COMP 416: Computer Networks - Project 3

Network Layer Analysis and DV Routing Simulator

Hasan Can Aslan - 60453

**Part 1. Network Layer Analysis**

**Part 1A. Network Interface Analysis**

**1.** We provide 2 parameters to netstat command. First -a parameter displays all active TCP connections and the TCP and UDP ports on which the computer is listening, -i parameter displays network interface status. So netstat -ai command shows the state of the network interfaces and their TCP connections.

Calendar

Description automatically generated with medium confidenceCalendar

Description automatically generated with medium confidence

**2.** Destination column: Shows the destination network.

Gateway column: Indicates the router through which packets are forwarded.

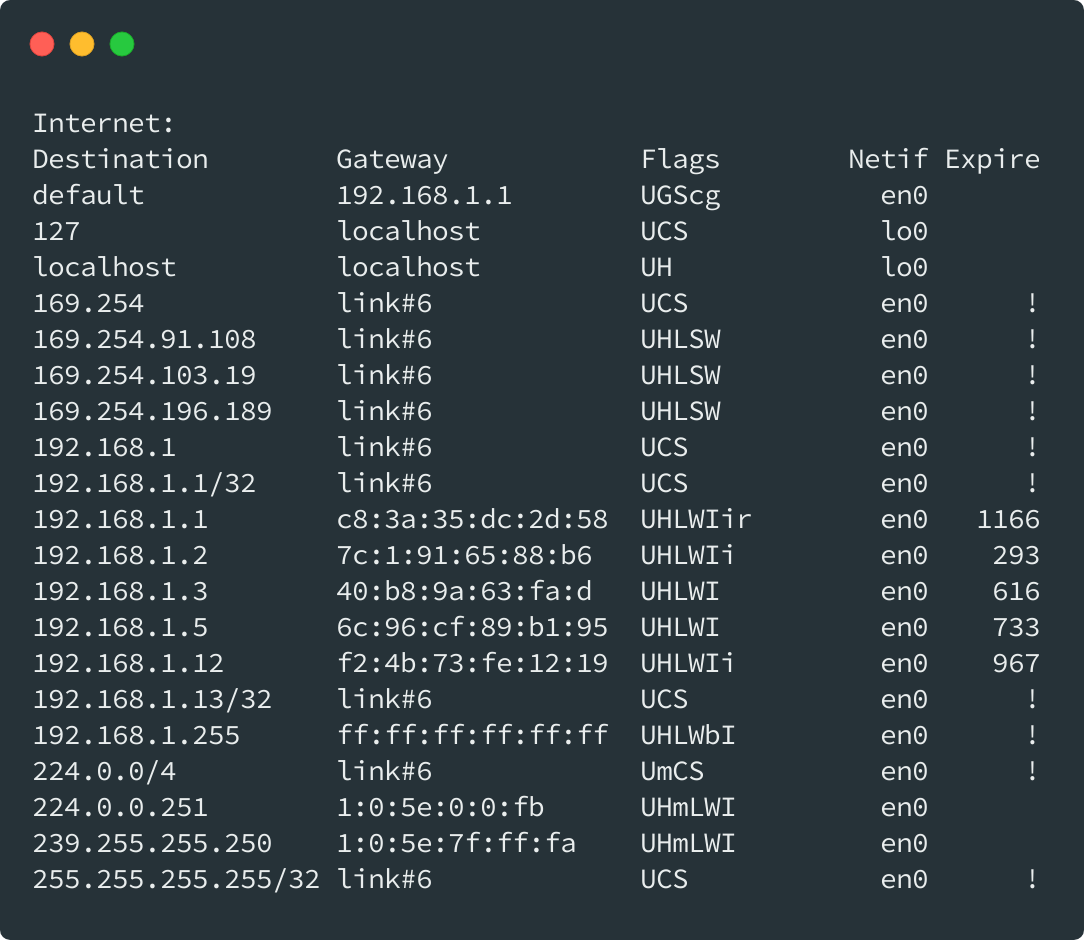
Flags column: The U flag indicates that the route is up route is valid.

The G flag indicates that the route is to a gateway.

The S flag indicates route added with the route command.

The H flag indicates route is to a host rather than to a network, where the destination address is a complete address.

Netif column: Indicates the interface on the local host that is the source endpoint of the transmission.

****

**3.** My client using ports 64532 and 64531 to connect nyu.edu. First, I get the IP address of the host, using nslookup.

Text

Description automatically generated with medium confidence

Then using netstat -P tcp command, we can find nyu.edu’s IP address in the list.

**A picture containing text

Description automatically generated**

**Part 1B. ICMP Analysis**

**4.** TTL, time to live is the number of hops or amount of time that a packet exists inside a network. When TTL exceeded, packet is discarded. Time to live is important for caching information. It determines how long the data can be cached and when the information should be updated. We can find TTL under the network layer, for example under the Internet Protocol.

**5.** ICMP protocol is designed to communicate between network layers. However, port numbers are used to communicate between application layers, so ICMP has neither destination nor source port number.

**6.** Minimum TTL is 1.

**Graphical user interface, text, website

Description automatically generated**

**7.** My computer sends first a packet with TTL = 1, the second packet has TTL = 2 so and so on. A router will decrement the packet’s TTL when the packet passes through. If the packet arrives the router with TTL value equals to 1, then router sends an ICMP packet to the source. In this way we can count the router along the path to the destination.

**8.** For each router along the path, traceroute sends 3 probes.

**Text

Description automatically generated**

**9**. I reach to the destination much faster, at 11 hops, when tracing 18.31.0.200.

Text

Description automatically generated

**10.** Routing Blackhole is the routing the all network traffic into blackhole and lost it. We can use such a system to prevent attacks to our networks like DDoS attack. However, normal traffic is also affected such prevention system.

**11.** We can arrange the output fields by using the -o flag. I use following command:

A picture containing logo

Description automatically generated

**Table

Description automatically generated**

These fields are described in man page of mtr as follows:

**Calendar

Description automatically generated**

**12.** mtr continuously updates times by polling a remote server and allows us to see how the latency and performance changes over time. On the other hand, traceroute allows us to discover the pathway to a host. When we compare two Wireshark record, we can see that mtr continuously sends probes whilst traceroute sends predetermined number of probes.

**Part 2. DV Routing Simulator**

**Part 2A. Implementation**

When constructing Node class, I initialize id from static counter, get cost and neighbors from static arrays of DVSimulator class. myDV is initially equals to cost of that node. I initialize bestPath array based on following conditions: If nodes id or neighbor’s id is equal to destination id, then use that id, otherwise get a random node from randomNeighbor method.

**Text

Description automatically generated**

For the notifyNeighbors method, I create a new packet for each neighbor that created with current node id as source, neighbor id as destination and current node’s myDV array as the dv. Then, I send the packet.

**Text

Description automatically generated**

**Citation:** I get the following method from <https://stackoverflow.com/a/34541755>. It is simple method for ease of use when searching a value in array**.**

**Text

Description automatically generated**

**Part 2B. Optional (Bonus)**

I compare for each value in the DV received from neighbor as new\_cost and current DV as current\_cost. If there is a cheaper path, I update myDV with corresponding cost and update bestPath with corresponding neighbor’s id. If there is such a change in current Node’s DV I update hasDVChanged value as true. If hasDVChanged value is true, then I notifyNeighbors and increment numUpdates. By doing that, I increment and notify neighbors just once for each packet.

**Text

Description automatically generated**