**14-740: Fundamentals of Computer and Telecommunication Networks**

Spring 2017
Quiz #2
Duration: 75 minutes

Name: ANSWER KEY Andrew ID: ______________________

Important:

- Each question is to be answered in the space provided. Material written on the back of the page or in space above or below the question will not be graded.
- This is a closed book exam -- you may not use any reference materials, crib sheets, or formula cards.
- Calculators are not needed, nor allowed.
- **Write legibly**. Unreadable work will be considered incorrect.
- At the end of the final duration, you will be told to "Cease Work." Immediately stop writing and turn in your paper. Any writing after this point will result in a zero grade.

| | |
|---|---|
| Page 2 | ____________ (22 possible) |
| Page 3 | ____________ (25 possible) |
| Page 4 | ____________ (28 possible) |
| Page 5 | ____________ (15 possible) |
| Page 6 | ____________ (10 possible) |
| Total | ____________ (100 possible) |

I understand that the CMU and course policies on cheating apply to this quiz.

______________________________________________________
signature date

1. Choose whether TCP or UDP is preferable for the following scenarios. Circle one or the other. Each scenario is independent. (12 points)

| | | |
|---|---|---|
| TCP | UDP | The application is sensitive to segment loss |
| TCP | UDP | A server must receive intermittent reports, each of which is small in size, from a large number of clients |
| TCP | UDP | Many segments need to be sent, but they are each independent and thus may be delivered out-of-order |
| TCP | UDP | The network tends to get very congested with messages from many other applications |
| TCP | UDP | The application has a very specific error recovery technique |
| TCP | UDP | The application must deliver a fairly steady bps (i.e. rate sensitive) |

2. The following statement is false. *Explain* why. (4 points)

Applications using UDP can never enjoy reliable end-to-end data transfer.

Applications can still use their own reliable data transfer protocol within the application.

(Such protocols have to be created by the application programmer).

3. Describe how the desire to limit the buffer size in the receiver affected the design of one of the RDT protocols discussed in class. (Part A is 1 point, B is 5 points. 6 points total)

Part A: Which protocol? Go-Back-N

Part B: How was the design affected? Any segment received out of order (i.e. caused by a loss of a previous segment) was discarded and not stored.

**Don't write down here. This is not considered "space provided." Anything you write down here will not be graded (nor even read).**

4. Suppose that you've been observing a single TCP connection using Wireshark during an HTTP file transfer. The total number of segments sent is 8327, while the total number received is only 4197. From the behavior of the application, you believe the data was transferred properly. Why are there so few segments sent in the reverse direction? (8 points)

Delayed ACKs! When things run properly, you only get half the number of ack segments.

5. Suppose Host A sends three segments to Host B with sequence numbers 1234, 1569 and 1822. The third segment contains 178 bytes of data. Host A then receives a segment with an acknowledgement number of 1569 and a receive window of 443. Assuming Host A has plenty of data to send, what will be the sequence number and number of bytes of the next segment Host A sends? Show your work. (8 points)

Host A currently has sent data up to byte 1822+178 -1 = 1999. Next sequence number will be 2000. Byte 1568 has been acknowledged, so there are 1999 - 1568 = 433 unacknowledged bytes already sent. Host A can then send 442 - 433 = 10 bytes.

Next Sequence Number: 2000

Number of Bytes: 10 bytes

6. On the following state diagram for a TCP Reno sender, identify the conditions for changing state (A - D) and the states E and F. (9 points)

A: 3-dup ACK

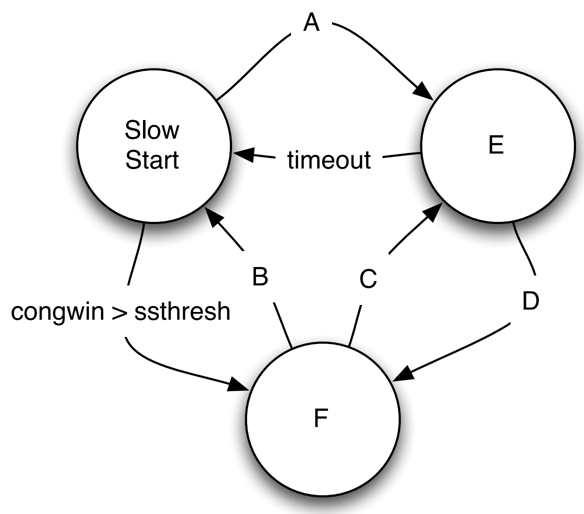
B: timeout

C: 3-dup ACK

D: ACK

E: Fast Recovery

F: Congestion Avoid/Add. Increase



7. List one advantage and one disadvantage of delay-based TCP Congestion Control algorithms. (The disadvantage can't be just the inverse of the advantage). (5 points each, 10 points total)

Advantage (One of the following) Congestion is detected and avoided earlier. Higher throughput. Smoother flow (i.e. fewer bursts).

Disadvantage: Lower throughput when competing with loss-based. More state needed to control (i.e. must keep records of RTTs to detect changes)

8. Explain what route aggregation is. Make sure to mention the rule we talked about that is important for route aggregation. (8 points)

Route Aggregation is when a single prefix can be used to advertise multiple networks (i.e. multiple smaller prefix ranges). The "Longest Matching Prefix Rule" allows route aggregation as many smaller prefix ranges (which have longer prefixes) can be grouped into a single, larger, not-necessarily-contiguous range (a shorter prefix).

9. I overheard a famous network researcher state that NAT is one of the biggest and ugliest hacks in internet history. Explain two possible objections he may have to the use of NAT (from the 4 objections we discussed in class). (10 points)

a. IPv6 should be used to solve the lack of addresses, not NAT. It is NAT's fault that IPv6 hasn't already been adopted.

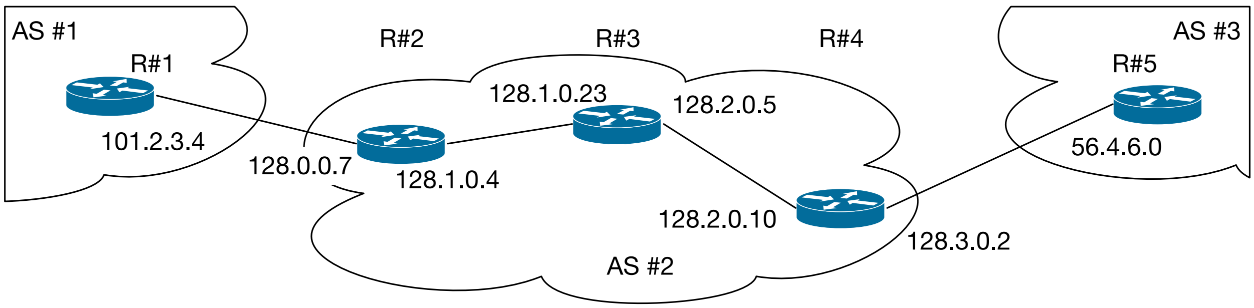
b. NAT violates the end-to-end principle by having devices in the middle of the network involved in maintaining state data relating to the connection between end hosts.

c. Routers shouldn't be processing upper level layer protocol data, as NAT routers must for FTP and SIP

d. Port numbers shouldn't be used to address hosts — they are supposed to address applications.

**Don't write down here. This is not considered "space provided." Anything you write down here will not be graded (nor even read).**

10. This diagram shows five routers (named R#1 through R#5). Each cloud is a different AS (named AS #1 through AS #3). In addition to the links shown, R#3 has no other connections. (Part A is 6 points, B is 9 points. 15 points total)




Part A: List all the TCP connections in the above diagram associated with BGP routing. For each connection, list the connecting *interfaces* of the sender and receiver. Specify the type of session it represents. You may not need all rows of the table.

| Router # | Interface | to | Router # | Interface | Session type |
|---|---|---|---|---|---|
| R#1 | 101.2.3.4 | to | R#2 | 128.0.0.7 | eBGP |
| R#2 | 128.1.0.4 | to | R#4 | 128.2.0.10 | iBGP |
| R#4 | 128.3.0.2 | to | R#5 | 56.4.6.0 | eBGP |
| | | to | | | |

Part B: R#2 receives a BGP UPDATE message from R#1 announcing the prefix 10.10.254/23. The message has the following attributes: AS-PATH (1, 9, 87), MED(100) and a NEXT-HOP. R#2 determines that it has a LOCAL-PREF of 42. Describe how the message is propagated to AS#3 (you can simply list each BGP message and contents sent to each router).

R#2 will send to R#4 with Prefix (10.10.254/23), AS-PATH ({2}, 1, 9, 87), NEXT-HOP (101.2.3.4), LOCAL-PREF(42). Maybe with MED(100).

R#4 will send to R#5 with Prefix (10.10.254/23), AS-PATH (2, 1, 9, 87), NEXT-HOP(128.3.0.2). MED will not be propagated. LOCAL-PREF will not be propagated.

11. The Distance Vector algorithm looks something like this, when executed on each router. N is the set of all destinations. (10 points)

```
Initialize stuff, which includes:
  For each y in set of neighbors:
    Dx(y) = link cost to y
  Send Dx to all neighbors

Loop forever:
  Wait (until a link cost changes to some neighbor W or
        until I receive a distance vector from some neighbor W)
  For each y in N:
    Dx(y) = minv{c(x,v) + Dv(y)}
    Update Forwarding table with {y, link to v} if changed
  End For
  If Dx(y) changed for any y
    For all neighbors i:
      send vector Dx = [Dx(y): y in N] to i
    End For
  End If
End Loop
```

Running this algorithm produces a distance vector (**$D_x$**) that indicates the minimum distance to all N. What modifications need to be made for Split Horizon? Describe modifications in terms of the pseudo-code above. Feel free to draw arrows indicating where code would be inserted or markup the code for deletions. (10 points)

In the body of the If block, add another If around the send:

```
  If Dx(y) changed for any y
    Lookup link L in forwarding table for destination y
    For each neighbors i in [list of all neighbors]:
      If L != link to i
        send vector Dx = [Dx(y): y in N] to i
      End If
    End For
  End If
```