

# TDTS06: Lab 2 – Sequence numbers and a NAK-free protocol

---

**Juha Takkinen, Ph.D.**

IDA, Dept. of Computer and Information  
Science

---

## 1.0 Overview

---

Your goal in lab 2 is to improve on the protocol from lab 1 so that it can handle corrupted control packets as well (and not only corrupted DATA packets). The protocol in lab 2 corresponds to the protocol rdt2.2 as described in the textbook (read the textbook and see Figure 3.13 in Chapter 3).

Your protocol will put sequence numbers on all packets and only use ACKs (no NAKs), in addition to the flow control and error detection functionality that you have already developed in lab 1. Also, ACKs will carry a checksum.

The network will be configured identical to the one in lab 1, that is, as partly unreliable, which means that it will deliver all packets but some of them will be corrupt when they arrive at the receiving side.

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 lab1 directory to the lab2 directory.

Then, update the protocol stack so that it includes your protocol that you will develop in lab 2: edit the `graph.comp` file and replace each occurrence of `lab1` with `lab2`.

Make sure that your new copy of the protocol in the `lab2` directory still works by recompiling `x-kernel` and running the server and the client as in lab 1.

---

## 3.0 Lab requirements

---

### 3.1 NAK-free protocol

Remove NAK from the protocol header and change your code so that the server only uses ACKs when responding to packets from the client.

### 3.2 Checksum on ACKs

Add a checksum field also to the ACKs and the code in the server client that can handle the calculation of the checksum.

### 3.3 Sequence numbers

Add sequence numbers to your DATA packets and code to keep track of the correct numbers being sent and received. Also, each ACK must contain information about what sequence number it acknowledges.

---

## 4.0 Follow-up questions

---

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

1. Repeat the experiments that you did in question 1 in lab 1, that is, vary the size of messages and also the size of the transmitted file. Briefly discuss the result, comparing it with lab 1.
2. The ACK packets flowing from the server to the client do not have sequence numbers, although they do have an ACK field that contains the sequence number of the DATA packet being acknowledged. Why is it that the ACK packets do not require sequence numbers?

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

## 5.0 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 2 before continuing with lab 3. When you have demonstrated lab 2 for your lab assistant, you can immediately continue with lab 3, while waiting for lab 2 to be passed by your lab assistant.