Computer Communications A3

Joakim Hjertholm (jhj010)

Introduction

In this project we are going to ensure reliable data transfer from the transport layer in the OSI-stack model to the application layer above. We will implement the Go-Back-N (GBN) which is the default mode in which TCP operates.

Background

OSI-model:

is a conceptual model that characterizes and standardizes the communication functions of a telecommunication or computing system without regard to its underlying internal structure and technology. Its goal is the interoperability of diverse communication systems with standard communication protocols.

Reliable Data Transport:

Is a network protocol that's meant to provide facilities for remote loading, debugging and bulk transfer of images and data. It is located on the Transport layer of the OSI model.

Design and Implementation

Initially when sending data from one source to another using TCP, there are several layers which we pass through before sending and then receiving an acknowledgment packet. But we are going to focus on transport and network layer. The data that is being transferred are placed into data packets which also contains a sequence number and acknowledgment state. This helps us differentiate between the data chunks and check if it's a data packet or acknowledgement packet.

When there is a request to send data from Alice the transport layer will place the data which it got from the application layer, into a packet and send the packet to the network layer. The network layer in our case will send the packet to Bob`s network layer and into his transport layer. Here the packet is gets checked to see if it's sequence number matchets the expected one, if so the data is stored and a new acknowledgement packet is sent to Alice. If the sequence numbers didn't match, we send back an acknowledgement packet with the sequence number of the one before Bob`s expected one. This tells Alice`s transport layer that there has happened packet loss, and all the packets already sent will be resent to ensure that none is lost.

One can also encounter corruption, this is when the data changes from its original form to another. This can occur while sending the data to the receiver, the solution we used where

to calculate a check sum which is then stored in the packet. If the receiver calculates the check sum again and it doesn't match with the store sum, we can conclude that the data is corrupted, and we send back an acknowledgment packet with the sequence number of the one before the expected.

After a packet is delayed with our implementation (GBN) the packet will arrive at the destination late, but will not impact the result as a timeout will be called and all the packets will be resent.

Evaluation

With no implications the implementation runs smoothly, it's only when we apply either a drop chance of 20% or delay chance of 20% the implementation runs to a halt. Implying that sometimes the packets are not resent or that some packets are acknowledged wrongly.

I have tried to implement a fix for the corruption aspect, but had some issues with the data being sent in as an int. I couldn't find the reason in time.

Conclusion

We have ensured reliable data transfer with little or no implications from the transport layer in the OSI-stack model. We implemented the Go-Back-N, but it doesn't work on corrupted data and a lot of packet loss or delay.

Sources

https://en.wikipedia.org/wiki/OSI_model

https://en.wikipedia.org/wiki/Reliable_Data_Protocol