University of Dhaka

CSE:3111 Computer Networking Lab

Lab Report

Title: Lab exercises on LAN configuration and troubleshooting tools

Diponker Roy
Roll: 28
Reg: 2020315638

Submitted to:

Dr. Md. Abdur Razzaque
Dr. Muhammad Ibrahim
Dr. Md. Redwan Ahmed Rizvee
Dr. Md. Mamun Or Rashid

Submitted on: 2021-08-31

Introduction

Provide an overview of the importance of LAN configuration and troubleshooting in maintaining a healthy network. State the objectives of the lab exercises and mention the tools you will be using.

Methodology

Detail the specific steps taken during the lab exercises, including the configuration changes made, the devices involved, and the scenarios tested.

LAN configuration exercises

0.1 PING

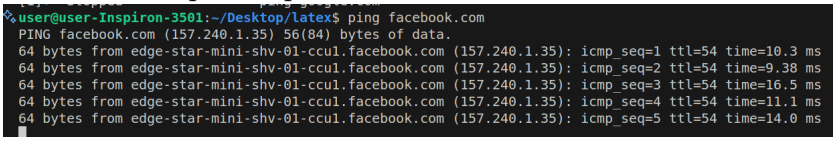
Using the `ping` command, test the connectivity between two devices. When the ping is successful, record the round-trip time (RTT) and the number of packets lost. When the ping is unsuccessful, record the error message. Repeat the test with different packet sizes and record the results.

Command Execution

To use PING, open the terminal and execute the following command:
$ping < destinationHost >$

Sample Output

Here is a sample output of a PING command:



Analysis

In the sample output, the PING command sends ICMP (Internet Control Message Protocol) packets to the specified host (in this case, `google.com`) and receives responses. The output includes information such as the round-trip time (rtt) for each packet and overall statistics.
By analyzing the PING output, you can assess the network connectivity and performance to the specified destination.

Limiting the number of PING requests

By default, the PING command sends an unlimited number of requests to the destination. To limit the number of requests, use the `-c` option:

Sample Output

$ping - c < num >< destinationHost >$

```
user@user-Inspiron-3501:~/Desktop/latex$ ping -c 4 du.ac.bd
PING du.ac.bd (103.221.255.104) 56(84) bytes of data.
64 bytes from 103.221.255.104 (103.221.255.104): icmp_seq=1 ttl=57 time=3.69 ms
64 bytes from 103.221.255.104 (103.221.255.104): icmp_seq=2 ttl=57 time=11.4 ms
64 bytes from 103.221.255.104 (103.221.255.104): icmp_seq=3 ttl=57 time=4.12 ms
64 bytes from 103.221.255.104 (103.221.255.104): icmp_seq=4 ttl=57 time=2.87 ms

--- du.ac.bd ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 2.871/5.525/11.422/3.433 ms
```

Analysis

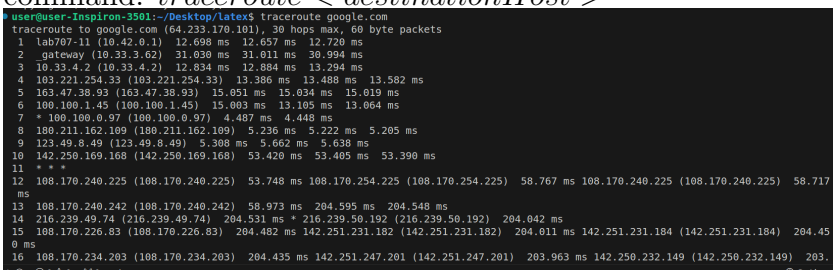
In the sample output, the PING command sends 4 ICMP packets to the specified host (in this case, `du.ac.bd`) and receives responses. The output includes information such as the round-trip time (rtt) for each packet and overall statistics.

0.2 TRACEROUTE

Using the `traceroute` command, test the connectivity between two devices. When the traceroute is successful, record the RTT and the number of hops. When the traceroute is unsuccessful, record the error message. Repeat the test with different packet sizes and record the results.

Command Execution

To use TRACEROUTE, open the terminal and execute the following command: $traceroute < destinationHost >$



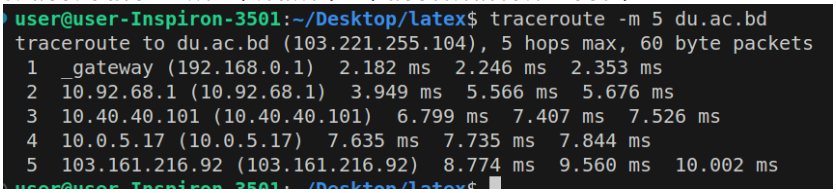
Number of hops is 30. It indicates the time to live (TTL) of the packet. The TTL value is decremented by one each time the packet is forwarded by a router. When the TTL value reaches zero, the packet is discarded and an ICMP error message is sent to the source. The source can use the ICMP error message to determine the IP address of the router that discarded the packet. The source can then use the IP address to determine the router name.

Limiting the number of hops

By default, the TRACEROUTE command sends packets with a TTL value of 1 and increments the TTL value by 1 for each subsequent packet. To limit the number of hops, use the `-m` option:

Sample Output

$traceroute - m < num >< destinationHost >$



We use 5 hops. It indicates the time to live (TTL) of the packet. The TTL value is decremented by one each time the packet is forwarded by a router.

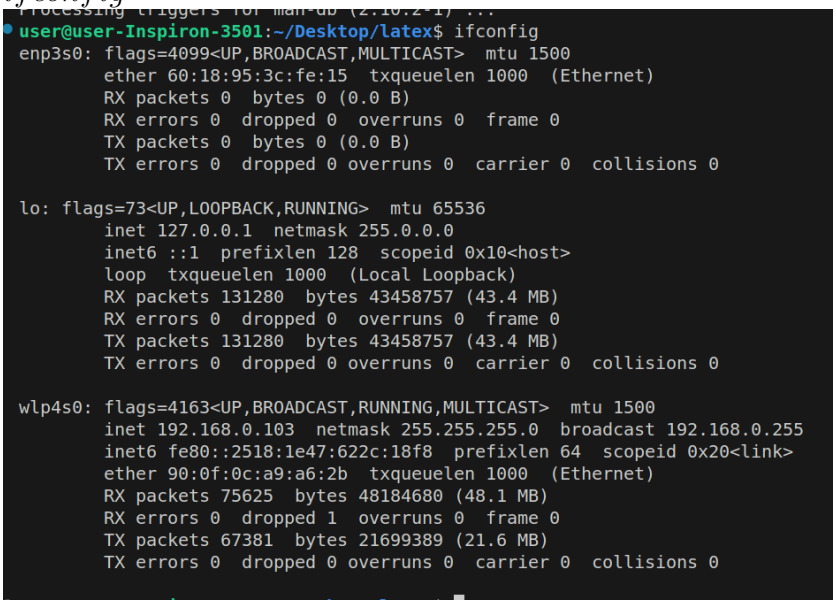
When the TTL value reaches zero, the packet is discarded and an ICMP error message is sent to the source. The source can use the ICMP error message to determine the IP address of the router that discarded the packet. The source can then use the IP address to determine the router name.

0.3 IFCONFIG

Using the `ifconfig` command, display the IP address, subnet mask, and default gateway for a device. Record the results. The "**ifconfig**" command with no arguments will display all the active network interface configuration details that includes their assigned IP addresses, netmasks, and other relevant information.

Command Execution

To use IFCONFIG, open the terminal and execute the following command:
ifconfig



Analysis

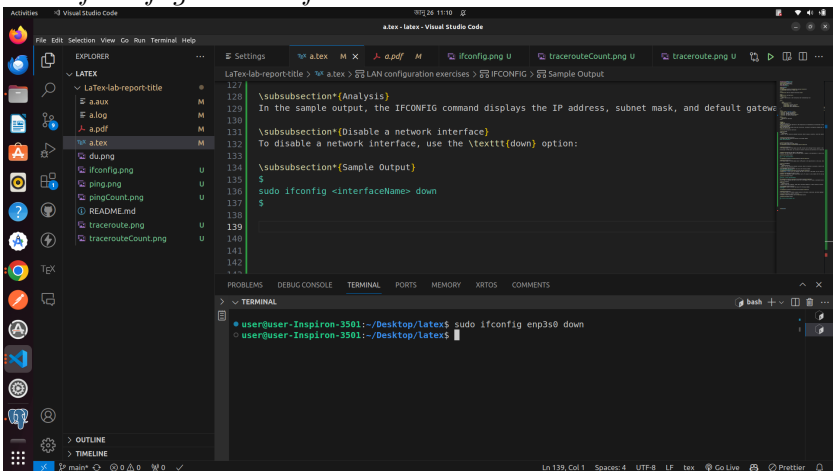
In the sample output, the IFCONFIG command displays the IP address, subnet mask, and default gateway for the device. The output also includes information such as the MAC address, MTU, and the number of packets transmitted and received.

Disable a network interface

To disable a network interface, use the `down` option:

Sample Output

$sudoifconfig < interfaceName > down$



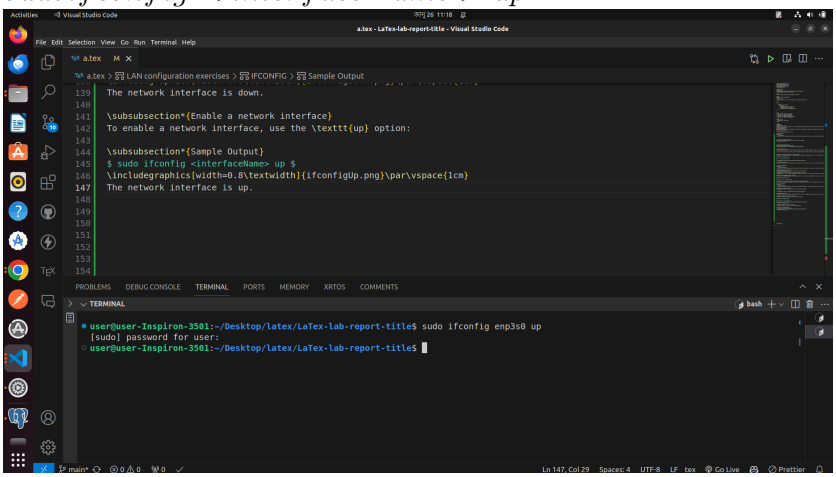
The network interface is down.

Enable a network interface

To enable a network interface, use the `up` option:

Sample Output

$sudoifconfig < interfaceName > up$



The network interface is up.

0.4 ARP

Using the `arp` command, display the ARP cache for a device. Record the results.

Command Execution

To use ARP, open the terminal and execute the following command: *arp*

```
user@user-Inspiron-3501:~/Desktop/latex/LaTex-lab-report-title$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   e0:1c:fc:a6:80:c6   C                     wlp4s0
169.254.169.254                  (incomplete)                              enp3s0
_gateway                 ether   e0:1c:fc:a6:80:c6   C                     enp3s0
user@user-Inspiron-3501:~/Desktop/latex/LaTex-lab-report-title$
```

Analysis

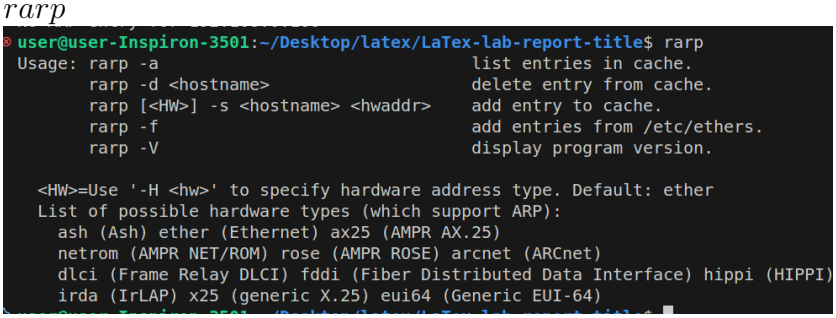
In the sample output, the ARP command displays the ARP cache for the device. The output includes information such as the IP address, MAC address, and type of each entry.

RARP

Using the `rarp` command, display the RARP cache for a device. Record the results.The Reverse Address Resolution Protocol (RARP) is a networking protocol that is used to map a physical (MAC) address to an Internet Protocol (IP) address. It is the reverse of the more commonly used Address Resolution Protocol (ARP), which maps an IP address to a MAC address.

Command Execution

To use RARP, open the terminal and execute the following command:
rarp



Analysis

In the sample output, the RARP command displays the RARP cache for the device. The output includes information such as the IP address, MAC address, and type of each entry.

NSLOOKUP

Using the `nslookup` command, display the DNS cache for a device. Record the results.

Command Execution

To use NSLOOKUP, open the terminal and execute the following command:
nslookup

```
user@user-Inspiron-3501:~/Desktop/latex/LaTex-lab-report-title$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.182.14
Name:   google.com
Address: 2404:6800:4007:819::200e
```

Analysis

In the sample output, the NSLOOKUP command displays the DNS cache for the device. The output includes information such as the IP address, hostname, and type of each entry. nslookup followed by the domain name will display the "A Record" (IP Address) of the domain. Use this command to find the address record for a domain. It queries domain name servers and gets the details.

Using type any

To display all the information in the DNS cache, use the `set type=any` option:

Sample Output

$nslookup - type = any < domainName >$

```
user@user-Inspiron-3501:~/Desktop/latex/LaTex-lab-report-title$ nslookup -type=any google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 2404:6800:4007:819::200e
google.com      nameserver = ns4.google.com.
google.com      nameserver = ns2.google.com.
google.com      nameserver = ns1.google.com.
google.com      nameserver = ns3.google.com.
google.com      rdata_65 = 1 . alpn="h2,h3"
Name:   google.com
Address: 142.250.182.14

Authoritative answers can be found from:
ns1.google.com  internet address = 216.239.32.10
ns2.google.com  internet address = 216.239.34.10
ns3.google.com  internet address = 216.239.36.10
ns4.google.com  internet address = 216.239.38.10
ns1.google.com  has AAAA address 2001:4860:4802:32::a
ns2.google.com  has AAAA address 2001:4860:4802:34::a
ns3.google.com  has AAAA address 2001:4860:4802:36::a
ns4.google.com  has AAAA address 2001:4860:4802:38::a
```

There are also available types of DNS records. The most common ones are:

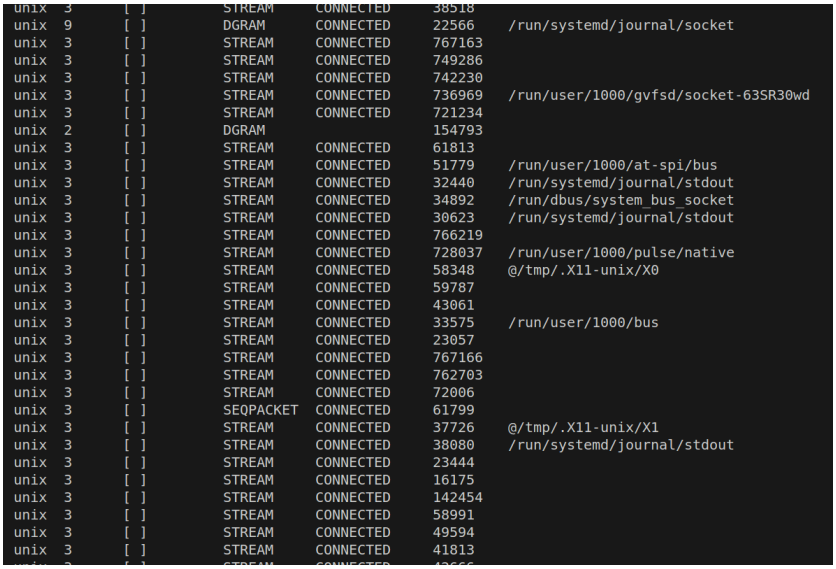
- A: Address record
- AAAA: IPv6 address record
- CNAME: Canonical name record
- MX: Mail exchange record
- NS: Name server record
- PTR: Pointer record
- SOA: Start of authority record
- TXT: Text record

NETSTAT

Using the `netstat` command, display the active TCP connections for a device. Record the results. The netstat command is like a special tool in Linux that helps you understand and check things about how your computer connects to the internet. It can tell you about the connections your computer is making, the paths it uses to send information, and even some technical details like how many packets of data are being sent or received. In simple terms, it's like a window that shows you what's happening with your computer and the internet. This article will help you learn how to use netstat, exploring different ways to get specific information and giving you a better idea of what's going on behind the scenes.

Command Execution

To use NETSTAT, open the terminal and execute the following command:
netstat



```
unix  3      [ ]         STREAM     CONNECTED     38518
unix  9      [ ]         DGRAM      CONNECTED     22566    /run/systemd/journal/socket
unix  3      [ ]         STREAM     CONNECTED     767163
unix  3      [ ]         STREAM     CONNECTED     749286
unix  3      [ ]         STREAM     CONNECTED     742230
unix  3      [ ]         STREAM     CONNECTED     736969   /run/user/1000/gvfsd/socket-63SR30wd
unix  3      [ ]         STREAM     CONNECTED     721234
unix  2      [ ]         DGRAM                    154793
unix  3      [ ]         STREAM     CONNECTED     61813
unix  3      [ ]         STREAM     CONNECTED     51779    /run/user/1000/at-spi/bus
unix  3      [ ]         STREAM     CONNECTED     32440    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     34892    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     30623    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     766219
unix  3      [ ]         STREAM     CONNECTED     728037   /run/user/1000/pulse/native
unix  3      [ ]         STREAM     CONNECTED     58348    @/tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     59787
unix  3      [ ]         STREAM     CONNECTED     43061
unix  3      [ ]         STREAM     CONNECTED     33575    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     23057
unix  3      [ ]         STREAM     CONNECTED     767166
unix  3      [ ]         STREAM     CONNECTED     762703
unix  3      [ ]         STREAM     CONNECTED     72006
unix  3      [ ]         SEQPACKET  CONNECTED     61799
unix  3      [ ]         STREAM     CONNECTED     37726    @/tmp/.X11-unix/X1
unix  3      [ ]         STREAM     CONNECTED     38080    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     23444
unix  3      [ ]         STREAM     CONNECTED     16175
unix  3      [ ]         STREAM     CONNECTED     142454
unix  3      [ ]         STREAM     CONNECTED     58991
unix  3      [ ]         STREAM     CONNECTED     49594
unix  3      [ ]         STREAM     CONNECTED     41813
```

Analysis

In the sample output, the NETSTAT command displays the active TCP connections for the device. The output includes information such as the protocol, local address, foreign address, and state of each connection.