Protocol Stack - Filoverførsel

TCP/IP SOCKET PROGRAMMING

Deltagere

201509378 – Niklas Meyer Møller Sørensen

201507686 – Jeppe Traberg Sørensen

20101979 – Henrik Søby Jørgensen

16. Maj 2017

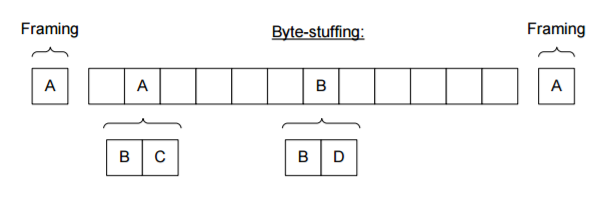
## Problemformulering:

Der skal udvikles en mulighed for at overføre en fil vha. den serielle kommunikationsport på en virtuel maskine. Systemet skal kunne overføre en fil af vilkårlig størrelse fra en virtuel maskine til en anden. Dette skal gøres via to applikationer en server og en client.

Clienten skal fortælle serveren hvad fil som skal overføres og serveren skal sende den pågældende fil tilbage med 1000bytes af gangen.

# Datalag - Link lag

Linklaget skal implementeres som en SLIP protokol. Karakteren ’A’ vil bliver brugt som start og stop karakter, også kaldet Framing. Dette betyder at når et rigtig ’A’ sendes skal dette erstattes med BC og ’B’ skal erstattes med BD.



I linklagets send funktion bliver et Byte[] kaldet sendBuff. Dette bruge stil at gemme det nye kodede byte array som skal sendes. Den buffer som ønskes sendt bliver løbet et foreach og de pågældende chars bliver udskiftet. Sidst bliver det nye array sendt via en SerialPort.

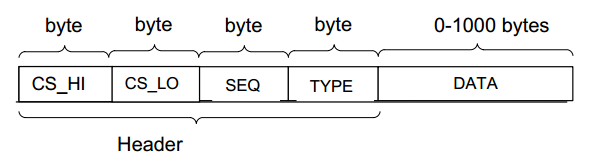


Recieve() i linklaget vil modtage en kodet buffer som skal dekodes. Her er et udsnit af det loop som læser af den medtagede buffer. Delimiter er A og er allerede taget ud af bufferen. Dette gør at der nu kan løbes igennem hele bufferen indtil det næste A bliver fundet. Under dette gennemløb bliver bufferen bliver dekodet og gemt i et byte[] temp.



# Transportlag

I transportlaget skal der implementeres en stop and wait protokol. Denne skal indeholde en 16 bit internet checksum så det kan detekteres hvis der opstår fejl. Der skal opsættes en payload på 1000 bytes og ligeledes overholde header formaterne:

[[1]](#footnote-1)

Her er CS\_HI og CS\_LO henholdsvis den mest betydende del af en checksumsberegning og mindst betydende del af checksumsberegning.   
SEQ er et sekvensnummer på det afsendte segment og typen indeholder enten et 1 eller 0. 1 for en ACK besked og 0 for vores data. Data indeholder vores 1000 byte payload.

Send



Recieve



# Applikationslag

### FileServer:

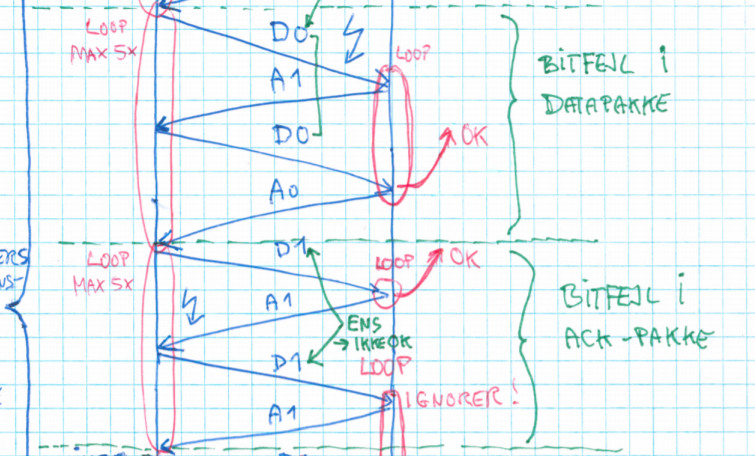


## FileClient:

## 

# Test

Vi har gennemtvunget nogle fejl for at kunne teste når der sker i fejl i Acknowledge og bitfejl i transportlaget. Fejlene er blevet introduceret ved hjælp af det udleverede materiale[[2]](#footnote-2) Skal simulere disse forskellige fejl

[[3]](#footnote-3)

Retultatet af disse er:

**INSÆT BILLEDE HER:**

# Konklusion

1. Øvelse-121.pdf [↑](#footnote-ref-1)
2. https://bb.au.dk/bbcswebdav/pid-755684-dt-content-rid-2002039\_1/xid-2002039\_1 [↑](#footnote-ref-2)
3. https://bb.au.dk/bbcswebdav/pid-755684-dt-content-rid-1969865\_1/xid-1969865\_1 [↑](#footnote-ref-3)