Network Systems (201300179), Test 4

April 4, 2014, 15:45–17:15

Brief answers

1. TCP congestion control

3 pt (a)

A can detect loss by means of a time-out. In that case it will reduce its sending rate to 1 MSS per RTT (congestion window of 1 MSS). Alternatively, A can detect loss after receiving 3 duplicate acknowledgements. In that case it will half it sending rate (half the congestion window).

2 pt (b)

If the initial packet from A is dropped, no successive packets are being sent to trigger the sending of (3) duplicate acks. So, this packet loss can only be detected by time out.

3 pt (c)

From the moment of packet loss (detection), the left side of the congestion window will not proceed until the acknowledgement for the retransmitted packet has been received. In the mean time, packets do leave the network, one for each ack received. In the course of 1 RTT, W packets will leave the network. In order to gradually decrease the number of packets in transit from W to W/2, a new packet must be sent for every two received acks. Since the left side of the congestion will not proceed, this means that the congestion windwo size must be increase by MSS/2 for every received ack. This also applies to the already received 3 duplicate acks, so at the moment of loss detection (3 dupl acks), the congestion window should already be incraesed with 3/2 MSS. At the moment the ack for the retranmitted packet is received, all of a sudden the left side of the congestion window will move with W MSS because the ack for the retransmitted packet will cumulatively acknowledge all W packets that have been received since the loss. To compensate for that, the congestion window size will have to be reset to W/2 at that moment.

---

2. Quality of Service

3 pt (a)

He could install a scheduler with WFQ, where the weight is such that a significant part of the bandwidth is for the ”high-priority” queue. The remaining capacity is then available for the other traffic. Alternatively, he can use priority queueing, but in that case, he has to install a policer that either shapes the traffic, or diverts excess traffic to the ”low-priority” queue. The policer could use a token bucket mechanism that uses a token generation rate that is below the link data rate.

2 pt (b)

In the worst case, bursts from all N sources are generated at the same time. If that is the case, the first byte of the burst can be transmitted immediately, since earlier bytes must have already been transmitted, since the token buffer is refilled slower than the link transmits data ($S > N \cdot r$). So, the maximum delay is the delay of the last packet of these bursts, which completes transmission after all $N \cdot B$ bytes have been transmitted, i.e., after $N \cdot B/S$ seconds

3 pt (c)

In the worst case, both high-priority sources send a burst of size B to the high priority queue at the same time, and also a low-priority packet has just started service. The last packet in the buffer (from one of the two high priority flows will finish transmission after $(2 \cdot B + 1)/S$ seconds.

---

3. Security

3 pt (a)

Verifying that a (public) key indeed belongs to the person/site it is used for.

2 pt (b)

Stateless is sufficient: just drop all outgoing packets whose source address is wrong.

3 pt (c)

Of course, there are many solutions.
To automate the keys, one could derive them from a master key, e.g., hash of masterkey and a sequence number.
To make sure nodes know which key to use right now, one could tie this to time (e.g., change exactly every second) or periodically announce the current key sequence number in an unencrypted packet.
But solutions where the last packet of the previous key announces what key to use next, can't work reliably as packets (e.g., a packet that announces that the key must be changed) can get lost. Also, unencryptedly broadcasting the new key itself (rather than e.g. a sequence number) is not a good idea, since evedroppers then would also learn the new key.

---

4. Time synchronization & localization

2 pt (a)

(1) Accuracy is the measure how close a measured/calculated value towards is actual value, and precision is a measure how much deviation is between the measured/calculated values.(2) Precision can be increased by adding more beacons and by having a faster duty cycle.

3 pt (b)

(1) APIT cannot operate with only one blind node, as it estimates its position based upon other blind nodes. What it then can do is mainly relying on distance estimates from the received beacons. (2) APIT operates by making a triangle between the beacons. This implies that a blind node cannot be located outside of the convex hull, but instead will be located on the convex hull.

3 pt (c)

(1) Using dead reckoning with the accelerometer and/or gyroscope; (2) Using fingerprinting of all available radio signals (wifi, GSM, etc.); (3) Using knowledge of the environment, like available maps, or making use of SLAM techniques to make maps and determine possible positions.