CS 39006: Networks Lab

Assignment 8: Designing ping using raw sockets

Date: 22th March, 2018

Objective:

The objective of this assignment is to understand how raw sockets are different from traditional sockets and using them to design a custom ping application.

Submission Instructions:

The code needs to be written in either C or C++ programming language. You need to prepare a report that will contain the followings.

1. **Documentation** of the code along with **compilation and running procedure**, **sample input** and **sample output**
2. **Observations** from the experiments.
3. Intuitive **justification** behind the observations

Also include a **makefile** to compile your code. You need to submit the report, code (only C/C++ source) and Makefile in a single compressed (tar.gz) file. Rename the compressed file as `Assignment_1_Roll1_Roll2.tar.gz`, where `Roll1` and `Roll2` are the roll numbers of the two members in the group. Submit the compressed file through Moodle by the submission deadline. The submission deadline is: **April 5, 2018 02:00 PM.** Please note that this is a strict deadline and no extension will be granted.

Please note that your submission will be awarded zero marks without further consideration, if it is find to be copied. In such cases, all the submissions will be treated equally, without any discrimination to figure out who has copied from whom.

Introduction:

The ping command got its name from the sound sonar makes when it "sees" something. In sonar, you send out a signal and measure the time it takes to get there. The ping command does the same thing. This tells if a computer or device is out there or not, which is the purpose of the command. It is basically a network diagnostic tool that's used to check if a host in a network is alive and responding. Ping uses ICMP messages. More particularly ICMP query messages.

ICMP:

ICMP (Internet Control Message Protocol) is a companion to the IP protocol. It compensates the IP protocol in error reporting since IP protocol doesn't have an error reporting method in place. ICMP only reports errors and expects higher layers of the TCP/IP architecture model to handle and correct the errors.

ICMP has two types of messages error reporting messages and query messages. Query messages are generally used to diagnose network problems.
There are two types of query messages

1. Echo-request message
2. Echo-reply message

Generic composition of an ICMP 32byte packet:

- **IP Header**: protocol set to 1 (ICMP) and Type of Service set to 0.
- **ICMP Header**:
  - Type of ICMP message (8 bits)
  - Code (8 bits)
  - Checksum (16 bits), calculated with the ICMP part of the packet (the IP header is not used). It is the 16bit one's complement of the one's complement sum of the ICMP message starting with the Type field
  - Header Data (32 bits) field, which in this case (ICMP echo request and replies), will be composed of identifier (16 bits) and sequence number (16 bits).
- **ICMP Payload**: payload for the different kind of answers; can be an arbitrary length, left to implementation detail. However, the packet including IP and ICMP headers must be less than the maximum transmission unit of the network or risk being fragmented

The ping process:

1. The source sends an ICMP echo-request message to the destination.
2. The ping program sets a sequence identifier which gets incremented with each echo-request message. It also sets a TTL (Time-to-live) period.
3. Ping also inserts the sending time in the data section of the message.

4. If the host is alive and responding, it sends an ICMP echo-reply message back to the source.
5. Ping notes the time of the arrival of the response message, uses the sending time in the message part and calculates the Round-trip time
6. It then increments the sequence identifier (as said above) and sends a new echo-request message. This goes on for the number of ping requests set by the user or the program is terminated.

The whole of the data is calculated to summarize the percentage of packet loss and other such information and the summarized data is then displayed, showing the number of packets transmitted, received, percentage of packet loss, total time taken, the minimum, average and maximum round-trip time. This of course, is in addition to the data displayed live when the program is running.

```
Sample: > ping www.google.co.in
PING www.google.co.in (74.125.68.94) 56(84) bytes of data.
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=1 ttl=43 time=98.8 ms
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=2 ttl=43 time=174 ms
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=3 ttl=43 time=131 ms
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=4 ttl=43 time=98.5 ms
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=5 ttl=43 time=107 ms
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=6 ttl=43 time=101 ms
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=7 ttl=43 time=100 ms
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=8 ttl=43 time=108 ms
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=9 ttl=43 time=101 ms
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=10 ttl=43 time=105 ms
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=11 ttl=43 time=102 ms
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=12 ttl=43 time=122 ms
64 bytes from scinf94.1e100.net (74.125.68.94): icmp_seq=13 ttl=43 time=106 ms
  www.google.co.in ping statistics
 13 packets transmitted, 13 received, 0% packet loss, time 12015ms
rtt min/avg/max/mdev = 98.593/112.265/174.998/20.325 ms
```

Output Explained:

1. **`PING www.google.co.in (74.125.68.94)`**: Ping only knows how to communicate with IP addresses, so the first thing it did when asked to ping “google.co.in” was to look up the corresponding IP address. This is one of the quickest ways to determine the IP address associated with a domain. Also, if this lookup fails, we know there’s a typo in the domain name, or the domain name lookup (DNS) is failing for some reason.
2. **`64 bytes from scinf94.1e100.net (74.125.68.94)`**: This tells you that the remote server at that IP address replied. What that means, though, is that the entire route across the internet, from your machine through routers and switches and networking equipment and whatever else, worked, as did the return path carrying the server’s reply. If this fails, (“timed out”) then something along the connection between you and the server might be broken, the server might be offline, or the server might not even exist. It’s also possible the server is explicitly configured not to respond to ping requests.

3. `time=98.8 ms`: This is the round trip time: the time between sending “Are you there?” and receiving “Yes I am!”. In this case, it took 98.8 milliseconds. Since the ping is repeated several times, you can see that this time is fairly consistent, which is good. The time varies depending on many factors, including how close you are to the remote server, how many routers and other networking equipment are between you and that server, and more.
4. `13 packets transmitted, 13 received`: One of the things TCP/IP is designed to deal with is packet loss. Ideally, every packet you send should get to where it’s going, but for various reasons, that doesn’t always happen. As long as the packets can get there after a retry or two, in normal usage you’d never notice. Ping sends multiple packets and reports specifically on the success rate, so you can see if a particular connection is prone to packet loss.
5. `rtt min/avg/max/mdev`: While on average the same kind of packet sent to the same destination should take roughly the same amount of time, that’s also not always the case. Some packets take longer than others, for reasons as diverse as the equipment involved and paths followed. Ping reports these statistics so you can see if a particular connection is prone to this type of problem. The information provided here are, minimum round trip time (rtt), maximum rtt, average rtt and mean deviation.

Objective:

Implement a standard ping application which uses raw sockets. The output should be similar as shown in the sample. Also, the ping should be general in nature, that is, it should be able to ping any machine.