Exercises: Basics of Networking – II

Experiential Learning Workshop

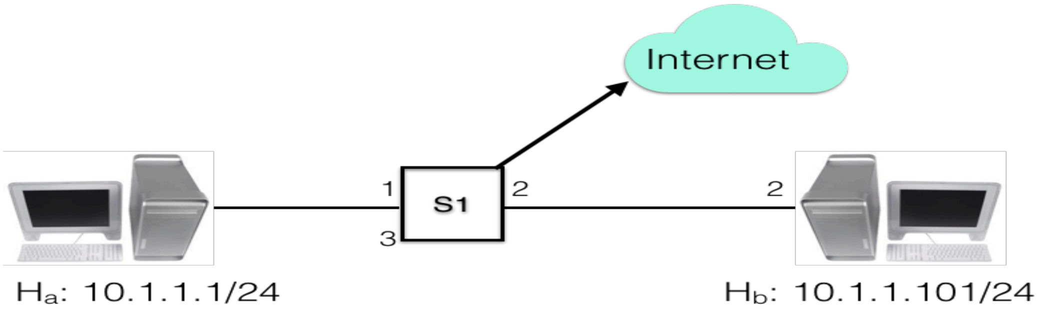
1 General Guidelines

1. Make a team of two or three unless stated otherwise.
2. For each exercise, use wireshark capture to verify contents
3. Ensure to use proper capture filter and don't capture irrelevant traffic
4. Where appropriate or applicable, use `wget` or `nc` to access the web server.
5. The default client for accessing web server is assumed to be browser, preferably firefox. You can use Chrome or any other browse as well.
6. The webserver in the example below is taken as 'myweb.com'. Please use your hostname or corresponding IP address instead in your exercise.
7. To kill any program in the linux terminal, please press **`Ctrl-C`** and not **`Ctrl-Z`**. The latter will suspend the program and not stop it.

Note: Appendix provides instructions on installing any package if not already installed.

2 Hands-on 1: Tools

The exercises below assume that client's IP address is 10.1.1.1 and other machine in your team has the IP address 10.1.1.101. Please use appropriate IP address in your setup.



2.1 Understand use of Wireshark to analyze network traffic

2.1.1 General usage and invocation.

1. Open wireshark (you might have to open in **sudo** mode)
2. Select the applicable interface e.g. `enp0s1` or `eth0`.
3. Specify the capture filter to capture traffic with other e.g. '`host 10.1.1.101`' if the IP address of other host is 10.1.1.101.
4. Click start
5. Access a web page from this website on the browser. For example, enter http://10.1.1.101.

2.1.2 Accessing websites

1. Browse your institute's website, e.g. `www.ieee.org` and capture traffic for this website. Use the appropriate capture filter (host http://www.ieee.org/). You should see traffic for this website only and no other traffic. Analyze the capture and look at how many packets are exchanged.

2. Access other websites e.g. vtu.ac.in or and other websites you prefer. Define appropriate capture filter for the same. Wireshark should show only relevant packets and not all kind of traffic.

2.1.3 Use other capture filters

1. Use a filter to exclude traffic from a web site e.g. host not 10.1.1.101 and analyze captured traffic.

2.1.4 Using other options

1. Use display filter to see traffic for a tcp stream.
2. Save some packets into a file and reopen the file
3. Explore various options of display time format.
4. Explore options of ordering packets by different fields e.g. by src address, by INFO field, by packet length etc.

2.1.5 Exercise Expectations

1. Launch wireshark and select the active interface.
2. Able to specify the appropriate capture filters and thus capture packets only of interest
3. Able to sort the packets w.r.t. different fields e.g. info, src, dst, time, etc.
4. Able to save packets to a file and open the file for later analysis.
5. Able specify display filter and see packets of interest.

2.2 Using ping

Note: Wherever count (-c option) is not specified, use Ctrl-C to abort.

1. Always Use wireshark to analyze all the traffic for below steps.
2. Ping `google.com` and `yahoo.com` by sending some fixed count packets e.g. 20. Analyze the response times and variation in response times.
3. Ping these sites again in quite mode (option -q). Analyze the packet loss.
4. Use ping with changing interval duration to 0.2s from the default of 1s as well as changing packet size from 56bytes to 1000 bytes.
5. Use flooding option (-f) to ping local m/c on the network. Analyze in wireshark the options of time difference between packets.
6. Use ping to send a packet of different size e.g. 1000 bytes with our own data pattern. Analyze the response time.
7. Use ping to use a different source IP address (e.g. of your neighbor) and analyze the response.

2.2.1 Exercise Expectations

1. Able to use ping with its various options.
2. Able to analyze ping response variation and packet loss statistics.

2.3 Using nc

1. Open terminal on two machines.
2. Identify each other's IP address. You can use the command `ip addr` in the linux terminal, to know the IP address of Ethernet interface. Do not the IP 127.0.0.1 for `lo` interface.

3. Run as TCP server on some port e.g. 2345 (`nc -l 2345`) in one terminal and UDP server (`nc -u -l 3456`) in another terminal.
4. Connect using clients (from another machine) to both TCP and UDP server and do chat.
5. Analyze wireshark capture of your chat conversation.
6. Transfer some files across machines e.g. `cat` “file”| `nc` “server IP” “server Port” on the client side and on server side (`nc -l` “port” >”file”)
7. Login in to remote machine without authentication

2.3.1 Exercise Expectations

1. Able to use nc for both TCP and UDP.
2. Able to do file transfer between two machines in the quickest possible way instead of using pendrive.
3. Able to use to communicate with a web server.

2.4 Using wget

1. Open terminal (command line. Preset Ctrl-Alt-T or from menu)
2. Mimic (option `-mk`) your college website (e.g. http://www.ksit.ac.in/), and access locally (turn off your internet).
3. Download a large file using the `--limit-rate=1m` e.g. http://rprustagi.com/workshops/web/media/movie.mp4, break the download by pressing **Ctrl-C** after about 5MB is downloaded and then download with resume option (-c). Ensure full download occurs and see if you can watch the movie after complete download.
4. Explore other options such as –d for debug headers, -O to save into a file,

2.4.1 Exercise Expectations

1. Able to use **wget** to download contents of a website for offline use.
2. Able to resume broken download.

3 Hands-on 2: IP and TCP Headers

Note: use wireshark capture to analyze all the 4 layers i.e. Application, Transport, Network and Link layers.

3.1 TCP Headers

1. Between two machines in your team, use `nc` to chat. Exchange few chat messages on the terminals
2. In the wireshark, analyze TCP headers. Look at sequence number, connection setup, data exchange and tear down after the connection is closed. Notice absence of any application layer protocol.
3. Do concurrent multiple chats and analyze corresponding TCP connections.
4. Analyze the TCP headers for HTTP based communication. Access a web page (e.g. www.ieee.org), and look at the application protocol header and data as TCP payload.
5. Analyze how to compute length of TCP payload.

3.2 UDP Headers

1. Between two machines in your team, use `nc -u` to chat. Exchange few chat messages on the terminals. Do concurrent chat with multiple clients.
2. In the wireshark, analyze UDP headers. Look at src and dstn port numbers, length and data exchange.
3. Identify the difference between concurrent UDP and TCP Chats.

3.3 Network layer analysis

1. Analyze IP addresses for `nc` chat both TCP and UDP.
2. Analyze IP header length and IP packet length.
3. Ping google with TTL value of 3 (use option `ping -c2 -t3 www.google.com`).
4. Analyze the ping request and reply for this request. Analyze the response received.

3.4 Exercise Expectations

1. Able to use **wireshark** to analyze layers of TCP/UDP/IP stack for a given application
2. Able to differentiate that use of ping is up to layer 3, use of `nc` is up to layer 4 and use of web make use for TCP/IP stack.

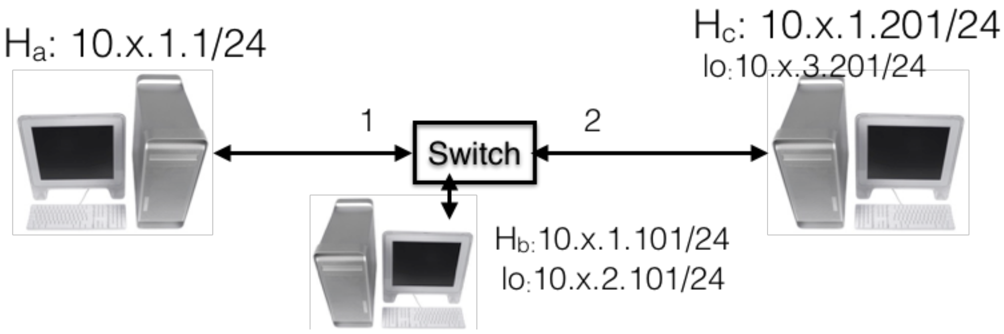
4 Hands-on 3: ICMP Errors

Note:

- The IP addresses specified in the below example are for illustrative purposes only. Use the IP Address assigned to your machines in the network.
- For this exercise you need to make a group of 3.

4.1 ICMP Redirect

1. Connect 3 m/cs in a network as shown in diagram. Assign the address manually to these systems Use the appropriate mask. Use value of x as per your team number e.g. 1 for team 1, 2 for team 2 and so on. The exercise can work without the use of small 4/8 port switch separately i.e. this exercise can be carried out in regular lab network.



2. On B and C, assign addresses from a different network on the loopback (`lo`) interface?
   a. `sudo ip addr add 10.x.2.101/24 dev lo` on B
   b. `sudo ip addr add 10.x.3.201/24 dev lo` on C

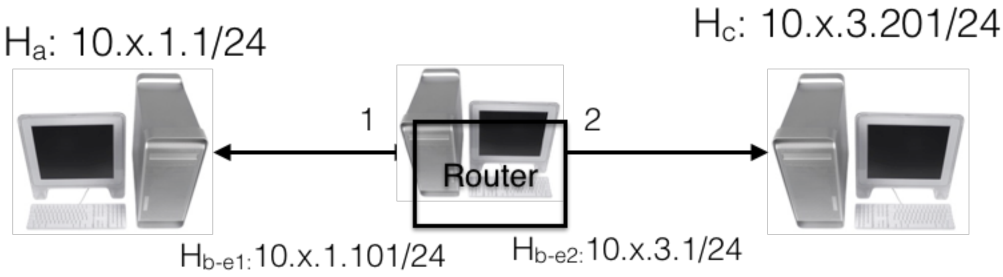
3. Disable acceptance of ICMP redirect on A to ensure that its routing table is not updated as per ICMP redirect. It is assumed that ethernet interface on the machine is `eth0` in the examples below. Replace it with appropriate values.
   a. `sudo sysctl -w net.ipv4.conf.all.accept_redirects=0`
   b. `sudo sysctl -w net.ipv4.conf.default.accept_redirects=0`
   c. `sudo sysctl -w net.ipv4.conf.eth0.accept_redirects=0`

4. Add mis-redirected (incorrect) routing entries on A.
   a. `sudo ip route add 10.x.2.0/24 via 10.x.1.201`
   b. `sudo ip route add 10.x.3.0/24 via 10.x.1.101`

5. Ping from A to addresses `10.x.2.101` and `10.x.3.201`
6. Analyze in wireshark the icmp redirect packets. Use the capture filter `icmp`.

4.2 PMTU Discovery

1. Connect 3 m/cs in a network as shown in diagram. Assign the address manually to these systems Use the appropriate mask. Use value of x as per your team number e.g. 1 for team 1, 2 for team 2 and so on. To get a second interface on B, use the USB to ethernet adaptor on B.



2. Convert machine B into a router as well as change the MTU value of the link between B and C.
   a. `sudo ip link set dev eth1 mtu 1000`
   b. `sudo sysctl -w net.ipv4.ip_forward=1`

3. Define appropriate routing on A to reach C
   a. `sudo ip route add 10.x.3.0/24 via 10.x.1.101`

4. Define appropriate routing on C to reach back A
   a. `sudo ip route add 10.x.1.0/24 via 10.x.3.1`

5. Send a ping packet bigger than 1000 bytes from A to C. Run following on A
   a. `ping -c1 -s 1200 -p 50515253 10.x.3.201`

6. Analyze wireshark capture to study ICMP error (fragmentation needed) and subsequent packet fragmentation (This will be studied later). How many packets are finally generated from A. Analyze the ping response. Are there two responses?

4.3 TTL Expiry

1. Connect 3 m/cs in a network as in above exercise of PMTU discovery.
2. Send a ping packet A to C with TTL=1.
3. Analyze the response with ICMP errors. Analyze how TTL Expiry works.
4. Analyze the source IP address in the ICMP error packet.

4.4 Exercise Expectations

1. Able to understand ICMP errors and source address of packet corresponding to ICMP errors.
2. Able to understand ICMP Redirect, PMTU Discovery and TTL expired.

5 Hands-on 4: Understanding ARP

5.1 Study ARP table maintenance

1. Setup three machines A, B, and C as shown in (4.1 **ICMP Redirect**). In this exercise we do not need 2nd and 3rd network i.e,. 10.x.2.0/24 and 10.x.3.0/24 and thus these addresses need not be assigned. Use of separate small switch is not recommended for this exercise
2. Study the existing ARP table and identify which machines are these.
   a. `arp -an`
3. Ping one live machine (i.e. machine which is reachable) and one unreachable machine. Study the ARP table. What does the entry show for unreachable machine.
4. Ping the broadcast address of your network. This requires sudo privilege.
   a. `sudo ping -b -c2  10.x.1.255`
5. Analyze in wireshark on number of ICMP response received as well as number of entries added in ARP Table.

5.2 Study Gratuitous ARP

1. Use the same setup as above. Note down MAC Address of 3 machines (A, B & C).
2. Install `arping` on the machine A (and may be others)
   a. `sudo apt install arping`
3. On A, add the IP address of A to that of C (replace eth0 with applicable interface name)
   a. `sudo ip addr add 10.x.1.201/24 dev eth0`
4. Issue arping with this new address to B
   a. `sudo arping -s 10.x.1.201 -c 2 10.x.1.101`
5. Study the ARP table at B. It should show updated MAC address of A for IP Address of C.

5.3 Exercise Expectations

1. Able to understand the use of ARP and Gratuitous ARP.

← end of exercise handout →