

中山大学计算机学院

计算机网络

本科生实验报告

(2022 学年春季学期)

课程名称：计算机网络

教学班级	计科二班	专业（方向）	计算机科学与技术
学号	20337263	姓名	俞泽斌

1. Ping 部分

首先是初始的 ping 命令

```
C:\Users\Aholi^y>ping www.baidu.com

正在 Ping www.a.shifen.com [39.156.66.18] 具有 32 字节的数据:
来自 39.156.66.18 的回复: 字节=32 时间=57ms TTL=49
来自 39.156.66.18 的回复: 字节=32 时间=55ms TTL=49
来自 39.156.66.18 的回复: 字节=32 时间=56ms TTL=49
来自 39.156.66.18 的回复: 字节=32 时间=62ms TTL=49

39.156.66.18 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 55ms, 最长 = 62ms, 平均 = 57ms
```

```
C:\Users\Aholi^y>ping stanford.edu

正在 Ping stanford.edu [171.67.215.200] 具有 32 字节的数据:
来自 171.67.215.200 的回复: 字节=32 时间=300ms TTL=231
来自 171.67.215.200 的回复: 字节=32 时间=307ms TTL=231
来自 171.67.215.200 的回复: 字节=32 时间=326ms TTL=231
来自 171.67.215.200 的回复: 字节=32 时间=332ms TTL=231

171.67.215.200 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 300ms, 最长 = 332ms, 平均 = 316ms
```

Ping 命令默认向网址发送四个大小为 32 字节的数据包，然后看看能否接收，以及接收延迟，可以看到 Stanford.edu 的延迟更高，其他的丢包什么的都没有出现，之后我们来看一下 ping 命令的几个选项，用 ping -h 的命令可以查看

选项:

- t Ping 指定的主机，直到停止。
若要查看统计信息并继续操作，请键入 Ctrl+Break;
若要停止，请键入 Ctrl+C。
- a 将地址解析为主机名。
- n count 要发送的回显请求数。
- l size 发送缓冲区大小。
- f 在数据包中设置“不分段”标记(仅适用于 IPv4)。
- i TTL 生存时间。
- v TOS 服务类型(仅适用于 IPv4。该设置已被弃用，
对 IP 标头中的服务类型字段没有任何影响)。
- r count 记录计数跃点的路由(仅适用于 IPv4)。
- s count 计数跃点的时间戳(仅适用于 IPv4)。
- j host-list 与主机列表一起使用的松散源路由(仅适用于 IPv4)。
- k host-list 与主机列表一起使用的严格源路由(仅适用于 IPv4)。
- w timeout 等待每次回复的超时时间(毫秒)。
- R 同样使用路由标头测试反向路由(仅适用于 IPv6)。
根据 RFC 5095，已弃用此路由标头。
如果使用此标头，某些系统可能丢弃回显请求。
- S srcaddr 要使用的源地址。
- c compartment 路由隔离舱标识符。
- p Ping Hyper-V 网络虚拟化提供程序地址。
- 4 强制使用 IPv4。
- 6 强制使用 IPv6。

```
C:\Users\AholiC^y>ping baidu.com -t
```

正在 Ping baidu.com [110.242.68.66] 具有 32 字节的数据:

来自 110.242.68.66 的回复: 字节=32 时间=58ms TTL=47
 来自 110.242.68.66 的回复: 字节=32 时间=59ms TTL=47
 来自 110.242.68.66 的回复: 字节=32 时间=62ms TTL=47
 来自 110.242.68.66 的回复: 字节=32 时间=63ms TTL=47
 来自 110.242.68.66 的回复: 字节=32 时间=65ms TTL=47
 来自 110.242.68.66 的回复: 字节=32 时间=90ms TTL=47
 来自 110.242.68.66 的回复: 字节=32 时间=59ms TTL=47
 来自 110.242.68.66 的回复: 字节=32 时间=60ms TTL=47
 来自 110.242.68.66 的回复: 字节=32 时间=60ms TTL=47
 来自 110.242.68.66 的回复: 字节=32 时间=60ms TTL=47
 来自 110.242.68.66 的回复: 字节=32 时间=60ms TTL=47
 来自 110.242.68.66 的回复: 字节=32 时间=60ms TTL=47
 来自 110.242.68.66 的回复: 字节=32 时间=62ms TTL=47

110.242.68.66 的 Ping 统计信息:
 数据包: 已发送 = 24, 已接收 = 24, 丢失 = 0 (0% 丢失),
 往返行程的估计时间(以毫秒为单位):
 最短 = 58ms, 最长 = 90ms, 平均 = 63ms

Control-C
 ^C

Ping -t 就是不断一直发数据包，一直等到 control +c 来强制结束并输出结果

```
C:\Users\Aholi^y>ping baidu.com -a
```

```
正在 Ping baidu.com [110.242.68.66] 具有 32 字节的数据:  
来自 110.242.68.66 的回复: 字节=32 时间=87ms TTL=47  
来自 110.242.68.66 的回复: 字节=32 时间=70ms TTL=47  
来自 110.242.68.66 的回复: 字节=32 时间=60ms TTL=47  
来自 110.242.68.66 的回复: 字节=32 时间=66ms TTL=47
```

```
110.242.68.66 的 Ping 统计信息:  
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
往返行程的估计时间(以毫秒为单位):  
最短 = 60ms, 最长 = 87ms, 平均 = 70ms
```

```
C:\Users\Aholi^y>ping stanford.edu -a
```

```
正在 Ping stanford.edu [171.67.215.200] 具有 32 字节的数据:  
来自 171.67.215.200 的回复: 字节=32 时间=264ms TTL=231  
来自 171.67.215.200 的回复: 字节=32 时间=371ms TTL=231  
来自 171.67.215.200 的回复: 字节=32 时间=407ms TTL=231  
来自 171.67.215.200 的回复: 字节=32 时间=408ms TTL=231
```

```
171.67.215.200 的 Ping 统计信息:  
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
往返行程的估计时间(以毫秒为单位):  
最短 = 264ms, 最长 = 408ms, 平均 = 362ms
```

Ping -a 能够将地址解析成主机名,如上图的 110.242.68.66 和 171.67.215.200, 可以看到这两个网站的主机名称不同

```
C:\Users\Aholi^y>ping baidu.com -n 2 -l 64
```

```
正在 Ping baidu.com [110.242.68.66] 具有 64 字节的数据:  
来自 110.242.68.66 的回复: 字节=64 时间=195ms TTL=47  
来自 110.242.68.66 的回复: 字节=64 时间=178ms TTL=47
```

```
110.242.68.66 的 Ping 统计信息:  
数据包: 已发送 = 2, 已接收 = 2, 丢失 = 0 (0% 丢失),  
往返行程的估计时间(以毫秒为单位):  
最短 = 178ms, 最长 = 195ms, 平均 = 186ms
```

```
C:\Users\AholiC^y>ping stanford.edu -n 2 -l 64
```

```
正在 Ping stanford.edu [171.67.215.200] 具有 64 字节的数据:  
来自 171.67.215.200 的回复: 字节=64 时间=192ms TTL=231  
来自 171.67.215.200 的回复: 字节=64 时间=212ms TTL=231
```

```
171.67.215.200 的 Ping 统计信息:  
    数据包: 已发送 = 2, 已接收 = 2, 丢失 = 0 (0% 丢失),  
    往返行程的估计时间(以毫秒为单位):  
        最短 = 192ms, 最长 = 212ms, 平均 = 202ms
```

Ping -n -l , -n 后的数据表示后面发几个数据包, -l 后的数据表明发送数据包的大小, 这里主要的区别也就是主机名称和延迟的区别了, Stanford.edu 的延迟略高。

```
C:\Users\AholiC^y>ping baidu.com -i 1
```

```
正在 Ping baidu.com [39.156.66.10] 具有 32 字节的数据:  
请求超时。  
请求超时。
```

```
39.156.66.10 的 Ping 统计信息:  
    数据包: 已发送 = 2, 已接收 = 0, 丢失 = 2 (100% 丢失),  
Control-C  
^C
```

```
C:\Users\AholiC^y>ping stanford.edu -i 20
```

```
正在 Ping stanford.edu [171.67.215.200] 具有 32 字节的数据:  
请求超时。  
请求超时。
```

```
C:\Users\AholiC^y>ping baidu.com -i 21
```

```
正在 Ping baidu.com [39.156.66.10] 具有 32 字节的数据:  
来自 39.156.66.10 的回复: 字节=32 时间=66ms TTL=49  
来自 39.156.66.10 的回复: 字节=32 时间=59ms TTL=49  
来自 39.156.66.10 的回复: 字节=32 时间=75ms TTL=49  
来自 39.156.66.10 的回复: 字节=32 时间=322ms TTL=49
```

```
39.156.66.10 的 Ping 统计信息:  
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
    往返行程的估计时间(以毫秒为单位):  
        最短 = 59ms, 最长 = 322ms, 平均 = 130ms
```

```
C:\Users\AholiC^y>ping stanford.edu -i 21
```

```
正在 Ping stanford.edu [171.67.215.200] 具有 32 字节的数据:  
来自 171.67.215.200 的回复: 字节=32 时间=369ms TTL=231  
来自 171.67.215.200 的回复: 字节=32 时间=239ms TTL=231  
来自 184.105.177.238 的回复: TTL 传输中过期。  
来自 184.105.177.238 的回复: TTL 传输中过期。
```

Ping -I 后设置的主要是生存的时间也就是 TTL，当设置的 TTL 过小的时候就会出现请求超时的现象，当 TTL 足够的时候才能发出信息。设置为 21 的时候可以发现 Stanford.edu 部分包会出现 TTL 传输过期的情况，说明要求的 TTL 大小比较大

```
C:\Users\Aholi^y>ping baidu.com -r 5

正在 Ping baidu.com [39.156.66.10] 具有 32 字节的数据:
来自 39.156.66.10 的回复: 字节=32 时间=181ms TTL=48
    路由: 10.44.36.202 ->
           10.44.16.202 ->
           10.10.1.41 ->
           120.197.11.6 ->
           183.233.109.86
```

```
C:\Users\Aholi^y>ping stanford.edu -r 5

正在 Ping stanford.edu [171.67.215.200] 具有 32 字节的数据:
请求超时。
请求超时。
```

Ping -r 后的数可以展示计数跃点的路由地址，如上图，但是访问 Stanford.edu 的时候就请求超时。

```
C:\Users\Aholi^y>ping baidu.com -w 60

正在 Ping baidu.com [110.242.68.66] 具有 32 字节的数据:
来自 110.242.68.66 的回复: 字节=32 时间=53ms TTL=47
来自 110.242.68.66 的回复: 字节=32 时间=54ms TTL=47
来自 110.242.68.66 的回复: 字节=32 时间=54ms TTL=47
来自 110.242.68.66 的回复: 字节=32 时间=50ms TTL=47

110.242.68.66 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 50ms, 最长 = 54ms, 平均 = 52ms
```

Ping -w 后的表示等待每次回复的超时时间，超时时间大于需要时间时就可以达到

2. Tracert 部分

首先是基础的 tracert 网址的命令

```
C:\Users\AholiC^y>tracert baidu.com
```

通过最多 30 个跃点跟踪
到 baidu.com [39.156.66.10] 的路由:

1	*	*	*	请求超时。
2	18 ms	11 ms	10 ms	10.44.36.201
3	18 ms	13 ms	17 ms	10.44.16.201
4	39 ms	12 ms	12 ms	10.10.1.42
5	13 ms	16 ms	19 ms	120.236.174.129
6	17 ms	14 ms	15 ms	120.197.11.5
7	18 ms	13 ms	15 ms	183.233.109.85
8	*	18 ms	15 ms	221.183.39.181
9	45 ms	43 ms	45 ms	221.183.37.221
10	*	*	*	请求超时。
11	44 ms	47 ms	40 ms	111.13.188.38
12	61 ms	50 ms	63 ms	39.156.27.1
13	*	*	*	请求超时。
14	*	*	*	请求超时。
15	*	*	*	请求超时。
16	*	*	*	请求超时。
17	*	*	*	请求超时。
18	47 ms	44 ms	64 ms	39.156.66.10

跟踪完成。

```
C:\Users\AholiC^y>tracert stanford.edu
```

通过最多 30 个跃点跟踪
到 stanford.edu [171.67.215.200] 的路由:

1	*	*	*	请求超时。
2	12 ms	16 ms	15 ms	10.44.36.201
3	4 ms	2 ms	3 ms	10.44.16.201
4	14 ms	14 ms	23 ms	10.10.1.42
5	21 ms	13 ms	15 ms	120.236.174.129
6	17 ms	14 ms	18 ms	120.197.11.5
7	*	15 ms	15 ms	183.233.109.81
8	*	*	*	请求超时。
9	25 ms	28 ms	22 ms	111.24.14.73
10	22 ms	20 ms	22 ms	111.24.4.242
11	19 ms	15 ms	17 ms	221.176.18.110
12	44 ms	39 ms	9 ms	221.176.19.46
13	12 ms	17 ms	9 ms	221.183.55.57
14	328 ms	305 ms	203 ms	223.120.13.221
15	238 ms	301 ms	201 ms	223.120.6.70
16	256 ms	203 ms	200 ms	10ge7-10.core1.sjc2.he.net [216.218.132.101]
17	*	201 ms	*	port-channel7.core2.paol.he.net [184.104.198.254]
18	297 ms	200 ms	202 ms	stanford-university.100gigabitethernet5-1.core1.paol.he.net [184.105.177.238]
19	222 ms	302 ms	203 ms	woa-west-rtr-v12.SUNet [171.64.255.132]
20	*	*	*	请求超时。
21	277 ms	304 ms	203 ms	web.stanford.edu [171.67.215.200]

跟踪完成。

可以看到 tracert 命令主要就是通过不断发包来找到最近的中继点, 然后不断向下进行, 直到最后的点为目标点, 区别上来说可能只是后面 Stanford.edu 追踪过程中延迟更高点

我们来看一下基本用法


```
C:\Users\Aholi^y>tracert
```

用法: tracert [-d] [-h maximum hops] [-j host-list] [-w timeout]
[-R] [-S srcaddr] [-4] [-6] target_name

选项:

-d	不将地址解析成主机名。
-h maximum_hops	搜索目标的最大跃点数。
-j host-list	与主机列表一起的松散源路由(仅适用于 IPv4)。
-w timeout	等待每个回复的超时时间(以毫秒为单位)。
-R	跟踪往返行程路径(仅适用于 IPv6)。
-S srcaddr	要使用的源地址(仅适用于 IPv6)。
-4	强制使用 IPv4。
-6	强制使用 IPv6。

```
C:\Users\Aholi^y>tracert -d baidu.com
```

通过最多 30 个跃点跟踪
到 baidu.com [39.156.66.10] 的路由:

1	*	*	*	请求超时。
2	10 ms	13 ms	10 ms	10.44.36.201
3	16 ms	13 ms	12 ms	10.44.16.201
4	16 ms	18 ms	11 ms	10.10.1.42
5	15 ms	13 ms	13 ms	120.236.174.129
6	14 ms	12 ms	13 ms	120.197.11.5
7	16 ms	15 ms	16 ms	183.233.109.85
8	*	*	9 ms	221.183.39.181
9	53 ms	55 ms	53 ms	221.183.37.221
10	*	*	*	请求超时。
11	47 ms	42 ms	43 ms	111.13.188.38
12	51 ms	50 ms	51 ms	39.156.27.1
13	*	*	*	请求超时。
14	*	*	*	请求超时。
15	*	*	*	请求超时。
16	*	*	*	请求超时。
17	*	*	*	请求超时。
18	47 ms	43 ms	46 ms	39.156.66.10

跟踪完成。

Tracert -d 就是不将地址解析成主机名, 基本与上面相同。

```
C:\Users\Aholi^y>tracert -h 5 baidu.com
```

通过最多 5 个跃点跟踪
到 baidu.com [110.242.68.66] 的路由:

1	*	*	*	请求超时。
2	26 ms	11 ms	11 ms	10.44.36.201
3	13 ms	11 ms	11 ms	10.44.16.201
4	15 ms	11 ms	11 ms	10.10.1.42
5	5 ms	2 ms	5 ms	120.236.174.129

跟踪完成。

Tracert -h 后的数字表示最大的跃点数, 这里设置为 5, 可以看到 5 次之后就不再跳转, 跟踪结束

```
C:\Users\AholiC`y>tracert -w 20 baidu.com
```

通过最多 30 个跃点跟踪
到 baidu.com [110.242.68.66] 的路由:

1	*	*	*	请求超时。
2	12 ms	10 ms	14 ms	10.44.36.201
3	13 ms	10 ms	11 ms	10.44.16.201
4	14 ms	11 ms	11 ms	10.10.1.42
5	13 ms	17 ms	13 ms	120.236.174.129
6	15 ms	12 ms	11 ms	120.197.11.5
7	20 ms	19 ms	13 ms	183.233.109.85
8	*	*	*	请求超时。
9	*	*	*	请求超时。
10	*	53 ms	53 ms	221.183.37.221
11	*	*	*	请求超时。
12	*	*	*	请求超时。
13	*	*	54 ms	219.158.109.185
14	*	*	*	请求超时。
15	57 ms	*	*	110.242.66.186
16	62 ms	*	57 ms	221.194.45.154
17	*	*	*	请求超时。
18	*	*	*	请求超时。
19	*	*	*	请求超时。
20	*	*	*	请求超时。
21	59 ms	58 ms	57 ms	110.242.68.66

跟踪完成。

```
C:\Users\AholiC`y>tracert -w 20 stanford.edu
```

通过最多 30 个跃点跟踪
到 stanford.edu [171.67.215.200] 的路由:

1	1 ms	2 ms	2 ms	172.26.127.254
2	2 ms	1 ms	1 ms	10.44.37.201
3	2 ms	3 ms	2 ms	10.44.16.201
4	3 ms	3 ms	2 ms	10.10.1.42
5	3 ms	3 ms	2 ms	120.236.174.129
6	3 ms	3 ms	2 ms	120.197.11.5
7	6 ms	4 ms	4 ms	183.233.109.85
8	*	*	*	请求超时。
9	12 ms	6 ms	7 ms	111.24.14.81
10	7 ms	6 ms	5 ms	111.24.5.190
11	8 ms	8 ms	7 ms	221.176.22.158
12	8 ms	8 ms	12 ms	221.176.19.38
13	14 ms	14 ms	26 ms	221.183.55.53
14	167 ms	168 ms	*	223.120.13.221
15	171 ms	*	171 ms	223.120.6.70
16	*	165 ms	*	10ge7-10.core1.sjc2.he.net [216.218.132.101]
17	*	*	*	请求超时。
18	164 ms	253 ms	*	100ge11-2.core1.nyc4.he.net [72.52.92.113]
19	*	175 ms	176 ms	woa-west-rtr-v12.SUNet [171.64.255.132]
20	*	283 ms	*	port-channel7.core2.paol.he.net [184.104.198.254]
21	*	252 ms	173 ms	web.stanford.edu [171.67.215.200]

跟踪完成。

Tracert -w 后的是等待每个回复的超时时间。

以上就是关于 ping 以及 tracert 命令的具体用法和实现过程，本次实验结束。