Computer Networks: Assignment 6

8 May 2016

Solving and submitting your assignment

Requirements about the delivery of this assignment:

- Submit via Blackboard (`http://blackboard.ru.nl`);
- Upload one pdf file for written answers and all supplemental files in a single zip file;
- The file should take the name of your student number, for example student *s0123456* should submit a file named *s0123456.pdf*.
- Write both your name and student number into the document (and only your student number in the filename).

Deadline: Wednesday, May 17, 20:00 p.m. sharp!

Goals: After completing these exercises successfully you should be able to:

- understand how reliability is implemented by various protocols
- better grasp with the TCP state model
- know how TCP congestion control operates

Marks: You will be graded with marks from 0 to 3 where 0 means not serious, 1 means serious but insufficient, 2 means sufficient and 3 means good. You can have at most 1 assignment graded 0. To get 1 or more, you MUST attempt to solve ALL exercises, even if the provided solution is not correct/complete. In other words, leaving an exercise out automatically turns your grade to 0. In your solution, please explain all answers clearly and concisely.

1 Reliable data transfer

Compare GBN, SR, and TCP (no delayed ACK). Assume that the timeout values for all three protocols are sufficiently long such that 6 consecutive data segments and their corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (Host A) respectively. Also assume large sender/receive windows. Suppose Host A sends 6 data segments to Host B, and the 3rd segment (sent from A) is lost. In the end, all 6 data segments have been correctly received by Host B.

a) How many segments does Host A send in total and how many ACKs does Host B send in total? What are their sequence numbers? Answer this question for all three protocols. Refer to packets and their ACKs by their sequence numbers, which range from 1 to 6.

b) If the timeout values for all three protocol are much longer than 6 RTT, then which protocol successfully delivers all five data segments in the shortest time interval? Explain your answer.

2 TCP State Machine

Figure 2 shows an *incomplete* state machine of an actual TCP client where the input and output messages of each transition are packets comprising flags, sequence and acknowledgement numbers, or high level socket calls. Packets carry no data and their flags are shorthanded by their starting letters. Numbers other than 0 are all relative. The socket calls are `connect` which connects to a TCP server and `close`, which closes the data connection socket. In practice, packet inputs are sent by the communicating server while socket calls are sent by the application. [1]

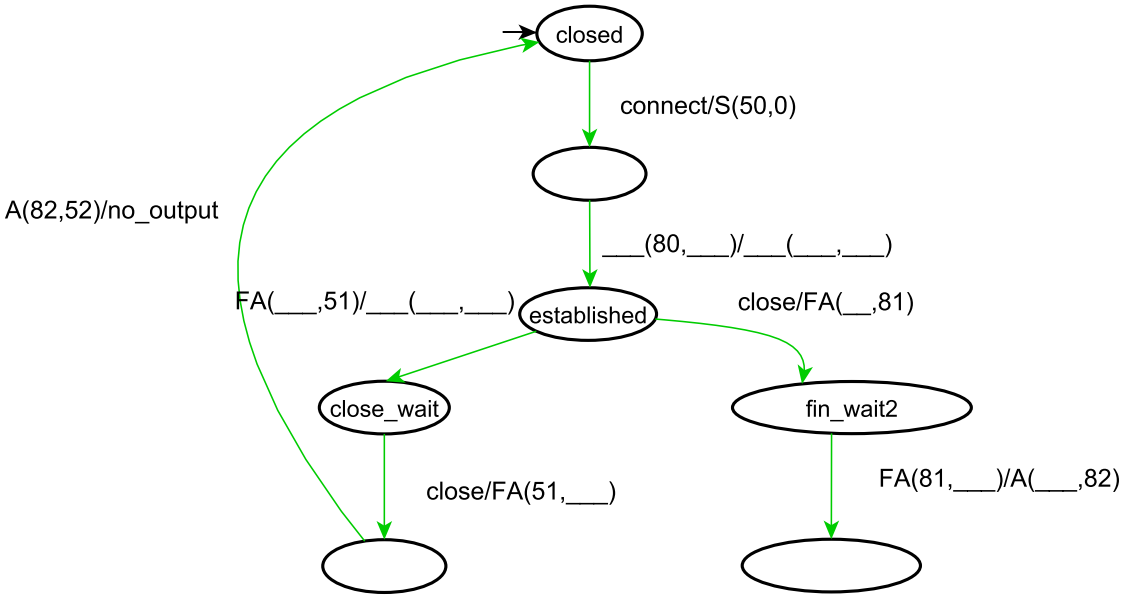


Figure 1: TCP Client State Machine

a) Complete the state model with the missing flags, state names and sequence numbers. Some of this information has been filled to help you out. For flags and state information, you can use the state machine RFC 793. To calculate numbers, think of packets carrying FIN and SYN flags as packets carrying data of one length. Acknowledgement numbers should acknowledge this data.

b) For each client state in the model, name the corresponding server state. Assume the server has processed the output of the transition leading up to the client state. Also assume no packet loss and normal behavior of the server (no abrupt termination, no denial of service...). The server starts in the `listening` state, thus for `closed`, the associated server state is `listening`. *(hint: Use the RFC state machine.)*

c) Define a resetting transition from the `established` state to the `closed` state. It is enough to only mention the input. *(hint: The transition could be made be sending a valid reset packet to the client. Look up Reset Processing in the RFC 793 on how to generate it.)*

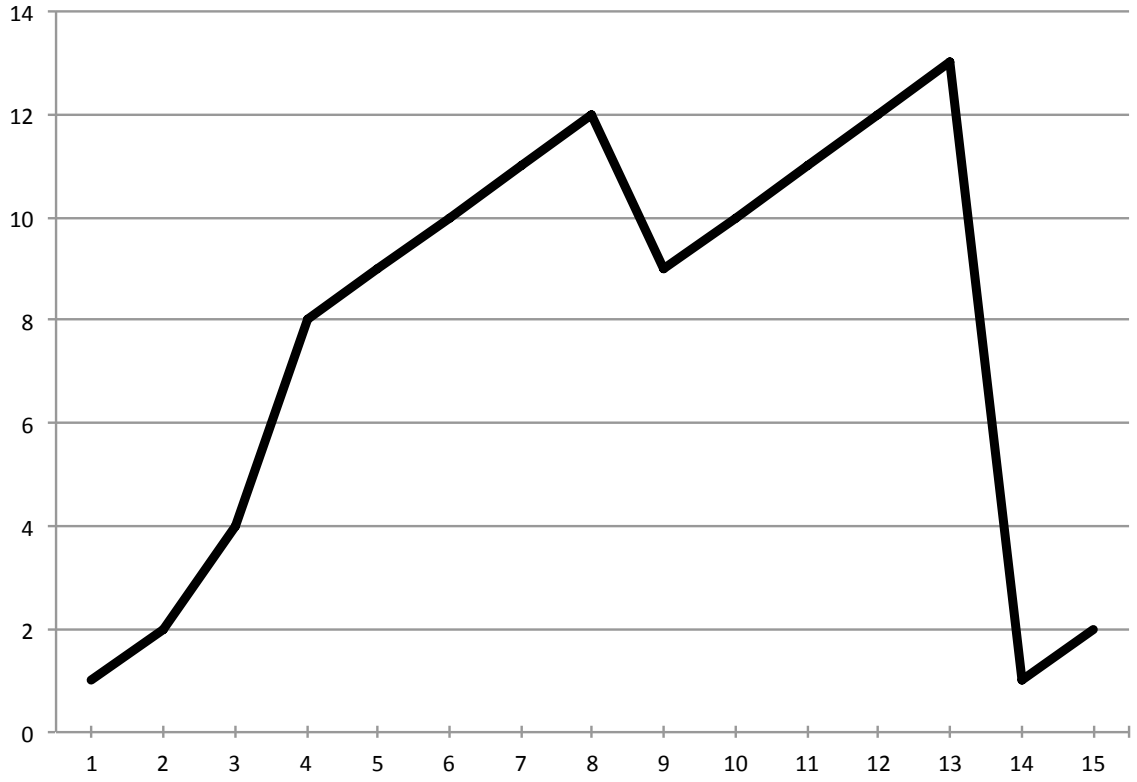
[1] You might wonder why the close action causes a FA packet instead of a F and why it does a direct transition to the `fin_wait_2` state. In practice, `close` actions are implemented as half duplex, that is, they close both the sending end and the receiving end of a connection, bypassing the `fin_wait_1` state (where the receiving end would still be open). Also, if you follow the RFC 793 event flow, you will see that packets containing only FIN and no ACK are dropped.

d) The model does not describe time behavior. Name a state in which if the client does not receive any input, it will eventually change state. Explain why it does so.

(hint: When in a non-synchronized state (`syn_sent`, `last_ack`, `fin_wait2`), the client is expecting an acknowledgement for a packet it has previously sent. This acknowledgement might not come. Also, there are transitions in the RFC 793 reference model triggered on a timeout.)

3 Congestion control evolution

Figure 3 describes the evolution of the congestion window size of a particular TCP implementation. The horizontal axis describes the transmission round in RTTs (round-trip-time), the vertical axis describes the congestion window size (or `cwnd`) in segments. Answer the following questions. Support each answer by an explanation. A problem similar to this one is solved online. The solution in the form of a video is found on the Student Resources site. You are advised to watch that video before solving this problem (if you haven't already).



a) Is this evolution characteristic of the Reno or of the Tahoe implementation of TCP congestion control? (*Hint:* While Reno distinguishes loss types, Tahoe does not.)

b) Identify the intervals of time when TCP slow start is operating.

c) Identify the intervals of time when TCP congestion avoidance is operating.

d) After the 13th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout? Why?

e) What is the initial value of `ssthresh` at the first transmission round?

f) What is the value of `ssthresh` at the 8th transmission round?

g) What will be the congestion window size in the 16th round if no loss happens?

h) Assuming a packet loss is detected after the 15th round by the receipt of a triple duplicate ACK, what will be the values of the congestion window size and of `ssthresh`?

i) During what transmission round is the 17th segment sent?