Assignment #2
Csci4211 Fall 2018
Due on Oct 24th, 2018

October 25, 2018

Notes: There are five questions in this assignment. Each question has 10 points.

1. (10 pt.) How does SMTP mark the end of a message body? How about HTTP? Can HTTP use the same method as SMTP to mark the end of a message body? Explain your answer.

**Answer:**

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.
No, HTTP cannot use the method used by SMTP, because HTTP message could be binary data, whereas in SMTP, the message body must be in 7-bit ASCII format.

Grading:
-3 If fail to show the mechanism of SMTP.
-3 If fail to show the mechanism of HTTP.
-1 If a clear answer (yes/no) is not given.
-2 If fail to give a correct answer to the third question. -2 If solid explanation is not given for the third question.

2. (10 pt) What is the Apache Web Server? Please describe this web server in a page of write-up. What functions does it currently have? What is the cost of this web server? You may want to look at Wikipedia to answer these questions.

**Answer: (From Wikipedia)**

The Apache HTTP Server, colloquially called Apache, is the world’s most used web server software. Originally based on the NCSA HTTPd server, development of Apache began in early 1995 after work on the NCSA code stalled. Apache played a key role in the initial growth of the World Wide Web, quickly overtaking NCSA HTTPd as the dominant HTTP server, and has remained most popular since April 1996. In 2009, it became the first web server software to serve more than 100 million websites.

Apache is an open source software available for free. The Apache server offers a number of services that clients might make use of. These services are offered using various protocols through different ports, and include: hypertext transfer protocol (HTTP), typically through port 80, simple mail transfer protocol (SMTP), typically through port 25, domain name service (DNS) for mapping domain names to their corresponding IP addresses, generally through port 53, and file transfer protocol (FTP) for uploading and downloading files, usually through port 21.

Grading:
-1~ -3 If the answer does not show the student’s understanding of the concept of Apache Web Server.
-2 If the cost of the Apache Web Server is not mentioned.
Partial points will be deducted if important functions of Apache Web Server are missing.

3. (10 pt.) What is the function of a DNS server? Please describe how an iterative query and a recursive query to be handled by DNS servers.

**Answer:**
The Domain Name System (DNS) is a hierarchical decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical IP addresses needed for locating and identifying computer services and devices with the underlying network protocols.

- Recursive queries will ask the DNS server to do all the job of fetching the final IP address for your request and return it directly to you. In the process, the queried DNS server may also query other DNS servers to get the final result.
- On the other hand, iterative queries means the DNS server will only resolve part of your request and refer you to other DNS servers for your final result.

**Grading:**
-1 ~ -4 If main function of the DNS is missing.
-1 ~ -6 If the difference between recursive query DNS and iterative DNS is not clear/missing.

4. (10 pt) Comparing HTTP 1.0 with HTTP 1.1, what are the major differences between the two? Which one will perform better from a client point of view? Do you see any condition that HTTP 1.0 will perform better from a server perspective?

**Answer:** (Stack Overflow: http://stackoverflow.com/questions/246859/http-1-0-vs-1-1 )

**Proxy support and the Host field:**

HTTP 1.1 has a required Host header by spec. HTTP 1.0 does not officially require a Host header, but it doesn’t hurt to add one, and many applications (proxies) expect to see the Host header regardless of the protocol version.

**Example:**
GET / HTTP/1.1
Host: www.blahblahblah.com

This header is useful because it allows you to route a message through proxy servers, and also because your web server can distinguish between different sites on the same server. So this means if you have blahblahblah.com and helohelohelo.com both pointing to the same IP. Your web server can use the Host field to distinguish which site the client machine wants.

**Persistent connections:**
HTTP 1.1 also allows you to have persistent connections which means that you can have more than one request/response on the same HTTP connection.

In HTTP 1.0 you had to open a new connection for each request/response pair. And after each response the connection would be closed. This lead to some big efficiency problems because of TCP Slow Start.

**OPTIONS method:**
HTTP/1.1 introduces the OPTIONS method. An HTTP client can use this method to determine the abilities of the HTTP server. It’s mostly used for Cross Origin Resource Sharing in web applications.

**caching:**
- HTTP 1.0 had support for caching via the header: If-Modified-Since.
- HTTP 1.1 expands on the caching support a lot by using something called ’entity tag’. If 2 resources are the same, then they will have the same entity tags.
- HTTP 1.1 also adds the If-Unmodified-Since, If-Match, If-None-Match conditional headers. There are also further additions relating to caching like the Cache-Control header.

100 Continue status:
There is a new return code in HTTP/1.1 100 Continue. This is to prevent a client from sending a large request when that client is not even sure if the server can process the request, or is authorized to process the request. In this case the client sends only the headers, and the server will tell the client 100 Continue, go ahead with the body.
HTTP 1.1 will perform better from a client point of view. Because HTTP 1.1 has PersistentConnection and Pipelining. Client can also get request resource from server more efficiently. There might be some scenarios that when server does not have enough resource, HTTP 1.1 will cause extra load for the server hence decrease the performance of the server.

Grading:
-1 ~ -4 If major differences are missing in the answer.
-2 If the answer is wrong in choosing the one with better performance.
-2 If the reason behind the choice is not clarified.
-1 ~ -2 If the answer to the last question is not justified. This is an open question; no points will be deducted if you have clear explanation.

5. (10 pt.) Alice @umn.edu plans to send an email to Bob @google.com. Please describe the sequence of actions of protocols to deliver this email to Bob’s mailbox. Please also describe the actions for Bob to access this email.

Answer:

Grading:
-3 if the SMTP protocol is missing/wrongly used or not included.
-3 if the POP3 or IMAP protocol is missing or not included.
-2 if POP3/IMAP is wrongly use. These protocols are used for email access.
-2 if description is not correct.