Suyog Joshi

Assignment 1

Computer Networks

09/21/2023

Question 1)

My computer doesn't support the virtual machine, so I used my computer as a host. But I tried to login to the university computer remotely using ssh(Secure Shell).

The Ip address of the host machine is 127.0.0.1/8.

```
sjoshi@prospero:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether e0:d5:5e:db:36:ae brd ff:ff:ff:ff:ff:ff
    inet 64.106.20.211/25 brd 64.106.20.255 scope global dynamic noprefixroute enp1s0
       valid_lft 3770sec preferred_lft 3770sec
    inet6 fe80::9608:5fa2:1f3e:9c03/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eno2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether e0:d5:5e:db:36:af brd ff:ff:ff:ff:ff:ff
    altname enp0s31f6
4: wlo1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether b4:6b:fc:53:e0:21 brd ff:ff:ff:ff:ff:ff
    altname wlp0s20f3
sjoshi@prospero:~$
```

OpenSSL (Open Secure Sockets Layer) is an open-source version of the SSL and TLS security protocols, which provide encryption and server authentication over the internet. It is used to implement the SSL and TLS protocols, for encrypting data transmitted over the internet. These protocols can establish encrypted connections between the client (i.e., the local machine) and the server (i.e., the remote system) ensuring that the data transmitted between them is secure and protected. It also provides strong user authentication which helps to verify the identity of both client and the server while remote access. When we use ssh to access the computers remotely it ensures secure communication as well using OpenSSL.

Virtual Machine is accessible remotely.

```
Last login: Mon Sep 18 15:51:20 on ttys004
suyogjoshi@Suyogs-MacBook-Pro Computer Networks % ssh sjoshi@moons.cs.unm.edu
sjoshi@moons.cs.unm.edu's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Sep 18 04:34:41 PM MDT 2023

  System load:  1.19921875          Temperature:             54.0 C
  Usage of /:   26.0% of 227.14GB   Processes:               476
  Memory usage: 18%                 Users logged in:         2
  Swap usage:   0%                  IPv4 address for enp1s0: 64.106.20.211

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

292 updates can be applied immediately.
185 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

34 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


Last login: Mon Sep 18 15:49:01 2023 from 98.60.72.53
sjoshi@prospero:~$
```

Question 2)

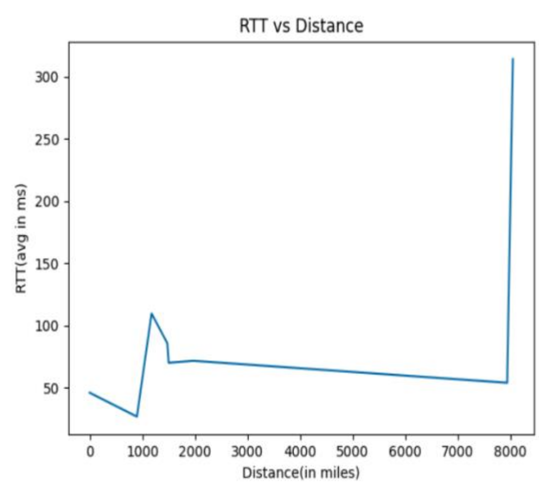
	Location	Distance (in miles)	RTT (in ms)
unm.edu	Albuquerque, New Mexico	0	45.823
berkley.edu	Berkeley, California	898.3	26.649
unam.mx	Mexico City, Mexico	1176.8	109.496
ufl.edu	Gainesville, Florida	1476.8	85.588
cmu.edu	Pittsburgh, Pennsylvania	1501.8	69.917
mit.edu	Cambridge, Massachusetts	1970.2	71.514
kaust.edu.sa	Thuwal, Saudi Arabia	7938.2	53.859
home.iitd.ac.in	Delhi, India	8046.2	313.828

```
import matplotlib.pyplot as plt
import numpy as np

y = [45.823, 26.649, 109.496, 85.588, 69.917, 71.514, 53.859, 313.828]
x = [0, 898.3, 1176.8, 1476.8, 1501.8, 1970.2, 7938.2, 8046.2]
plt.plot(x, y)

plt.xlabel('Distance(in miles)')
plt.ylabel('RTT(avg in ms)')
plt.title('RTT vs Distance')

plt.show()
plt.plot(x, y)
plt.show()
```



I observed that the distance is directly proportional to the RTT meaning as the distance increases the RTT also seems to increase and if the distance decreases the RTT also seems to decrease. So, if data has to travel greater distance, it takes more time (higher RTT) to reach its destination and return. We can also see some variation in the graph as well which could possibly be because of network congestion, routing changes or other network conditions. So, we can also say that there might be communication delay if the distance increasing resulting in higher RTT. The graph illustrates a non-linearity in the relationship.

Question 3)

1)

We have,

File size = 3 MB

$= 3 * 2^{10}$ KB

$= 3 * 2^{10} * 2^{10}$ MB

$= 3 * 2^{20}$ bytes

$= 3 * 2^{20}$

= 3145728

One way delay = 40 ms

So, RTT = 2 * 40 ms

= 80 ms

= 0.08 s

Packet size = 1 KB

$= 2^{10}$ bytes – 40 bytes

= 984 bytes

Now,

Number of packets = File Size / Packet size

$$= \frac{3145728}{984}$$

= 3196.878049 packets

= 3197 packets

Since, the last packet is less than 1KB so the total data to be sent = 3196 * 2^{10} + 40 + (3145728 – 3144864)

= 3196 * 2^{10} + 40 + 864

= 3279656 bytes

= 3273608 * 8 bits

= 26188864 bits

Transmission Rate = 1MBbps

= 10^6 bits/s

Total data = 3 * 2^{20} * 8

= 24 * 2^{20} bits

Data transmission time = Total data / Transmission Rate

$$= \frac{26188864}{10^6}$$

= 26.188864 s

Total time = RTT + One way delay + Data transmission time

= 0.08 + 0.04 + 26.188864

= 26.308864 s

So, it will take around 26.308864 s to transmit the file.

2)

Everything is similar to (1) but here we must account for one RTT before transmitting a packet. So,

Each packets waits 1 RTT.

So, we have 3072 packets. But 1 RTT is already counted. So, we have 3071 packets left.

3196 * 0.08 = 245.68 s

Total time = 26.188864 + $\frac{864*8}{10^6}$ + 3197 * 0.08 + 0.08

= 282.035776 s

Question 4)

DNS is the Domain Name System which translates the human understandable domain names to machine understandable IP addresses allowing browsers to get to the website over the internet.

a) Major Components of DNS

1) Domain name space and resource records

   This is the Domain Hierarchy. It forms a hierarchical tree with a starting point called root, which is a null node. And, below the root domain, you have atop-level domains, which are then subdivided into second-level domains, third level and so forth. Each of these nodes contains DNS records used to associate IP addresses with domain names and vice versa. This record holds information such as IP addresses, mail server details and other information.

2) DNS Servers

This element is responsible for storing and overseeing DNS records to particular domains. A name server is designed to receive and reply to inquiries from resolvers and other name servers. Additionally, in order to expedite future searches for same data, name servers/DNS servers typically store data they have previously retrieved in a cache.

3) Resolver

A resolver is a software installed on a client computer or network devices and its primary function is to kickstart DNS queries when necessary. Its role is it dispatches DNS queries to name servers, enabling the conversion of domain names into IP addresses. The resolver triggers DNS queries, name servers await and reply to queries initiated by resolver and name spaces serve as storage points where server seek information to establish connection between domain names and IP addresses making them vital components that enables the translation of human-friendly domain names into IP addresses and facilitating communication over the network.

b) Root

The root can be seen as the pinnacle of the domain tree hierarchy represented by a single dot(.). It marks the initiation point for all DNS resolution processes. Also, when a DNS resolver needs to resolve a domain name into an IP address, it starts at the root level. Within the root, you'll find the name servers for TLDs (Top Level Domains) like .org, .com, .net and more. Below these TLDs are second-level domains, typically for identifying organizations. Further down the hierarchy, you encounter third-level domains, fourth-level domains and so on. For instance, the domain name "facebook.com" comprises the second-level domain "facebook" and the top-level domain ".com".

c) Zones

As we know the domain database is vast and it would be impractical for a single organization to manage its efficiency. So, to address this challenge, the responsibility for maintaining the domain databse is decentralized and distributed amongst the independent organizations which leads to more effective management of DNS records. These decentralized subdivisions are known as "zones". Zones represent administrative divisions within the DNS namespace, with each zone corresponding to a specific portion of the domain name space managed by a particular organization. It serves the role of organizing and facilitating the DNS records for individual domains and subdomains by allowing different to independently oversee and maintain their respective sections of the database. For example if a organization owns the domain unm.edu, they would be responsible for managing the DNS records for unm.edu and its subdomain.

d)

Suppose if I want to find the address of “isi.edu”, the components role and why they are needed are as follows:

The resolver on the client’s computer initiates the DNS queries. The DNS resolver begins by sending a query to one of the root servers and the root server provides the information for the top-level domain (TLDs)i.e., looking for “.edu”. This “.edu” TLD is consulted to pinpoint the name server for “isi.edu”. This progression continues recursively, traversing the DNS hierarchy until the requisite name server is identified. Once the name server for “isi.edu” is found, a query is directed to this specific name server to fetch the IP address linked to the domain. Finally, the name server for “isi.edu” responds with the requested IP address, which is then transmitted back to the resolver.

Subdomains are hierarchical extensions of a domain name. In the case of “isi.edu”, there might be subdomains like “enginnering.isi.edu”, “rec.isi.edu” and each of them may have their own authoritative name servers responsible for DNS records specific to the domain. It helps manage complex applications more efficiently because it can be directed to different servers instead of single server handling all the requests.