ICMP

Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) is a system by which hosts and routers can send status and error messages to each other. There are many different ICMP message types. Here we will cover the most important types.

Type	Name	Use
0	Echo Reply	This is the reply sent to a ping.
3	Destination Unreachable	This can happen in many different ways. A router may send this if you try to access a nonexistent host on its network or if you try to access a port on a host that is not listening for traffic. Another way is if a packet needs to be fragmented but the DF flag is set.
5	Redirect Message	This allows someone to inform a router of a more direct route to a destination.
8	Echo Request	This is a ping, used to send a quick message to another machine to see if it is listening.
11	Time Exceeded	This is sent if the TTL of a packet reaches 0.

Attackers often use the ping and traceroute tools to map out a network. Those tools use ICMP to do what they do, so network administrators will often configure their routers to not send ICMP messages. ICMP has also sometimes been used in denial of service attacks. The drawback of blocking ICMP is that it makes it harder to do legitimate networking things, such as path MTU discovery or diagnosing network connectivity problems.

Note: There is also ICMPv6, which is the version of ICMP that goes along with IPv6.

Ping

Ping is a simple and extremely useful networking tool. Its most common use is to see if a host is reachable. Here is the output of running a simple ping in Windows PowerShell.

```
Pinging google.com [172.217.13.78] with 32 bytes of data:
Reply from 172.217.13.78: bytes=32 time=26ms TTL=112
Reply from 172.217.13.78: bytes=32 time=26ms TTL=112
Reply from 172.217.13.78: bytes=32 time=26ms TTL=112
Reply from 172.217.13.78: bytes=32 time=28ms TTL=112

Ping statistics for 172.217.13.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 26ms, Maximum = 28ms, Average = 26ms
```

Ping sends a small amount of data in an ICMP message to the destination. If the destination is reachable and is configured to respond to pings, then it sends back a reply. On Windows, ping automatically sends four pings. On Mac and Linux, the pings are generated continuously until you tell it to stop (via Ctrl+C). Ping also provides timing information, which is useful for diagnosing networking problems. Here are a few of the most useful options.

Description	Windows	Mac	Linux
number of pings	`-n`	`-c`	`-c`
size of ping	`-l`	`-s`	`-s`
TTL	`-i`	`-T`	`-t`
do not fragment	`-f`	`-D`	`-m do`

For instance, on Windows, to send two 500-byte pings with TTL 255, we could use the following:

```
ping google.com -n 2 -l 500 -i 255~.
```

Ping is one of the first things I try if I'm having network connectivity problems. I'll try to ping Google and my router to see which ones, if any, I'm able to reach. Note that some network administrators disable their machines from responding to pings. This is to prevent people from discovering those machines.

Traceroute

Traceroute is a useful networking tool for seeing the path a packet takes from source to destination. Here is an example traceroute I ran from my house to Google.

```
 1     1 ms     2 ms    <1 ms  192.168.1.1
 2    15 ms    13 ms    12 ms  96.120.9.9
 3    38 ms    16 ms    27 ms  162.151.69.213
 4    24 ms    24 ms    14 ms  96.108.5.201
 5    21 ms    22 ms    23 ms  be-34-ar01.mckeesport.pa.pitt.comcast.net [69.139.168.141]
 6    26 ms    23 ms    25 ms  be-31641-cs04.pittsburgh.pa.ibone.comcast.net [96.110.42.173]
 7    24 ms    22 ms    24 ms  be-1411-cr11.pittsburgh.pa.ibone.comcast.net [96.110.38.142]
 8    42 ms    28 ms    29 ms  be-302-cr12.ashburn.va.ibone.comcast.net [96.110.32.101]
 9    29 ms    28 ms    40 ms  be-1112-cs01.ashburn.va.ibone.comcast.net [96.110.32.201]
10    28 ms    36 ms    28 ms  be-2111-pe11.ashburn.va.ibone.comcast.net [96.110.32.122]
11    34 ms    30 ms    27 ms  50.248.119.106
12    27 ms    27 ms    27 ms  108.170.229.246
13    30 ms    31 ms    32 ms  209.85.251.83
14    26 ms    26 ms    27 ms  iad23s63-in-f14.1e100.net [172.217.15.78]
```

We see that the first hop is my home router at 192.168.1.1. Then second hop is 96.120.9.9. A little while later we see some addresses that are part of the internet backbone, which is sort of like the internet equivalent of a major interstate highway. After 14 hops, the packet reaches its destination. The names you see for some of the hops are gotten by a reverse DNS lookup (using the PTR resource record). Not all the routers on the network have names. This name lookup is the slowest part of traceroute. See a little later for how to disable it and just see the IP addresses.

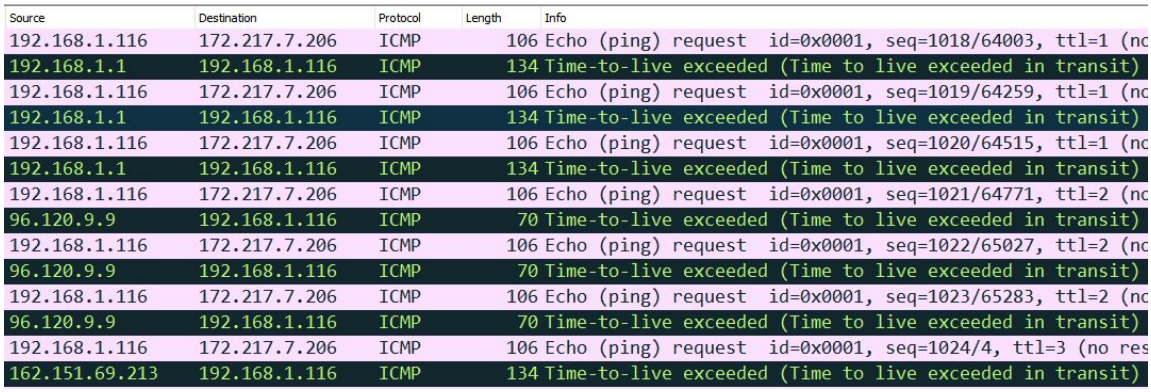
Sometimes, when you do a traceroute, you may see a hop that gives no information. It will look like `* * * Request timed out`. That is often because the router at which the packet's TTL hit 0 is configured for security reasons to not send back ICMP Time Exceeded messages.

How Traceroute works Traceroute works by making clever use of the TTL field in the IP header. The first three steps are shown below.

1. Send a packet with TTL=1. This packet will get to the router at the first hop, and the packet's TTL will drop to 0. The packet is dropped by the router, and the router sends an ICMP Time Exceeded message back to us. The IP address of the router will be visible in that message.
2. Send a packet with TTL=2. This packet will get to the router at the second hop before its TTL reaches 0. It's dropped there, and we'll get a Time Exceeded message from that router.

3. Send a packet with TTL=3. This packet gets to the router at the third hop before its TTL reaches 0. It's dropped, and the router at that third hop sends us a Time Exceeded message.

Traceroute continues this by sending packets with TTL 4, 5, 6, etc. Here is a Wireshark screenshot showing the first few steps of the process. Notice near the right the TTL increasing from 1 to 2 to 3. Traceroute by default repeats each step 3 times.



Source	Destination	Protocol	Length	Info
192.168.1.116	172.217.7.206	ICMP	106	Echo (ping) request id=0x0001, seq=1018/64003, ttl=1 (nc
192.168.1.1	192.168.1.116	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
192.168.1.116	172.217.7.206	ICMP	106	Echo (ping) request id=0x0001, seq=1019/64259, ttl=1 (nc
192.168.1.1	192.168.1.116	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
192.168.1.116	172.217.7.206	ICMP	106	Echo (ping) request id=0x0001, seq=1020/64515, ttl=1 (nc
192.168.1.1	192.168.1.116	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
192.168.1.116	172.217.7.206	ICMP	106	Echo (ping) request id=0x0001, seq=1021/64771, ttl=2 (nc
96.120.9.9	192.168.1.116	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
192.168.1.116	172.217.7.206	ICMP	106	Echo (ping) request id=0x0001, seq=1022/65027, ttl=2 (nc
96.120.9.9	192.168.1.116	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
192.168.1.116	172.217.7.206	ICMP	106	Echo (ping) request id=0x0001, seq=1023/65283, ttl=2 (nc
96.120.9.9	192.168.1.116	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
192.168.1.116	172.217.7.206	ICMP	106	Echo (ping) request id=0x0001, seq=1024/4, ttl=3 (no res
162.151.69.213	192.168.1.116	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)

We can do a manual traceroute using pings. Here is a short example using the Windows version of ping, with a few lines removed from the output for clarity. The "Reply from…" lines tell us the IP addresses of the various routers each packet reaches.

```
> ping google.com -n 1 -i 1
Pinging google.com [172.217.15.78] with 32 bytes of data:
Reply from 192.168.1.1: TTL expired in transit.

> ping google.com -n 1 -i 2
Pinging google.com [172.217.9.206] with 32 bytes of data:
Reply from 96.120.9.9: TTL expired in transit.

> ping google.com -n 1 -i 3
Pinging google.com [172.217.9.206] with 32 bytes of data:
Reply from 162.151.69.213: TTL expired in transit.
```

Using Traceroute On Mac and Linux, the tool is called `traceroute`, but on Windows it's called `tracert`. The Windows version sends out ICMP pings by default, while the Mac/Linux version uses UDP and has options to use ICMP or TCP. A useful option is `-d` on windows and `-n` on Mac/Linux, which tells traceroute to not try to find the domain names associated with the IP addresses. This considerably speeds up the result. There are also options that allow you to set a limit so that it never searches past a certain number of hops. On Mac and Linux (but not Windows), you can set the TTL value to skip some hops, to start right at hop 5, for instance. The Mac/Linux version has some other interesting options that aren't available on the Windows version.

Path MTU discovery

The *maximum transmission unit* (MTU) of a network is the largest packet size that can get through from a sender to a receiver. Often there are multiple types of network media that a packet will pass through on its way to its destination. It might first travel over Wi-Fi to a router, then over ethernet to another router, then maybe over a fiber optic line to another router, etc. Some media can handle larger packet sizes than others. The MTU is limited by whatever the smallest of these sizes is.

A sender can find the MTU of a path to a receiver by sending packets (usually pings) of varying sizes and seeing what gets through. The trick is to set the DF flag to true so that if a packet is too large, it will be dropped and the router dropping it will send back an ICMP error message. Note that for security reasons, some people set

their network devices to not respond to pings or send ICMP error messages, which makes path MTU discovery more difficult.

As an example of how path MTU discovery works, maybe the sender tries to send a 600-byte packet and it gets through. Then they try a 2000-byte packet and maybe it doesn't get through. So they know the path MTU must be somewhere between 600 and 2000 bytes. A reasonable next try would be the middle value of these, which is 1300 bytes. They continue this process to determine the exact value. A very common value is 1472 bytes. This is due to the 1500-byte ethernet MTU. Specifically, 1472 bytes of data plus 20 bytes of the IP header, plus 8 bytes of the ICMP header. Here is an example of the first two steps of the process using Windows, with a few lines removed from the output for clarity.

```
> ping google.com -n 1 -l 600 -f
Pinging google.com [172.217.9.206] with 600 bytes of data:
Reply from 172.217.9.206: bytes=68 (sent 600) time=6ms TTL=114

> ping google.com -n 1 -l 2000 -f
Pinging google.com [172.217.9.206] with 2000 bytes of data:
Packet needs to be fragmented but DF set.
```

A final note, there you can use `netsh interface ipv4 show subinterfaces` on Windows and `sudo ifconfig | grep MTU` on Mac/Linux to see the MTU of the various interfaces on your machine.