CMSC 417 Computer Networks **Spring 2017**

Second Third-Term Exam

Closed book and notes; In class *Thursday, April 27*

⊕ *Do not forget to write your name on the first page. Initial each subsequent page.*
⊕ *Be* **neat** *and* **precise**. **I will not grade answers I cannot read.**
⊕ *You should draw simple figures if you think it will make your answers clearer.*
⊕ *When working with hex dumps, it may be helpful to visually identify the components.*
⊕ *Please indicate if your response to a question is continued on the back of the page.*
⊕ *Good luck and remember, brevity is the soul of wit*

- All problems are mandatory
- I cannot stress this point enough: **Be precise**. If you have written something incorrect along with the correct answer, you should **not** expect to get all the points. I will grade based upon what you **wrote**, not what you **meant**.
- Maximum possible points: ~~50~~. 42

Name: _____

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536

1. TCP

(a) What is SACK, and what does it do?

(2pts)

Selective ACK -- ACKs blocks of data with gaps

(b) Describe the difference between flow control and congestion control.

(3pts)

Flow Control -- avoid overloading receiver
Congestion Control -- avoid overloading network

(c) What is the largest advertised window supported by TCP?

(2pts)

(2^16-1)(2^14) = 2^30 - 2^14

(d) How does TCP cope with an advertised window size of 0?

(2pts)

Sender sends periodic 1-byte segment
Receiver ACKs w/ current AdvWin
Sender waits for AdvWin >= threshold (usually MSS)

(e) What is the purpose of the NOP option?

(1pt)

Pads individual options to 4-byte alignment

2. Domain Name System

(a) Basic Concepts

i) How do DNS zones relate to nameservers? (2pts)

Zones are areas of authority, with at least 1 (usually 2 or more) nameservers, and one nameserver is the authoritative source of info for the zone

ii) When might a DNS request result in an additional record in the response? (2pts)

The requested record is not cached, but the server has an A record for the relevant NS record, it will add the authoritative nameserver's A record as a glue record

(b) Name Encoding

Given the following byte sequences as query names in a DNS payload, what are the corresponding FQDNs? (1pt each):

i) `06 74 6f 6d 63 61 74 06 61 70 61 63 68 65 03 6f 72 67 00`

tomcat.apache.org

ii) `03 77 77 77 06 67 69 74 68 75 62 03 63 6f 6d 00`

www.github.com

iii) `09 77 69 6b 69 70 65 64 69 61 03 6f 72 67 00`

wikipedia.org

(c) Request and Response Payloads

What, if anything is wrong with the following DNS payload (lines begin with byte offsets, IP/UDP headers are not included, and both queries and resource records are highlighted), and what might happen to a naively written host attempting to interpret it? (3pts)

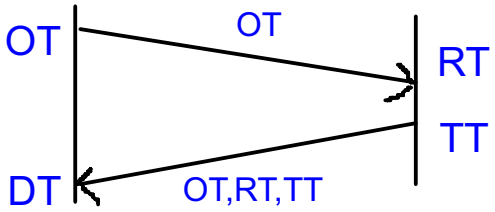
```
0000  cb 85 81 80 00 01 00 01  00 00 00 00 09 77 69 6b ···
0010  69 70 65 64 69 61 03 6f  72 67 00 00 01 00 01 c0 ···
0020  1f 00 01 00 01 00 00 02  3b 00 04 d0 50 99 e0
```

RR query name is a pointer (starts with "11")
The pointer refers to itself (starts at offset 001F)
Could cause an infinite loop when decoding

3. Network Time Protocol

(a) Diagram

Draw a diagram of the client/server message exchanges. Label everything, and use these labels for subsequent parts of this section. You are not obligated to use the notation from lecture, only to make your diagram clear and sufficient. (1pt)



OT = OT'+s DT = DT'+s+d
Tq = RT-OT' Tr = DT'-TT
offset = [(RT-OT)+(TT-DT)]/2
= [(Tq-s)+(-Tr-s-d)]/2
= [Tq - Tr - 2s - d]/2
delay = (RT-OT) + (DT-TT)
= Tq-s+Tr+s+d = Tq+Tr+d

(b) Clock Skew

Assume the client's clock is *skewed* with respect to the server, so that it is s seconds ahead. Assuming no *drift* (that is, the client's clock advances one second for each second that the server's clock advances), and that the message latencies are identical in both directions, what is the offset that the client will apply after receiving a response from the server? Be sure to show your work. (3pts)

d=0, Tq=Tr
offset = -s

(c) Clock Drift

Building off of the previous question, assume that the client's clock *drifts* as well, gaining an additional time d over the course of the message exchange. What will the offset be now? (3pts)

Tq=Tr
offset = -s - d/2

(d) Asymmetric Paths

Now assume that there is neither skew nor drift between the clocks, but that the response from the server takes twice as long to reach the client as the request takes to reach the server. What would the computed offset and delay be, and what does this tell us about choosing an NTP server? (3pts)

s=d=0
Tr=2Tq
offset = -Tq/2
delay = 3Tq
Choosing a server with lower RTT minimizes errors from asymmetry

4. SMTP and HTTP

(a) Spam Prevention

On receiving a `HELO` command, what basic check could an MTA perform, assuming the connection is not from an authenticated user's MUA? (2pts)

HELO includes a server name
Does the client's IP match this name?
Is this name the MX record for its zone?

(b) Mail Forwarding

Users often specify forwarding rules for email, particularly when they have multiple addresses or move from one organization to another. How might an MTA employ envelope headers to detect and prevent infinite forwarding loops? (3pts)

Add the receiving mailbox to a delimited header, if it is already in that header, there's a loop

(c) HTTP Versions

What is one major advantage of HTTP/1.1 over HTTP/1.0? (2pts)

Persistent connections -- can send multiple requests with one TCP session

(d) HTTP GET vs POST

GET typically supports a query format for URLs, which often takes the form:

```
GET path/to/location.php?param1=val1&param2=val2
```

Given this, provide three examples of when it would make sense to use POST rather than GET. (3pts—1pt for each)

Too many key/val pairs (URL too long)
Complex values (structures)
Using SOAP or WSDL (requires XML)

5. Distributed Hash Tables

(a) Hash-Based Addresses vs Chosen Addresses

Distributed hash tables typically assign addresses to nodes and items based on a hash of some relevant data. If, instead, nodes could choose their own addresses, and items were assigned addresses based on some simpler (and more predictable) scheme, how could a malicious party take advantage of this? (2pts)

Adversary could flood a particular part of the ring with nodes (Sybil attack), effectively controlling access to a particular item

(b) Message Latencies in Chord

If a node in Chord were free to choose any node "close enough" to a target identifier in its finger table, how might it improve the latency of message delivery? (2pts)

Choose a node a little short of the target (except successor), but with lower RTT
Most hops are relatively low latency, so only the last hop might need to travel a large distance

(c) Tapestry

Tapestry is another DHT, which employs a different form of routing table. Instead of treating the address space as a ring, it holds a routing table based on longest-prefix matching. Consider a node with the (20-bit) address `01FA3`. It would have routing table entries for the following addresses:

```
10000 20000 30000 40000 50000 60000 70000 80000 90000 A0000 B0000 C0000 D0000 E0000 F0000
00000 02000 03000 04000 05000 06000 07000 08000 09000 0A000 0B000 0C000 0D000 0E000 0F000
01000 01100 01200 01300 01400 01500 01600 01700 01800 01900 01A00 01B00 01C00 01D00 01E00
01F00 01F10 01F20 01F30 01F40 01F50 01F60 01F70 01F80 01F90 01FB0 01FC0 01FD0 01FE0 01FF0
01FA0 01FA1 01FA2 01FA4 01FA5 01FA6 01FA7 01FA8 01FA9 01FAA 01FAB 01FAC 01FAD 01FAE 01FAF
```

Note, in particular, that matching prefixes are successively lengthened by one hex digit for each hop through the overlay. For m-bit addresses, and assuming no redundant links:

i) In the worst case, how many hops through the overlay network are required in Chord? (1pt)

m

ii) In the worst case, how many hops through the overlay network are required in Tapestry? (2pts)

m/4

iii) How many entries are in the Chord finger table? (1pt)

m

iv) How many entries are in the Tapestry routing table (you may round)? (2pts)

15m/4 or 4m