Network Systems (201300179/201400431), Test 4

April 2, 2015, 13:45–15:15

Brief answers

1. TCP Congestion Control

2 pt (a)

By a timeout; it can't be a triple duplicate ack, since then the cwnd wouldn't be set to 1 but to ssthr.

4 pt (b)

For brevity, I'm omitting the time-sequence diagram here, but it was required for full score:
1st RTT: cwnd=1, 1 packet sent, 1 ack.
2nd RTT: cwnd=2; 2 packets sent and acked; total 3 packets so far.
3rd RTT: cwnd=4; total 7 packets so far.
Now we leave slow start and go to the linear increase of cwnd.:
4th RTT: cwnd=5; total 12 packets so far.
5th RTT: cwnd=6; total 18 packets so far.
So, it takes 5 RTTs for the first 15 datapackets.

2 pt (c)

For each received ACK, check how many *new* bytes it acknowledges (i.e., by how much its acknowledgement-number is higher than the previous ACK), and call this n. Then increase the cwnd by MSS × (n/CongestionWindow). As a result, ACKs for a total of cwnd bytes are needed for cwnd to advance by 1 MSS, regardless of how many ack packets are used for this.

> Some suggested increasing by $n\times$ (n/CongestionWindow), i.e., replace both MSSs in the formula by n, but that would increase the cwnd by only n bytes per RTT, rather than by MSS.
>
> Note that you were asked to give an *algorithm*. Therefore, vague descriptions like "only increase the cwnd when a full packet has been acknowledged" got only part of the points, since it is left rather unclear how to determine whether a full packet has been acknowledged.

2. QoS

2 pt (a)

Using only two flows is the solution here, because then the bandwidth is *equally* divided between VoIP and web, which is exactly what the network administrator wanted.

If we would treat each connection as a separate flow, the bandwidth would be divided equally among all connections, so if there are more many web connections than VoIP connections, the VoIP connections together would not get half the bandwidth.

> Many seem to have totally overlooked the introductory paragraph which told that the administrator wanted to reserve half the bandwidth for VoIP.

3 pt (b)

Worst case is that we're looking at the last packet of a maximum size VoIP burst. Such a burst can be at most 400 bytes long, consisting of 4 packets. Furthermore, the worst case is that just before this burst arrived, the router had just started transmitting a 1000 byte web packet. According to bit-by-bit round-robin, our 4 VoIP packets *would* finish *before* that web packet, but because the web packet won't be interrupted, all 4 VoIP packets will be scheduled immediately after the web packet. If more web packets arrive, they will be scheduled after the 4 VoIP packets, again because that's what bit-by-bit round-robin would do.
Thus, the maximum total delay is 1400 bytes at 10 Mbit/s, i.e., 1120 μs.

> Many students calculated the delay based on how long it would take for enough tokens to accumulate in the bucket. That's wrong, because we're not asking for the delay the token bucket device introduces: the token bucket system is only a *model* describing the *characteristics* of the VoIP traffic. See also slide 27 of lecture 14.

3. Security

3 pt (a)

Confidentiality: not. Dropping illegal packets doesn't really help confidentiality since those packets presumably didn't contain sensitive information anyway. (And if they were used by an intruder to smuggle out sensitive information, that intruder could just as well use a legitimate source address in his smuggling packets.)

Authentication: only a little bit. If *all* networks would use such filtering, then it would help more: then we could be sure that the source address in every packet correctly identifies from which network it comes (well assuming we trust the backbone routers not to manipulate the addresses). If only one network uses such filtering, then packets with source addresses in that network's range could still be coming from another network that doesn't have source address filtering yet.

Availability: yes, since it makes (D)DoS attacks harder; attacks made from this network now have to use a valid source address, so can be more easily tracked down, and can no longer use tricks like DNS amplification.

2 pt (b)

The client can be sure about the identity of the server, through the use of certificates signed by a CA that guarantees that the public key indeed belongs to this particular host. There is no authentication the other way around; in particular, the clients don't need to have public keys with certificates.

> The book tells that the underlying security protocol for HTTPS, namely TLS, has the possibility to authenticate both the client and the server. However, in actual HTTPS practice, client-side authentication is not used (or it is done through some other means at the application layer, such as username/password).

2 pt (c)

No. Indeed, a m.i.t.m. can replace the session key and the message, but the message in PGP also has a signature. The m.i.t.m. cannot generate the correct signature for his new message, since that requires the sender's private key; thus, the recipient will notice that something is wrong.

> Many claimed the m.i.t.m. can't decrypt the session key because that requires having the recipient's private key; that's true, but the question spoke about *replacing* the key.
>
> Many wrote that the message is *encrypted* with the recipient's public key; no, it's only *signed* with that key.

4. Time synchronization and localization

2 pt (a)

Although conceptually suitable, it is in practice not because 1) NTP assumes a rather stable connection to its timekeepers in a hierachical topology, 2) NTP is causing quite some communication overhead, and 3) NTP is consuming relative much processing power, all of which is typically not available on wireless sensor nodes.

3 pt (b)

ToF measures the transmission delay between sender and receiver, whereas TDoA is using one sender (with unknown position) and multiple receivers (with known position).

ToF has to measure pairs of devices, and each multiple times to achieve sufficient accuracy, whereas TDoA can do a measurement in one transmission. TDoA is thus more scalable and has less communication overhead. However, TDoA requires a highly synchronised network of receivers, limiting the scalability due to deployment issues.

3 pt (c)

(i) Basically the transmission times are high, in which time both the earth and the moon have changed their positions, and thus their relative distance.

(ii) The transmission times are high due to the use of acoustic transmissions. Additionally, due to the dynamic wireless channel the transmission delays will be impacted significantly, making the measurements less accurate.

End of this exam.