Kevin Nash
kjn33
EECS 325

Homework 2

1.

a. The minimum number of DNS servers is **one**, if the user's local DNS server has the requested hostname cached, then the user may receive the appropriate IP address after only one DNS request.

b. If no DNS server caches any DNS records, the minimum number of DNS servers involved is **four:** The requesting host sends a query message to its **local DNS server**. The local DNS server forwards the query message to a **root DNS server**, which returns addresses for TLD servers responsible for edu. The local DNS server then resends the query message to one of those **TLD server**s. The TLD server will respond with the IP address of the **authoritative DNS server** for CWRU.

c.
```
; <<>> DiG 9.9.5-3ubuntu0.5-Ubuntu <<>> www.case.edu @a.root-servers.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50970
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 8
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.case.edu.                  IN    A

;; AUTHORITY SECTION:
edu.              172800      IN    NS    l.edu-servers.net.
edu.              172800      IN    NS    g.edu-servers.net.
edu.              172800      IN    NS    f.edu-servers.net.
edu.              172800      IN    NS    d.edu-servers.net.
edu.              172800      IN    NS    c.edu-servers.net.
edu.              172800      IN    NS    a.edu-servers.net.

;; ADDITIONAL SECTION:
l.edu-servers.net.      172800      IN    A     192.41.162.30
g.edu-servers.net.      172800      IN    A     192.42.93.30
g.edu-servers.net.      172800      IN    AAAA  2001:503:cc2c::2:36
f.edu-servers.net.      172800      IN    A     192.35.51.30
d.edu-servers.net.      172800      IN    A     192.31.80.30
c.edu-servers.net.      172800      IN    A     192.26.92.30
a.edu-servers.net.      172800      IN    A     192.5.6.30

;; Query time: 19 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Thu Oct 01 14:39:58 EDT 2015
;; MSG SIZE  rcvd: 276

; <<>> DiG 9.9.5-3ubuntu0.5-Ubuntu <<>> www.case.edu @a.edu-servers.net
```

```
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48996
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 5, ADDITIONAL: 6
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.case.edu.                  IN    A

;; AUTHORITY SECTION:
case.edu.          172800      IN    NS    ns1.oar.net.
case.edu.          172800      IN    NS    ns2.cwru.edu.
case.edu.          172800      IN    NS    ns2.oar.net.
case.edu.          172800      IN    NS    ns.cwru.edu.
case.edu.          172800      IN    NS    ns3.cwru.edu.

;; ADDITIONAL SECTION:
ns1.oar.net.             172800      IN    A     192.88.193.144
ns2.cwru.edu.            172800      IN    A     129.22.4.3
ns2.oar.net.             172800      IN    A     192.88.195.10
ns.cwru.edu.             172800      IN    A     129.22.4.1
ns3.cwru.edu.            172800      IN    A     129.22.105.54

;; Query time: 38 msec
;; SERVER: 192.5.6.30#53(192.5.6.30)
;; WHEN: Thu Oct 01 15:39:48 EDT 2015
;; MSG SIZE  rcvd: 222

; <<>> DiG 9.9.5-3ubuntu0.5-Ubuntu <<>> www.case.edu @ns.cwru.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40517
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 11, ADDITIONAL: 16

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.case.edu.                  IN    A

;; ANSWER SECTION:
www.case.edu.            120   IN    A     129.22.12.21
www.case.edu.            120   IN    A     129.22.108.21

;; AUTHORITY SECTION:
case.edu.          86400 IN    NS    ns.CWRU.edu.
case.edu.          86400 IN    NS    ns2.CWRU.edu.
case.edu.          86400 IN    NS    ns3a.tis.CWRU.edu.
case.edu.          86400 IN    NS    galvatron.CNS.CWRU.edu.
case.edu.          86400 IN    NS    ncnoc.NCREN.NET.
case.edu.          86400 IN    NS    ns1.OAR.NET.
case.edu.          86400 IN    NS    ns2.OAR.NET.
case.edu.          86400 IN    NS    ns4.OAR.NET.
case.edu.          86400 IN    NS    ns5.OAR.NET.
```

```
case.edu.         86400 IN    NS    ns6.OAR.NET.
case.edu.         86400 IN    NS    ns7.OAR.NET.

;; ADDITIONAL SECTION:
ns.CWRU.edu.            86400 IN    A     129.22.4.1
ns2.CWRU.edu.           86400 IN    A     129.22.4.3
ns3a.tis.CWRU.edu.      604800      IN    A     129.22.105.54
galvatron.CNS.CWRU.edu. 604800      IN    A     129.22.104.56
ncnoc.NCREN.NET.  44606 IN    A     192.101.21.1
ns1.OAR.NET.            44606 IN    A     192.88.193.144
ns2.OAR.NET.            44606 IN    A     192.88.195.10
ns4.OAR.NET.            44606 IN    A     199.18.199.2
ns4.OAR.NET.            44690 IN    AAAA  2610:a8:2004::2
ns5.OAR.NET.            44610 IN    A     199.218.199.2
ns5.OAR.NET.            44690 IN    AAAA  2610:a8:2005::2
ns6.OAR.NET.            44610 IN    A     157.134.199.2
ns6.OAR.NET.            44690 IN    AAAA  2610:a8:2006::2
ns7.OAR.NET.            44610 IN    A     206.244.199.2
ns7.OAR.NET.            44690 IN    AAAA  2610:a8:2007::2

;; Query time: 1 msec
;; SERVER: 129.22.4.1#53(129.22.4.1)
;; WHEN: Thu Oct 01 15:41:12 EDT 2015
;; MSG SIZE  rcvd: 593
```

2.

a. Non-persistent HTTP with 4 parallel TCP connections, the total RTT is **6T**

T	Open the TCP connection
T	Retrieve the container object
T	Open four parallel connections
T	Retrieve four embedded objects
T	Open three parallel connections
T	Retrieve three embedded objects

b. HTTP without pipelining with a single TCP connection: the total RTT is **9T**

T	Open the TCP connection
T	Retrieve the container object
7T	Retrieve seven embedded objects

HTTP without pipelining with 2 parallel TCP connections: the total RTT is **7T**

T	Open the TCP connection
T	Retrieve the container object
T	Open a second parallel connection
3T	Retrieve six embedded objects (3 round trips over two connections)
T	Retrieve one embedded object

c. HTTP with pipelining with a single TCP connection: the total RTT is **3T**

T	Open the TCP connection
T	Retrieve the container object
T	Retrieve seven embedded objects without waiting for responses

HTTP with pipelining with 2 parallel TCP connections: the total RTT is **3T**

T	Open one TCP connection
T	Retrieve the container object
T	Retrieve seven embedded objects without waiting for responses

Note: Using a second TCP connection would not reduce our RTT in this case, so the above model uses only one. Opening a second connection would add another round trip for a total of 4T.

d.

i. Multiple parallel TCP connections can reduce the overall round trip time when also using non-persistent HTTP, if the reduction from distributing objects between connections is greater than the overhead cost of doing so. For pages with less embedded content, the overhead of more connections reduces the

ii. Persistent HTTP can reduce loading time and thus it is very widely used. There isn't much of a disadvantage to a persistent connection: only that the server keeps the connection open for a brief period once the client is finished with the connection, which prevents those resources from be utilized by another client during that period.

iii. Pipelining can dramatically reduce loading time of web pages, especially over high latency connections. Because pipelined connections do not immediately wait for a server response, its effectiveness is tied to latency. As such, it has a great effect on satellite Internet but a lesser effect on broadband connections. Despite its potential benefits, many browsers do not support pipelining or have it disabled by default to avoid issues with misbehaving servers.

3. There are three cases possible:

Download source	Response time	Probability
Origin server	1100 ms	0.6
Proxy cache	100 ms	0.4
Direct download	1000 ms	--

Therefore the average response time is given by:

$$1100 \text{ ms} * 0.6 + 100 \text{ ms} * 0.4 = \mathbf{700 \text{ ms}}$$

Because the above 700 ms is less than the direct download time of 1000 ms, using a proxy does make sense from a performance perspective.

4.

a. Without logging into the site store.steampowered.com, users can select a language. This preference will persist for future sessions by using a locally stored cookie. As a test, I loaded the page in my browser, set my language to Greek. I then received the following cookie:

Name: `Steam_Language`

Content: `greek`

Domain: `store.steampowered.com`

b. Users logging into the site newgrounds.com can instruct the site to remember their credentials. Visiting other pages on the site will not prompt another login, but revisiting the site after closing the browser will. As a test, I logged in to the site with username “Garage” and chose “Remember me.” I then received the following cookies:

Name: `NG_GG_username`
Content: `Garage`
Domain: `.newgrounds.com`

Name: `remember_me_checkbox`
Content: `1`
Domain: `.newgrounds.com`

c. Cookies that track user access patterns rarely have recognizable content in the same way that user preferences and time/dates are stored. However, according to amazon.com’s Privacy Notice, “cookies […] transfer to your device to enable our systems to recognize your device and to provide features such as […] Recommended for You,” which is used to recommend items similar to ones the user has previously viewed or searched for.

d. Regarding cookies, amazon.com’s Privacy Notice also states that it serves “personalized advertisements on other Web sites. There are various cookies given to the browser by Amazon which may contain advertisement tracking information.

5.

a. SMTP uses dot stuffing. Wherever a dot begins a message, an additional dot is inserted directly afterward by the client. When the message is read by the server, it first removes every first dot, mutating the body back into its original form.

b. HTTP has several specific rules for denoting the end of a message body, either terminated explicitly or once it reaches its transfer-length. For sake of brevity, I will simply list the possibilities here:
terminated by an empty line
transfer-length defined by “chunked” transfer-coding
transfer-length defined by Content-Length header field
transfer-length defined by the “multipart/byteranges” media type
terminated by the server closing the connection

LAB

Sites covered:

004	`baidu.com`	504	`surveymonkey.com`
104	`china.com`	604	`sh.st`
204	`w3schools.com`	704	`torcache.net`
304	`mashable.com`	804	`clixsense.com`
404	`tube8.com`	904	`ampclicks.com`

Note: a Python file and requisite trace files are included in the Blackboard submission alongside this PDF

a. A total of 57 DNS resolutions were performed by my host over the 10 page loads.
The average number of resolutions is therefore 5.7 per page, including those due to embedded objects.

b. My program estimated that nine of the ten sites used a load balancer. This may not be an exact number, as a server does not broadcast explicitly that it is using load balancing. Instead I searched for multiple resolutions for the same request as my metric. If multiple resolvers were given, I counted that as indicative of load balancing.

c. The average number of embedded objects had an extremely broad range. One site (`ampclicks.com`) had extremely few embedded objects, while another (`china.com`) used so many that the page took several seconds to finish loading completely, even after the majority of the page's content was visible.
Ampclicks sent about 100 packets versus China's 12,000.
On average, 752 packets were sent.