LEBANESE AMERICAN UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS
CSC430 – COMPUTER NETWORKS
FALL 25-26

PROBLEM SET 2

Problem 1

Consider the following string of ASCII characters that were captured by Wireshark when the browser sent an HTTP GET message (i.e., this is the actual content of an HTTP GET message). The characters *<cr><lf>* are carriage return and line-feed characters (that is, the italized character string *<cr>* in the text below represents the single carriage-return character that was contained at that point in the HTTP header). Answer the following questions, indicating where in the HTTP GET message below you find the answer.

```
GET /cs453/index.html HTTP/1.1<cr><lf>Host: gai
a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (
Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec
ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ex
t/xml, application/xml, application/xhtml+xml, text
/html;q=0.9, text/plain;q=0.8,image/png,*/*;q=0.5
<cr><lf>Accept-Language: en-us,en;q=0.5<cr><lf>Accept-
Encoding: zip,deflate<cr><lf>Accept-Charset: ISO
-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr>
<lf>Connection:keep-alive<cr><lf><cr><lf>
```

a. What is the URL of the document requested by the browser?
   The document request was http://gaia.cs.umass.edu/cs453/index.html. The Host : field indicates the server's name and /cs453/index.html indicates the file name.
b. What version of HTTP is the browser running?
   The browser is running HTTP version 1.1, as indicated just before the first pair
c. Does the browser request a non-persistent or a persistent connection?
   The browser is requesting a persistent connection, as indicated by the Connection: keep-alive
d. What is the IP address of the host on which the browser is running?
   This is a trick question. This information is not contained in an HTTP message anywhere. So there is no way to tell this from looking at the exchange of HTTP messages alone. One would need information from the IP datagrams (that carried the TCP segment that carried the HTTP GET request) to answer this question.
e. What type of browser initiates this message? Why is the browser type needed in an HTTP request message?
   Mozilla/5.0. The browser type information is needed by the server to send different versions of the same object to different types of browsers

Problem 2

The text below shows the reply sent from the server in response to the HTTP GET message in the pervious Problem. Answer the following questions, indicating where in the message below you find the answer.

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)
<cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46
GMT<cr><lf>ETag: "526c3-f22-a88a4c80"<cr><lf>Accept-
Ranges: bytes<cr><lf>Content-Length: 3874<cr><lf>
Keep-Alive: timeout=max=100<cr><lf>Connection:
Keep-Alive<cr><lf>Content-Type: text/html; charset=
ISO-8859-1<cr><lf><cr><lf><!doctype html public "-
//w3c//dtd html 4.0 transitional//en"><lf><html><lf>
<head><lf> <meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"><lf> <meta
name="GENERATOR" content="Mozilla/4.79 [en] (Windows NT
5.0; U) Netscape]"><lf> <title>CMPSCI 453 / 591 /
NTU-ST550A Spring 2005 homepage</title><lf></head><lf>
<much more document text following here (not shown)>
```

a. Was the server able to successfully find the document or not? What time was the document reply provided?
   The status code of 200 and the phrase OK indicate that the server was able to locate the document successfully. The reply was provided on Tuesday, 07 Mar 2008 12:39:45 Greenwich Mean Time.
b. When was the document last modified?
   The document index.html was last modified on Saturday 10 Dec 2005 18:27:46 GMT
c. How many bytes are there in the document being returned?
   There are 3874 bytes in the document being returned
d. What are the first 5 bytes of the document being returned? Did the server agree to a persistent connection?
   The first five bytes of the returned document are : <!doc. The server agreed to a persistent connection, as indicated by the Connection: Keep-Alive field

Problem 3

Suppose a client wants to download from a web server a web page that includes links to 5 images to be displayed on the page. If the roundtrip time is 5 milliseconds and the transmission delays of the web page and the 5 images are equal to 7, 12, 18, 23, 33, and 37 milliseconds respectively.

a. How long (in milliseconds) will it take the client to fully display the page when it is using non-persistent HTTP connection?
b. What if it is using a persistent HTTP connection?

a) For non-persistent HTTP, we would need 2RTT per object, in addition to each objects transmission delay.
   Objects = web page + 5 images = 6 total objects
   Time = 2N*(RTT) + transmission delays = 2*6*5ms + 7ms + 12ms + 18ms + 23ms + 33ms + 37ms
   Time = 190 ms
b) For persistent HTTP, connection remains open.
   Time = (N+1)*RTT + transmission delays= 7*5ms + 7ms + 12ms + 18ms + 23ms + 33ms + 37ms
   Time = 165 ms

Problem 4

Suppose a server has been contacted by 5 hosts (clients) to send them a file *F* of size 1500 B

- Suppose the server has an upload rate u_s=16Kbps and a download rate of d_s=8Kbps.
- The upload rates of the hosts are as follows: u_1=3Kbps; u_2=4Kbps; u_3=5Kbps; u_4=6Kbps; u_5=7Kbps
- The download rates of the hosts are as follows: d_1=7Kbps; d_2=8Kbps; d_3=9Kbps; d_4=10Kbps; d_5=11Kbps
  a. If the hosts are operating in client-server model, what will be the time for all the clients to download the file?
  b. If the hosts are operating in a peer-to-peer model (such as *BitTorrent*), what will be the time for all the clients to download the file?

a) For client server architecture, we know that the time needed to distribute the file to all hosts is the maximum between 1-how fast the server can upload N times the file, 2- the slowest time for a host to download the file.

So, time = MAX { $\frac{N \times F}{\mu_s}$, $\frac{F}{\min(d1,d2,d3,d4,d5)}$ }

$\frac{N \times F}{\mu_s} = \frac{1500 \times 5 \times 8}{16 \times 1000}$ = 3.75 seconds

$\frac{F}{\min(d1,d2,d3,d4,d5)} = \frac{1500 \times 8}{7000}$ = 1.714 seconds

Time = MAX{3.75,1.714} = 3.75 seconds

b) For a peer-to-peer model, we need to compare the time for the server to upload the file once, with the time for the hosts to upload and download amongst themselves.

So, time = MAX { $\frac{F}{\mu_s}$, $\frac{F}{\min(d1,d2,d3,d4,d5)}$, $\frac{N \times F}{\mu_s + (\mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5)}$ }

$\frac{F}{\mu_s} = \frac{1500 \times 8}{16000}$ = 0.75 seconds

$\frac{F}{\min(d1,d2,d3,d4,d5)}$ = 1.714 seconds

$\frac{N \times F}{\mu_s + (\mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5)} = \frac{5 \times 1500 \times 8}{16000 + (3000 + 4000 + 5000 + 6000 + 7000),}$ = 1.463 seconds.

Time = MAX{0.75, 1.714, 1.463} = 1.714 seconds.

Problem 5

Consider that initially the client has the following cookies: {Yahoo 2342 – Amazon 8367 – Google 9110}. Now the client wants to contact eBay.

- What will be the exchanged messages between the client and the eBay server?

- If the client wants to contact the eBay server again, what will be the exchanged messages between the client and the eBay server?

- Upon first initiation with the server, the client sends a normal request message (GET request). The eBay server then replies with the usual response (200 OK) with an additional Cookie ID (set-cookie) specific to the client.
- When the client wants to contact the eBay server again, the client will send the request message with an additional (cookie ID), the one that was given to the client by eBay. The server will reply with the usual response message.

Problem 6

Suppose that Alice wants to send an email message to Bob. This will involve four entities: Alice's mail client MCA (for email composition and sending), Alice's outgoing mail server OSA, Bob's incoming mail server ISB, and Bob's mail client MCB (for email retrieval and viewing). Explain the steps, including the protocols and servers used, from the time when the email is sent until it is retrieved by Bob.

- Alice uses her mail client (MCA) to compose an email to Bob.
- The email is then sent to Alice's outgoing mail server (OSA) using the SMTP protocol.
- OSA opens a TCP connection to Bob's incoming mail server (ISB) after resolving Bob's domain using a DNS query. Then using SMTP OSA will send Alice's mail to Bob where it will be saved in the inbox.
- Finally Bob must invoke IMAP to access his received mail from his ISB.

Problem 7

If you registered a new website name: www.networks.org. The IP address of your webserver is 111.212.101.10, and that of your DNS server is 111.212.101.11
What DNS records should you add and where? Write down any assumptions.

We assume that the requests needed for the TLD servers are already found in the root server.
In the org TLD server, we will add two types of DNS records:
- A type: returns the hostname of our DNS server (dns.networks.org) as well as the IP address of our DNS server (111.212.101.11)
- NS type: returns the domain name (networks.org) and the hostname of authoritative name server for this domain (dns.networks.org)

In our DNS server, we will add one DNS record:
- A type: returns the IP address of our web server (111.212.101.10)

Problem 8

You have a startup company.
Choose its name and TLD, with or without the cTLD.

Then show all the steps to have your website up and running completely, giving server names and IP addresses, the RRs that must be inserted into DNS servers, and the actual DNS registrar that must be contacted.

Startup Company Details:
Name: networks.com
IP address: 74.178.1.110
TLD: .org

1. Create an authoritative DNS server with **name:** dns.networks.com and **IP address:** 74.178.1.111. This DNS server will hold the IP address of our website
   RR in DNS:
   (networks.com, 74.178.1.110, A, ttl)
2. We initialize into the com TLD server the name and the IP of the authoritative DNS server.
   RR in the TLD:
   (networks.com, dns.networks.com, NS, ttl)
   (dns.networks.com, 74.178.1.111, A, ttl)

Problem 9

a. What is a *whois* database?
   A WHOIS database is a public directory that stores information about the registered owners of Internet resources such as domain names and IP address blocks. For domain names it typically lists the registrar, registration and expiration dates, name servers (authoritative DNS), and contact information (registrant, admin, technical). For IP addresses it lists the Regional Internet Registry (RIR) record (ARIN/RIPE/APNIC/AFRINIC/LACNIC), the organization that holds the allocation, and the allocated IP range.
b. Use various whois databases on the Internet to obtain the names of two DNS servers. Indicate which whois databases you used.
c. Use nslookup on your local host to send DNS queries to three DNS servers: your local DNS server and the two DNS servers you found in part (b). Try querying for Type A, NS, and MX reports. Summarize your findings.
d. Use nslookup to find a Web server that has multiple IP addresses. Does the Web server of your institution (school or company) have multiple IP addresses?
e. Use the ARIN whois database to determine the IP address range used by your university.
f. Describe how an attacker can use whois databases and the nslookup tool to perform reconnaissance on an institution before launching an attack.
   An attacker can use the whois database and nslookup tool to determine the IP address ranges, DNS server addresses, etc., for the target institution. So, WHOIS + nslookup provide passive reconnaissance data that is legal to collect and very useful for attackers to plan targeted campaigns; thus organizations should limit public exposure of unnecessary contact info and secure their DNS infrastructure.
g. Discuss why whois databases should be publicly available.
   By analyzing the source address of attack packets, the victim can use whois to obtain information about domain from which the attack is coming and possibly inform the administrators of the origin domain

Arguments for public WHOIS:

- **Accountability & abuse handling:** Victims and network operators can identify the owner of a domain/IP involved in abuse (spam, DDoS) and contact them or their registrar. This enables remediation and cross-operator coordination.
- **Network troubleshooting:** Administrators use WHOIS to find point(s) of contact for routing and incident response.
- **Transparency & trust:** Public registration data helps law enforcement and improves trust on the Internet.

Counterpoints / Privacy concerns:

- **Privacy & safety:** Publicly exposing personal contact info risks harassment, doxxing, and privacy invasion. Individuals and small organizations may prefer proxy or privacy protection.
- **GDPR and regulation:** Legal frameworks (like GDPR in EU) require that personal data be protected; WHOIS systems have evolved (RDAP, redaction) to accommodate lawful access while protecting personal data.

WHOIS should be **available in a controlled manner** — public enough to allow abuse reporting and technical troubleshooting, but privacy protections (WHOIS privacy services, redaction, RDAP access controls) should be supported so personal data is not unnecessarily exposed.

h. Discuss some methods for hiding/protecting DNS & WHOIS information.

- Use a **WHOIS privacy protection service** (registrars often provide this to hide personal contact info).
- Use a **DNS proxy/CDN like Cloudflare** to hide the origin server's real IP address behind Cloudflare's network.
- Configure **restricted zone transfers** (AXFR) so DNS records aren't leaked.
- Consider **split-horizon DNS** (different views for internal vs. external queries).