Data Networking

Layering and Protocol Design for Milestone 2

Alexander Bunte (2518686)
Jonas Bürse (2519546)
Stefan Tombers (2521218)

September 20, 2011

1 Introduction

The goal of this project is to transfer textfiles as sequence of characters through a network with reliable data transfer. In this paper we will present our layering and protocol design of milestone 2.

2 Layering and Protocol Design

Beside the two layers, physical and application layer, cnet provides we implement three additional layers between them. We call them *link layer*, *network layer*, and *transport layer*, from bottom to top. Data packets in the link layer are called *frame*, in the network layer *datagram* and in the transport layer *segment*. To data packets from the application layer we refer to as *messages*. Figure 1 gives an overview how the different layers call each other.

2.1 Link Layer

The link layer provides basic "point2point" connections between neighboring nodes.

2.1.1 Framementation

It can handle the transmission of arbitrary large packets (limited by the buffer size, which is large enough to handle all messages in this scenario) independently of the MTU of the link because it has the ability of fra(g)mentation. This means that data from the upper layer is split into several frames, which fit to the MTU, and put together on the receiver side. Figure 2 illustrates the receiver side of the link layer. The ordering of the frames is checked by sequence numbers and identification numbers (ID). We use an additional *isLast* flag to identify when a datagram ends. This mechanism is guarded by the fact that no reordering of frames on one link occur.

2.1.2 Corruption Detection

This layer also guarantees that all datagrams which are handed to the upper layer are *free of corruption*. Corrupt frames are recognized by computing a checksum and all frames with errors are dropped. For milestone 3 it is also possible to implement error correction in this layer. Finally, the data in this layer are handed to the physical layer in a first come first serve fairness.

2.2 Network Layer

In the second milestone the network layer has no function. It only hands the data from the upper layer to the lower one and vice versa. It will get the function of maintaining and building a routing / forwarding table as well as route datagrams through the network.

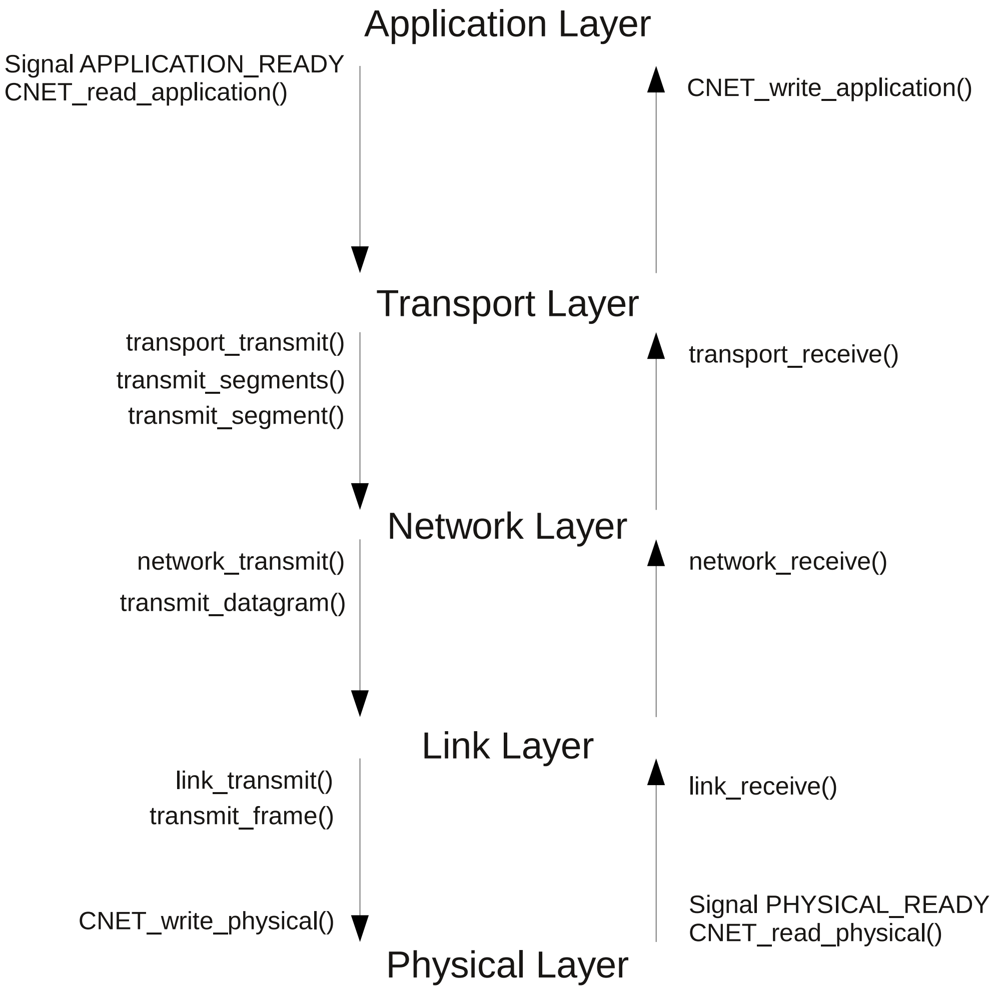


Figure 1: Overview Layers. Overview of the call hierarchy of the different layers.

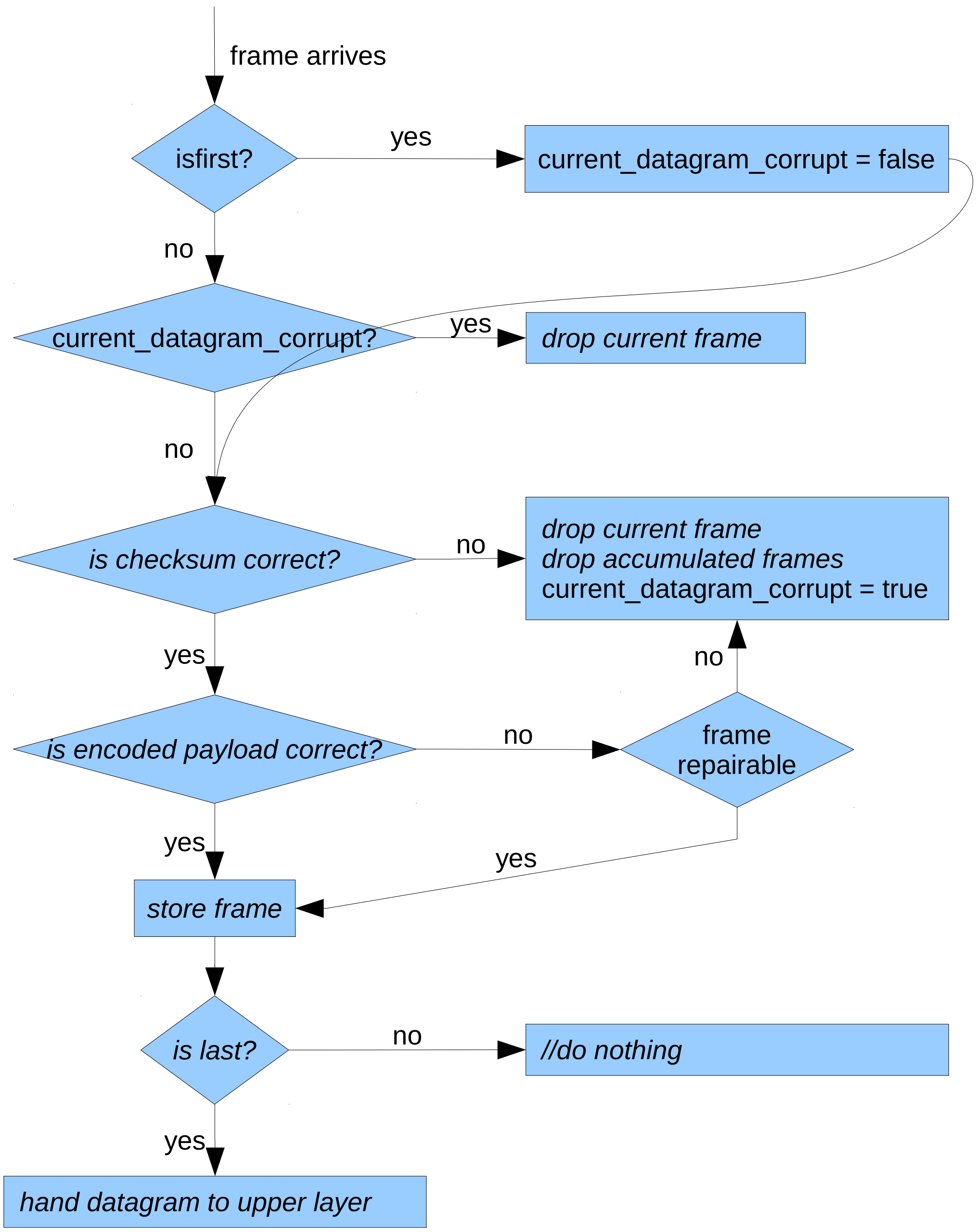


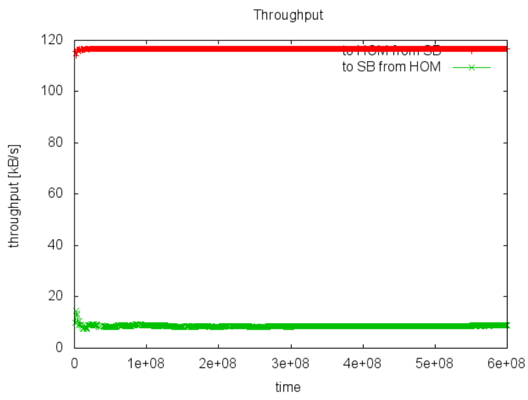
Figure 2: Receiver side of link layer. The flow graph shows how data on the receiver side of the link layer are handled.

2.3 Transport Layer

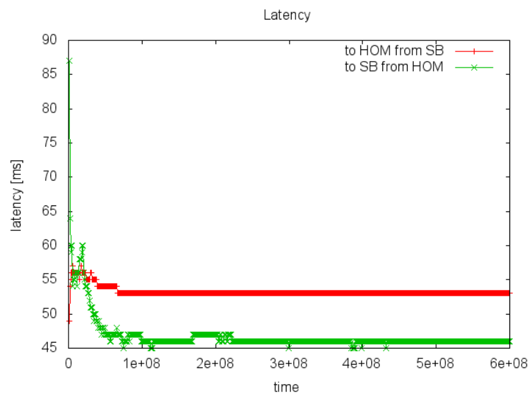
In the second milestone the transport layer has no function. It only hands the data from the upper layer to the lower one and vice versa. It will get the function to guarantee reliable data transfer, congestion and flow control and to segment messages to decrease the number of resend messages in case of corruption. To improve throughput it will buffer reordered messages.

3 Results

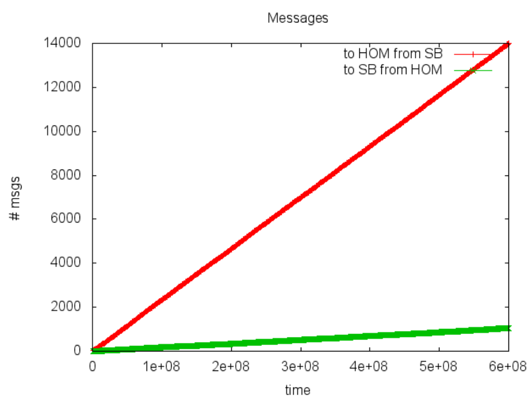
Our analysis shows that throughput and latency stays constant after a short period of time and the number of send / received messages increases linearly as assumed. Figure 3 shows the behavior of the `saarnet2.txt` network for a simulation period of 10 min. The fluctuations at the beginning are due to the not filled queues after initialization. With our implementation we reach a throughput from Homburg to Saarbrücken of 9.1 kB/s and 116.8 kB/s the other way around.



(a) Throughput



(b) Latency



(c) Messages

Figure 3: Analysis of milestone 2. The plots show the behavior for a 10 min simulation of `saarnet2.txt`.