

Tin Can Skype

RB-MRO3 - Gruppe 3

Uddannelse og semester:

Robotteknologi - 3. semester

Afleveringsdato:

18. December 2015

Vejleder:

Ib Refer (refer@mami.sdu.dk)

Gruppemedlemmer:

Anders Ellinge (aelli14@student.sdu.dk)

Anders Fredensborg Rasmussen (andra14@student.sdu.dk)

Daniel Holst Hviid (dahvi14@student.sdu.dk)

Mathias Elbæk Gregersen (magre14@student.sdu.dk)

Rasmus Skjerning Nielsen (rasni14@student.sdu.dk)

René Tidemand Haagensen (rehaa14@student.sdu.dk)

Sarah Darmer Rasmussen (srasm14@student.sdu.dk)



*Det Tekniske Fakultet
Syddansk Universitet*

1 Abstract

2 Forord

Denne rapport er udarbejdet af gruppe 3, på andet semester på Civilingenør i Robotteknologi på Syddansk Universitet. Rapporten er blevet skrevet i forbindelse med dette semesters projekt og beskriver hvordan denne gruppe har valgt at løse opgaverne i det valgte projekt, "Tin Can Skype", som er et chatprogram, der bruger DTMF-toner og indeholder bla. et simpelt log-in og historik system.

Formålet med denne rapport er, at læseren skal være i stand til at læse og forstå projektet ved blot at have grundlæggende viden om C++ og datakommunikation, og ved at læse rapporten.

I forbindelse med dette projekt, blev følgende udstyr stillet til rådighed:

- To mikrofoner
- To højtalere

Indhold

1	Abstract	2
2	Forord	3
3	Indledning	5
3.1	Projektbeskrivelse	5
3.1.1	Krav til produktet	5
3.1.2	Metodebeskrivelse	6
3.1.3	Afgrænsning	6
3.2	Workload (product backlog)	7
4	Det Fysiske Lag	8
4.1	Fysisk Lag Ind	8
4.1.1	Teori	8
5	Data Link Laget	10
5.1	Teori	10
6	Kontrolfunktioner	11
7	Konklusion	12
8	Perspektivering	13
9	Litteraturliste	14
9.1	Bøger	14
9.2	Hjemmesider	14

3 Indledning

3.1 Projektbeskrivelse

I dette projekt er to højtalere og to mikrofoner blevet stillet til rådighed. Formålet med dette projekt er, at kunne sende data vha. DTMF-toner.

Det valgte projekt er et chatprogram, der udvikles i C++, og skal have de primære funktioner:

- Overførsel af tekst.
- Log-in funktioner.
- Historik af chat-samtale.

Desuden er disse sekundære funktioner blevet overvejet:

- Filoverførsel
- Gruppe chat
- Spil
- Redigering af tidligere beskeder
- Video streamings funktioner
- Humørikoner

Herudover er der desuden blevet overvejet at bruge en tredje computer, som kan bruges som en server. Her vil mindst to computere altså være i stand til at kommunikere med hinanden vha. DTMF-toner.

3.1.1 Krav til produktet

Følgende krav blev stillet til projektet:

- Bærbare computere skal kommunikere med hinanden, eller evt. et embedded system, ved udveksling af lyd
- Der skal anvendes DTMF toner, og der skal designes en kommunikationsprotokol
- Der skal udvikles en distribueret applikation i C++
- Der skal anvendes en lagdelt softwarearkitektur
- Arkitekturen kunne være client/server med f.eks. tykke klienter

For at fuldføre dette projekt skal der anvendes to computere som skal være i stand til at kommunikere med hinanden ved hjælp af lyd i form af DTMF-toner. Derudover skal dette programmeres i C++, her bruges klasser.

3.1.2 Metodebeskrivelse

Vi har i dette projekt valgt at benytte SCRUM, da alle gruppe medlemmer således er i stand til at arbejde med den metode der passer dem bedst. Vi har valgt at bruge brainstorm, som vores primære form for idégenererings-teknik. Desuden prøver gruppen så vidt muligt at beregne alle de ting, der kan beregnes på forhånd.

3.1.3 Afgrænsning

Her ses de emner, som gruppen har overvejet at arbejde med. De primære funktioner er de funktioner som skal løses først, mens de sekundære løses efter tidsbegrænsning.

Primære funktioner:

- Overførsel af tekst
 - Protokol
 - Karakter definition
 - Størrelse
- Historik
 - Tidspunkt
 - Størrelse

- Log-in

Sekundære funktioner:

- Fil-overførsel
 - Protokol
 - Queue
- Gruppe chat
 - Protokol
- Spil database
 - Protokol
 - Funktion
- Rediger tidligere beskeder
 - Protokol
 - Funktion
- Stream funktion
 - Protokol
 - Funktion

- Humørikoner
Char def.
Database
- GUI
Agil
- Sky "server"
Funktion

3.2 Workload (product backlog)

Der laves en product backlog i stedet for en tidplan (se figur 1).

Requirement	Status	Priority	Estimate (Hr)	
Overførsel af tekst	Not started	1	70	
Log-in	Not started	1	40	
Historik	Not started	1	40	
Fil-overførsel	Not started	2	120	
Gui	Not started	2	70	
Rediger tidligere besked	Not started	3	40	
Gruppe chat	Not started	3	40	
Streaming	Not started	4	120	
Cloud/server (tredje computer)	Not started	4	100	
Smileys	Not started	4	30	
Spil	Not started	4	100	
		I alt	770	

Figur 1: Product backlog

En product backlog er et værktøj inden for metoden SCRUM, som viser hvor langt tid en opgaver tager i mandetimer og hvilken status opgaven har (Not started, In process og Finished).

4 Det Fysiske Lag

4.1 Fysisk Lag Ind

dsf
dkdk idkfgfdggfd

4.1.1 Teori

Send-and-Wait protokol

Stop-and-Wait er et special tilfælde af Go-Back-N. Stop-and-Wait har to sekvensnumre, mens Go-Back-N har flere

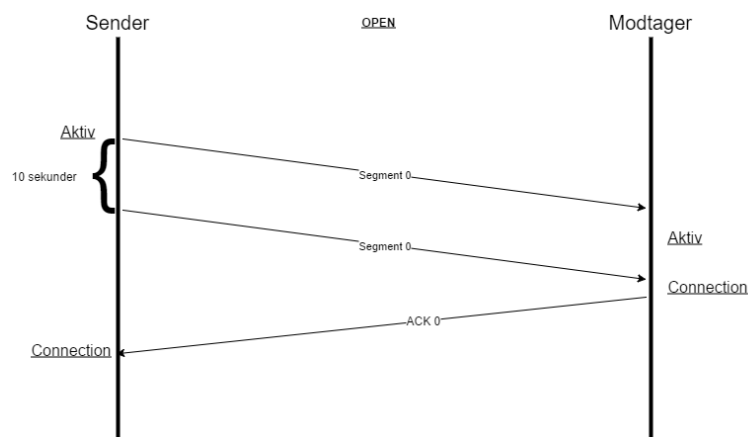
I stop-and-wait protokollen tilføjes en fejl mekanisme, hvor der tilføjes redundante bit for at finde og rette fejl i sendte pakker, dette gøres ved hjælp af CRC checket, som forklares senere.

Ved hjælp af sekvensnummerering af pakkerne, er modtageren i stand til at konstatere om den modtagne pakke er den korrekte, eller om den har et forkert nummer i forhold til rækkefølgen.

Automatic repeat request (ARQ) benyttes når mistede eller fejlbehæftede pakker skal retransmitteres. Pakker retransmitteres hvis en ny ACK ikke er modtaget inden timeren udløber.

Open

Som der ses i figur 2, sendes Segment 0, når Sender er aktiv. Hvis Modtager er aktiv oprettes der forbindelse og Modtager sender ACK 0 (acknowledgement 0) og Sender ved at der er forbindelse. Hvis Modtager derimod ikke er aktiv vil Sender ikke modtage ACK 0, og Sender vil derfor vente 10 sekunder før igen at sende Segment 0. Dette vil Sender gøre op til tre gange, hvis nødvendigt.

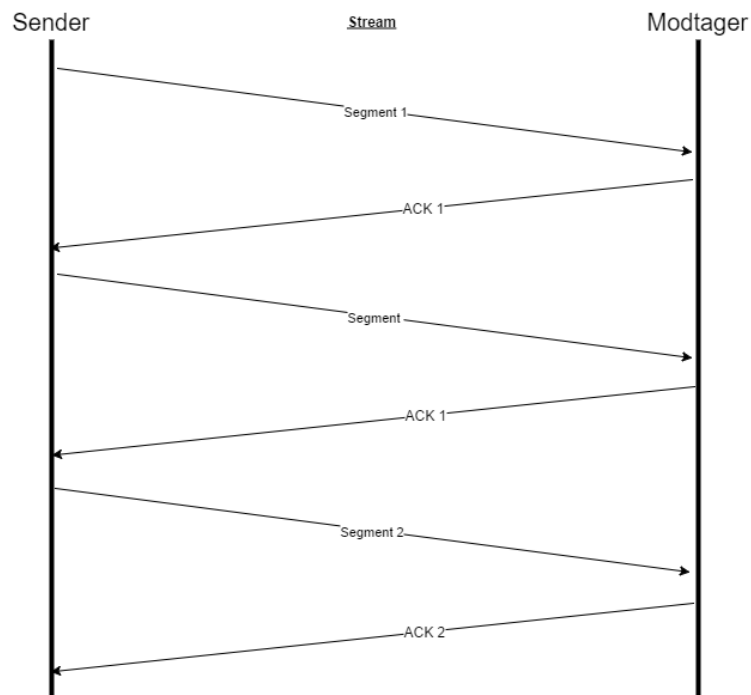


Figur 2: Open

Stream

Stream delen, som kan ses på figur 3, er den del hvor data sendes.

Her sender Sender først Segment 1. Hvis dette er blevet korrekt modtaget vil Modtager sende ACK 1. Dette vil forstærke indtil Close.

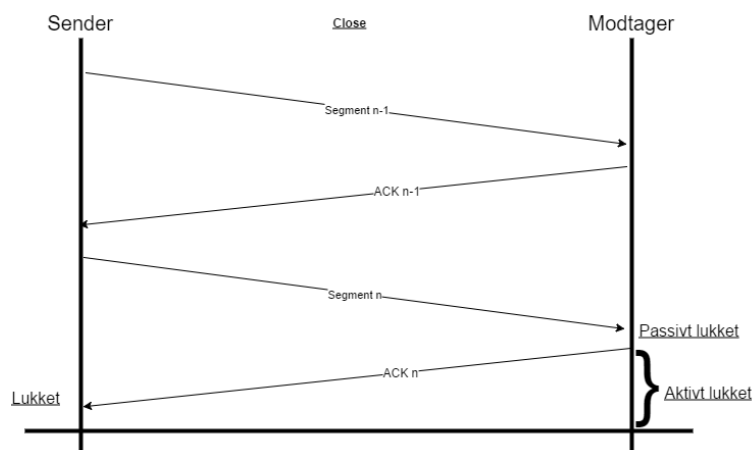


Figur 3: Stream

Close

Close delen kan ses på figur 4.

I denne del gør Sender Modtager opmærksom på at der ikke sendes mere.



Figur 4: Close

Flag

I projekt koden er der benyttet tre flag til at fortælle om beskeden er en probe, accept, eller sidste besked. Der benyttes 3 bit, til at definere flaget. 001 er probe, 010 er accept og 100 er last.

Der blev valgt ikke at bruge fuld duplex, da dette kræver tråde og flere bytes.

5 Data Link Laget

5.1 Teori

CRC

CRC står for Cyklisk Redundant Check, og bruges til fejl-detektering. Denne metode bruges generelt både til LAN og WAN netværk.

Modulo-2 Binary Division

Modulo-2 binary division benytter XOR regnemethoden.

Stuffing

Bitstuffing tilføjer ubrugte bits for at gøre datastrukturer nemmere at manipulere.

Stuffing tilføjes bitstrengen efter resten er fundet, ved hjælp af modulus. Hvis resten er 0 tilføjes "100", hvis resten er 1 tilføjes "10" og hvis resten er 2 tilføjes "1" giver mening i forhold til dataformat.

6 Kontrolfunktioner

Vi valgte at samle de vigtigste funktioner i programmet i klassen Controller. Der

7 Konklusion

8 Perspektivering

9 Litteraturliste

9.1 Bøger

- [1] John W. Dower *Readings compiled for History 21.479*. 1991.
- [2] The Japan Reader *Imperial Japan 1800-1945* 1973: Random House, N.Y.

9.2 Hjemmesider

- [3] E. H. Norman *Japan's emergence as a modern state* 1940: International Secretariat, Institute of Pacific Relations.
- [4] Bob Tadashi Wakabayashi *Anti-Foreignism and Western Learning in Early-Modern Japan* 1986: Harvard University Press.