V.2.4 March · 2025

Practicals on Computer Networks and Distributed Systems

Host and LAN access tests (WIP)

Testing for IP connectivity

IP connectivity can be tested by using Internet's ICMP protocol. This protocol offers two basic echo messages: the request and the reply. Each is encapsulated as the payload to an ICMP PDU. Whenever an IP host wants to establish whether connectivity is available to a given IP host, an ICMP Echo Request message is sent to the other host, which is expected to respond to the received echo request with an ICMP Echo Reply Message. A generally available ping utility is included in the Linux installation that sends ICMP Echo Requests and expects to receive the corresponding ICMP Echo Reply. The ping utility will serve us to learn the basics about the ICMP protocol.

The Linux/UNIX ping command is used to see whether an Internet host has IP connectivity to another Internet host. The ping name stands for *Packet Internet Gopher* and is an old-time utility in Internet hosts; it functions by using the ICMP (Internet Control Management Protocol) protocol which is an essential IP control-plane protocol. Depending on the options included by the user, ping can send and receive different ICMP protocol messages. The simplest form of the ping command sends the ICMP ECHO message to the specified destination IP address and the destination (receiving) IP module[1] will react by sending back the same ECHO message. The sending ICMP can measure the Rtt (Round Trip Time) upon receipt of the ICMP ECHO back message. Below is an example of ping executed in OS-X: (If you want to stop the ping command, compose the key combination **ctrl-C**). See the following example executed at an OS-X machine:

```
$ ping www.telefonica.net
PING www.telefonica.net (213.4.130.95): 56 data bytes
64 bytes from 213.4.130.95: icmp_seq=0 ttl=120 time=43.752 ms
64 bytes from 213.4.130.95: icmp_seq=1 ttl=120 time=41.791 ms
64 bytes from 213.4.130.95: icmp_seq=2 ttl=120 time=43.660 ms
64 bytes from 213.4.130.95: icmp_seq=3 ttl=120 time=43.293 ms
64 bytes from 213.4.130.95: icmp_seq=4 ttl=120 time=43.070 ms
64 bytes from 213.4.130.95: icmp_seq=5 ttl=120 time=42.248 ms
64 bytes from 213.4.130.95: icmp_seq=6 ttl=120 time=43.548 ms
64 bytes from 213.4.130.95: icmp_seq=7 ttl=120 time=43.364 ms
64 bytes from 213.4.130.95: icmp_seq=8 ttl=120 time=43.014 ms
64 bytes from 213.4.130.95: icmp_seq=9 ttl=120 time=42.474 ms
64 bytes from 213.4.130.95: icmp_seq=10 ttl=120 time=43.837 ms
^C
--- www.telefonica.net ping statistics ---
11 packets transmitted, 11 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 41.791/43.096/43.837/0.635 ms
```

[1] Module refers to the software organization that implements the TCP/IP protocols in the Linux operating system, including ICMP

Depending on the specific platform, the output may differ but essentially ping tells whether there exists IP connectivity between two Internet hosts. As we will review, the Internet architecture has a single, central protocol in the network layer, its name is IP (Internetwork Protocol), there is no other network protocol in this architecture. From an implementation standpoint, the IP module belonging to the TCP/IP stack of a typical Linux/Unix/Window must contain other two modules in turn: the ARP and the ICMP. As we mentioned above, the ARP module maps internetwork addresses (IP addresses) to level-2 MAC addresses (The hardware address of an Ethernet NIC, for example), the ICMP protocol belongs to the control plane, that is, it is not involved in effective data transfers but cares for those transfers by coordinating the network equipment in ways that we will take up later. Together, the three protocol modules: IP, ARP and ICMP constitute what is normally known as the *IP module.* The technical specifications corresponding to the protocols that govern the Internet are called RFCs (Request For Comments) by the IETF (Internet Engineering Task Force), for instance, the RFC that documents the ICMP protocol is RFC 792. It's common that some RFCs affect other future RFCs, sometimes to the point of being superseded by them. At the beginning of an RFC, a list of affected RFCs appears alongside its past history. To finish this brief foray into the realm of Internet protocols we must recall that ICMP messages are encapsulated in IP datagrams; we will delve into these important concepts in chapter 1 and further.

Exercise 1. ICMP protocol messages

a. Search the Internet for "RFC 792". RFC documents are just an Internet search away and are available in several formats for improved viewing, internal searching, etc. Skim RFC 792 and search for the ICMP message **types** and **codes** that define the **ICMP Echo** and **ICMP Echo Reply** messages.

b. Can the response to an ICMP echo request be dropped at some router as it travels to its destination? In other words, can an ICMP packet be lost? Recall that an ICMP Echo Reply message is encapsulated into an IP packet. See the following ICMP protocol stack:

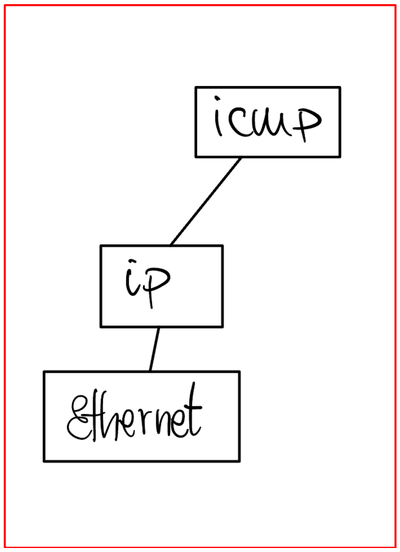

Figure 1. The protocol stack of the ICMP protocol

c. Now, more specifically, what should be the behavior of the host that sends an ICMP Echo to another and, the ICMP Reply sent back by the latter, is lost amid the Internet path that leads backward to the sender? Explain what you consider a *reasonable* behavior by the sender whenever it doesn't receive the corresponding ICMP Echo Reply. Tell what you would do should you program a utility like ping and an ICMP response is not received.

d. Observe the following example and try to reproduce it in your own computer. Use the IP address of a host which you are sure that is not powered up. For example, LabB6 host 192.168.1.49 shouldn't be powered up at the time you'll be doing this practice. Let the command run for about 10 seconds, then, stop it by simultaneously pressing the keys ctrl-C (This control combination sends SIGINT signal to the process which by default will cause the process to die). Explain what might have occurred that made the ping utility to report a 100% of responses missed:

```
administrator@debian-ule:~$ ping 192.168.1.49
PING 192.168.1.49 (192.168.1.49) 56(84) bytes of data.
From 192.168.1.89 icmp_seq=1 Destination Host Unreachable
From 192.168.1.89 icmp_seq=2 Destination Host Unreachable
From 192.168.1.89 icmp_seq=3 Destination Host Unreachable
From 192.168.1.89 icmp_seq=4 Destination Host Unreachable
From 192.168.1.89 icmp_seq=5 Destination Host Unreachable
From 192.168.1.89 icmp_seq=6 Destination Host Unreachable
^C
--- 192.168.1.49 ping statistics ---
8 packets transmitted, 0 received, +6 errors, 100% packet
loss, time 7168ms
```

e. Linux hosts, and virtually all mainstream operating systems can be set to not respond to ICMP Echo requests. In specific environments, this is useful, that is, from a cybersecurity standpoint. If a host won't respond to ICMP Echo, then, somehow it is keeping itself concealed, and there be less prone to be attacked. The procedure to have a **host not respond to ICMP echo requests** consists of **setting kernel parameter** to the right integer value (1) as is illustrated below. Use ping to test connectivity with a host that has disabled the echo responses and check whether other Internet services such as the Web keep functioning.

```
# List the kernel parameters that contain echo_ignore in their name
$ sysctl -a | grep echo_ignore
net.ipv4.icmp_echo_ignore_all = 0
net.ipv4.icmp_echo_ignore_broadcasts = 1

# su

# Disable ICMP echo responses at this host
# sysctl -w net.ipv4.icmp_echo_ignore_all=1

# Enable ICMP echo responses at this host
# sysctl -w net.ipv4.icmp_echo_ignore_all=0
```

Exercise 2. Sending icmp echo to your localhost and to your own IP public address

a. Find your system's current IP address with the ifconfig command, and send ping to it. Are you receiving echo responses? If that is true, observe the RTTs (Round-trip times) and check that those RTTs are substantially smaller than those received when pinging or www.princeton.edu or www.cisco.com, for instance. Give an explanation to those small RTTs.

b. In the preceding exercise you contacted your own IP in order to establish whether connectivity from your computer to itself were possible. Often a client application wants to connect with a server application that is running on the same system as the client. What would happen if effectively the network to which your system is connected were down and yet you needed to have client and server communicate, exchange protocol messages? Testing a client and a server within the same Internet host is possible even in the case your network is down, to that end, it is mandatory that the IP host have a special IP address (127.0.0.1, or host name localhost) configured to a *loopback network device*. This device plumbs the messages outgoing from the IP layer back onto itself, thereby making a physical network connection unnecessary. See the following example and execute the same test it in your system:

```
$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.066 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.139 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.181 ms
^C
--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.066/0.129/0.181/0.048 ms
```

```
$ ping localhost
PING localhost (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.050 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.179 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.179 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.179 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.174 ms
^C
--- localhost ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.050/0.152/0.179/0.051 ms
```

Exercise 3. Taking an IP traffic sample at your host with the tcpdump sniffer

```
# su
```

```
# tcpdump -i eno1 -n -vvv -eX icmp
```

Follow the explanation offered by the teacher on the board.

Exercise 4. Other options of the ping command

Explore other ping options (Execute $ man ping) and flags in your specific operating system –these options and flags are not the same across the sundry of mainstream operating systems of today. Tell us which ones seem useful.

Exercise 5. Check the guidelines for composing the practicals log book (LabBook) published in paloalto:

http://paloalto.unileon.es/LabBook-Guidelines.pdf

Carefully read the document, and henceforth, use it for composing the documentation to each practical (Disregard any mention to any submission dates included in that document. Submission dates for the LabBook will be published and advertised in the agora in the form of a Homework submission for late May/25).