CMSC 417 Computer Networks **Spring 2013**

Second Third-Term Exam

Open book and notes; In class *Thursday, April 11th*

⊕ *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.*
⊕ *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.

Name: _____

Problem	Points
1	
2	
3	
4	
5	
Total	

1. Nomenclature

   (a) Describe the following terms: (2 points each)

   - Authoritative Answer (in DNS)

   - Forward Error Correction

   - FQDN

   - Silly Window Syndrome

   - Congestion Collapse

2. Reliable Transfer Protocols

   (a) State the invariant a sliding windows sender maintains. Explain your notation. (2 points)

   (b) What is a SACK? Give an example demonstrating its efficacy. (3 points)

   (c) Suppose you implement reliable transfer using sliding windows using 3-bit sequence numbers, send/receive window sizes of 5 (frames can have sequence numbers 0, 1, 2, 3, 4). Assume that you have a *perfect* forward channel, but a lossy reverse channel, i.e. data can never be lost, but ACKs can be lost or delayed. Argue why your choice of parameters is safe or show with an example how the protocol fails. *Don't state that data cannot be lost therefore acks are not required — Give an example where the protocol fails even when data is not lost in the forward direction.* (5 points)

TCP Details

3. (a) Without window scaling or the timestamp option, what is the fastest safe rate at which a TCP sender can send data? How is it (most) limited? (2 points)

   (b) How do modern TCPs (Reno, Tahoe, but not Vegas) implement congestion *avoidance*? Describe in terms of how the *cwnd* variable is evolved. (2 points)

   (c) Can both ends of a TCP end up storing TIME-WAIT state? If no, explain why. If yes, show an example with a diagram. (3 points)

   (d) The original TCP timeout was set to twice the estimated RTT, where estimated RTT was computed as a low pass filter over the RTT samples. Describe one fundamental problem with this technique (3 points)

4. DNS

(a) How does a DNS server notify a resolver that a response packet is incomplete? (2 points)

(b) How would you look up the DNS name corresponding to the IP address 128.8.128.8? (2 points)

(c) What is a DNS zone? How is it related to a cut in the namespace? (3 points)

(d) Suppose `cs.umd.edu` is a delegated zone from `umd.edu`. Describe the RRs that allow this relationship to be expressed within the DNS. (3 points)

5. Design (Choices)

(a) TCP allows receivers to delay acks by up to 500ms and send one ack for every other segment received. When would/should a TCP receiver delay an ack? (3 points)

(b) The DNS protocol specifies pointers within a DNS packet by setting two MSBs of the left byte to one. Why? (3 points)

(c) Describe the DNS calls, authorities (owners of DNS servers), nameservers, and RRs involved during a reverse lookup (with +trace on dig) of 128.8.128.8. (4 points)