CS 5110 Network Architecture I

Assignment 1 Solution

Sri Harsha Chennavajjala

sc9v9@mail.umkc.edu

1. Suppose two hosts, A and B are separated by 20,000 kilometers and are connected by a direct link of R=1 Mbps. Suppose the propagation speed over the link is $2*10^8$ meters/sec. Consider sending a file of 2,000,000 bits from Host A to Host B.

a. Suppose the file is sent continuously as one big message. How long does it take to send the file, assuming it is sent continuously?

b. Suppose now the file is broken up into 1000 packets with each packet containing 2,000 bits. Suppose that each packet is acknowledged by the receiver and the transmission time of an acknowledgement packet is negligible. Finally, assume that the sender cannot send a packet until the preceding one is acknowledged. How long does it take to send the file?

c. Calculate the bandwidth-delay product, R*t_{prop}. What does it mean?

(Provide an interpretation of the bandwidth-delay product.)

d. If there are two routers between Host A and B (rather than a direct link), and all three links have 1 Mbps links, how long does it take to send the file? (use the assumptions in 1.b)

Solution:

1.a:

The total time taken to transfer the file from source to destination can be calculated using

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

Where d_{nodal} = Total time taken to transfer the file

d_{proc} = Time taken to process the packets

d_{queue} = Time spent in keeping the packets in transmission queue

d_{trans} = Time taken to transfer the file to the transmission medium

d_{prop} = Time taken by packet to propagate from source to destination

Here processing delay and queuing delay are zero as the file is being sent as a single packet.

We need to calculate transmission delay and propagation delay.

Transmission delay can be calculated using

$$d_{trans} = \frac{No.of\ bits\ transmitted\ (N)}{Rate\ of\ transmission\ of\ the\ medium\ (R)}$$

Here

N = 2,000,000 bits

R = 1Mbps = 10^6 bits per second

$$\therefore \text{Transmission delay} = \frac{2000000}{1000000} = 2 \text{ sec}$$

$$d_{prop} = \frac{distance\ (d)}{speed\ of\ medium\ (s)}$$

Here

$$\text{d} = 20{,}000 \text{ km} = 2 * 10^7 \text{ meters}$$

$$\text{s} = 2 * 10^8 \text{ m/s}$$

$$\therefore \text{Propagation delay} = \frac{2 * 10^7}{2 * 10^8} = 0.1 \text{ sec}$$

$$\therefore \text{Total delay} = 2 + 0.1 = 2.1 \text{ sec}$$

∴ The total time taken to transfer a 2,000,000 bits file over a link with R=1Mbps is 2.1 seconds

1.b :

In this case, we need to calculate the transmission delay of each packet as a packet can be sent only after the acknowledgement of successful receiving of previous packet at the destination. But here the transmission time for acknowledgement is negligible. The given values are:

Length of each packet (L) = 2,000 bits

Number of packets = 1,000

Distance (d) = 20,000 km

Data transmission rate (R) = 1Mbps

Propagation speed (s) = $2 * 10^8$ m/s

Propagation time of each packet

$d_{prop} = \frac{d}{s}\ sec$

$= \frac{2 * 10^7}{2 * 10^8} = 0.1$ sec

Propagation time of acknowledgement is same as propagation time of packet.

Transmission time of each packet

$d_{trans} = \frac{L}{R}$

$= \frac{2000}{10^6} = 0.002$ sec

$\therefore$ The total time taken to transfer a packet is $= 0.1 + 0.002 + 0.1$

$= 0.202$

Thus there are 1000 such packets. Therefore total time taken to transfer these packets is

$1000 * 0.202 = 202$ sec

1.c :

Bandwidth - Delay product can be calculated using:

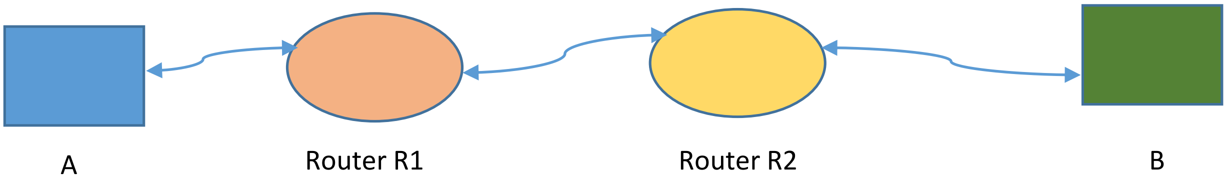
BDP (bits) = $\left(total\ available\ bandwidth\ \left(\frac{bits}{sec}\right)\right) * \left(round\ trip\ time\ (sec)\right)$

$$\therefore \text{BDP} = 10^6 * 10^{-1} = 10^5 \text{ bits}$$

Therefore, the transmission medium can have 10^5 bits in-transit at any given point of time. Using this value we can estimate the full throughput of the transmission link.

1.d :

Consider the connectivity as shown below



The propagation delay remains same as the distance between A & B is constant. Transmission delay will be changed because the packets are being transferred through different devices.

The transmission delay will be equal to the sum of transmission time from A to R1, R1 to R2 and R2 to B.

$$\therefore d_{trans} = \frac{L}{R} + \frac{L}{R} + \frac{L}{R} = \frac{3L}{R} = 3 * 0.002 = 0.006\ sec$$

$$d_{prop} = 0.1\ sec$$
$$d_{ack} = 0.1\ sec$$

Total time taken to transfer the file = $1000 * (0.006 + 0.1 + 0.1)$ = 206 sec

Internet Pioneers

Paul Baran:

I've chosen Paul Baran as one of the best Internet Pioneers because of his enormous contribution towards the building a distributed network and packet switching data transmission technique. These two are the most required features of today's Internet.

There are two qualities of Baran that inspired me. Frist, he is from a middle class family. He struggled a lot and worked hard in order to complete his masters. He was a hard worker as he used to work in Hughes Aircraft Company and also he is used to teach night classes at UCLA. Second, Baran was self-confident person. All the ideas that he shared were look like utter nonsense to his colleagues. But he did not lose hope and he wrote a series of technical papers to answer criticisms of his ideas.

During the World War II, communication played most important role in deciding the winner. So US government planned to establish a communication network which could withstand the enemy attack. At this time, Baran came up with the theory of "Distributed Networks". He also proposed the ground-breaking idea "Packet Switching". Packet switching showed promising results at that time. Later Baran joined ARPANET project as an informal consultant.

Vint Cerf:

I've chosen Vint Cerf as one of the best Internet Pioneers because he derived the most important communication protocol for the Internet - TCP/IP Protocol. That's the reason, he earned the nickname of "father of the Internet".

Cerf was grew up in Los Angeles. As a child, his interest towards computers is what I liked from him. Though he majored in mathematics from Stanford, he continued to grow more interest in computing. Best lines told by him about computer programming - "You created your own universe and you were master of it. The computer would do anything you programmed it to do."

In early 1970s, the main obstacle of communication among computers of different networks through Interface Message Processors (IMPs). Cerf joined with other universities and formed a group called Network Working Group (NWG). NWG implemented the standards for communications called "Protocol". In 1970, the group released the universal protocol for host-to-host communication called Network Control Protocol (NCP). Later Cerf and Kahn published a paper on Transmission Control Protocol (TCP). TCP allowed networks to be joined into a network of networks. In 1976, Cerf joined ARPA. In 1978, Cerf with his colleagues, split TCP into two parts. They separated the part of TCP that is responsible for routing packages and formed a separate protocol called Internet Protocol (IP). The new protocol was called TCP/IP. This became the standard for today's internet communications

Computer Virus:

Computer virus is a malware program. When we executed this program, it replicates itself into other data files or boot sector of hard drive. Computer virus can steal the private user data or spamming the user's contacts or even it can render the computer useless.

E.g. Trojan, Netsky and Sasser

Malware:

Malware or malicious software is a software. When we run it, it has the ability used to disrupt computer operations or display unwanted advertising or gather sensitive information.

E.g. Adware, spyware, virus, worm etc.,

Worm:

A computer worm is a self-replicating computer program that penetrates an operating system with the intent of spreading malicious code. Worms utilize networks to send copies of the original code to other computers, causing harm by consuming bandwidth or possibly deleting files or sending documents via email.

E.g. Jerusalem, Storm Worm, ILOVEYOU etc..,

Spyware:

Spyware is a type of malware that is installed on a computer without the knowledge of the owner. Spyware collects the owner's private information and sends it to its author. Spyware can also install additional software.

E.g. Zango, Huntbar etc.

Trojan Horse:

Trojan Horse is a malicious computer program which misrepresents itself to appear useful in order to persuade a victim to install it.

Botnet:

Botnet is the generic name given to any collection of compromised PCs controlled by an attacker remotely. Botnets generally are created by a specific attacker or small group of attackers using one piece of malware to infect a large number of machines. The individual PCs that are part of a botnet often are called “bots” or “zombies” and there is no minimum size for a group of PCs to be called a botnet.

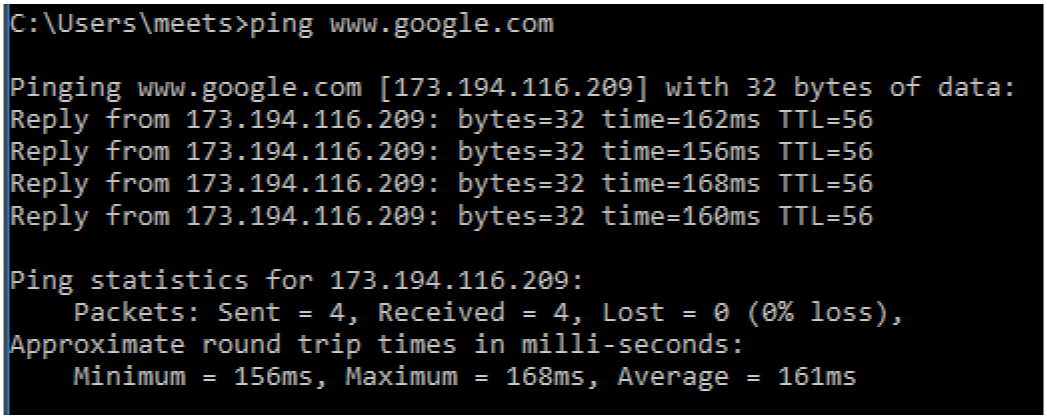
Ping:

The ping command is a Command Prompt command used to test the ability of the source computer to reach a specified destination computer. The ping command is usually used as a simple way verify that a computer can communicate over the network with another computer or network device.

Syntax:

```
ping <host address>
```

Pinging www.google.com

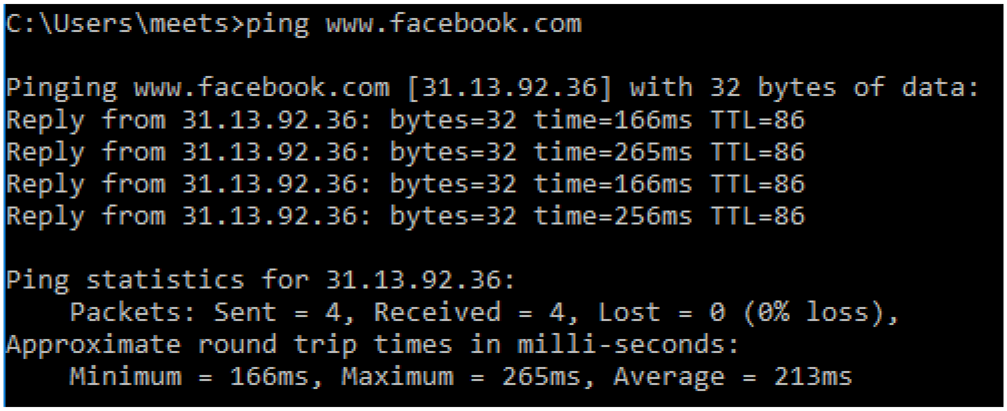


```
C:\Users\meets>ping www.google.com

Pinging www.google.com [173.194.116.209] with 32 bytes of data:
Reply from 173.194.116.209: bytes=32 time=162ms TTL=56
Reply from 173.194.116.209: bytes=32 time=156ms TTL=56
Reply from 173.194.116.209: bytes=32 time=168ms TTL=56
Reply from 173.194.116.209: bytes=32 time=160ms TTL=56

Ping statistics for 173.194.116.209:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 156ms, Maximum = 168ms, Average = 161ms
```

Pinging www.facebook.com

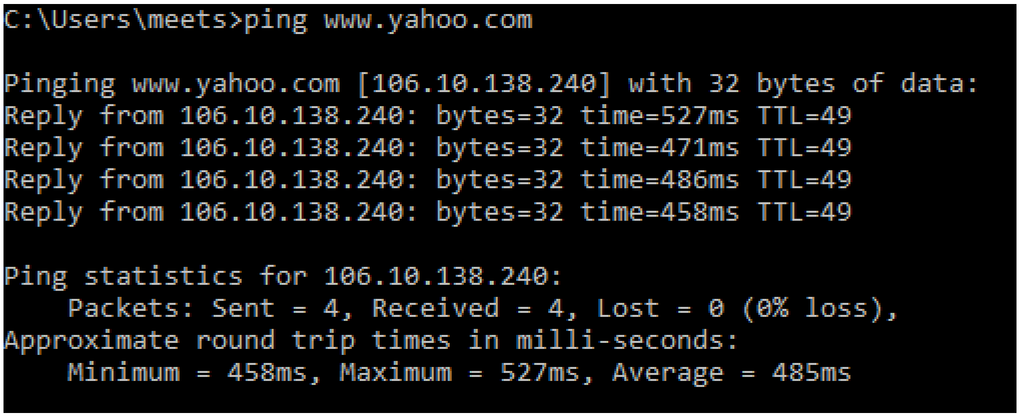


```
C:\Users\meets>ping www.facebook.com

Pinging www.facebook.com [31.13.92.36] with 32 bytes of data:
Reply from 31.13.92.36: bytes=32 time=166ms TTL=86
Reply from 31.13.92.36: bytes=32 time=265ms TTL=86
Reply from 31.13.92.36: bytes=32 time=166ms TTL=86
Reply from 31.13.92.36: bytes=32 time=256ms TTL=86

Ping statistics for 31.13.92.36:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 166ms, Maximum = 265ms, Average = 213ms
```

Pinging www.yahoo.com



```
C:\Users\meets>ping www.yahoo.com

Pinging www.yahoo.com [106.10.138.240] with 32 bytes of data:
Reply from 106.10.138.240: bytes=32 time=527ms TTL=49
Reply from 106.10.138.240: bytes=32 time=471ms TTL=49
Reply from 106.10.138.240: bytes=32 time=486ms TTL=49
Reply from 106.10.138.240: bytes=32 time=458ms TTL=49

Ping statistics for 106.10.138.240:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 458ms, Maximum = 527ms, Average = 485ms
```

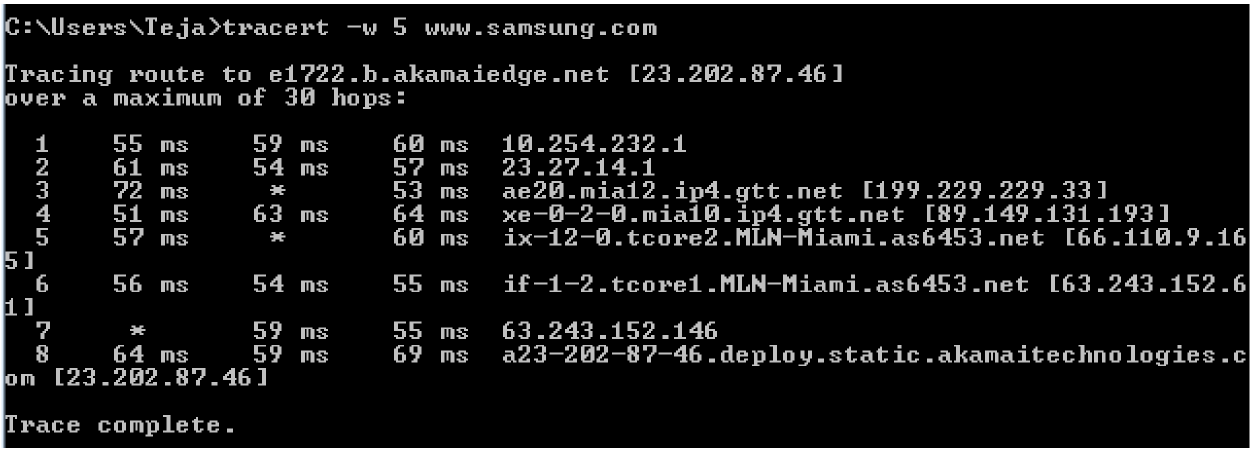
Traceroute:

Traceroute is a computer network diagnostic tool for displaying the path taken by the packets and calculating the transit delays of these packets across the IP network. Traceroute calculates RTTs for each successive host in the route, whereas on the other hand ping only computes only the final RTTs from destination. By default traceroute takes 30 hops (intermediate hosts).

Syntax:

```
tracert <host address>
```

Traceroute command outputs:

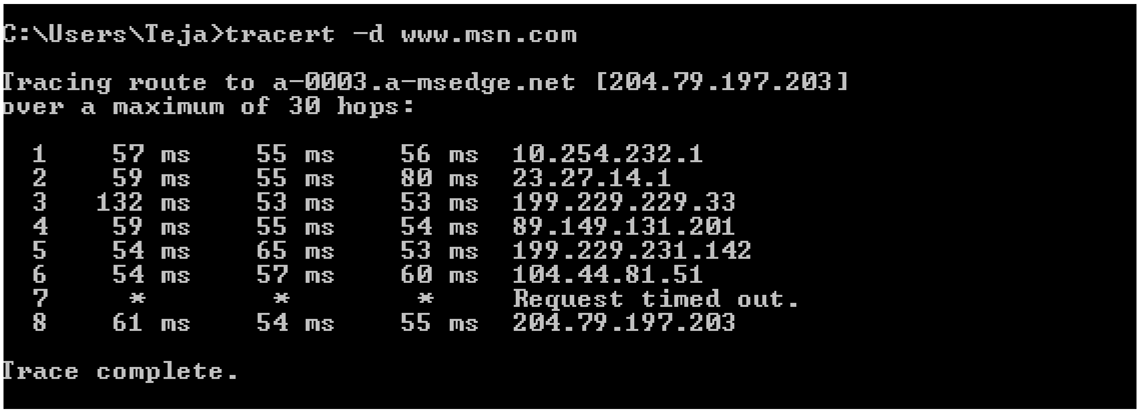


```
C:\Users\Teja>tracert -w 5 www.samsung.com

Tracing route to e1722.b.akamaiedge.net [23.202.87.46]
over a maximum of 30 hops:

  1    55 ms    59 ms    60 ms  10.254.232.1
  2    61 ms    54 ms    57 ms  23.27.14.1
  3    72 ms     *       53 ms  ae20.mia12.ip4.gtt.net [199.229.229.33]
  4    51 ms    63 ms    64 ms  xe-0-2-0.mia10.ip4.gtt.net [89.149.131.193]
  5    57 ms     *       60 ms  ix-12-0.tcore2.MLN-Miami.as6453.net [66.110.9.16
5]
  6    56 ms    54 ms    55 ms  if-1-2.tcore1.MLN-Miami.as6453.net [63.243.152.6
1]
  7     *       59 ms    55 ms  63.243.152.146
  8    64 ms    59 ms    69 ms  a23-202-87-46.deploy.static.akamaitechnologies.c
om [23.202.87.46]

Trace complete.
```

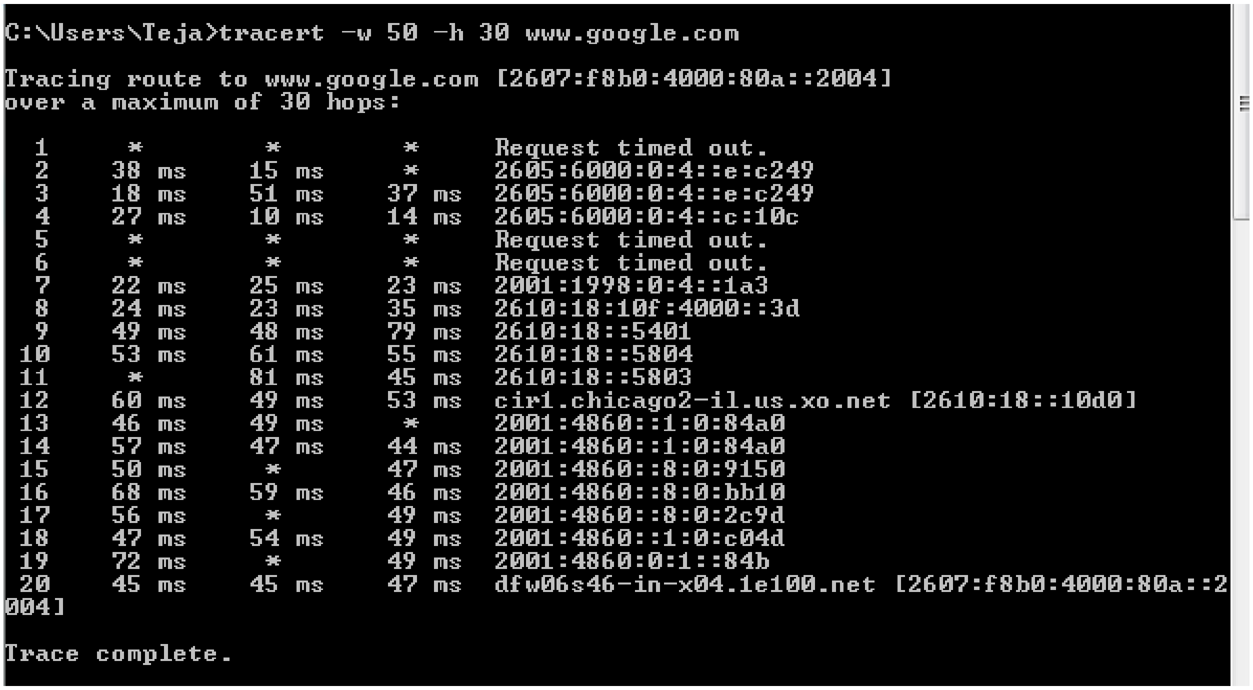


```
C:\Users\Teja>tracert -d www.msn.com

Tracing route to a-0003.a-msedge.net [204.79.197.203]
over a maximum of 30 hops:

  1    57 ms    55 ms    56 ms  10.254.232.1
  2    59 ms    55 ms    80 ms  23.27.14.1
  3   132 ms    53 ms    53 ms  199.229.229.33
  4    59 ms    55 ms    54 ms  89.149.131.201
  5    54 ms    65 ms    53 ms  199.229.231.142
  6    54 ms    57 ms    60 ms  104.44.81.51
  7     *        *        *     Request timed out.
  8    61 ms    54 ms    55 ms  204.79.197.203

Trace complete.
```



```
C:\Users\Teja>tracert -w 50 -h 30 www.google.com

Tracing route to www.google.com [2607:f8b0:4000:80a::2004]
over a maximum of 30 hops:

  1     *        *        *     Request timed out.
  2    38 ms    15 ms     *     2605:6000:0:4::e:c249
  3    18 ms    51 ms    37 ms  2605:6000:0:4::e:c249
  4    27 ms    10 ms    14 ms  2605:6000:0:4::c:10c
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7    22 ms    25 ms    23 ms  2001:1998:0:4::1a3
  8    24 ms    23 ms    35 ms  2610:18:10f:4000::3d
  9    49 ms    48 ms    79 ms  2610:18::5401
 10    53 ms    61 ms    55 ms  2610:18::5804
 11     *       81 ms    45 ms  2610:18::5803
 12    60 ms    49 ms    53 ms  cir1.chicago2-il.us.xo.net [2610:18::10d0]
 13    46 ms    49 ms     *     2001:4860::1:0:84a0
 14    57 ms    47 ms    44 ms  2001:4860::1:0:84a0
 15    50 ms     *       47 ms  2001:4860::8:0:9150
 16    68 ms    59 ms    46 ms  2001:4860::8:0:bb10
 17    56 ms     *       49 ms  2001:4860::8:0:2c9d
 18    47 ms    54 ms    49 ms  2001:4860::1:0:c04d
 19    72 ms     *       49 ms  2001:4860:0:1::84b
 20    45 ms    45 ms    47 ms  dfw06s46-in-x04.1e100.net [2607:f8b0:4000:80a::2
004]

Trace complete.
```

4. Explore ‘nslookup’ which is a program to query Internet domain name servers.Particularly,

a. find out the ip address(es) of www.yahoo.com

b. find out the name servers and their IP addresses of yahoo.com domain.

c. find out the email servers and their IP addresses of yahoo.com domain.

d. Try two other options (same server, different command parameters).

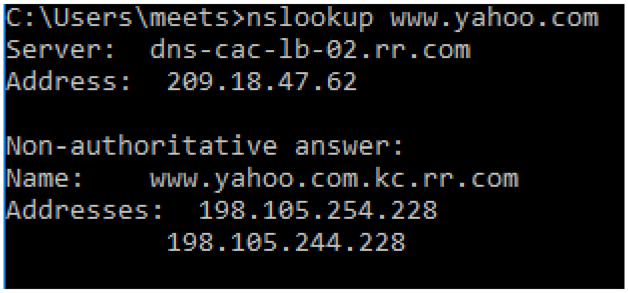
Record the commands and their output.

Solution:

nslookup is a network administration command-line tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or for any other specific DNS record.

The output of nslookup can have two types. Any answer that originates from the DNS Server which has the complete zone file information available for the domain is said to be authoritative answer. In many cases, DNS servers will not have the complete zone file information available for a given domain. Instead, it maintains a cache file which has the results of all queries performed in the past for which it has gotten authoritative response. When a DNS query is given, it searches the cache file, and return the information available as “Non-Authoritative Answer”.

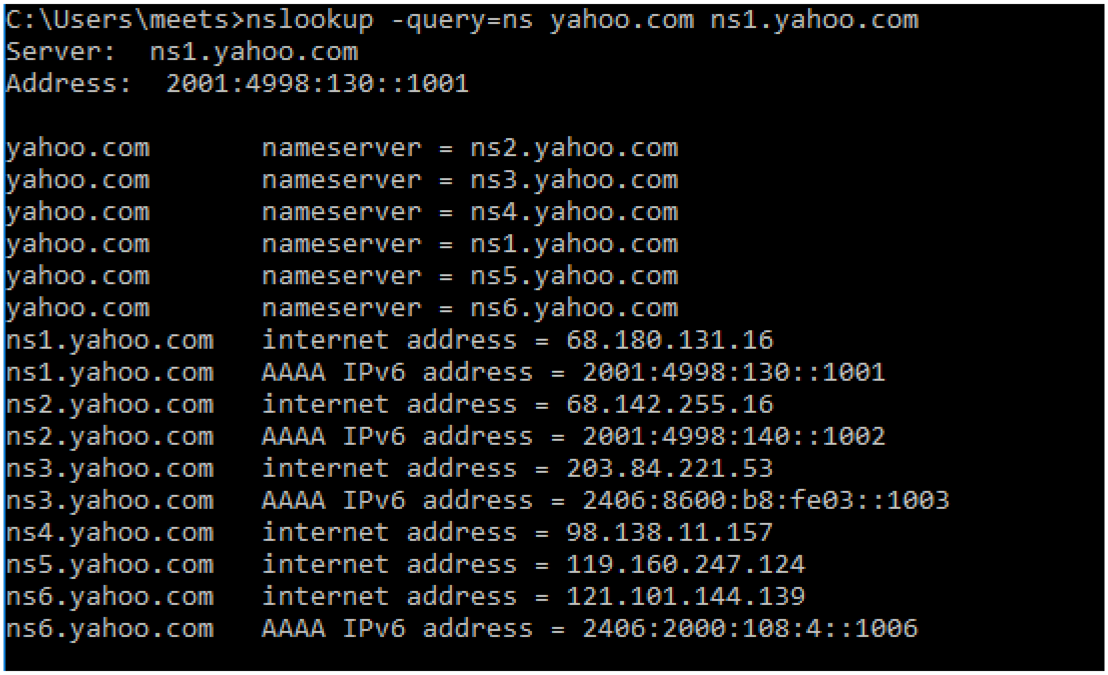
IP addresses of yahoo.com:



```
C:\Users\meets>nslookup www.yahoo.com
Server:  dns-cac-lb-02.rr.com
Address:  209.18.47.62

Non-authoritative answer:
Name:    www.yahoo.com.kc.rr.com
Addresses:  198.105.254.228
          198.105.244.228
```

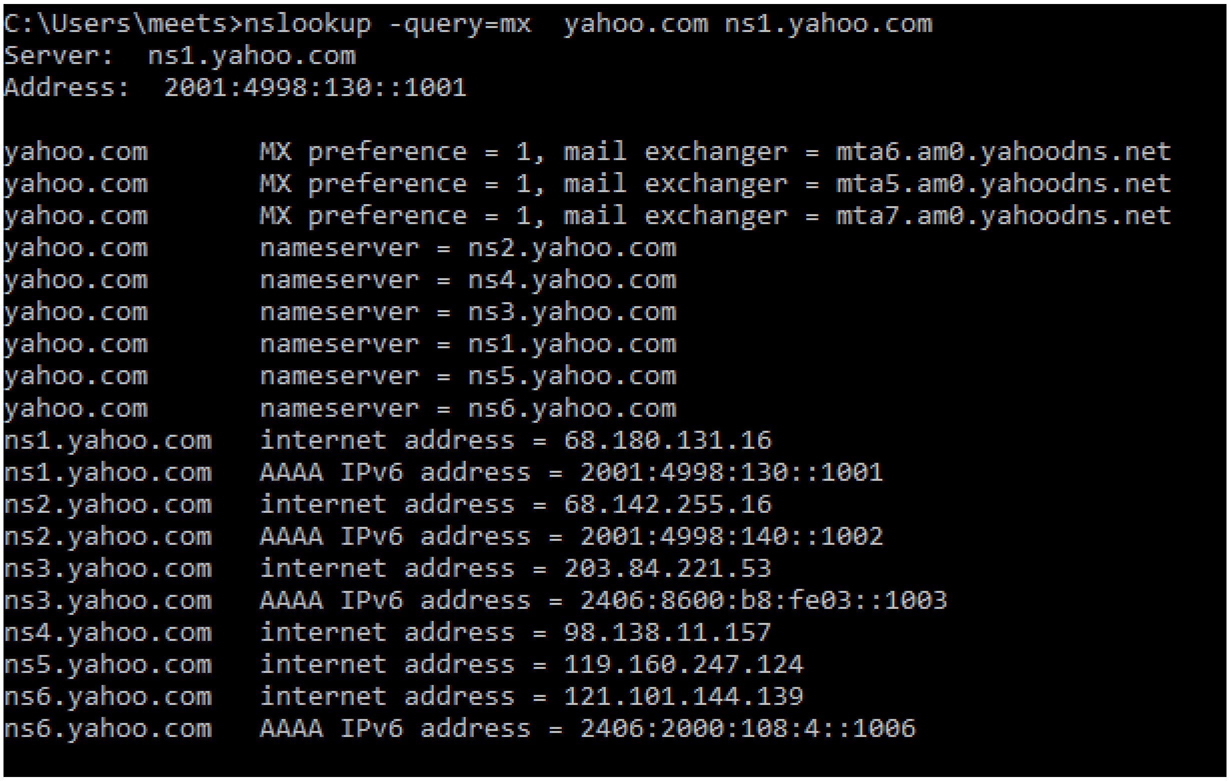
Name servers of yahoo.com:



```
C:\Users\meets>nslookup -query=ns yahoo.com ns1.yahoo.com
Server:  ns1.yahoo.com
Address:  2001:4998:130::1001

yahoo.com       nameserver = ns2.yahoo.com
yahoo.com       nameserver = ns3.yahoo.com
yahoo.com       nameserver = ns4.yahoo.com
yahoo.com       nameserver = ns1.yahoo.com
yahoo.com       nameserver = ns5.yahoo.com
yahoo.com       nameserver = ns6.yahoo.com
ns1.yahoo.com   internet address = 68.180.131.16
ns1.yahoo.com   AAAA IPv6 address = 2001:4998:130::1001
ns2.yahoo.com   internet address = 68.142.255.16
ns2.yahoo.com   AAAA IPv6 address = 2001:4998:140::1002
ns3.yahoo.com   internet address = 203.84.221.53
ns3.yahoo.com   AAAA IPv6 address = 2406:8600:b8:fe03::1003
ns4.yahoo.com   internet address = 98.138.11.157
ns5.yahoo.com   internet address = 119.160.247.124
ns6.yahoo.com   internet address = 121.101.144.139
ns6.yahoo.com   AAAA IPv6 address = 2406:2000:108:4::1006
```

Mail servers of yahoo.com:

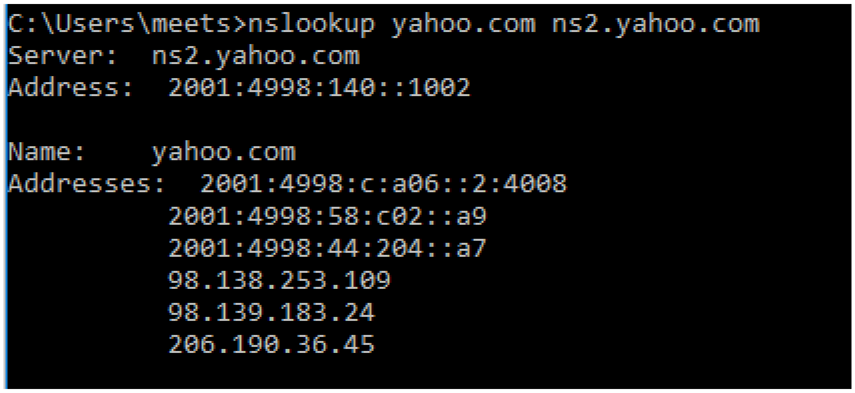


```
C:\Users\meets>nslookup -query=mx  yahoo.com ns1.yahoo.com
Server:  ns1.yahoo.com
Address:  2001:4998:130::1001

yahoo.com       MX preference = 1, mail exchanger = mta6.am0.yahoodns.net
yahoo.com       MX preference = 1, mail exchanger = mta5.am0.yahoodns.net
yahoo.com       MX preference = 1, mail exchanger = mta7.am0.yahoodns.net
yahoo.com       nameserver = ns2.yahoo.com
yahoo.com       nameserver = ns4.yahoo.com
yahoo.com       nameserver = ns3.yahoo.com
yahoo.com       nameserver = ns1.yahoo.com
yahoo.com       nameserver = ns5.yahoo.com
yahoo.com       nameserver = ns6.yahoo.com
ns1.yahoo.com   internet address = 68.180.131.16
ns1.yahoo.com   AAAA IPv6 address = 2001:4998:130::1001
ns2.yahoo.com   internet address = 68.142.255.16
ns2.yahoo.com   AAAA IPv6 address = 2001:4998:140::1002
ns3.yahoo.com   internet address = 203.84.221.53
ns3.yahoo.com   AAAA IPv6 address = 2406:8600:b8:fe03::1003
ns4.yahoo.com   internet address = 98.138.11.157
ns5.yahoo.com   internet address = 119.160.247.124
ns6.yahoo.com   internet address = 121.101.144.139
ns6.yahoo.com   AAAA IPv6 address = 2406:2000:108:4::1006
```

Other command options:

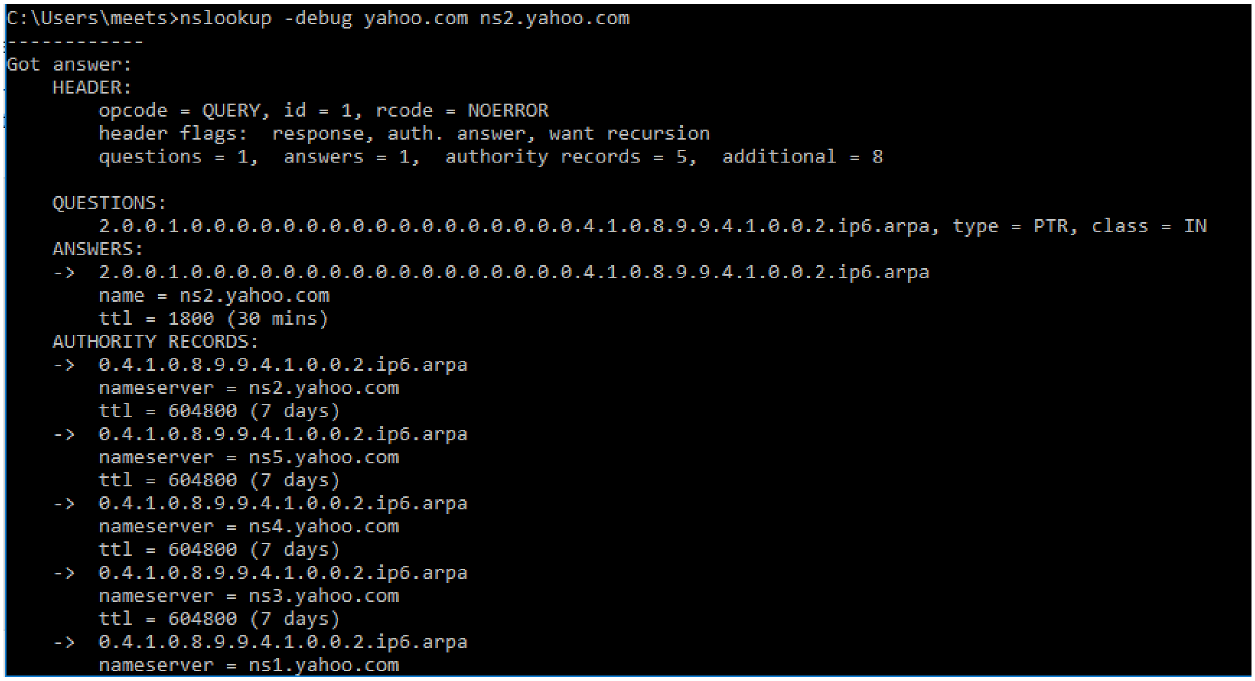
We can specify the domain name server to lookup for the website details instead of using the default DNS server using the below command. This will give us authoritative details instead of cached details.



```
C:\Users\meets>nslookup yahoo.com ns2.yahoo.com
Server:  ns2.yahoo.com
Address:  2001:4998:140::1002

Name:    yahoo.com
Addresses:  2001:4998:c:a06::2:4008
          2001:4998:58:c02::a9
          2001:4998:44:204::a7
          98.138.253.109
          98.139.183.24
          206.190.36.45
```

nslookup comes with debug option. We can use this option to get a keen observation on what’s going on in the background.



```
C:\Users\meets>nslookup -debug yahoo.com ns2.yahoo.com
------------
Got answer:
    HEADER:
        opcode = QUERY, id = 1, rcode = NOERROR
        header flags:  response, auth. answer, want recursion
        questions = 1,  answers = 1,  authority records = 5,  additional = 8

    QUESTIONS:
        2.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.4.1.0.8.9.9.4.1.0.0.2.ip6.arpa, type = PTR, class = IN
    ANSWERS:
    ->  2.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.4.1.0.8.9.9.4.1.0.0.2.ip6.arpa
        name = ns2.yahoo.com
        ttl = 1800 (30 mins)
    AUTHORITY RECORDS:
    ->  0.4.1.0.8.9.9.4.1.0.0.2.ip6.arpa
        nameserver = ns2.yahoo.com
        ttl = 604800 (7 days)
    ->  0.4.1.0.8.9.9.4.1.0.0.2.ip6.arpa
        nameserver = ns5.yahoo.com
        ttl = 604800 (7 days)
    ->  0.4.1.0.8.9.9.4.1.0.0.2.ip6.arpa
        nameserver = ns4.yahoo.com
        ttl = 604800 (7 days)
    ->  0.4.1.0.8.9.9.4.1.0.0.2.ip6.arpa
        nameserver = ns3.yahoo.com
        ttl = 604800 (7 days)
    ->  0.4.1.0.8.9.9.4.1.0.0.2.ip6.arpa
        nameserver = ns1.yahoo.com
```

International Engineering Task Force (IETF):

The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet.

Requests for Comments (RFC) document series contain technical and organizational notes about the Internet. They cover many aspects of computer networking, including protocols, procedures, programs, and concepts, as well as meeting notes, opinions. There are a total of 7771 RFCs till date.

IETF **Working Groups (WGs)** are the primary mechanism for development of IETF specifications and guidelines, many of which are intended to be standards or recommendations. Working Groups are typically created to address a specific problem or to produce one or more specific deliverables (a guideline, standards specification, etc.). Working Groups are generally expected to be short-lived in nature. Upon completion of its goals and achievement of its objectives, the Working Group is terminated. Each Working Group has a charter. WG charters state the scope of work for group, and lay out goals and milestones that show how this work will be completed.

Some of the work groups are…

GeoJSON: GeoJSON is a geospatial data interchange format based on JavaScript Object otation (JSON). It defines several types of JSON objects and the manner in which they are combined to represent data about geographic features, their properties, and their spatial extents.

WebPush: A message encryption scheme is described for the Web Push protocol. This scheme provides confidentiality and integrity for messages sent from an Application Server to a User Agent.

CalExt: This document specifies a new iCalendar calendar component that allows the publication of available and unavailable time periods associated with a calendar user. This component can be used in standard iCalendar free-busy lookups, including iTIP free-busy requests, to generate repeating blocks of available or busy time with exceptions as needed.

P2PSIP: This document defines concepts and terminology for the use of the Session Initiation Protocol in a peer-to-peer environment where the traditional proxy-registrar and message routing functions are replaced by a distributed mechanism.

RTCWeb: Application Layer Protocol Negotiation (ALPN) labels are defined for use in identifying Web Real-Time Communications (WebRTC) usages of Datagram Transport Layer Security (DTLS). Labels are provided for identifying a session that uses a combination of WebRTC compatible media and data, and for identifying a session requiring confidentiality protection.

WebPush:

A message encryption scheme is described for the Web Push protocol. This scheme provides confidentiality and integrity for messages sent from an Application Server to a User Agent. The Web Push protocol is an intermediated protocol by necessity. Messages from an Application Server are delivered to a User Agent via a Push Service.

Objective: Main goal of this protocol is to secure the messages against inspection or modification by a Push Service. Multiple users of Web Push often share a central agent that aggregates push functionality. This agent can enforce the use of this encryption scheme by applications that use push messaging. An agent that only delivers messages that are properly encrypted strongly encourages the end-to-end protection of messages.

A web browser that implements the Web Push API can enforce the use of encryption by forwarding only those messages that were properly encrypted.

Key Generation:

For each new subscription that the User Agent generates for an application, it also generates an asymmetric key pair for use in Diffie-Hellman (DH) [DH] or elliptic-curve Diffie-Hellman (ECDH) [ECDH]. The public key for this key pair can then be distributed by the application to the Application Server along with the URI of the subscription. The private key MUST remain secret.

A User Agent MUST generate and provide a public key for the scheme. The public key MUST be accompanied by a key identifier that can be used in the "keyid" parameter to identify which key is in use. Key identifiers need only be unique within the context of a subscription.

Key Distribution:

The application using the subscription distributes the key identifier and public key along with other subscription information, such as the subscription URI and expiration time. The communication medium by which an application distributes the key identifier and public key MUST be confidentiality protected. Most applications that use push messaging have a pre-existing relationship with an Application Server.

Message Encryption:

An Application Server that has the key identifier, public key, group and format information can encrypt a message for the User Agent. The Application Server generates a new DH or ECDH key pair in the same group as the value generated by the User Agent. From the newly generated key pair, the Application Server performs a DH or ECDH computation with the public key provided by the User Agent to find the shared secret.

The Application Server then encrypts the payload. Header fields are populated with URL-safe base-64 encoded values:

- the salt is added to the "salt" parameter of the Encryption header field; and
- the public key for its DH or ECDH key pair is placed in the "dh" parameter of the Encryption-Key header field.

Message Decryption:

A User Agent decrypts messages. The value of the "keyid" parameter is used to identify the correct key pair, if there is more than one possible value for the corresponding subscription.

References:

[DH] Diffie, W. and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, V.IT-22 n.6 , June 1977.

[ECDH] SECG, "Elliptic Curve Cryptography", SEC 1 , 2000, .

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/ RFC2119, March 1997, .