山东大学 计算机 学院

计算机网络 课程实验报告

学号：202000130143	姓名： 郑凯饶	班级： 2020 级 1 班
实验题目：ICMP		
实验学时：2	实验日期： 2022-5-15	

实验目的：

探索 ICMP 协议：

【1】 Ping 程序生成的 ICMP 消息

【2】 Traceroute 程序生成的 ICMP 消息

【3】 ICMP 消息的格式与内容

硬件环境：

Dell Latitude 5411

Intel(R) Core(TM) i5-10400H CPU @ 2.60GHz(8GPUs),~2.6GHz

软件环境：

Windows 10 家庭中文版 64 位（10.0，版本 18363）

Wireshark-win64-3.6.2

实验步骤与内容：

1. 问题：

（1） 本机 IP 以及目标 IP。

（2） 为什么 ICMP 数据包没有端口信息？

（3） 请求 ICMP 的类型和代码。还有哪些其他字段？校验和，序列号和标识符字段所占字节数。

（4） 响应。

（5） 本机 IP 以及目标 IP。

（6） 如果 ICMP 发送了 UDP 数据包，那么探测数据包的 IP 协议号仍然是 01 吗？否则它是什么？

（7） 响应数据包和 ping 有何不同？

（8） 错误数据包比 ICMP 有更多字段，具体是？

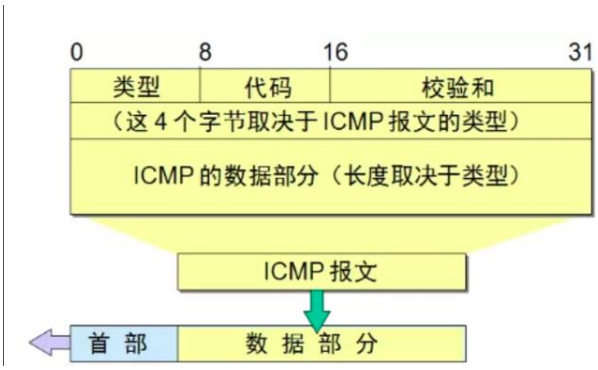
（9） 源主机最后收到的 3 个数据包和 ICMP 错误数据包有何不同？

（10）在 tracert 的跟踪中，是否有一个连接的延迟比其他连接长得多？这个连接末端的 2 个路由器的位置？

2. 阐述基本方法

ICMP 提供简明的出错报告信息，发送的错误报文返回到发送原数据的设备。从技术角度来说，ICMP 是一个“错误侦测与回报机制”，可以通过其检测网络的连线状况。

报文格式：

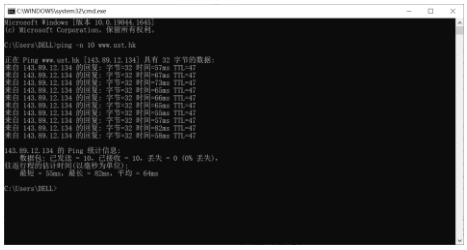


通过 type 和 code 标识消息类型：

TYPE	CODE	Description	Query	Error
0	0	Echo Reply——回显应答（Ping应答）	x	
3	0	Network Unreachable——网络不可达		x
3	1	Host Unreachable——主机不可达		x
3	2	Protocol Unreachable——协议不可达		x
3	3	Port Unreachable——端口不可达		x
3	4	Fragmentation needed but no frag. bit set——需要进行分片但设置不分片比特		x
3	5	Source routing failed——源站选路失败		x
3	6	Destination network unknown——目的网络未知		x
3	7	Destination host unknown——目的主机未知		x
3	8	Source host isolated (obsolete)——源主机被隔离（作废不用）		x
3	9	Destination network administratively prohibited——目的网络被强制禁止		x
3	10	Destination host administratively prohibited——目的主机被强制禁止		x
3	11	Network unreachable for TOS——由于服务类型TOS，网络不可达		x
3	12	Host unreachable for TOS——由于服务类型TOS，主机不可达		x

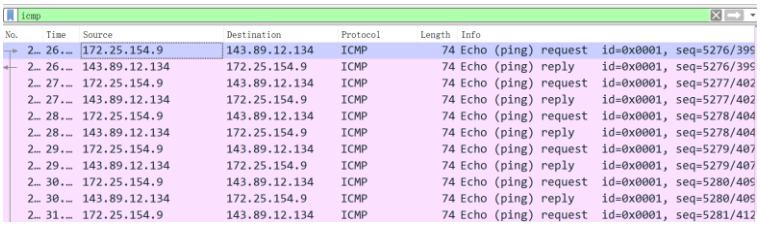
3. 实验结果展示与分析

Ping 操作：



（1） Source Address: 172.25.154.9

Destination Address: 143.89.12.134

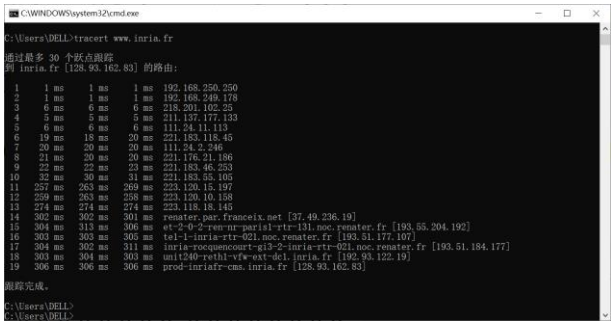


（2） ICMP 报文仅仅传送到指定主机，而不是指定程序，所以没有端口号概念。

（3） 类型 8，代码 0，还包括校验和、序列号、标识符各 2B

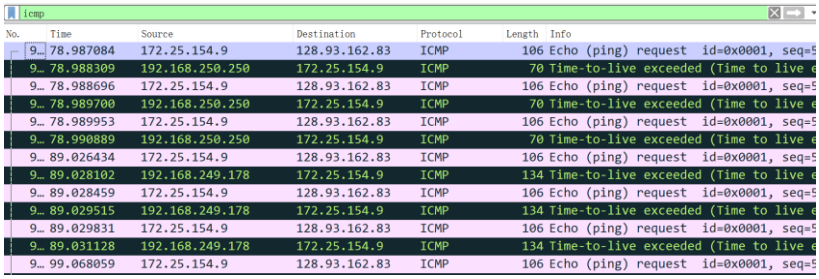
（4） 类型 0，代码 0，还包括校验和、序列号、标识符各 2B

Tracert 操作：



（5） Source Address: 172.25.154.9

Destination Address: 128.93.162.83



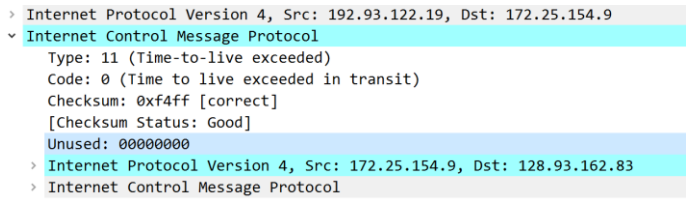
（6） 不是，应该是 UDP 的协议号为 17

（7） 具体字段内容不同

（8） Type: 0 (Echo (ping) reply) -> 11 (Time-to-live exceeded)

增加：unused 字段，包含请求的 IP 以及 ICMP 数据包的首部

减少：标识符 Identifier 以及序列号 Sequence Number 字段



（9） 最后 3 个接收到的数据包是目标主机的 reply，表示成功接收 request

（10）观察到第 11 跳相比前面的往返时延陡增，猜测是国际之间的路由连接。

结论分析与体会：

这次实验深入了 ICMP 协议，学习了之前略有眼缘的 ping 与 tracert 命令，明白了为什么我们可以用其来检查网络状况。有意思的是，我们可以用 tracert 探测出网络通路，探明消息如何从大洋彼岸发送到我们的电脑上。

例如 ping 一下 github.com:

可以“正常”访问，但是却 ping 不通，查询了可能是 DNS 解析的问题。

在 hosts 中加入：

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#   127.0.0.1       localhost
#   ::1             localhost
192.30.255.112  github.com git
185.31.16.184 github.global.ssl.fastly.net
```

好了，成功了！

```
C:\Users\DELL>ping github.com

正在 Ping github.com [192.30.255.112] 具有 32 字节的数据:
来自 192.30.255.112 的回复: 字节=32 时间=235ms TTL=46
来自 192.30.255.112 的回复: 字节=32 时间=237ms TTL=46
来自 192.30.255.112 的回复: 字节=32 时间=246ms TTL=46
来自 192.30.255.112 的回复: 字节=32 时间=252ms TTL=46
```

看看 github 服务器距离我有多少路由“跳”：

```
C:\Users\DELL>tracert github.com

通过最多 30 个跃点跟踪
到 github.com [192.30.255.112] 的路由:

  1     2 ms     2 ms     6 ms  192.168.250.250
  2     2 ms     1 ms     3 ms  192.168.249.178
  3     8 ms     6 ms     8 ms  218.201.102.25
  4     8 ms     7 ms     6 ms  211.137.177.133
  5    11 ms     9 ms     6 ms  111.24.11.113
  6    25 ms    29 ms    28 ms  221.183.109.133
  7    23 ms    24 ms    23 ms  221.183.89.41
  8    26 ms    36 ms    24 ms  221.183.89.70
  9     *       29 ms     *     221.183.89.181
 10   198 ms   199 ms   197 ms  223.120.12.177
 11   196 ms   195 ms   203 ms  223.120.6.218
 12   195 ms   198 ms   198 ms  las-b23-link.ip.twelve99.net [62.115.171.216]
 13   232 ms   224 ms   203 ms  sjo-b23-link.ip.twelve99.net [62.115.116.40]
 14   227 ms   248 ms   251 ms  sea-b2-link.ip.twelve99.net [62.115.118.168]
 15   234 ms   229 ms   247 ms  github-ic345014-sea-b2.ip.twelve99-cust.net [62.115.175.97]
 16     *        *        *     请求超时。
 17     *        *        *     请求超时。
 18     *        *        *     请求超时。
 19   260 ms     *      252 ms  github.com [192.30.255.112]

跟踪完成。
```

19 跳，对于计算机网络我们似乎可以给出一个确切结论：任意两台通讯设备的距离不超过 K 台路由！