Lab 9: CRC Error Detection

Objectives

- Practice client/server programming.
- Understand the CRC Error Detection Algorithm

You may complete this lab individually or with one colleague from this class using pair programming rules (see the course Canvas page).

1 Introduction

In this lab you will incorporate an error detection mechanism for data sent over the network. You will use the CRC algorithm that is widely used in modern protocols.

2 The Idea

The idea behind error detection is as follows. Both the sender and the receiver have a predetermined method of a special check value for the transmitted data. Before the data is sent, the sender computes the check value. This check value is appended to the data and transmitted along with the data. When the receiver receives the augmented data, it separates the data and the check value and recomputes the check value based on the data it received. If the received check value matches the computed check value, the data is assumed to have been transmitted correctly.

3 Assignment (20 Points)

Incorporating CRC into Client/Server Communication

You will use the starting client and server programs from the previous lab. In this lab, the client will always be the sender, and the server will always be the receiver.

The data to be transmitted by a client will be a sequence of 0's and 1's entered by a user. The client will have to read in the data, compute the check value, form and send a new message. The server will receive a message, validate it and print the result of validation – ok if data transmitted correctly and error message if an error is detected.

In our case, communication between client and server is reliable and no errors will be introduced during actual transmission. How shall we test that the error detection mechanism is working? Before you send the data, but after the check value is computed, you should with probability 0.3 flip one randomly chosen bit in the data (if this flipping occurs, print the original data and modified data at the client side). You will join the erroneous data and the original check value and send them to the server who will then should be able to detect the error.

CMPS 2301 Introduction to Computer Systems & Networking

Lab 9: CRC Error Detection

The format of transmitted message is up to you, i.e. you can glue data and check value in any way that allows you to split them correctly at the server side. The user's data and check value should travel together (in one joint message).

Phases of Development

There are two principle steps in implementing error-detection mechanism: 1) implementing the algorithm for computing a check value using CRC algorithm given a message and 2) making the client and server communicate with augmented messages and performing message validation.

Part 2: Client/Server Communication (10 Points)

Implementing Client/Server communication is independent from the algorithm used for computing check value. You may find it easier to begin with implementing part 2). If you wish, you can even test it with another check value algorithm (for instance, use the sum of all bits in the message as a check value).

Part 1: The CRC Algorithm (10 Points)

The CRC algorithm has been implemented for you, but during transmission, all lines containing comments were lost. Your first task in using CRC algorithm will be annotating each line in crc.c file with a comment explaining what the line is doing.

About CRC

You can read more about CRC on Wikipedia http://en.wikipedia.org/wiki/Cyclic_redundancy_check or in Ch. 2.4.3. You won't need to implement dividing polynomials for this lab, but you should read 2.4.3 to understand how to intelligently use the provided code.

Submission

Upload your client, server, and annotated crc.c files to Canvas.