Homework 2

Suyi Liu

February 12 2017

1 R1

Web applications: use HTTP
Mobile applications: use HTTP
File transfer applications: use FTP
Email applications: use SMTP
Remote accessing applications: use Telnet

2 R4

I do not agree. Because from page 89 of textbook, it says ”In the context of a communication session between a pair of processes, the process that initiates the communication (that is, initially contacts the other process at the beginning of the session) is labeled as the client. The process that waits to be contacted to begin the session is the server.”
So in this case, of P2P file-sharing application, we can label the user requesting for file as client and another host sending the file as server.

3 R5

They use IP address to identify the host the other process is residing at, and a port number to identify the process. They communicate through sockets by choosing from TCP or UDP transport layer protocols.

4 R7

An important instant messaging application that involves real time decision based on data received from other side. For example, a real time rocket launching messaging system cannot tolerate any data loss(needs to make calculations based on accurate data) and is time sensitive to allow scientists and engineers to communicate in real time.

5 R9

SSL operates at the application layer.
When an application uses SSL, the sending process passes cleartext data to the SSL socket; SSL in the sending host then encrypts the data and passes the encrypted data to the TCP socket.
If an application developer wants to use the services of SSL, the application needs to include SSL code (existing, highly optimized libraries and classes) in both the client and server sides of the application. (According to textbook page 94)

6 R15

Because FTP uses a separate control connection, FTP is said to send its control information out-of-band. More specifically, FTP uses two parallel TCP connections to transfer a file, a control connection(sending control information between the two hosts) and a data connection(actually send a file).

7 R20

From jhu.edu: I can see the IP address of the sender host by examining the message details.
From Gmail: I can see the IP address of the sender host by examining the message details.

8 R21

It is not necessary for Bob to return the favor and provide chunks to Alice in this same interval. Because if Bob is not satisfied with the data rate, he can choose not to send chunks back to Alice. (Alice has to be in the top 4 uploaders of Bob for Bob to send out chunks to her).

9 R23

An overlay network is the network in which the peers form an abstract logical network which resides above the “underlay” computer network consisting of physical links, routers, and hosts. The links are simply virtual liaisons between pairs of peers.
It does not include routers.
Edges in the overlay network are virtual links connecting two nodes.

10 R27

For the client-server application over TCP: TCP is a connection oriented protocol. The server has to be executed first so that the client can successfully connect to the server at the time it needs to make such connection.
For the client-server application over UDP: UDP is not a connection oriented protocol. Client does not need to make connection with the server, so that server does not need to be executed before the client does.

11 P1

(a) False. The client sends four requests and receives four responses.

(b) True. They can be sent over the same persistent connection since the two files are from the same server.

(c) False. In non-persistent connections, new segment is created every time there is a new request.

(d) False. Header line indicates the time and date when the HTTP response was created and sent by the server. It is the time when the server retrieves the object from its file system, inserts the object into the response message, and sends the response message.

(e) False. For example, when the server failed to find the requested object, the message body in the response is empty.

12 P2

USER NAME (USER), PASSWORD (PASS), ACCOUNT (ACCT), CHANGE WORKING DIRECTORY (CWD), CHANGE TO PARENT DIRECTORY (CDUP), STRUCTURE MOUNT (SMNT), REINITIALIZE (REIN), LOGOUT (QUIT), DATA PORT (PORT), PASSIVE (PASV), REPRESENTATION TYPE (TYPE), FILE STRUCTURE (STRU), TRANSFER MODE (MODE), RETRIEVE (RETR), STORE (STOR), STORE UNIQUE (STOU), APPEND (with create) (APPE), ALLOCATE (ALLO), RESTART (REST), RENAME FROM (RNFR), RENAME TO (RNTO), ABORT (ABOR), DELETE (DELE), REMOVE DIRECTORY (RMD), MAKE DIRECTORY (MKD), PRINT WORKING DIRECTORY (PWD), LIST (LIST), NAME LIST (NLST), SITE PARAMETERS (SITE), SYSTEM (SYST), STATUS (STAT), HELP (HELP), NOOP (NOOP)

13 P6

(a) From In HTTP/1.1, connections are the default behavior of any HTTP connection. If the server or client chooses to close the connection, it SHOULD send a Connection header including the connection-token close.(Connection: close)

(8.1.2 and 8.1.2.1)
The client, the server, or both can signal the close of a connection. And if either the client or the server sends the close token in the Connection header, that request becomes the last one for the connection.

(b) HTTP/1.1 does not provide any encryption services.

(c) A client cannot open three or more simultaneous connections with a given server.
(From RFC2616) Clients that use persistent connections SHOULD limit the number of simultaneous connections that they maintain to a given server. A single-user client SHOULD NOT maintain more than 2 connections with any server or proxy.

(d) Yes. A client, server, or proxy MAY close the transport connection at any time. For example, a client might have started to send a new request at the same time that the server has decided to close the "idle" connection. From the server's point of view, the connection is being closed while it was idle, but from the client's point of view, a request is in progress. (From RFC2616)

14 P14

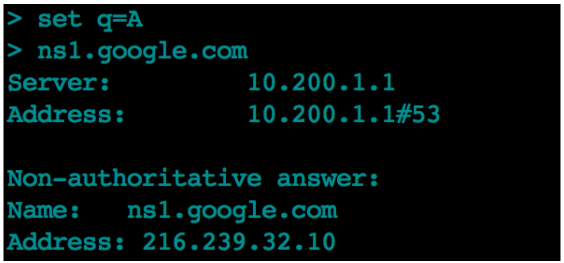
SMTP: uses a line containing only a period to mark the end of a message body.
HTTP: uses "Content-Length header field" to indicate the length of a message body so that receiver knows when it ends.
HTTP cannot use the same method as SMTP to mark the end of a message body because the HTTP message is in binary data while SMTP message is in 7 bit ASCII format, HTTP cannot use a special mark to indicate the end of a message body.

15 P18

(a) A whois database is a query and response protocol that is widely used for querying databases that store the registered users or assignees of an Internet resource, such as a domain name, an IP address block, or an autonomous system, but is also used for a wider range of other information. (From Wikipedia)

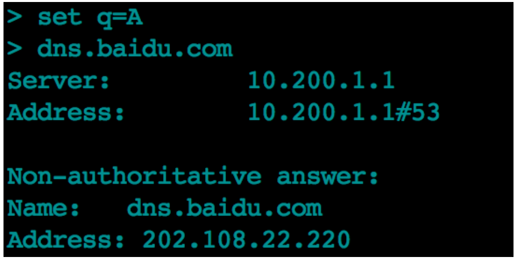
(b) I used who.is and found the name of a DNS server: dns.baidu.com
I used www.whois.net and found the name of another DNS server: ns1.google.com

(c) My dns server: 10.200.1.1 using cat /etc/resolv.conf(whois.arin.net)
What I found(screenshots):



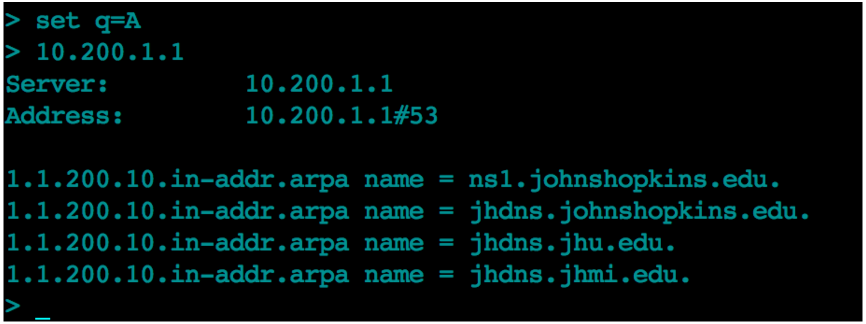
```
> set q=A
> ns1.google.com
Server:         10.200.1.1
Address:        10.200.1.1#53

Non-authoritative answer:
Name:   ns1.google.com
Address: 216.239.32.10
```



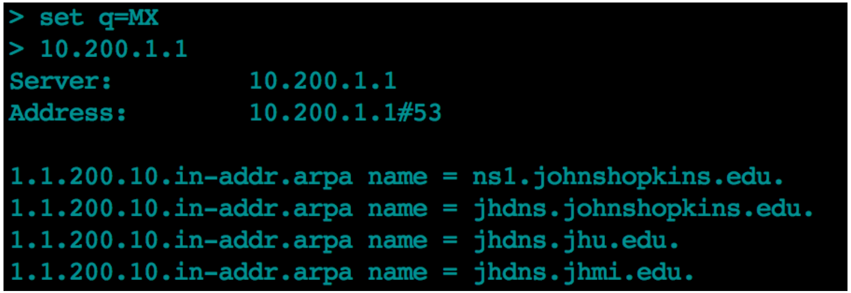
```
> set q=A
> dns.baidu.com
Server:         10.200.1.1
Address:        10.200.1.1#53

Non-authoritative answer:
Name:   dns.baidu.com
Address: 202.108.22.220
```



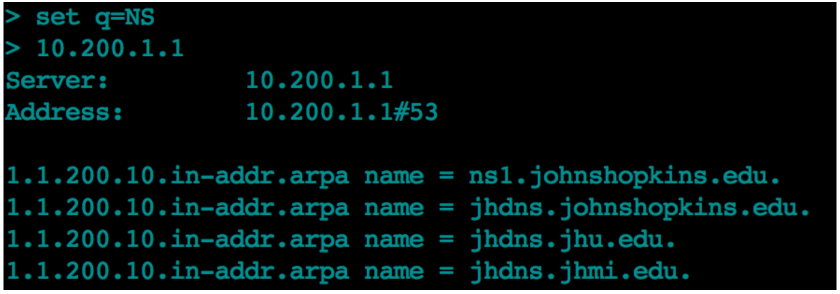
```
> set q=A
> 10.200.1.1
Server:         10.200.1.1
Address:        10.200.1.1#53

1.1.200.10.in-addr.arpa name = ns1.johnshopkins.edu.
1.1.200.10.in-addr.arpa name = jhdns.johnshopkins.edu.
1.1.200.10.in-addr.arpa name = jhdns.jhu.edu.
1.1.200.10.in-addr.arpa name = jhdns.jhmi.edu.
> _
```



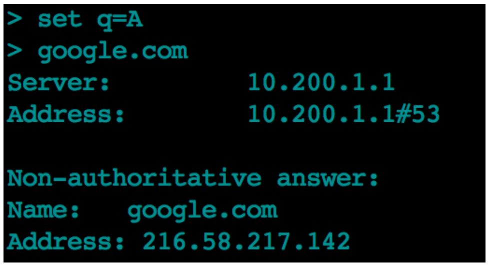
```
> set q=MX
> 10.200.1.1
Server:         10.200.1.1
Address:        10.200.1.1#53

1.1.200.10.in-addr.arpa name = ns1.johnshopkins.edu.
1.1.200.10.in-addr.arpa name = jhdns.johnshopkins.edu.
1.1.200.10.in-addr.arpa name = jhdns.jhu.edu.
1.1.200.10.in-addr.arpa name = jhdns.jhmi.edu.
```



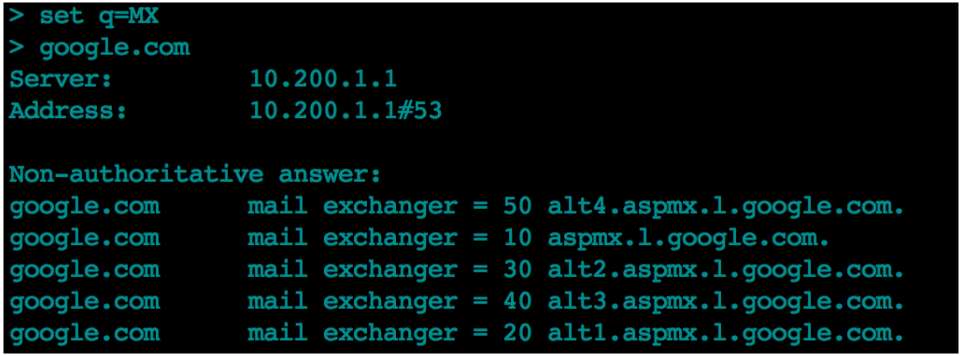
```
> set q=NS
> 10.200.1.1
Server:         10.200.1.1
Address:        10.200.1.1#53

1.1.200.10.in-addr.arpa name = ns1.johnshopkins.edu.
1.1.200.10.in-addr.arpa name = jhdns.johnshopkins.edu.
1.1.200.10.in-addr.arpa name = jhdns.jhu.edu.
1.1.200.10.in-addr.arpa name = jhdns.jhmi.edu.
```



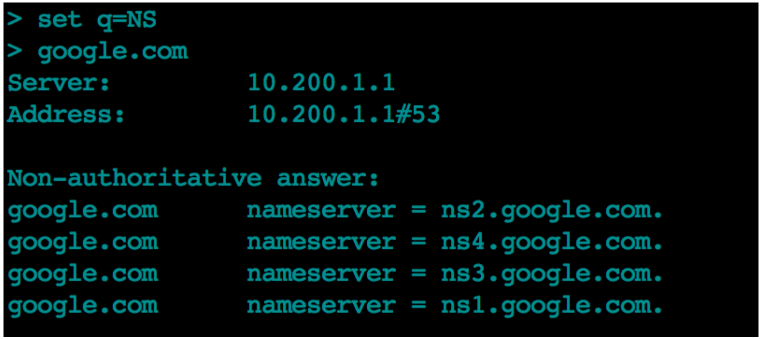
```
> set q=A
> google.com
Server:         10.200.1.1
Address:        10.200.1.1#53

Non-authoritative answer:
Name:   google.com
Address: 216.58.217.142
```



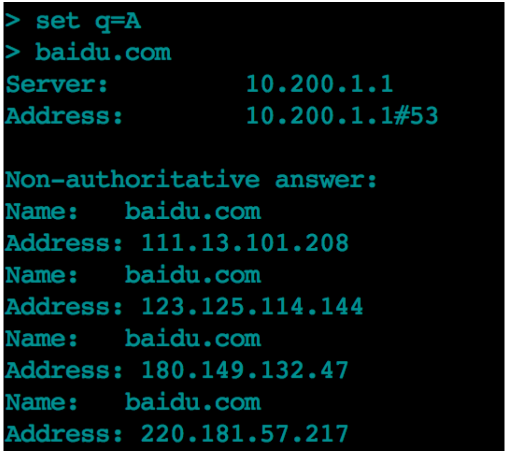
```
> set q=MX
> google.com
Server:         10.200.1.1
Address:        10.200.1.1#53

Non-authoritative answer:
google.com      mail exchanger = 50 alt4.aspmx.l.google.com.
google.com      mail exchanger = 10 aspmx.l.google.com.
google.com      mail exchanger = 30 alt2.aspmx.l.google.com.
google.com      mail exchanger = 40 alt3.aspmx.l.google.com.
google.com      mail exchanger = 20 alt1.aspmx.l.google.com.
```



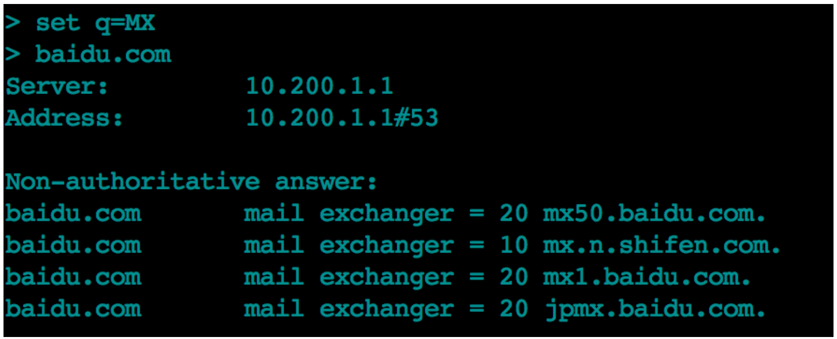
```
> set q=NS
> google.com
Server:         10.200.1.1
Address:        10.200.1.1#53

Non-authoritative answer:
google.com      nameserver = ns2.google.com.
google.com      nameserver = ns4.google.com.
google.com      nameserver = ns3.google.com.
google.com      nameserver = ns1.google.com.
```



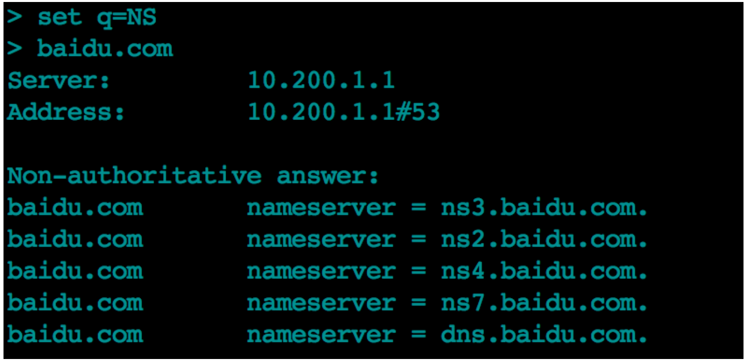
```
> set q=A
> baidu.com
Server:         10.200.1.1
Address:        10.200.1.1#53

Non-authoritative answer:
Name:   baidu.com
Address: 111.13.101.208
Name:   baidu.com
Address: 123.125.114.144
Name:   baidu.com
Address: 180.149.132.47
Name:   baidu.com
Address: 220.181.57.217
```



```
> set q=MX
> baidu.com
Server:         10.200.1.1
Address:        10.200.1.1#53

Non-authoritative answer:
baidu.com       mail exchanger = 20 mx50.baidu.com.
baidu.com       mail exchanger = 10 mx.n.shifen.com.
baidu.com       mail exchanger = 20 mx1.baidu.com.
baidu.com       mail exchanger = 20 jpmx.baidu.com.
```

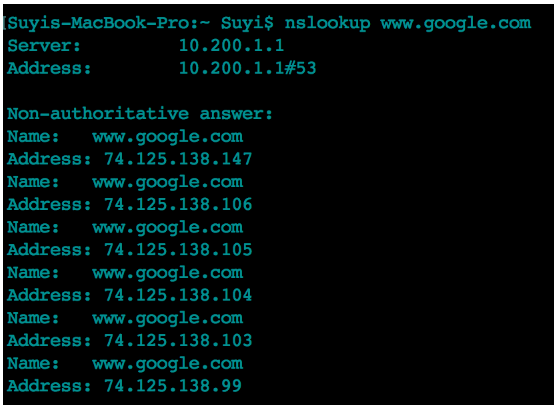


```
> set q=NS
> baidu.com
Server:         10.200.1.1
Address:        10.200.1.1#53

Non-authoritative answer:
baidu.com       nameserver = ns3.baidu.com.
baidu.com       nameserver = ns2.baidu.com.
baidu.com       nameserver = ns4.baidu.com.
baidu.com       nameserver = ns7.baidu.com.
baidu.com       nameserver = dns.baidu.com.
```

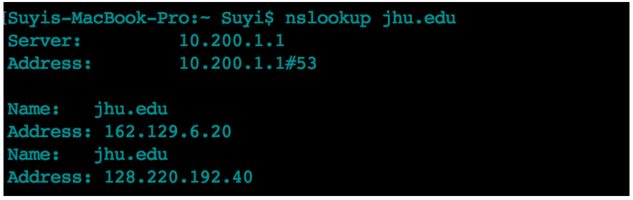
I found the record confirms what is said in the textbook:
If type is A: then Name is a hostname and Value is the IP address for the hostname. (I got those dns servers' IP address 216.239.32.10 and 202.108.22.220 respectively)
If type is NS: then Name is a domain (such as foo.com) and Value is the hostname of an authoritative DNS server that knows how to obtain the IP addresses for hosts in the domain. Using q=NS, I obtained what I got from question (b) again
If type is MX: then Value is the canonical name of a mail server that has an alias hostnameName.Asanexample. I my case, I obtained google and baidu's mail servers
So in conclusion, both big servers such as baidu.com and google.com have multiple ip addresses and dns servers, and provide mail services

(d) www.google.com has multiple IP addresses(nslookup www.google.com)
My school(jhu.edu) also has multiple ip addresses(nslookup www.jhu.edu)
Screenshots:



```
[Suyis-MacBook-Pro:~ Suyi$ nslookup www.google.com
Server:         10.200.1.1
Address:        10.200.1.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 74.125.138.147
Name:   www.google.com
Address: 74.125.138.106
Name:   www.google.com
Address: 74.125.138.105
Name:   www.google.com
Address: 74.125.138.104
Name:   www.google.com
Address: 74.125.138.103
Name:   www.google.com
Address: 74.125.138.99
```



```
[Suyis-MacBook-Pro:~ Suyi$ nslookup jhu.edu
Server:         10.200.1.1
Address:        10.200.1.1#53

Name:   jhu.edu
Address: 162.129.6.20
Name:   jhu.edu
Address: 128.220.192.40
```

(e) By looking up jhu according to ARIN whois, I found the range is:
128.220.0.0 - 128.220.255.255
192.12.13.0 - 192.12.13.255
192.12.14.0 - 192.12.14.255

(f) Using whois database and nslookup, the attacker can get the dns address, and ip addresses associated to the institution before the attack, and perform DDoS(for example) attack using the IP addresses obtained.

(g) Who is databases should be publicly available because it serves as a good tool for determining if the address of the website we visit is real. Also it helps people, for example developers, to lookup IP address and domain name respectively as information of registration of an internet resource.

16 P21

Yes. In a query chain, when a DNS server receives a DNS reply (containing, for example, a mapping from a host- name to an IP address), it can cache the mapping in its local memory. (From textbook page 139)
So the ordinary user can just query the same web site again, if it results in no query time, it means that the web site was accessed from a computer in the department few seconds ago.
Because of caching, the local DNS server will be able to immediately return the IP address of such web site to this second requesting host without having to

query any other DNS servers. (From textbook page 139)

17 P25

Since overlay network does not include routers. So there are N nodes and $\frac{N(N-1)}{2}$ edges since each pair of peers have an active TCP connection.

18 P26

(a) Yes it is possible. Because Bob can always receive file through optimistic unchoking by other peers if there are sufficient peers staying in that swarm for enough time.

(b) He can run clients on different machines in his lab, and also do free-riding, through optimistic unchoking. He can let each client ask for different parts of the file, and combine them into a whole when receiving finishes.

19 P27

peer 3 doesn't change its first successor since peer 4 hasn't left. So its new first successor is still peer 4.
For the new second successor of peer 3, peer 3 asks peer 4 for its new immediate successor for identifier and IP address.(Which in peer 8) Then peer 3 updates its new second successor as peer 8 according to that information.

20 P34

Advantage:
Some application layer protocols(such as HTTP, TELNET) do not restrict boundaries, so a stream of data is more suitable. Also the sequence of data bytes are ordered by sequence id associated with each byte.
Disadvantage:
It's hard for byte-oriented API to distinguish the end of previous message(which is also the start of the new message). Also such protocol is less efficient as they take more time to transmit and process byte one by one.