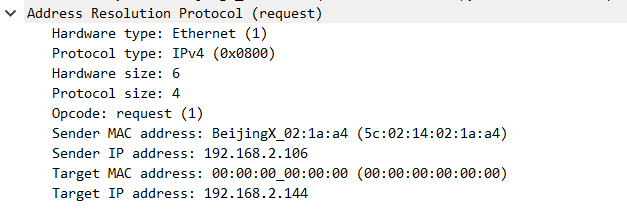
### ARP协议分析

* 断网一分钟后重连，ARP报文捕获

进入DOC窗口，通过arp -d 192.168.2.241删除主机B的ARP记录，断网一分钟后重连，Wireshark捕获ARP报文如下：

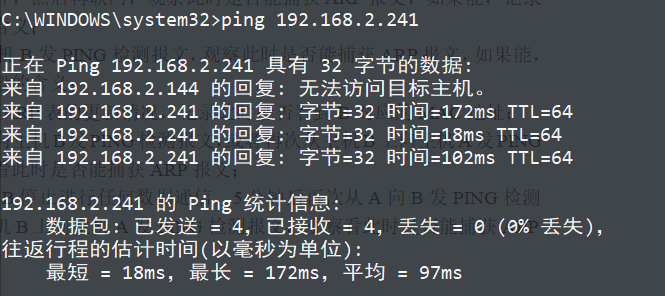


选取其中一个分析ARP协议报文字段：

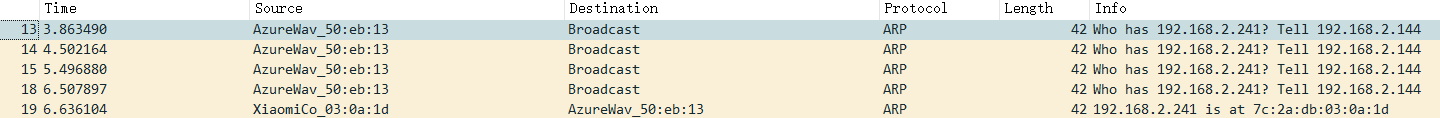


|  |  |
| --- | --- |
| **字段名** | **说明** |
| Hardware type | 硬件类型，图中1即0x0001（以太网） |
| Protocol type | 协议类型 |
| Hardware size | 硬件地址长度（byte），一般为6（即6字节长的MAC地址） |
| Protocol size | 协议地址长度（byte），一般为4（即4字节长的IPv4地址） |
| Opcode | 操作码，1为ARP请求；2为APR应答；3为RARP请求；4为RARP应答 |
| Sender MAC address | 源硬件地址，图中为WIFI路由器的MAC地址 |
| Sender IP address | 源协议地址，图中为WIFI路由器的IP地址 |
| Target MAC address | 目标硬件地址，图中全0表示待补充 |
| Target IP address | 目标协议地址，图中为主机B的IP地址 |

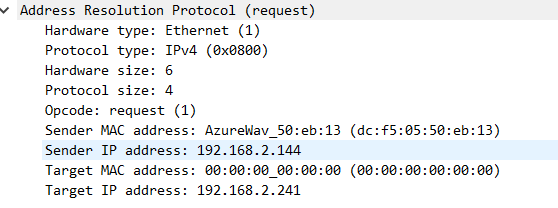
* 主机A Ping主机B，ARP报文捕获



Wireshark捕获ARP报文如下：

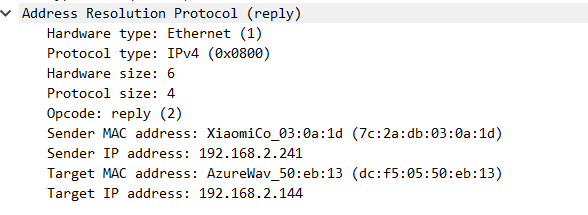


选取其中一个ARP请求报文进行分析：



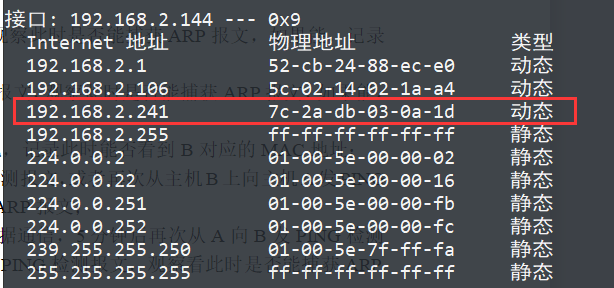
注意到其中源MAC地址与主机A的MAC地址是对应的，源IP地址、目标IP地址分别与主机A、B对应，其中目标MAC地址全0表示待补充，此时还未收到主机B的响应。

对于ARP响应报文：

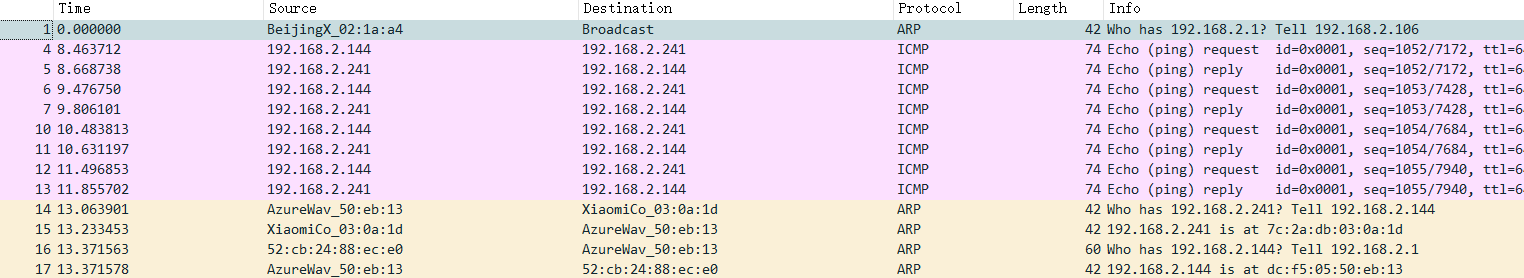


报文各字段含义与请求报文相同。内容上的唯一区别是填写了主机B的MAC地址作为源MAC地址。

* 通过arp -a查看ARP表，可以看到主机B对应的MAC地址：

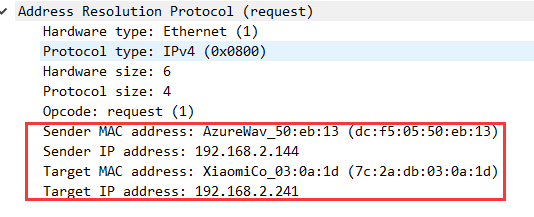


* 主机A再次Ping主机B，查看ARP报文捕获情况



注意到与第一次Ping时的区别：主机A与主机B之间的ARP请求和响应不再在Ping之前发生，而是出现在Ping之后。

对于出现在Ping之后的ARP请求与响应报文（如下图所示），发现报文中双方MAC地址均是已知的：



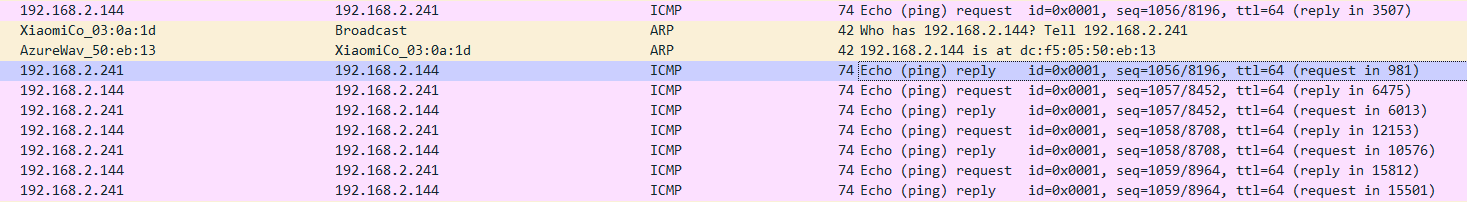
这是因为刷新ARP缓存条目的机制之一——单播轮询（Unicast Poll），该机制定期向远程主机发送点对点的ARP请求，主动轮询远程主机，如果连续N（通常为2）次轮询都未收到ARP响应，则删除ARP缓存条目，以避免无效资源的占用。

所以，第一次Ping时，主机A在ARP表中写入了主机B的缓存条目；第二次Ping时，因为已有缓存条目可知主机B的MAC地址，不再发送ARP请求。

* 主机A与主机B停止数据通信五分钟后再Ping

停止数据通信五分钟实质上是为了等待ARP缓存条目刷新，由于前文所述的单播轮询机制，将主机B断网以保证其不再响应ARP。

ARP报文捕获情况如下：

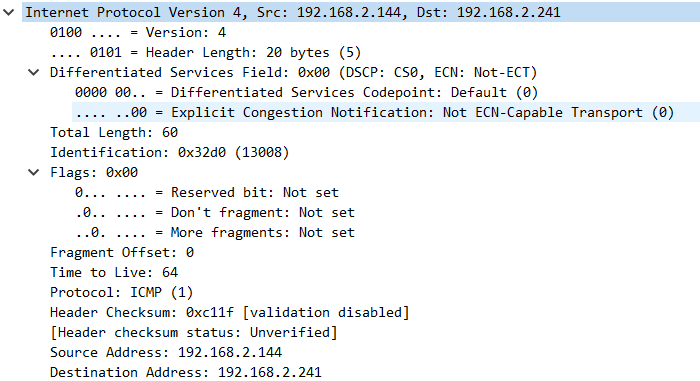


注意到当主机A发送Ping请求后，由主机B发出了ARP请求报文，请求主机A的MAC地址。主机A回复MAC地址后收到主机B的Ping响应。这表明停止数据通信期间主机A中对应的缓存条目仍未删除，而主机B中的缓存条目则已过期，因此重新发起了ARP请求。

### IP协议分析

* 从主机A上向主机B发PING检测报文，捕获IP数据包，记录并分析各字段的含义，并与IP数据包格式进行比较;

捕获IP数据包如下图所示：



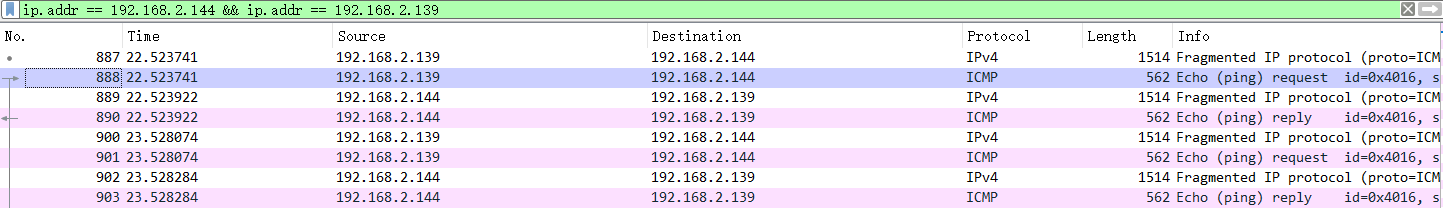
分析如下：

|  |  |
| --- | --- |
| **字段名** | **说明** |
| Version | 0-3位，版本名。0100表示IP协议的版本4 |
| Header Length | 4-7位，首部长度。0101表示首部长度为5\*32bit，即20字节（160位）。这也是首部长度的最小值。若报文中存在可选字段（第160位及以后），则会大于20字节。 |
| Differentiated Services Field | 8-15位，区分服务域。其中：   * 8-13位：区分服务，一般不适用。0为默认 * 14-15位：显式堵塞通告，允许在不丢弃报文的同时通知对方网络拥塞的发生。0表示不可用 |
| Total Length | 16-31位，总长度。60表示报文总长度为60字节 |
| Identification | 32-47位，标识符。0x32d0即十进制的13008，用于唯一地标识一个报文的所有分片 |
| Flags | 48-50位，标志。其中：   * 第48位：保留且必须为0 * 第49位：DF位（Don’t Fragment，禁止分片），为0时表示允许分片 * 第50位：MF位（More Fragment，更多分片），为1时表示后面还有分片   此处3位均为0，表明允许分片且后面没有分片 |
| Fragment Offset | 51-63位，分片偏移。指明了这个分片相对于原始报文开头的偏移量。图中该值为0，表明这是该报文的第一个分片（结合MF位为0可知，这也是唯一一个分片，因为报文总长度为60字节，小于MTU值1500，不需要分片） |
| Time to Live | 64-71位，存活时间（即TTL）。指明了报文的存活时间避免其陷入环路时永远存在。原本以秒为单位，但实际上指的是跳数：每经过一个路由器都将减1，为0时报文被丢弃。 |
| Protocol | 72-79位，协议。指明了报文数据段使用的协议。此处协议号为1，代表的是ICMP。专门的[协议号列表](https://zh.wikipedia.org/wiki/IP%E5%8D%8F%E8%AE%AE%E5%8F%B7%E5%88%97%E8%A1%A8)由IANA/ICANN维护。 |
| Header Checksum | 80-95位，首部检验和。报文所经过的路由器都要根据报文首部重新计算首部检验和并与该字段比对，若不一致则说明出现差错，丢弃该报文。图中的[validation disabled]是因为Wireshark不自动做检验和，右键该字段，勾选validate the IPv4 checksum if possible后显示为[correct]表示检验和正确。 |
| Source Address | 96-127位，源IP地址。发送IP数据报的主机的IP地址，图中为主机A的IP地址。 |
| Destination Address | 128-159位，目的IP地址。接收IP数据报的主机的IP地址，图中为主机B的IP地址。 |

* 使用ping命令，指定数据包长度，如ping -l 2000，使用嗅探器观察IP分片情况，并分析**分片和重组**过程。

主机B的IP地址更改为192.168.2.139，其他不变。

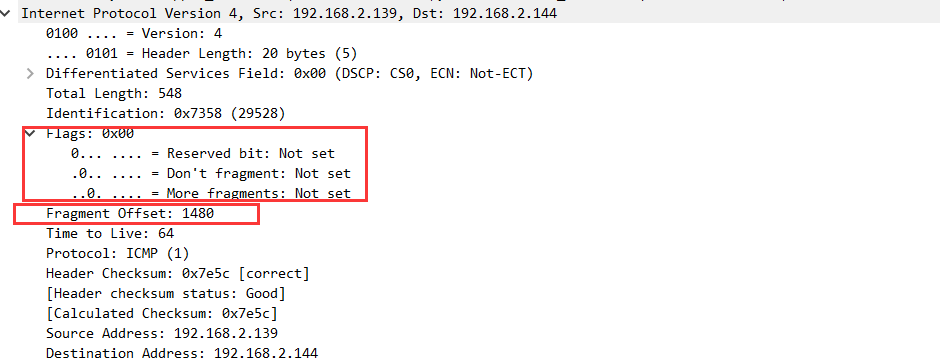
主机B ping主机A，Wireshark捕获IP分片，如下图所示：



选取第一个数据包，分析第一个分片：

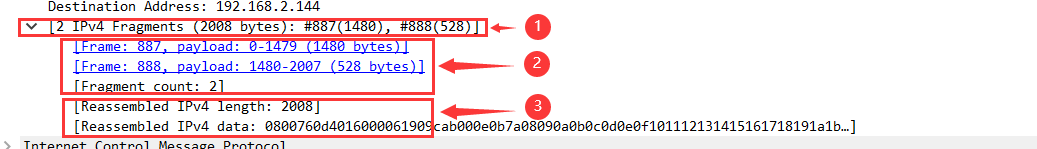


分析该数据包的第二个分片：



注意到首部的Flag字段与第一个分片不同，MF位为0表示没有更多分片，该分片已是最后一个分片。

Fragment Offset字段指出了该分片的偏移量，表示距离原始报文开头偏移了1480字节（理论上来说偏移量应该以8字节为单位，但结合实际发送数据的大小来看此处显示的值是以1字节为单位的），前1480字节已在第一个分片中。



|  |  |
| --- | --- |
| ① | 表示共2个IPv4分片，数据段共2008 字节，第一个分片数据段1480字节，第二个分片数据段528字节。此外，数据段总长度多出8字节是因为包含了8字节长的ICMP首部。 |
| ② | 分片详情 |
| ③ | 重组信息，分别表示重组后的IPv4报文长度、重组后的IPv4的数据段内容 |

综上，IP数据包的每一个分片都有大致相同的IP首部，对于同一个报文的不同分片，首部中的标识符字段相同；而标志字段的MF位则根据当前分片的在报文中的顺序而不同，对于最后一个分片，MF位为0，其他分片则为1。分片偏移字段负责指出当前分片在报文中的具体位置。需要注意的是，进行分片后，总长度字段中的值是当前分片的总长度，而非报文的总长度。

此外，对于最后一个分片，其分片偏移量（经过换算）再加上该分片的总长度即可得出报文的总长度（包含首部）。图中总长度548+分片偏移量1480=2028=2000（原始数据字节大小）+8（ICMP报文首部）+20（IP报文首部）。接收方可以据此判断出分片是否全部收齐。

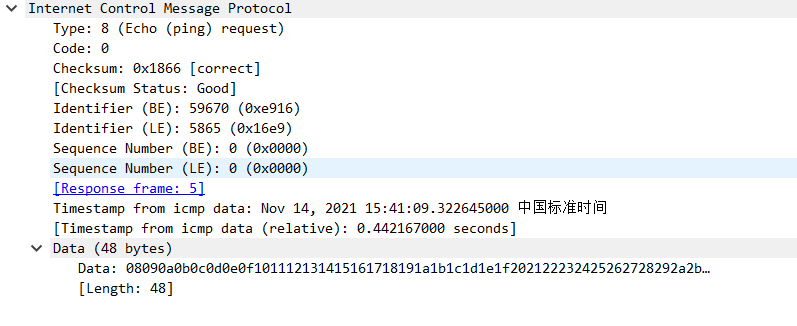
分片收齐后，即可按照首部相关字段的值按序进行重组，还原出IP报文数据段的内容。

### ICMP协议分析

通过ping和tracet命令，了解ICMP协议的使用。

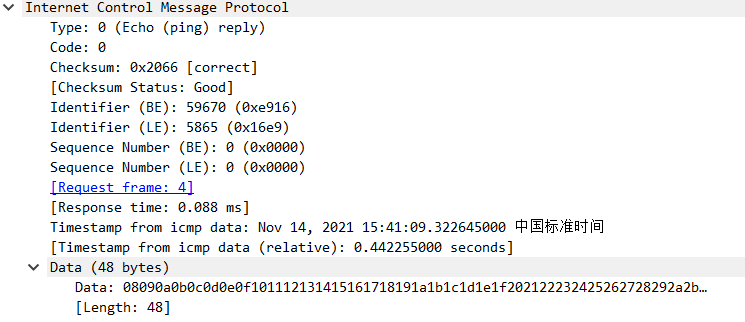
* 从主机A上向主机B发PING检测报文，捕获ICMP请求数据包和应答数据包，记录并分析各字段的含义，并与ICMP数据包格式进行比较；如果返回的差错信息，请分析是由于什么差错引起的。

为了方便分析，主机B ping主机A，Wireshark捕获ICMP请求数据包如下：



|  |  |
| --- | --- |
| Type | 8位，ICMP报文类型。8表示询问报文中的Echo请求。 |
| Code | 8位，代码字段，进一步区分某种类型中的几种不同情况。注意要结合Type来查找对应的详细信息。图中类型8+代号0表示Echo请求（类型8也这一种情况）。 |
| Checksum | 16位，检验和，用于检验整个ICMP报文。此处检验和比对结果为[correct] |
| Identifier(BE) | 标识符，用于区分不同的Ping进程，确定该报文的发送者。BE和LE是Wireshark考虑到发送ping报文时，Windows系统下为little-endian byte order，而Linux系统下为 big-endian format，为容易理解特别注明的。 |
| Identifier(LE) |
| Sequence Number(BE) | 序列号，关联对应的请求报文和应答报文（请求报文和其对应的响应报文的序列号相同)） |
| Sequence Number(LE) |
| Timestamp form icmp data | 时间戳 |

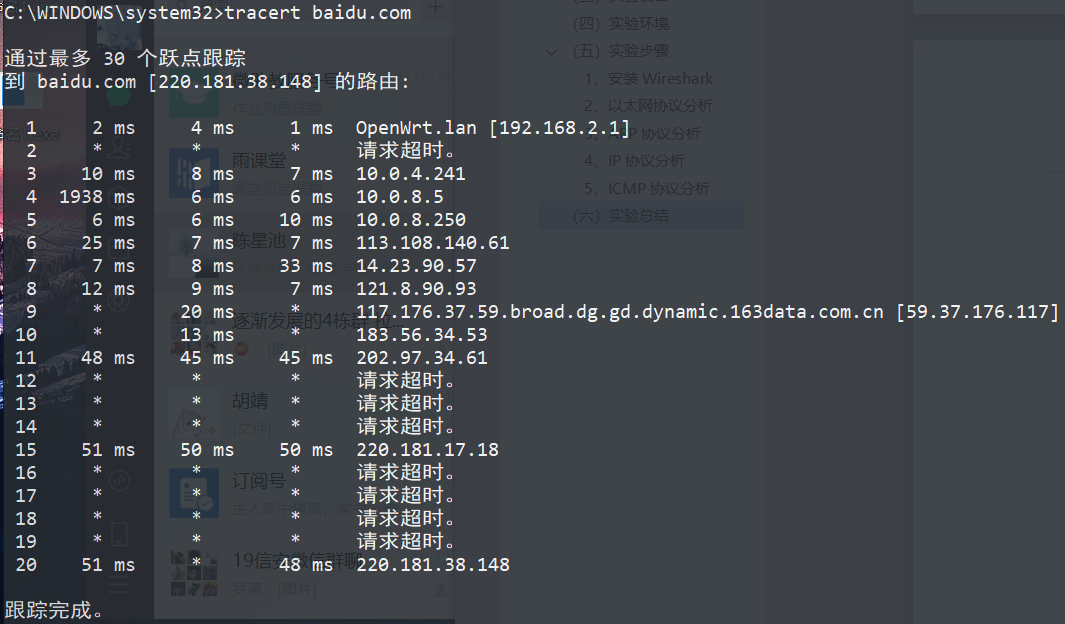
捕获ICMP响应数据包如下：



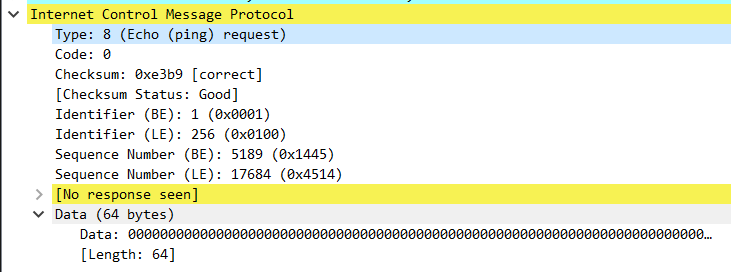
各字段含义与请求数据包相同，内容上也大致相同，仅需注意Type字段以0表示该报文为echo响应报文。

捕获ICMP差错报文如下（令主机B处于休眠状态后，主机A ping 主机B）：

* 使用tracert命令，跟踪某台主机，使用wireshark捕获数据包，分析不同类型ICMP响应数据包格式（如type=8,type=0,type=11）。分析tracert工作原理。

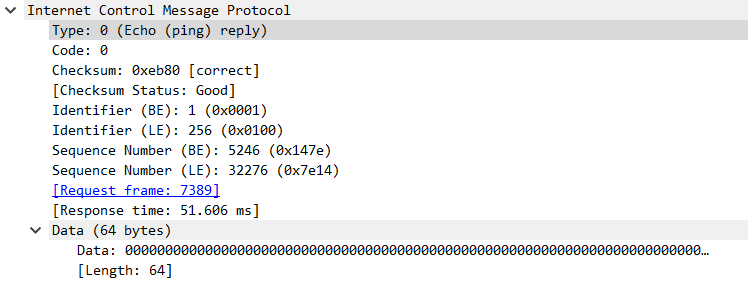


捕获到type=8的ICMP请求数据包：



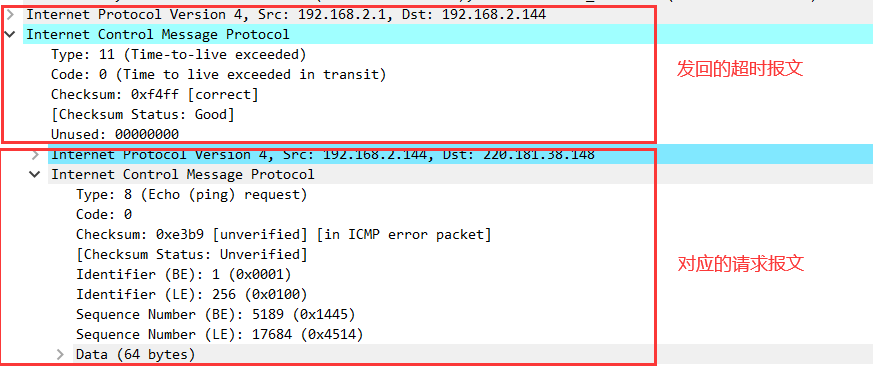
此时Type为8，Code为0，表示Echo请求。

捕获到type=0的ICMP响应数据包：



此时Type为0，Code为0，表示Echo响应。

捕获到type=11的ICMP响应数据包：



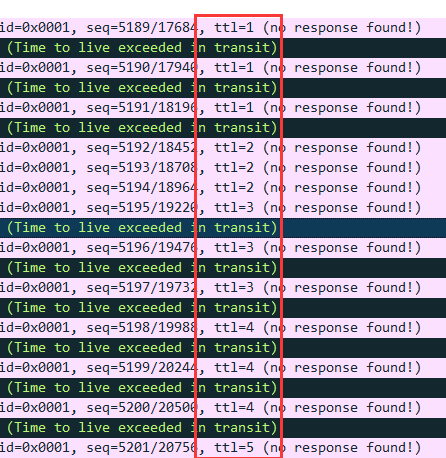
Type为11，Code为0，表示TTL超时。

源地址为192.168.2.1（网关），目的地址为主机A的IP地址，即网关产生并发送给源地址的ICMP超时报文，通知源地址有数据报因TTL递减至0而被丢弃。

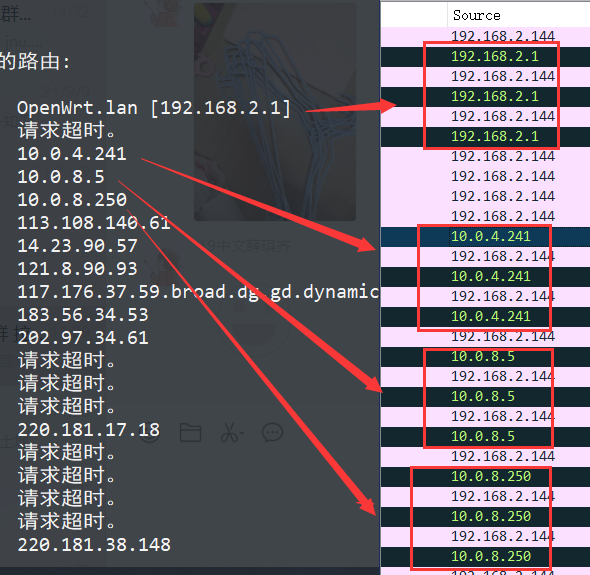
下方另一个ICMP报文即先前主机A发送的ICMP请求报文。

tracert工作原理：

首先观察到tracert命令会发送一系列TTL递增的ICMP请求报文：



并且注意到tracert命令所跟踪到的一系列路由IP地址的出现顺序与ICMP超时报文发送方的出现顺序是相同的：



这说明tracert命令实际上是通过发送一系列TTL递增的ICMP请求报文，利用路由丢弃报文时会向源地址发送超时报文的机制，间接获取路径上的路由IP地址，从而实现跟踪的效果。

### 【思考题】

1. 在ARP包分析实验过程中，为什么A有时能捕获ARP报文，有时却不能捕获ARP报文？

经过多次测试，发现不能捕获到ARP报文的情形出现在两次时间间隔较短的Ping命令之间。这是因为在之前的通信过程中主机A将主机B的匹配MAC地址添加到本地ARP缓存，在缓存未过期之前，在后续通信中将直接使用缓存中的MAC地址，而不需要再发送ARP请求，此时不能捕获到ARP报文。因此主机A有时能捕获ARP报文，有时不能捕获ARP报文。

1. 为什么运行ping 127.0.0.1时，不能捕获到ICMP报文？如果运行ping 本机IP地址能收到报文吗？为什么？

Wireshark捕获数据包的基本原理是捕获本机网卡进出的流量。Ping 127.0.0.1时，由于该IP为本地环回地址，数据直接返回到本机，不经过网卡，因此不能捕获到ICMP报文。

Ping 本机IP地址能收到报文，发出去的请求报文会通过环路返回本机。但同样不能捕获到报文，因为同样未经过本机网卡。

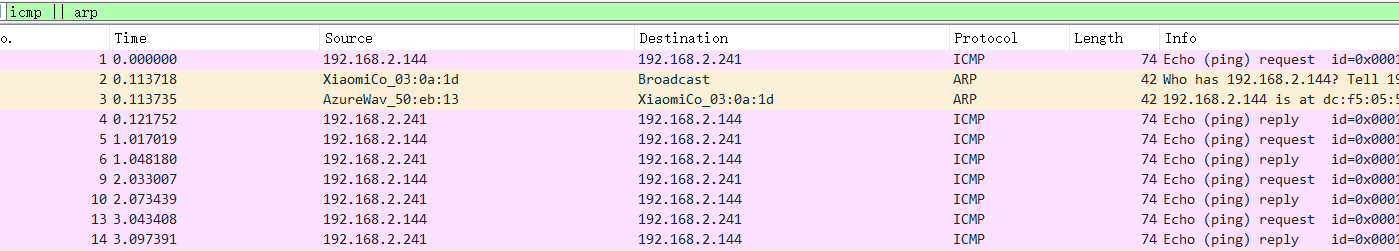
1. 在ping 的过程中，返回信息“Request timed out” 和“Destination Host Unreachable”分别是由哪些情况引起的？

Ping命令的实现依靠ICMP报文。当返回信息Request timed out时，表明ICMP请求报文未得到响应，进一步的原因可能是目的主机防火墙拦截或目的主机已下线。但此时存在通往目的主机的路由，只是因为其他原因未得到响应。

Destination Host Unreachable表明找不到通往目的主机的路由，一般是收到了表示目标不可达的ICMP差错报文，进一步的原因可能有目的主机不存在、物理连接失效、协议项有误等。

4、请通过实验**验证**：

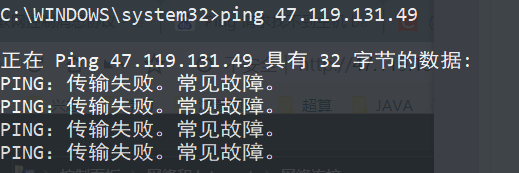
* 主机如果不设置“网关”，同一网段内的主机可以相互通信。用ping命令测试，用嗅探器测试可以捕获8个ICMP数据包，2个ARP数据包。



嗅探器捕获结果：8个ICMP数据包，2个ARP数据包。

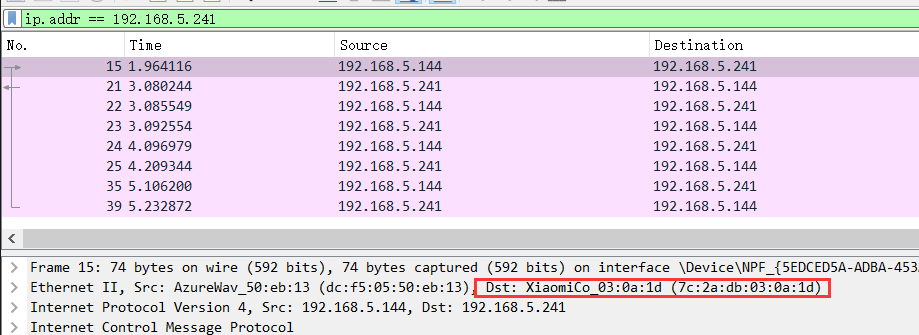
* 不同网段的主机不能通信，用PING命令测试，会显示“ Destination Host Unreachable”，因为没有指明网关，无法发送出去，因此显示“目的主机不可达”，用嗅探器捕获不到任何信息。

主机A ping 个人云服务器47.119.131.39，如下图所示：

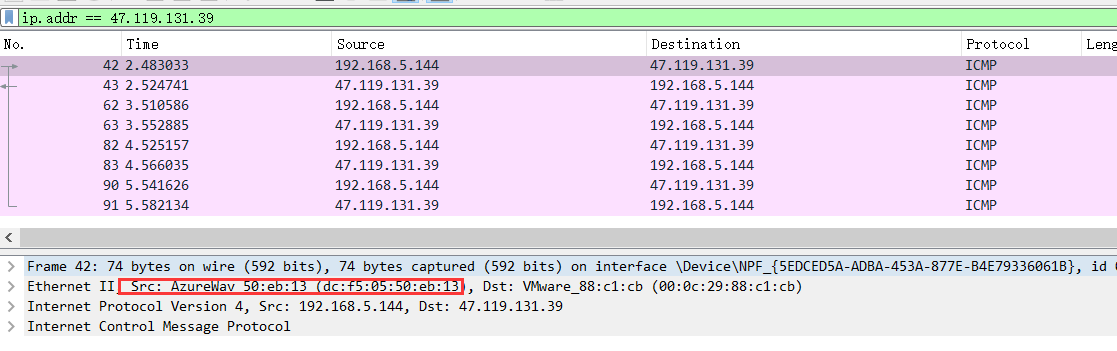


* 主机如果设置“网关”，同一网段的主机通信不通过网关转发，用ping命令测试，用嗅探器可以捕获所有测试数据包，能看到对方主机的MAC地址。不同网段的主机之间通信需要网关转发，用ping命令测试，能看到网关的MAC地址（包括能通信或不能通信）。

设置网关，主机A与主机B处于同一网段，主机A Ping主机B，嗅探器捕获能看到主机B的MAC地址：



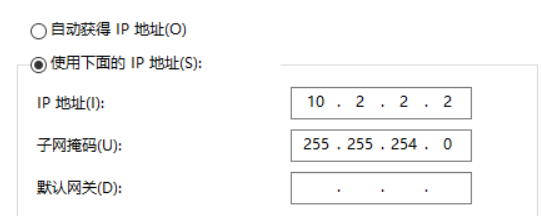
主机A Ping 个人云服务器47.119.131.39，嗅探器捕获能看到网关的MAC地址：



5、通过下面实验**理解网关**

* 假设主机A的IP地址为10.2.2.2/23，主机B的IP地址为10.2.3.3/23，两台主机均不设置网关，用ping命令测试两主机的连通性，用ARP命令查看物理地址。对结果进行分析。

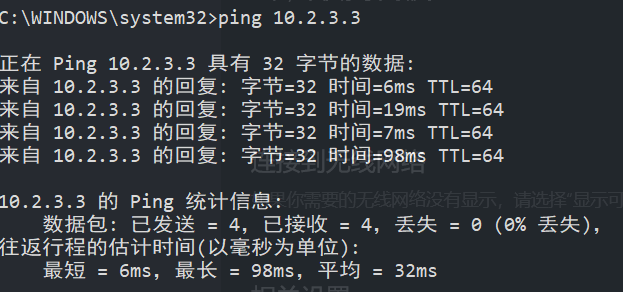
主机A配置如图所示：



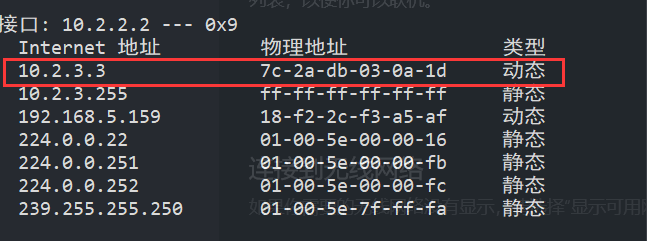
主机B配置如图所示：



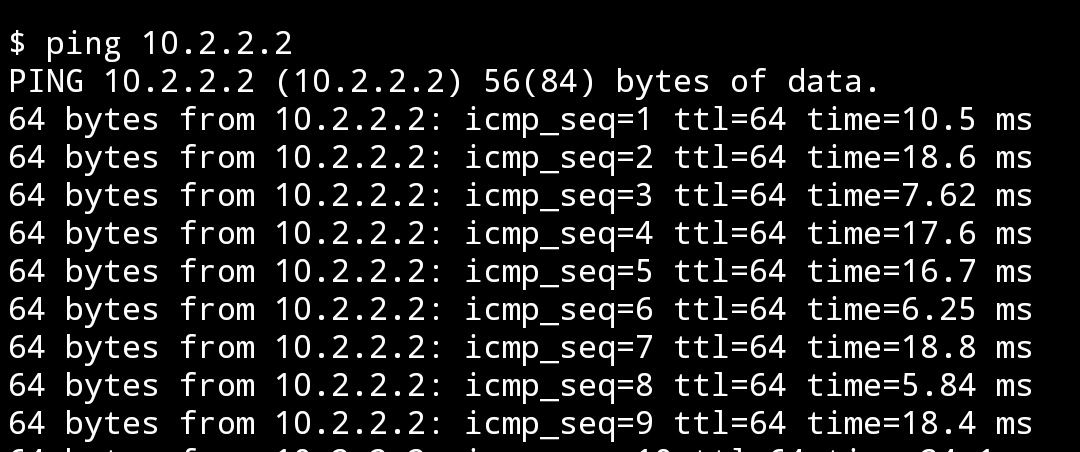
主机A ping 主机B，可以正常Ping通：



查看物理地址，10.2.3.3对应物理地址与主机B的MAC地址相同：



主机B Ping主机A，可以正常Ping通：



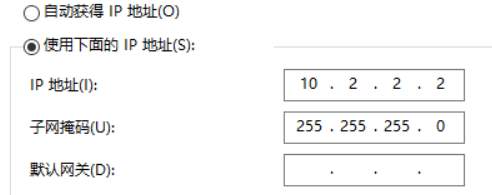
手机终端执行arp命令会提示无权限，所以未能查看arp表中物理地址是否能对应。

分析：10.2.2.2/23与10.2.3.3/23属于同一网段，相互通信不需要通过网关，因此即使不设置网关也能相互Ping通。

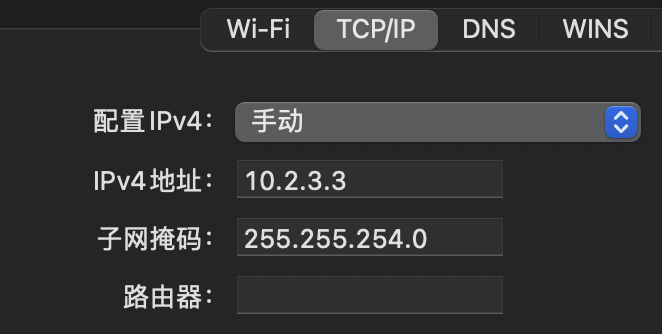
* 假设主机A的IP地址为10.2.2.2/24，主机B的IP地址为10.2.3.3/23，两主机不设置网关，分别在主机A和主机B上用ping测试与对方的连通性，用ARP查看物理地址。对测试结果进行分析。

针对上述情况，分别将主机的网关设置为本机地址，观察测试结果，并分析原因。

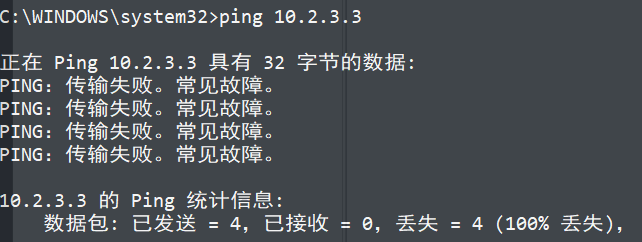
主机A设置：



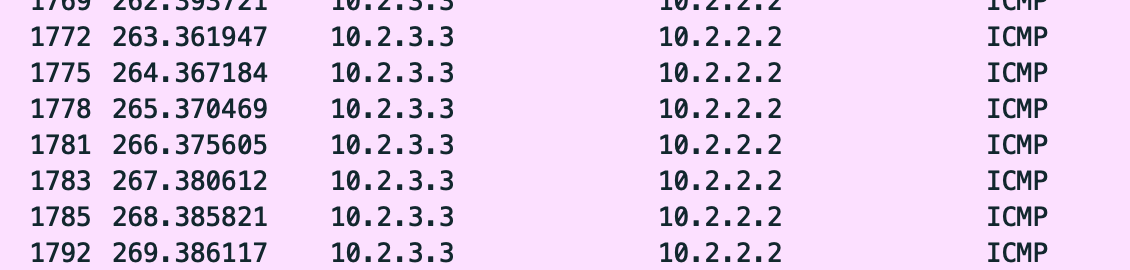
主机B设置：



主机A Ping主机B，不通，且捕捉不到ICMP请求报文：



主机B Ping主机A，显示请求超时，但能捕捉到ICMP请求报文：



在主机A（10.2.2.2），也能捕捉到来自主机B的ICMP请求报文，但不响应：

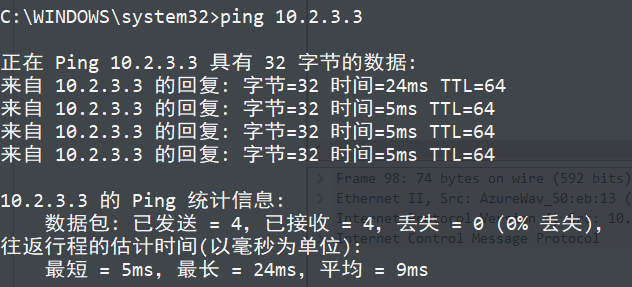


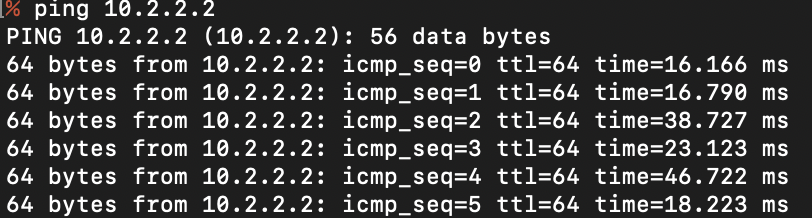
分析：

在无网关的情况下，虽然主机A和主机B相互之间都不能Ping通，但原因是不同的：

1. 主机A Ping不通主机B，其原因是主机A用主机B的IP 10.2.3.3与自己的子网掩码24进行与运算，发现主机B网络号为10.2.3.0，与自己所在子网10.2.2.0**不一致**，因此未发送ICMP请求报文，故嗅探不到ICMP请求报文，Ping命令从发送这一步骤就失败了。
2. 主机B Ping不通主机A，其原因是主机B用主机A的IP 10.2.2.2与自己的子网掩码23进行与运算，发现主机A网络号为10.2.2.0，与自己所在子网10.2.2.0**一致，**因此正常发送ICMP请求报文。对于主机A，虽然收到了ICMP请求报文，但因为前文所述的原因同样不发送响应报文，因此主机B执行的Ping命令是因为未收到ICMP响应而失败的，故提示的是“请求超时”。

将主机A与主机B的**网关分别设置成本机IP地址**，相互能正常Ping通：





## 实验总结

当信息被高层协议层层封装，到最后封装成以太网数据帧时，需要添加的信息仅有最直接的源MAC地址与目的MAC地址，标明数据的来路与去处，以及标明所封装的上层协议类型。于是就产生了一个问题：从高层协议里只能获知IP地址，如何找到对应MAC地址？于是就引入了ARP协议。ARP协议其实与DNS协议类似，都是维护一个映射表，此处是IP地址到MAC地址的映射。ARP协议获知MAC地址的方式也很简单：广播，然后等待IP地址对应的主机主动告知。

与以太网帧相比，IP所需要包含的信息比较多，但其中与以太网帧关系最密切的就是分片。为何要分片？因为以太网帧有MTU限制，而MTU值是传输效率与传输时间两相权衡下的折中结果，IP只能服从并且将数据划分成多个不超过MTU值的数据包发送。因为分片的存在，IP需要几个字段来保存相关信息：这个分片是属于哪个报文的？（Identification）这个分片是报文中的哪一段？（Fragment Offset）怎么知道一个报文的所有分片有没有收全？（Flags+Fragment Offset+Total Length）。

ICMP为IP补上了差错报告的功能：报文被丢弃了，通知发送方；报文到不了目的地，通知发送方……然后就利用这一机制产生了ping命令与tracert命令，作为检测网络的常用方法。

实验过程中碰到了本机收到了ICMP请求报文却不响应的问题，起初还以为实验室路由设置有误，最后才发现原来是Windows防火墙设置里没启用响应。以后网络实验遇到理论可行而实际不行的情况时，应该还是先检查一下防火墙。