Network Systems (201600146/201600197), Test 4

April 6, 2017, 13:45–15:15

- This is an open-book exam: you are allowed to use the book by Peterson & Davie and the reader that belongs to this module. Furthermore, use of a dictionary is allowed. Use of a simple (non-graphical) calculator is allowed.
- Other written materials, and laptops, tablets, graphical calculators, mobile phones, etc., are not allowed. ***Please remove any such material and equipment from your desk, now!***
- Visiting the toilet without explicit permission of the supervisor is not allowed. During the last 30 minutes of the exam, no toilet visits are allowed.
- Write your answers on this paper, in the provided boxes, and hand this in.
- Total number of pages: 5.

Your name:

(please underline your family name (i.e., the name on your student card), so that we know how to sort)

Your student number:

1. Congestion Control

1 pt (a) Consider TCP's congestion control algorithm, and assume it has at some point `CongestionWindow` = 10 MSS and `SlowstartThreshold` = 16 MSS (see footnote[1]). Then some packets are lost, resulting in a timeout. What will the new values of `CongestionWindow` and `SlowstartThreshold` be after the timeout?

`CongestionWindow` = and `SlowstartThreshold` =

1 pt (b) Consider "our" TCP connection sharing a bottleneck link with one other TCP connection, both trying to send as fast as allowed by the usual TCP congestion control algorithm. Now suppose the other TCP connection has finished transmitting all its data. How does "our" TCP detect that it can now use more bandwidth?

A. Our TCP will get a triple non-duplicate ack.
B. Our TCP will only detect this if we force it to go back to slow start.
C. The router in front of the bottleneck link notifies our host explicitly.
D. Our TCP finds it can increase its window further before packet loss occurs.
E. Our TCP doesn't detect this, and thus will continue sending as if the other connection is still there.

[1] By `SlowstartThreshold`, we mean the same thing which is called `CongestionThreshold` in the book. The former name is much more common though.

1 pt (c) According to the standards, TCP should start its "slow start" procedure with a congestion window of 1 MSS, but many web servers nowadays use a larger value, like 4 MSS.
Assume the RTT is much larger than the transmission delay of the packets, a large file needs to be sent, and no packet loss occurs (so we don't leave slow start).
What is (approximately) the effect of starting with an initial window of 4 MSS instead of 1 MSS on the time needed to transfer the file?

A. It saves 1 RTT.
B. It saves 2 RTTs.
C. It saves 3 RTTs.
D. It saves 4 RTTs.
E. It halves the number of RTTs needed.
F. It divides the number of RTTs needed by 4.
G. It doubles the number of RTTs needed.

1 pt (d) TCP does a fast retransmit when it sees 3 duplicate acknowledgements. This number 3 is somewhat arbitrary. What would be the consequence(s) of choosing it *lower*? (one or more)

A. The congestion window would grow faster.
B. It would take longer for packet loss to be detected.
C. More packets would be lost before the retransmission.
D. Fast retransmission becomes possible at smaller window sizes.
E. Packet reordering would be more likely to cause an unnecessary retransmission.

1 pt (e) Consider a TCP flow (using the standard TCP congestion control algorithms) which shares a 1 Mbit/s bottleneck link with a UDP flow, using First In First Out scheduling. In order to avoid congestion, the UDP application monitors the round-trip time its packets undergo: it reduces its sending rate if the RTT increases, and vice versa.
Which flow will get most of the 1 Mbit/s bottleneck capacity?

A. Each gets half, because both use a fair congestion-control algorithm.
B. Each gets half, because the FIFO scheduling enforces fairness.
C. TCP, because its self-clocking mechanism is optimal for FIFO.
D. TCP, because the UDP sending rate decreases as TCP fills the queue.
E. UDP, because it avoids filling up the queue too much.
F. UDP, because its sending rate can increase even when there is packet loss.

Continued on next page...

2. QoS

1 pt (a) Suppose we have a 100 Mbit/s link, which is shared by a high-quality video data stream of 60 Mbit/s and normal web-browsing traffic. Which of the following is true?

A. Both traffic types are elastic.
B. The video traffic is elastic, the web traffic is not.
C. The video traffic is not elastic, the web traffic is.
D. Neither is elastic.

1 pt (b) Continuing with the situation from the previous question, we want to ensure good quality (low packet loss) for the video stream, regardless of how much web traffic is offered.
Which (one or more) of the following scheduling algorithms can guarantee this?

A. First in, first out.
B. Priority, with the video given high prioriy and web low priority.
C. Priority, with the video given low prioriy and web high priority.
D. Round robin.

2 pt (c) Consider the situation of the previous question again; could Weighted Fair Queueing work?
If no, then explain why not.
If yes, then tell how the weights should be assigned and why that works.

2 pt (d) Consider a source (unrelated to the previous questions) sending packets of 1000 bytes each. The intervals between the packets alternate between 2 ms and 8 ms; so it sends packets at t = 0, 2, 10, 12, 20, 22, 30, ... ms.
Which (one or more) of the following token-bucket specifications (with 1 byte per token) does this source obey?

A. bucket size = 500 tokens; token rate = 2000 tokens/ms.
B. bucket size = 1000 tokens; token rate = 200 tokens/ms.
C. bucket size = 1500 tokens; token rate = 200 tokens/ms.
D. bucket size = 1500 tokens; token rate = 250 tokens/ms.
E. bucket size = 2000 tokens; token rate = 200 tokens/ms.
F. bucket size = 4000 tokens; token rate = 100 tokens/ms.
G. bucket size = 4000 tokens; token rate = 500 tokens/ms.

Continued on next page...

3. Security

2 pt (a) Complete the following table, in which we explore what information is hidden from an eavesdropper when using several security protocols; write "yes" or "no" (or abbreviate as "Y" or "N") in each table cell:

	What is hidden?		
	destination IP	**destination port**	**user data**
SSH			
SSL			
IPSec in tunnel mode			
IPSec in transport mode			

1 pt (b) When connecting to a server using SSH for the first time, your SSH client program may ask you to check the correctness of the "server key fingerprint", which is basically a hash of the key that the server just sent to your client. What authentication does this provide?

A. It authenticates you to the server.
B. It authenticates the server to you.
C. It authenticates the server and you both ways.
D. It does not provide authentication; this is the so-called "first-time risk".

1 pt (c) Cryptographic protocols often use timestamps to thwart replay attacks. One side sends a timestamp, the other encrypts or signs it and sends it back. As the book explains, there is no need for the clocks on both sides to be synchronized. However, do both clocks need to run at the same speed?

A. No, the only requirement is that both clocks run forward.
B. No, since timestamps take from each clock are only compared to that same clock later on.
C. Yes, otherwise an attacker could overload a server by running his/her own clock very fast.
D. Yes, otherwise an attacker could replay messages after an arbitrary time by running his/her own clock very slow.

2 pt (d) A company has a 50 Mbit/s Internet connection, and is suffering from a DDoS attack on its web server with a total volume of 300 Mbit/s. Can the company protect itself by setting some rules in its firewall (which sits between the Internet link and all of the company's hosts, including the web server)? If so, what rule(s)? If not, why not?

4. Localisation and timing

2 pt (a) All methods to achieve a synchronized time among networked nodes rely on message exchange between nodes. What are the message delay uncertainties that can happen, which can lead to inaccuracies in the synchronisation?

2 pt (b) In localisation ranging between nodes is commonly used. What techniques exist to determine the range between nodes that are wirelessly connected?

2 pt (c) DV-Hop is a method that allows a system wide positioning without the need for ranging. How does this protocol work?

End of this exam.