Network Systems (201600146/201600197), Test 4

April 5, 2019, 13:45–15:15

Answers

1. Congestion control

1 pt (a) F.

After a timeout, the congestion window is reset to 1, and the slowstart threshold set to half the previous congestion window. Then slow start (i.e., exponential increase) is performed until the slowstart threshold is reached; after the threshold, TCP does AIMD, i.e., linear growth. This is what curve F shows: exponential growth until the congestion window is half of what it was just before t_1.

1 pt (b) E.

1 pt (c) A.

1 pt (d) D.

1 pt (e) B.

If someone increases her congestion window faster than the others, she will get a larger share of the bandwidth, leaving less for the rest. This can be shown e.g. using the kind of argument used on slides 13 and 14 of lecture 13.

1 pt (f) E.

1 pt (g) F.

See section 6.5.4 in the book.

1 pt (h) Already at window W=10, the RTT increases, so queueing apparently occurs (there was no queueing at W=1 or 2 yet, since there the RTT was constant).
Thus, the increase in RTT from W=10 to W=20 must be solely due to queueing.
Apparently then, transmission of 10 packets takes 20 ms, that's a link speed of 10 × 1000 bits / 20 ms = 500 kbit/s.

Note that this exercise relates to the TCP Vegas exercise from the first tutorial in week 7.

1 pt (i) Extrapolating from the RTTs at W=10 and W=20, the queue would be empty at a window of 5 packets, so out of the 20 packets, 15 must be in the queue.

2. QoS

1 pt (a) B.

2 pt (b) Worst case, our source doesn't transmit for a while, allowing the bucket to fill up, and then empties the bucket to send a burst of maximum size, namely B, which equals 200 bits. Transmitting this at 2000 bits/s gives a maximum delay of $\frac{200}{2000} = 0.1$ s = 100 ms.

This is actually the simplest case of delay calculation in a token-bucket context. Many students made it far too complicated (and wrong), e.g. by also including the delay the packets undergo in the token-bucket system itself (that's wrong, because the token bucket model only *describes* the traffic), or by trying to include the tokens that accumulate over some time interval (yes, that was needed in last year's exam question, but not in this one).

1 pt (c) G.

Three sources like this can, on average, send up to $3r = 3000$ bits per second, which could overwhelm the link which may not be able to transmit more than 2000 bits per second. Thus, the queue could grow indefinitely, making a delay guarantee impossible.

1 pt (d) D.

Quite many chose option G, but that's not right. Both IntServ and DiffServ aim to give good QoS to flows that need it, they only differ in how they do it.

1 pt (e) B.

1 pt (f) C.

1 pt (g) C.

1 pt (h) D.

Fair queueing gives each flow a "right" to 50 Mbit/s, but if one flow doesn't use its entire share, the other flow can use the leftover. Since the total traffic offered by red and blue is less than the available 100 Mbit/s, both flows can send at their respective full speed, leaving 10 Mbit/s unused (so 10 % of the time, the link is idle).
The second most popular option was E, where the flows get 66 and 33 Mbit/s. This is a weird option (and wrong). What would it mean, letting the blue flow get 33 Mbit/s even though the blue source sends only 30 Mbit/s: where are the extra 3 Mbit/s coming from?

---

3. Security

1 pt (a) D.

You need ESP to actually encrypt the data (rather than just authenticate), and you need tunnel mode to access all machines on your home network.

1 pt (b) E.

Second most popular choice was F, but there is no "symmetric PGP key of a person". Symmetric keys need to be specific to a pair of communicating persons. If a person would publish "his" symmetric key, everyone could decrypt everything encrypted by that key, so it wouldn't give security.

1 pt (c) F.

See the book: SSH-TRANS authenticates the server to the client, and then SSH-AUTH authenticats the client to the server.

1 pt (d) Personal / Pre-Shared Key (PSK)
Extensible Authentication Protocol (EAP) / Enterprise mode / Authentication Server (AS) mode

2 pt (e) Mentioning at least two of the following aspects is enough for full points:

- EAP has a 'personal' credential check while PSK has the same key for everybody.
- Advantage of PSK is ease of deployment (esp. for simple home networks) while EAP allows more fine-grained control of who has access.
- Security in EAP is better (one reason is that the individual users cannot snoop on each other's traffic).

2 pt (f) HTTPS runs on port 443, so clearly you cannot directly access the bank's secure website.
However, port 22 is for SSH, and we've learnt that SSH can do "port forwarding". This requires having an SSH server outside the company's network, and then first setting up an SSH connection to that server, and configuring it to forward a TCP connection to port 443 of your bank.

1 pt (g) D.

1 pt (h) B.

If there's an outgoing connection with port 1234 as its source port, then the replies will be incoming packets to port 1234. The only way a firewall can distinguish between the packets to port 1234 which do and do not belong to that outgoing connection, is by keeping track of which outgoing connections there are. Therefore, a stateful firewall is needed.