Course: Communication Networks
Date: 9/29
Topics Covered: HW #2

- MATTHEW KRUEGER -

1. You wish to download the Amazon homepage at URL `www.amazon.com/index.html`.

   (a) Write an HTTP request message or messages sufficient to induce this download.

   (b) Once `www.amazon.com` downloads this base page, it typically populates the page with objects tailored to the requesting browser (e.g., personalized purchase recommendations). Briefly explain how the HTTP feature that enables this personalization works. What is a security vulnerability of this feature? *(Hint: See Sec. 2.2.4 in text).*

   (c) Suppose that you work for a company that has just begun selling merchandise on `amazon.com`. Due to the resulting spike in inquiries and orders (most originating from Amazon servers) your company's ISP connection becomes congested. To help reduce this congestion you propose the use of a web cache. Where should this web cache ideally be located? Can all of your site's web content be cached? Briefly explain.

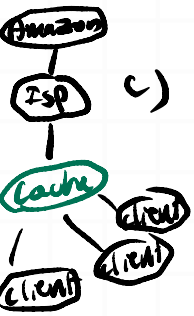
   (d) Explain the role played by "conditional GETs" in your web caching solution. How are the conditional GETs initiated? What header must be present in your company server's response to all GET requests if your web caching solution is to work?

   (e) What role does the HTTP "Host:" header play in web caching?

a) Request: 1. GET /index.html HTTP/1.1
host: 2. Host www.amazon.com
user: 3. user-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS...
accept: 4. Accept: text/html
language 5. Accept-language: en-US, en; q=0.9

b) Cookies allows for personalized information to be shown. Cookies store authentication, recommendations, and user-state. If a malicious actor were to gain access to the cookie session, this would pose a huge security threat as user-state & personal items would be exposed.

c) With an increase of traffic to your site, a web-cache should be placed close to the client; near the company's network edge. This allows for the cache to store frequently requested content, quickening the response time to client requests.



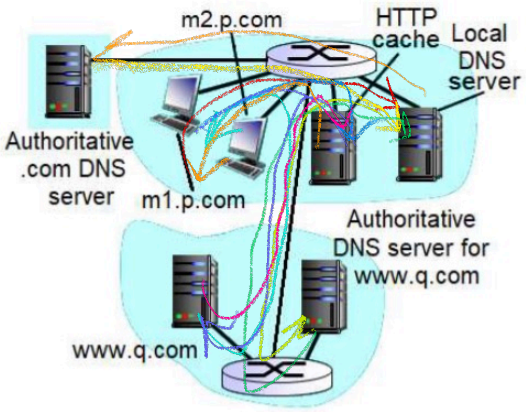
d) The conditional GET checks to see if the cache contains current content for user request. If the cache contains the most up-to-date content, the conditional get will simply take the content from the cache. If the cache is NOT updated the conditional get will get the most recent content from the origin server and update the cache.

e) The host is the server to the cache & user. Stores content in cache between host and client & updates cache when current information is requested by client. Host specifies DNS request

Course: Communication Networks
Date: 2/29
Topics Covered: HW #2

- MATTHEW KRUEGER -

2. Answer the following assuming that the network below follows: (1) all HTTP requests originating in `p.com` are first directed to `p.com`'s HTTP cache; (2) that both `p.com`'s HTTP cache and DNS server caches are initially empty; and (3) that all DNS queries are iterative.



(a) A user at `m1.p.com` types the URL `www.q.com/largefile.html` into a browser to retrieve a large file from `www.q.com`. List, in order, all messages (DNS or HTTP, request or response) generated by this browser request. Please use the following format:

1) {DNS or HTTP} {request or response} from {A} to {B}
2) {DNS or HTTP} {request or response} from {C} to {D}
3) etc.

(b) Just after the large file is delivered to `m1.p.com`, a user at `m2.p.com` requests the same file from `www.q.com`. List, in order and in the format displayed above, all the HTTP and DNS messages (requests and responses) generated by this request.

(c) If all the DNS queries were recursive would your answer to (a) change? If yes, rework (a) in this case. What about (b)? If yes, rework (b) in this case.

(d) It is often said that recursive DNS queries are "more burdensome" to DNS servers than iterative DNS queries. Why? *(Hint: Compare Figs. 2.19 and 2.20 in the textbook and/or the work done by the DNS servers in each case in this problem).*

a)

① DNS request : m1.p.com → Local DNS server
② DNS response : Local DNS server → Authoritative .com server
③ DNS request : Authoritative .com server → Local DNS server
④ DNS response : Local DNS server → Authoritative .com server for www.q.com
⑤ DNS request : Authoritative .com server for www.q.com → Local DNS server
⑥ DNS response: www.q.com → m1.p.com
⑦ HTTP request: m1.p.com → HTTP cache @ p.com MISS
⑧ HTTP request: HTTP cache @ p.com → www.q.com
⑨ HTTP response: www.q.com → HTTP cache @ p.com
⑩ HTTP response: HTTP cache @ p.com → m1.p.com

Course: Communication Networks
Date: 9/29
Topics Covered: Hw #2

- MATTHEW KRUEGER -

b)
① DNS request: m2.p.com → local DNS server @ p.com
② DNS response: local DNS server @ p.com → m2.p.com
③ HTTP rqst: m2.p.com → HTTP cache @ p.com — Hit
④ HTTP response: HTTP cache @ p.com → m2.p.com

c) Part a changes, part b does not.

Part a) yes change!
① DNS request: m1.p.com → local DNS server
② DNS : local DNS server → Authoritative .com server
③ DNS : Authoritative .com server → Authoritative .com server for www.q.com (recursive replies)
④ DNS : Authoritative .com server for www.q.com → www.q.com
⑤ DNS : www.q.com → Authoritative .com server for www.q.com
⑥ DNS : Authoritative .com server for www.q.com → Authoritative .com server (recursive replies)
⑦ DNS : Authoritative .com server → local DNS server
⑧ DNS response : local DNS server → m1.p.com

Part b) No change!
① DNS request: m2.p.com → local DNS server @ p.com
② DNS response: local DNS server @ p.com → m2.p.com

d) Recursive DNS is more costly because the load is kept in DNS rather than client.
Iterative: provides final answer or referral to another DNS server; client does "lookup"
Recursive: provides final answer; DNS does "lookup"

Course: Communication Networks
Date: 9/24
Topics Covered: HW#2

- MATTHEW KRUEGER -

3. The base page of a website is of size F_0 bits. The page references six objects of size F_1 bits located on the same server. The round-trip time to the server is T_0 ($< min\{F_0/R, F_1/R\}$) seconds, and the bottleneck rate of the server is R bps. *(Hint: Recall that a RTT is involved both in setting up a connection AND in making file requests).* Suppose that three DNS servers are visited before the host receives the base page IP address, with successive DNS visits incurring delays of T_1, T_2, and T_3 seconds. Assuming that the DNS packet transmission time and all other processing delays and overheads are negligible, how much time elapses when:

HTTP 1.0: NON-persistent
1. TCP connection opened
2. can send only one object over TCP
3. TCP closed

HTTP 1.1: persistent
1. TCP connection opened
2. multiple objects sent
3. TCP closed

(a) HTTP/1.0 is used?

(b) Persistent, non-pipelined HTTP/1.1 is used?

(c) Persistent, pipelined HTTP/1.1 is used?

(d) Suppose persistent, pipelined HTTP/1.1 is to be used with n parallel connections being opened after the base page is read. Is it always better for a user to open as many parallel HTTP connections as possible when downloading their webpages? Why or why not?

a) HTTP 1.0: NON persistent $\longrightarrow$ 2RTT + file transmission time

1. DNS time: $T_1 + T_2 + T_3$ +
2. Download base page: $2 \cdot T_0 + \frac{F_0}{R}$ + $\longrightarrow$ $T_1 + T_2 + T_3 + 2T_0 + \frac{F_0}{R} + 12T_0 + \frac{6F_1}{R}$
3. Download objects: $6(2 \cdot T_0 + \frac{F_1}{R})$ +

b) HTTP 1.1: Persistent non-pipelined

1. DNS time: $T_1 + T_2 + T_3$ +
2. Download base page: $2T_0 + F_0/R$ + $\longrightarrow$ $T_1 + T_2 + T_3 + 2T_0 + \frac{F_0}{R} + 6T_0 + 6\frac{F_1}{R}$
3. Download objects: $6(F_1/R + T_0)$ +

c) HTTP 1.1: Persistent pipelined

1. DNS time: $T_1 + T_2 + T_3$ +
2. Download base: $2T_0 + \frac{F_0}{R}$ + $T_1 + T_2 + T_3 + 2T_0 + \frac{F_0}{R} + T_0 + \frac{6F_1}{R}$
3. Download objects: $T_0 + 6\frac{F_1}{R}$ +

d) Not necessarily:

- each new connection requires an additional RTT, adding many parallel increases this.
- persistent pipelining only uses one connection.

This opening some parallel may be beneficial, but as the number of parallel connections increase, there is reduced benefit, which can make the process less efficient.

Course: Communication Networks
Date: 9/29
Topics Covered: hw #2

- MATTHEW KRUEGER -

4. Log on to any Linux/UNIX machine that supports the `dig` and `whois` commands. You can use ssh, OpenSSH, PuTTY, SecureCRT, etc. to connect to `login.engineering.uiowa.edu`, for example; however, you may need to be connected to the uiowa/engineering VPN in order for this connection to establish (if you have issues just let us know). Peek at the `man` pages for the commands (e.g., man dig, man whois) and then use them to answer:

   (a) Type `dig www.wsj.com` and then hit enter; wait a few seconds and then do it again.

      i. How long did the first query take? How about the second? Why did the time change?
      ii. Does there exist a host named `www.wsj.com`? Justify your answer.
      iii. Did your two queries return different/multiple IP addresses for the same host? Why?
      iv. Does the WSJ rely on a CDN to distribute its web page? *(Hint: Use `whois` on an IP address(es) found in (i)-(iii) to justify your answer.)*
      v. If the host were to change, what is the minimum time it would take "The Web" to learn this?

   (b) Determine the DNS host name delegation chain for `wsj.com` by recursively invoking `dig` on hosts ".", ".com", "wsj.com", and "wsj.com" with: the name server set to a server discovered in the previous invocation and the query type set to NS. Continue until you stop getting new delegations (indicated by a DNS server returning a SOA record or a delegation pointing to itself). List the chain you obtain. Example commands may be...

   ```
   dig NS .
   dig @<root name server> NS com
   etc.
   ```

   (c) Determine, using `dig`, the host name of the host that would most likely process email sent to `[user]@wsj.com`. Justify your answer.

a) i) 1st time: 65 ms
2nd time: 17 ms

This change is because of a web-cache; fewer DNS steps taken to resolve connection therefore information received quicker.
* fewer intermediate resolves required.

ii) yes, there exists a host named www.wsj.com, with the CNAME: dipoylmxy0v3w.cloudfront.net
The CNAME is the alias www.wsj.com points to. Canonical Name record maps to IP addr.
4 in this case:
52.84.52.48
52.84.52.119
52.84.52.28
52.84.52.69

iii) No, they both returned the same set of IP addresses. See i)

iv) yes. WSJ relies on Amazon's Content Delivery Network (ACDN) to map the WSJ to its CNAME Domain

v) TTL for wsj (via dig) is 180 seconds.

Course: Communication Networks
Date: 9/22
Topics Covered: Hw #2

- MATTHEW KRUEGER -

b)

1. dig NS (query root nameserver)
2. dig @a.gtld-servers.net NS wsj.com (query a .com nameserver for wsj)
3. dig @ns-705.awsdns-24.net NS wsj.com (query AWS nameserver)
4. end.

c) using "dig mx wsj.com" : mxa-00596a01.gslb.pphosted.com

mx: mail exchange.

Course: Communication Networks

Date: 9/29

Topics Covered: few #2

- MATTHEW KRUEGER -

a) screenshots

```
❯ dig www.wsj.com

; <<>> DiG 9.10.6 <<>> www.wsj.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3673
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.wsj.com.            180     IN      CNAME   dlp0y1mxy0v3u.cloudfront.net.
dlp0y1mxy0v3u.cloudfront.net. 60 IN     A       52.84.52.48
dlp0y1mxy0v3u.cloudfront.net. 60 IN     A       52.84.52.119
dlp0y1mxy0v3u.cloudfront.net. 60 IN     A       52.84.52.28
dlp0y1mxy0v3u.cloudfront.net. 60 IN     A       52.84.52.69

;; Query time: 65 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Sun Sep 29 18:01:16 CDT 2024
;; MSG SIZE  rcvd: 146

❯ dig www.wsj.com

; <<>> DiG 9.10.6 <<>> www.wsj.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55439
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.wsj.com.            55      IN      CNAME   dlp0y1mxy0v3u.cloudfront.net.
dlp0y1mxy0v3u.cloudfront.net. 55 IN     A       52.84.52.48
dlp0y1mxy0v3u.cloudfront.net. 55 IN     A       52.84.52.119
dlp0y1mxy0v3u.cloudfront.net. 55 IN     A       52.84.52.28
dlp0y1mxy0v3u.cloudfront.net. 55 IN     A       52.84.52.69

;; Query time: 17 msec
```

```
❯ whois 52.84.52.48
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object

refer:        whois.arin.net

inetnum:      52.0.0.0 - 52.255.255.255
organisation: Administered by ARIN
status:       LEGACY

whois:        whois.arin.net

changed:      1991-12
source:       IANA

# whois.arin.net

NetRange:       52.84.0.0 - 52.95.255.255
CIDR:           52.88.0.0/13, 52.84.0.0/14
NetName:        AT-88-Z
NetHandle:      NET-52-84-0-0-1
Parent:         NET52 (NET-52-0-0-0-0)
NetType:        Direct Allocation
OriginAS:       AS16509, AS14618
Organization:   Amazon Technologies Inc. (AT-88-Z)
RegDate:        1991-12-19
Updated:        2022-03-21
Ref:            https://rdap.arin.net/registry/ip/52.84.0.0


OrgName:        Amazon Technologies Inc.
OrgId:          AT-88-Z
Address:        410 Terry Ave N.
City:           Seattle
StateProv:      WA
PostalCode:     98109
Country:        US
RegDate:        2011-12-08
Updated:        2024-01-24
Comment:        All abuse reports MUST include:
Comment:        * src IP
```

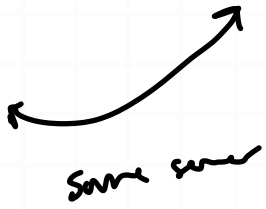
b) screenshots

```
❯ dig NS

; <<>> DiG 9.10.6 <<>> NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46391
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1400
;; QUESTION SECTION:
;.                              IN      NS

;; ANSWER SECTION:
.                       34542   IN      NS      e.root-servers.net.
.                       34542   IN      NS      d.root-servers.net.
.                       34542   IN      NS      c.root-servers.net.
.                       34542   IN      NS      g.root-servers.net.
.                       34542   IN      NS      a.root-servers.net.
.                       34542   IN      NS      k.root-servers.net.
.                       34542   IN      NS      m.root-servers.net.
.                       34542   IN      NS      h.root-servers.net.
.                       34542   IN      NS      i.root-servers.net.
.                       34542   IN      NS      f.root-servers.net.
.                       34542   IN      NS      b.root-servers.net.
.                       34542   IN      NS      j.root-servers.net.
.                       34542   IN      NS      l.root-servers.net.

;; Query time: 7 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Sun Sep 29 18:35:29 CDT 2024
;; MSG SIZE  rcvd: 239
```

```
❯ dig @e.root-servers.net NS com

; <<>> DiG 9.10.6 <<>> @e.root-servers.net NS com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60597
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 27
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;com.                           IN      NS

;; AUTHORITY SECTION:
com.                    172800  IN      NS      a.gtld-servers.net.
com.                    172800  IN      NS      b.gtld-servers.net.
com.                    172800  IN      NS      c.gtld-servers.net.
com.                    172800  IN      NS      d.gtld-servers.net.
com.                    172800  IN      NS      e.gtld-servers.net.
com.                    172800  IN      NS      f.gtld-servers.net.
com.                    172800  IN      NS      g.gtld-servers.net.
com.                    172800  IN      NS      h.gtld-servers.net.
com.                    172800  IN      NS      i.gtld-servers.net.
com.                    172800  IN      NS      j.gtld-servers.net.
com.                    172800  IN      NS      k.gtld-servers.net.
com.                    172800  IN      NS      l.gtld-servers.net.
com.                    172800  IN      NS      m.gtld-servers.net.

;; ADDITIONAL SECTION:
a.gtld-servers.net.     172800  IN      A       192.5.6.30
```

```
❯ dig @a.gtld-servers.net NS wsj.com


; <<>> DiG 9.10.6 <<>> @a.gtld-servers.net NS wsj.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57654
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;wsj.com.                       IN      NS

;; AUTHORITY SECTION:
wsj.com.                172800  IN      NS      ns-705.awsdns-24.net.
wsj.com.                172800  IN      NS      ns-450.awsdns-56.com.
wsj.com.                172800  IN      NS      ns-1588.awsdns-06.co.uk.
wsj.com.                172800  IN      NS      ns-1291.awsdns-33.org.

;; ADDITIONAL SECTION:
ns-450.awsdns-56.com.   172800  IN      A       205.251.193.194

;; Query time: 47 msec
;; SERVER: 192.5.6.30#53(192.5.6.30)
;; WHEN: Sun Sep 29 18:39:44 CDT 2024
;; MSG SIZE  rcvd: 189
```

```
❯ dig @ns-705.awsdns-24.net NS wsj.com


; <<>> DiG 9.10.6 <<>> @ns-705.awsdns-24.net NS wsj.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53771
;; flags: qr aa rd; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;wsj.com.                       IN      NS

;; ANSWER SECTION:
wsj.com.                1800    IN      NS      ns-1291.awsdns-33.org.
wsj.com.                1800    IN      NS      ns-1588.awsdns-06.co.uk.
wsj.com.                1800    IN      NS      ns-450.awsdns-56.com.
wsj.com.                1800    IN      NS      ns-705.awsdns-24.net.

;; Query time: 39 msec
;; SERVER: 205.251.194.193#53(205.251.194.193)
;; WHEN: Sun Sep 29 18:40:27 CDT 2024
;; MSG SIZE  rcvd: 173

~ ❯
```



c) screenshot

```
❯ dig mx wsj.com

; <<>> DiG 9.10.6 <<>> mx wsj.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11928
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;wsj.com.                       IN      MX

;; ANSWER SECTION:
wsj.com.                300     IN      MX      10 mxa-00596a01.gslb.pphosted.com.
wsj.com.                300     IN      MX      10 mxb-00596a01.gslb.pphosted.com.

;; Query time: 39 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Sun Sep 29 18:46:30 CDT 2024
;; MSG SIZE  rcvd: 108
```

Course: Communication Networks
Date: 9/29
Topics Covered: HW#2

- MATTHEW KRUEGER -

5. A 120 Mb file is to be distributed to N peers via the Internet. The 120 Mb file's file server has an upload rate of $u_{server} = 50$ Mbps. All peers have upload rates of $u_{peer} = 10$ Mbps and download rates of $d_{peer} = 30$ Mbps. The connections between the server and the peers' uplinks and downlinks all have 1 Gbps or greater capacity.

   (a) Compute lower bounds for the time (in minutes) required to fully distribute the file to $N = 25$ peers via (1) client-server and (2) peer-to-peer (P2P) distribution architectures.

   (b) Repeat (a) for the case of $N = 500$ peers and briefly comment on what your computations suggest about the relative merits of the two architectures.

   (c) Suppose that the P2P protocol employed is BitTorrent. Is it possible for a host to join the torrent, never upload any data, but still eventually receive a complete copy of a shared file? Explain. *(Hint: See text Sec. 2.5).*

a)

N = 25
F = 120 mb
u_s = u_{server} : 50 mbps
d_{min} = d_{peer} : 30 mbps
u_p = u_{peer} : 10 mbps
connections ≥ 1 Gbps

u_i : u_p · N

$D_{p2p} = \max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum u_i}\right\}$

(min client download time; as aggregate; time for one copy; limiting max download rate)

$D_{CS} = \max\left\{\frac{NF}{u_s}, \frac{F}{d_{min}}\right\}$

(file to send N copies; min client download time)

$= \max\left\{\frac{120}{50}, \frac{120}{30}, \frac{25 \cdot 120}{50 + (10 \cdot 25)}\right\}$

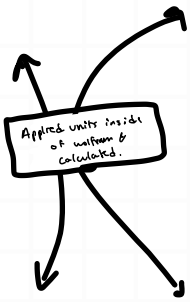
$= \max\{2.4\,(s), 4\,(s), 10\,(s)\}$

D_{p2p} = 10(s) seconds

$D_{CS} = \max\left\{\frac{25 \cdot 120}{50}, \frac{120}{30}\right\}$

$= \max\{60\,(s), 4\,(s)\}$

D_{CS} = 60 seconds



b) N = 500

$D_{p2p} = \max\left\{\frac{120}{50}, \frac{120}{30}, \frac{500 \cdot 120}{50 + (10 \cdot 500)}\right\}$

$= \max\{2.4(s), 4(s), 11.8\,(s)\}$

D_{p2p} = 11.88 seconds
scales efficiently;
↑ peers = ↓ load on server

$D_{CS} = \max\left\{\frac{500 \cdot 120}{50}, \frac{120}{30}\right\}$

$= \max\{1200\,(s), 4(s)\}$

D_{CS} = 1200 seconds
Does not scale efficiently;
↑ peers = ↑ load on server

c) NO; Bit torrent is Tit-for-Tat, meaning at any given time, peers have subsets of chunks to be distributed across other peers. Peers that do not upload may be deprioritized; i.e. not be able to download copies.

Course: Communication Networks
Date: 9/24
Topics Covered: HW #2

- MATTHEW KRUEGER -

6. Content distribution networks (CDNs) deliver content from multiple locations. Typically a client's request is directed to the appropriate location by returning custom answers to content server DNS queries (e.g., by conditioning the responses on the address of the requesting DNS).

   (a) When using DNSs to direct clients to replicas, CDNs typically set the DNS Time-to-Live (TTL) to a small value. Why might this be done? List one negative implication of having a small TTL value.

   (b) List two reasons why a DNS might return different IP addresses in response to the same DNS address query from different clients.

   (c) Suppose that the replica to be shared is a 7 minute video. If each location delivers the video using a DASH system that stores the video in 10 second segments at four different bit rates, how many URLs will be listed in the video's manifest in each location?

a) setting a small TTL ensures clients receive frequently updated data.
An implication is that a small TTL increases the # of DNS queries to frequently re-query DNS; high DNS load.

b)
1. Location in world: geographical location of the client affects the routing of CDN.
2. Load balancing: if traffic in one location is high, CDN will load balance by providing content from different server.

c) DASH: Dynamic, Adaptive, Streaming over HTTP

$$\frac{7\ \text{sec}}{1\ \text{min}} \cdot \frac{60\ \text{sec}}{1} \cdot \frac{1\ \text{segment}}{10\ \text{sec}} \cdot \frac{4}{1} = 168\ \text{urls}$$