Exercises Day 01 – Basics of Networking

Experiential Learning Workshop

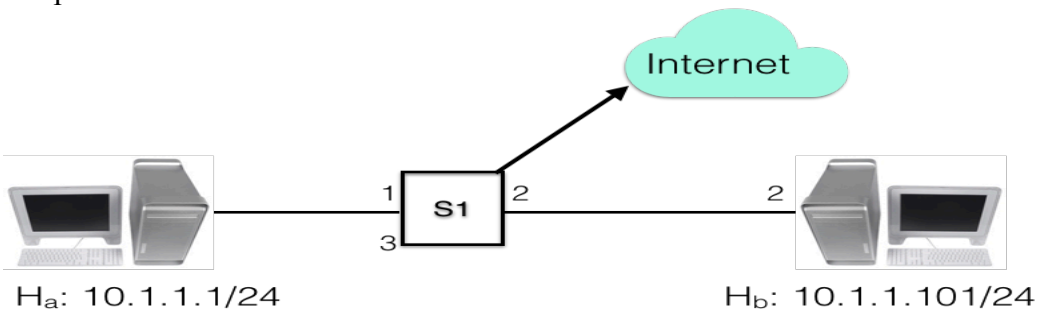
1 General Guidelines

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 `www.dr-ait.org` and capture traffic for this website. Use the appropriate capture filter (host www.dr-ait.org). You should see traffic for this website only and no other traffic. Analyse 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 neigbbour) 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" "serverPort" 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/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: TCP/IP Stack

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

3.1 Application layer analysis

1. Open the browser to access your institute website.
2. Analyze the HTTP headers, status codes of all the requests
3. For a given web page access, analyze number of HTTP requests made and responses received.
4. Analyze the number of TCP connections used.

3.2 Transport layer analysis

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. Analyze the TCP headers for HTTP based communication. Look at the application protocol header and data as TCP payload.
4. Think analyze how to compute length of TCP payload.
5. Use nc (option -u) to chat using UDP and analyze the UDP headers. Identify the fields in UDP protocols.

3.3 Network layer analysis

1. Analyze IP addresses for nc chat.
2. Analyze header length and IP packet length.
3. Ping google with TTL value of 3 (use option `-t 3`). Analyze the ping request and reply for this request. Analyze the response received.

3.4 Ethernet layer analysis

1. For the above ping packets to google, analyze src and destination MAC address and type.
2. For the nc chat between two machines in your network, analyze the MAC address and type field..
3. Think about how to find the length of ethernet packet.

3.5 Exercise Expectations

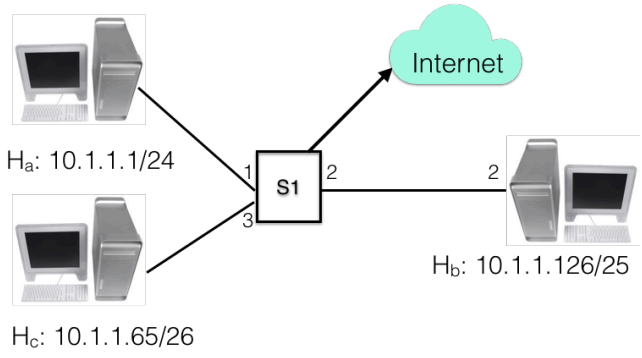
1. Able to use **wireshark** to analyze layers of TCP/IP stack for a given application
2. Able to differentiate that use of ping is up to layer 3, use of nc is up to layer 4 and use of web make use for TCP/IP stack.

4 Hands-on 3: IP Subnetting

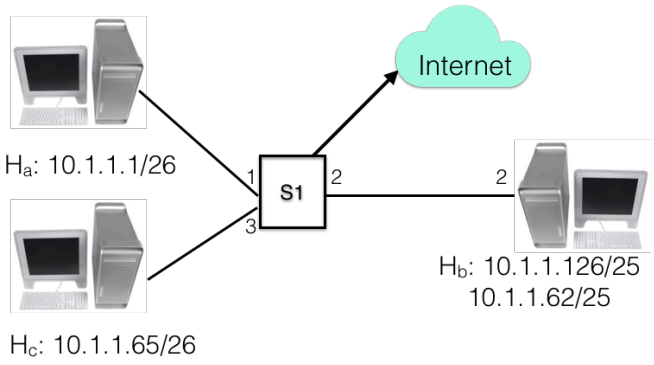
Note:

- 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.

1. Connect 3 m/cs in a network as shown in diagram with addresses as 10.1.1.1/24 (A), 10.1.1.126/25 (B), and 10.1.1.65/26 (C) for team 1. Assign the address manually to these systems since subnet mask is to be assigned differently. Use the appropriate mask. For the team 2, assign the addresses as 10.2.1.1/24, 10.2.1.126/25, and 10.2.1.65/26. Similarly, for Kth team, use the IP Addresses as 10.K.1.1/24, 10.K.1.126/25, and 10.K.1.65/26.



2. Ping from A to B, and A to C. A should see a response from B but not from C. Explain why?
3. Ping from B to C. Does it work properly. Change the subnet mask of C to /27 instead of /26. Does ping still work. Analyze the IP packet behavior.
4. Change the subnet mask of A to /26 and assign another IP Address 10.1.1.62/25 to B as shown below
    a. Use the the command "`sudo ip add 10.1.1.62/25 dev enp1s0`".
    b. Verify that B has two IP addresses using command "`ip addr show`"



5. Explore which of these two IPs of C can be successfully pinged from A and which can be successfully pinged from B. Explore and explain.

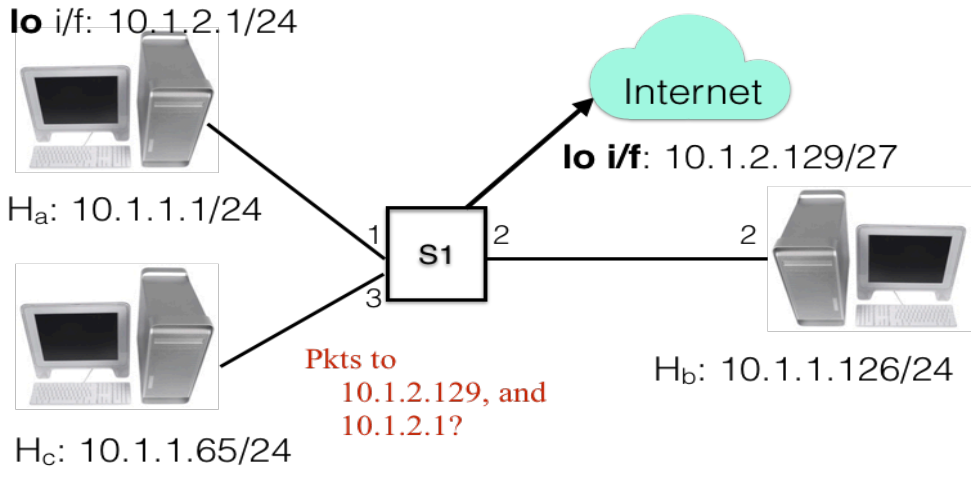
4.1 Exercise Expectations

1. Able to understand the role of subnets in routing of packets.
2. Able to understand the range of IP addresses in a given subnet.

5 Hands-on 4: Longest Prefix Match

5.1 Forwarding table and longest prefix match

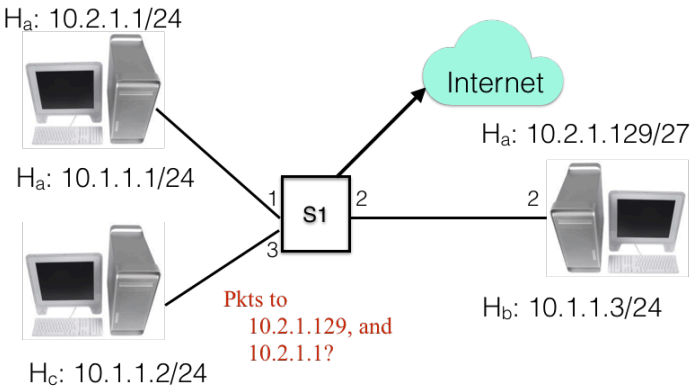
1. Setup three machines A, B, and C as shown below. Both A and B have another IP address in another network. Assign the respective addresses to their loopback interface i.e. **lo** (it is the interface having 127.0.0.1 IP address)



2. The IP subnet of new IP addresses 10.1.2.1/24 of A overlaps IP subnet of new IP address 10.1.2.129/27 of B.

3. Build the routing (forwarding table) at C (having IP 10.1.1.65) corresponding to these two overlapping networks. Use the following commands to build the routing table and verify the routing table
   a. `sudo ip route add 10.1.2.0/24 via 10.1.1.1`
   b. `sudo ip route add 10.1.2.128/27 via 10.1.1.126`
   c. `ip route show`

4. Ping these IP addresses from B. Both should be successful.
5. Verify in the wireshark capture that two responses are from two different machines i.e. A and B. Verify this with MAC Addresses at link layer.



6. Ping two unassigned IP addresses such 10.2.1.130, and 10.2.1.161. Verify that ICMP packet for 10.2.1.130 goes to C and ICMP packet for 10.2.1.161 goes to A. Explore and analyze.

5.2 Exercise Expectations

1. Able to understand the use of longest prefix match in packet routing..
2. Able to understand how overlapping subnet works where one subnet is subsumed in another subnet.

← end of exercise 01 handout →