TDTS06: Lab 3 – Retransmission and handling lost packets

Juha Takkinen, Ph.D.

IDA, Dept. or Computer and Information Science

1.0 Overview

Your goal in lab 3 is to improve on the protocol from lab 2 so that it can handle lost packets as well (and not only corrupted packets). The protocol in lab 3 corresponds to the protocol rdt3.0 as described in the textbook (read the textbook and see Figure 3.15 in Chapter 3).

Your protocol will handle lost packets by using a retransmission timer. When timer times out, the sender will retransmit the lost packet.

The network will by fully unreliable, which means that it will both loose, delay and corrupt packets.

You must also, again, answer a number of follow-up questions concerning the lab assignment.

2.0 What to do first

Copy the files from the lab2 directory to the lab3 directory.

Then, update the protocol stack so that it includes your protocol that you will develop in lab 3: edit the graph.comp file and replace each occurrence of lab2 with lab3.

Make sure that your new copy of the protocol in the lab3 directory still works by recompiling x-kernel and running the server and the client as in lab 2.

3.0 Lab requirements

3.1 Retransmission timer

Because packets can be lost or delayed for a very long time, you must also implement a retransmission function. This function will retransmit the old packet when a timer expires. Specifically this means temporarily storing a copy of each sent packet, starting a timer each time a packet is sent, and then stopping the timer when it is known that the packet has been received correctly by the server.

Retransmissions are scheduled as events in x-kernel: use evSchedule to schedule the retransmission event. One of the parameters to evSchedule is the length of the timer in milliseconds. Another one of the parameters to evSchedule is the name of the function that is to be called when the timer expires, typically named “retransmit” or something similar. As said above, you must implement this function so that it is called and a previously saved packet can be retransmitted when the timer expires because you have not received the ACK that you were waiting for. In order to cancel the scheduled event, for example when an ACK was received and everything went well, use evCancel.

The sender has to save the last sent packet for possible retransmissions. Be aware that after pushing a message (using the xPush(&msg) function) the handle to that message is no more valid ... Therefore you need to make a copy (for retransmission) of the data you send before calling xPush. Note: ACKs are not retransmitted!

You may optionally implement a flexible timer that is set to a variable timeout value depending on the number of retransmissions. See examples from TCP’s retransmission timer in the textbook.

4.0 Testing the protocol

Configure the protocol stack in graph.comp so that it includes all three virtual protocols VDISTORT, VDELAY and VDROP. However, it is recommended that you test with only one virtual protocol at a time, to ensure that the correct mechanisms are in place. Add one virtual protocol at a time.

5.0 Follow-up questions

When you have finished coding and testing the protocol, you must answer the following questions:

1. Draw the FSM for the receiver side of the protocol.
2. Consider the retransmission timer as used in TCP and the estimation of RTT in the calculation. How often does TCP measure the sample RTT?

Explain all of your answers and list correct sources, such as RFCs, if any.

6.0 Demonstrating the solution

Before handing in a lab report you must first demonstrate the resulting protocol to your lab assistant. Remember to remove unnecessary trace printouts and only keep the most important ones. You are recommended to slow down the execution of your protocol, using delays, for an even clearer demonstration.

Before you demonstrate your solution to the lab assistant, give him/her a copy of the code on paper.

In the demonstration you must show that your protocol works according to the requirements, that is, the protocol can handle a partly unreliable channel (vdistort installed), which is shown by clear trace printouts of the packets that are corrupted and which ones are received correctly. Be prepared to answer questions on specifics in the code and your solution.

When your lab assistant has seen your demo and approved of the functionality, you can hand in a lab report containing the source code of the changed files on paper (see “Coding guidelines” on the course web site) and also the answers to all of the follow-up questions.

You must finish lab 3 before continuing with lab 4. When you have demonstrated lab 3 for your lab assistant, you can immediately continue with lab 4, while waiting for lab 3 to be passed by your lab assistant.