

CDIO 1 – forår 2014

02325 Datakommunikation

Gruppe nr.: **55**

Afleveringsfrist: **mandag den 25/02 2012 Kl. 15:00**

Denne journal er afleveret via Campusnet (der skrives ikke under)

Denne journal indeholder **16** sider



s133991, Larsen, Anders
Kontaktperson (projektleder)



s133988, Magnussen, Malte



s133974, Hansen, Nicolai



s134010, Jensen, Lars
Kontaktperson (suppleant)



s133980, Liang, Jiahua



s133970, Hansen, Kristin

Indholdsfortegnelse

Spørgsmål	3
1	3
2	4
3	6
4	6
5	8
6	8
7	10
8	12
Test af konsol	12
Forbind	12
Read	13
Tare	14
Zero	14
Message	15
Disconnect	15
Konklusion	16

Spørgsmål

1

This program is listening on a TCP-socket – which? (& how to find out if it did not state so?)

Figur 1 - 2 computere forbundet til vægten

Scale-programmet benytter en TCP-forbindelse. Det betyder kort fortalt at der først kan overføres data, når der er etableret en forbindelse mellem klient og server. Når scale-programmet er oppe at køre, vil der i første linje stå beskrevet hvilken port serveren kører over. Dette er tydeligt illustreret på billedet ovenfor, hvor det kan ses at weight-serveren kører over port 4567.

Nu er vi så heldige at port nummeret står klart og tydeligt i programmet. Skulle dette ikke være tilfældet, er der andre måde at finde ud af hvilket port nummer serveren bruger:

Figur 2 - porten til scale.exe fundet gennem cmd.exe

Ved at bruge Command Prompt (cmd.exe) kan man finde frem til hvilket port weight-server kører over. Det gør men i cmd.exe ved at skrive netstat -a -b, hvorefter en lang liste af forbindelser dukker op på skærmen. Kører scale-programmet på klienten, vil det være muligt at finde scale.exe på listen over forbindelser, og her vil der stå at det er en TCP-forbindelse, som kører over port 4567.

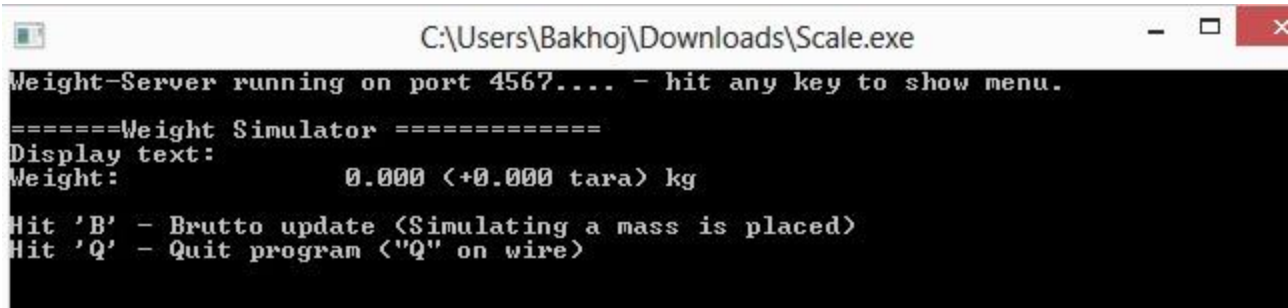
Figur 3 - porten fundet gennem wireshark

Man kan også med lidt held og god vilje finde port nummeret ved brug af Wireshark. Scale-programmet skal være startet og der skal desuden være forbindelse til en klient. Wireshark vil som standard finde en hel masse forbindelse, og ved at indstille filteret, kan man nå frem til det man søger.

Vi forsøgte os frem og fandt en weight-serveren med Wireshark (se nederst på billedet ovenfor).

2

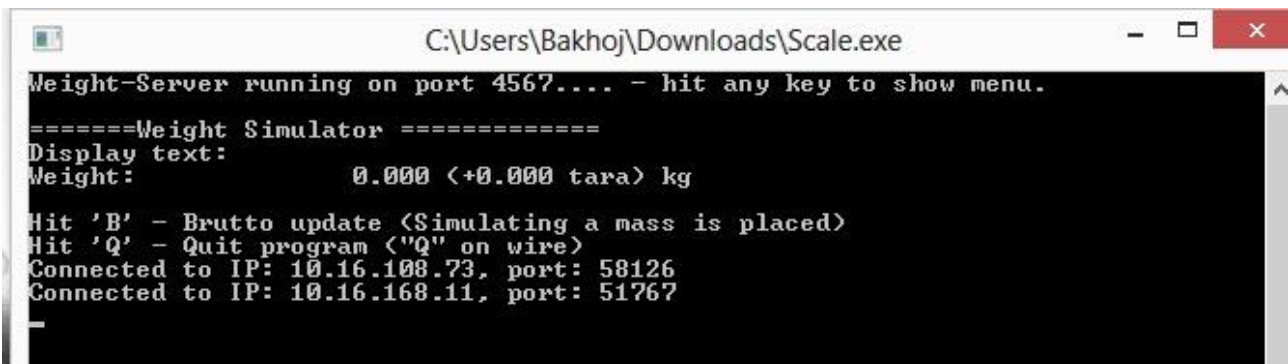
Use a terminal-program or telnet, to verify that the scale-simulator works according to the specification.



```
C:\Users\Bakhoj\Downloads\Scale.exe
Weight-Server running on port 4567.... - hit any key to show menu.
=====Weight Simulator =====
Display text:
Weight:          0.000 (<+0.000 tara> kg)
Hit 'B' - Brutto update (Simulating a mass is placed)
Hit 'Q' - Quit program ('Q' on wire)
```

Figur 4 – Simulatorvindue ved opstart.

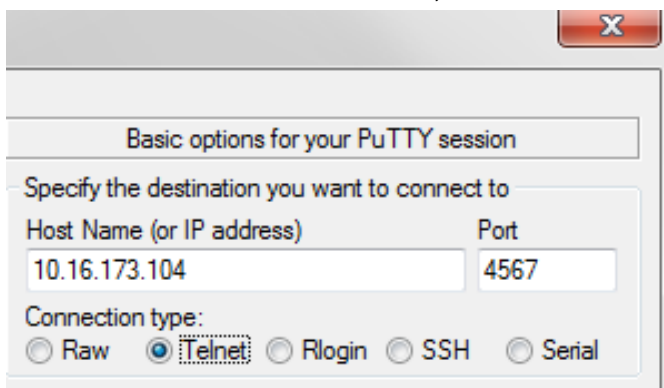
Simulator er startet, og den lytter på socket 4567.



```
C:\Users\Bakhoj\Downloads\Scale.exe
Weight-Server running on port 4567.... - hit any key to show menu.
=====Weight Simulator =====
Display text:
Weight:          0.000 (<+0.000 tara> kg)
Hit 'B' - Brutto update (Simulating a mass is placed)
Hit 'Q' - Quit program ('Q' on wire)
Connected to IP: 10.16.108.73, port: 58126
Connected to IP: 10.16.168.11, port: 51767
```

Figur 5 – Simulatorvindue, med forbundne klienter (multiple clients)

Vi ser, at der er 2 klienter, der har forbundet til simulatoren. Deres ip og hvilken port de forbinder igennem er noteret. Der er altså her bevis for, at simulatoren modtager mere end en enkelt klient af gangen.

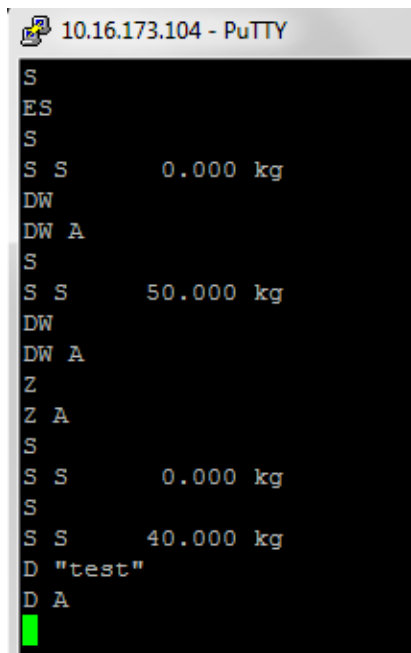


Figur 6 – PuTTY indstillinger (basale)

Indstillinger for PuTTY. Der indskrives IP for server (simulator) og porten serveren lytter på, og der vælges Telnet som internetprotokol.



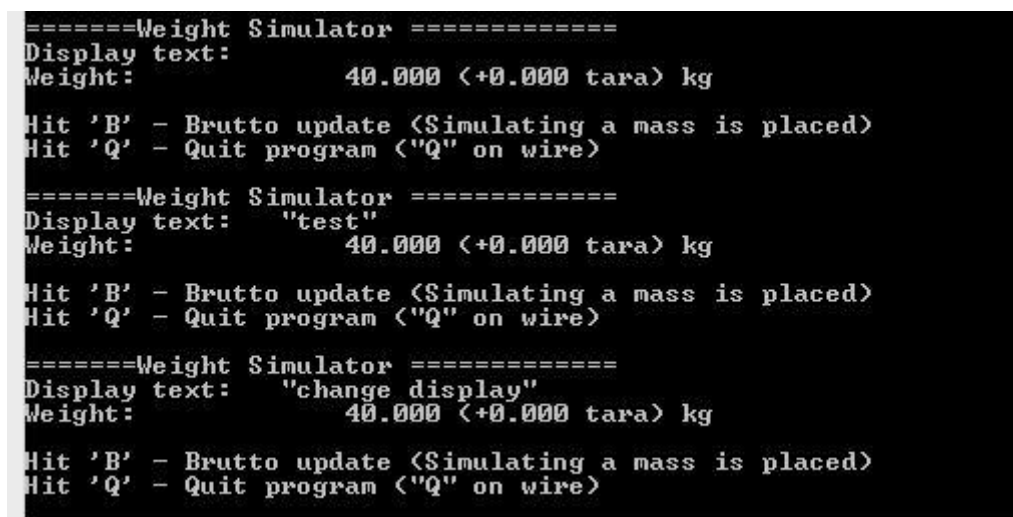
Figur 7 – Tomt PuTTY-vindue, efter der er blevet forbundet til simulator



Figur 8 - Testkommandoer

Der bliver testet nogle forskellige kommandoer i PuTTY. På figur 6, kan et udpluk af serverens udførelse af hver kommando ses, mens PuTTY modtager diverse respons efter forespørgslerne, f.eks.:

Kommando	Respons	Beskrivelse
S	S S 0.000 kg	Klient: Hvad er den nuværende stabile vægt? Server: Den nuværende stabile vægt er lig 0.000 kg
DW	DW A	Klient: På dit eget display, vis nuværende stabile vægt Server: Jeg har vist nuværende stabile vægt på display.

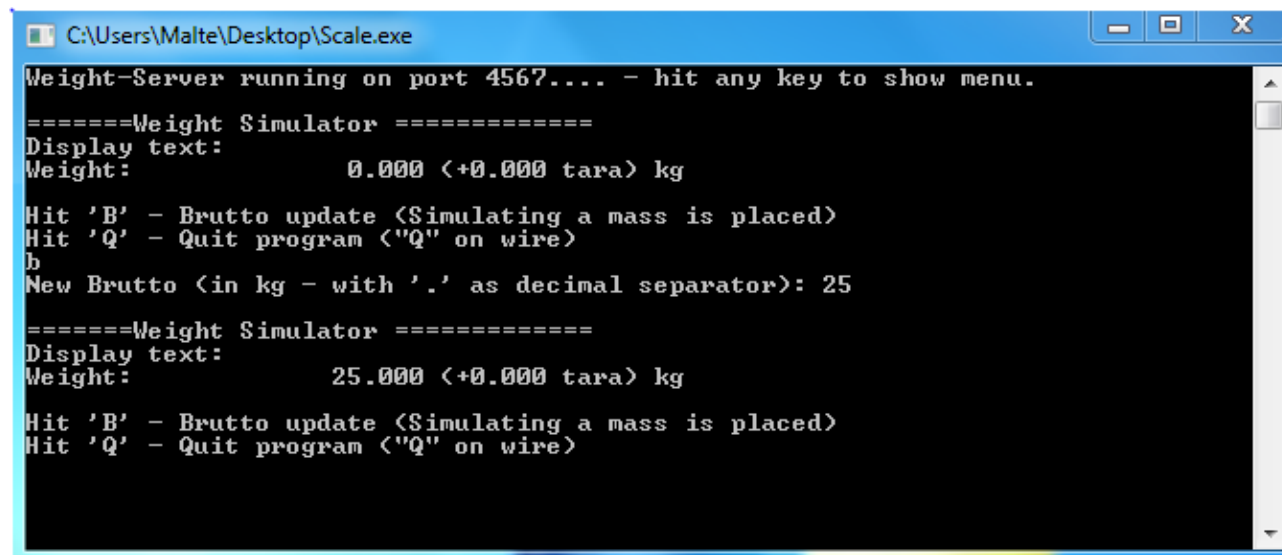


Figur 9 – Simulatoren udfører diverse kommandoer den har modtaget fra PuTTY-klienten.

3

To what extend does the scale-simulator (server) support multiple clients? Is this similar to a real physical scale?

Scale-simulatorens funktioner opfylder alle krav, da den simulerer den fysiske vægt, med samme antal funktioner og med samme kommandoer. De kommandoer som simulatoren har, er f.eks. read, tare og zero, dette er alle sammen kommandoer den fysiske vægt også har. Alle disse kommandoer fungerer ligesom på vægten, f.eks. tare gemmer vægten man har sat på simulatoren og derefter sættes vægten til 0. På den måde kan man simulere at man sætter en skål på vægten, og hælder noget op i den som man så kan se vægten på. Den eneste forskel der er på vægten og simulatoren, er at to personer kan skrive en besked til vægtens display samtidig, simulatoren vil dog slette den første besked lige efter den har modtaget den anden.



```

C:\Users\Malte\Desktop\Scale.exe
Weight-Server running on port 4567.... - hit any key to show menu.
=====Weight Simulator =====
Display text:
Weight:          0.000 (<+0.000 tara> kg
Hit 'B' - Brutto update (<Simulating a mass is placed>
Hit 'Q' - Quit program (<"Q" on wire>
b
New Brutto <in kg - with '.' as decimal separator>: 25
=====Weight Simulator =====
Display text:
Weight:          25.000 (<+0.000 tara> kg
Hit 'B' - Brutto update (<Simulating a mass is placed>
Hit 'Q' - Quit program (<"Q" on wire>

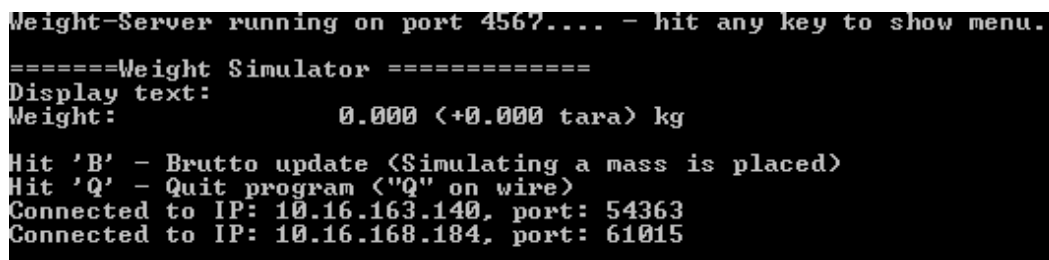
```

Figur 10 - billede af vægten med en angivet vægt på 25 kg

4

Use Wireshark to catch the communication between the two programs.

I løbet af følgende figurer bliver der oprettet forbindelse fra en computer til scale programmet, ved brug af puTTY. Der anvendes derefter Wireshark til at opfange data'en mellem de to programmer. Alle funktioner der er lavet, mens Wireshark har kørt, ses kun på scale programmet, da alle funktioner er vist.



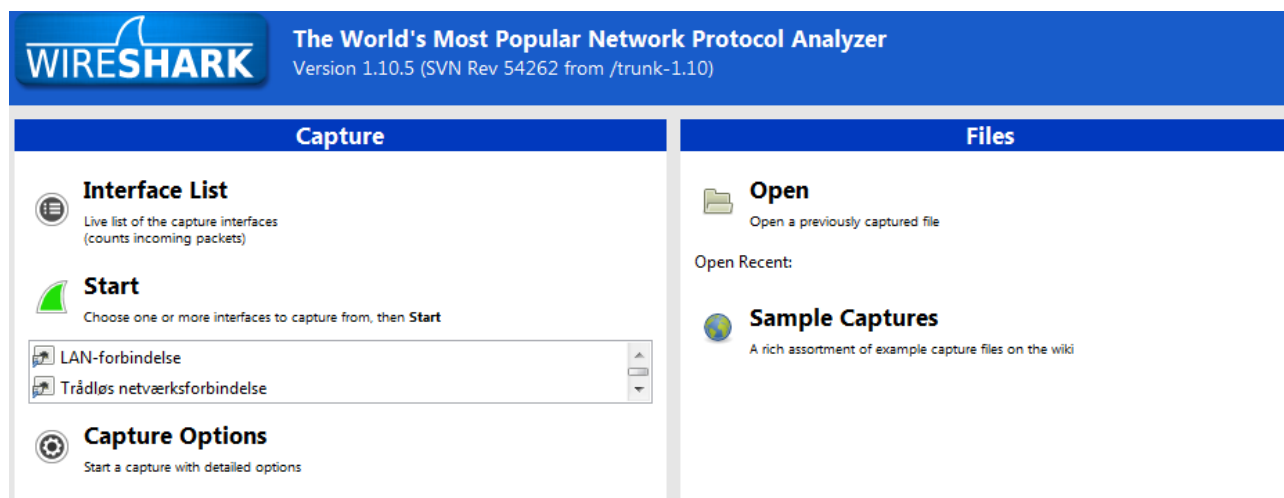
```

Weight-Server running on port 4567.... - hit any key to show menu.
=====Weight Simulator =====
Display text:
Weight:          0.000 (<+0.000 tara> kg
Hit 'B' - Brutto update (<Simulating a mass is placed>
Hit 'Q' - Quit program (<"Q" on wire>
Connected to IP: 10.16.163.140, port: 54363
Connected to IP: 10.16.168.184, port: 61015

```

Figur 11 - 2 forbundet til scale, vha. puTTY

I ovenstående billede er der to forbundet til scale. Vi arbejder med den første der er forbundet hvilket vil kunne ses i Wireshark-billedet taget fra dennes computer.



Figur 12 - wireshark opstart

I opstarten af Wireshark vælges trådløse forbindelser og herefter start, dette vil efter diverse kommandoer er givet mellem puTTY og scale give en række forbindelser mellem alt hvad der snakker på internettet, hvis dette filtreres kan "samtalerne" mellem disse programmer findes. Dette kan ses på nedenstående billede.

19378	51.21003800	10.16.163.140	10.16.165.73	TCP	66	54363	>	tram	[SYN]	Seq=0	win=8192	Len=0	MSS=1460	ws=4	SACK_PERM=1	
19380	51.21794700	10.16.163.73	10.16.163.140	TCP	66	tram	>	54363	[SYN, ACK]	Seq=0	Ack=1	win=8192	Len=0	MSS=1460	ws=256	SACK_PERM=1
19381	51.21806700	10.16.163.140	10.16.165.73	TCP	54	54363	>	tram	[ACK]	Seq=1	Ack=1	win=17520	Len=0			
19382	51.21897900	10.16.163.140	10.16.165.73	TCP	75	54363	>	tram	[PSH, ACK]	Seq=1	Ack=1	win=17520	Len=21			
19397	51.42193800	10.16.165.73	10.16.163.140	TCP	54	tram	>	54363	[ACK]	Seq=1	Ack=22	win=17408	Len=0			
37038	96.31398800	10.16.163.140	10.16.165.73	TCP	55	54363	>	tram	[PSH, ACK]	Seq=22	Ack=1	win=17520	Len=1			
37039	96.31414200	10.16.163.140	10.16.165.73	TCP	56	54363	>	tram	[PSH, ACK]	Seq=23	Ack=1	win=17520	Len=2			
37041	96.31939400	10.16.165.73	10.16.163.140	TCP	54	tram	>	54363	[ACK]	Seq=1	Ack=25	win=17408	Len=0			
37042	96.31955900	10.16.165.73	10.16.163.140	TCP	58	tram	>	54363	[PSH, ACK]	Seq=1	Ack=25	win=17408	Len=4			
37063	96.51888500	10.16.163.140	10.16.165.73	TCP	54	54363	>	tram	[ACK]	Seq=25	Ack=5	win=17516	Len=0			
37204	97.84020500	10.16.163.140	10.16.165.73	TCP	55	54363	>	tram	[PSH, ACK]	Seq=25	Ack=5	win=17516	Len=1			
37205	97.84038800	10.16.163.140	10.16.165.73	TCP	56	54363	>	tram	[PSH, ACK]	Seq=26	Ack=5	win=17516	Len=2			
37206	97.86315900	10.16.165.73	10.16.163.140	TCP	54	tram	>	54363	[ACK]	Seq=5	Ack=28	win=17408	Len=0			
37207	97.86388700	10.16.165.73	10.16.163.140	TCP	73	tram	>	54363	[PSH, ACK]	Seq=5	Ack=28	win=17408	Len=19			
37209	98.06397500	10.16.163.140	10.16.165.73	TCP	54	54363	>	tram	[ACK]	Seq=28	Ack=24	win=17496	Len=0			
49277	128.8351230	10.16.163.140	10.16.165.73	TCP	55	54363	>	tram	[PSH, ACK]	Seq=28	Ack=24	win=17496	Len=1			
49278	128.8353100	10.16.163.140	10.16.165.73	TCP	56	54363	>	tram	[PSH, ACK]	Seq=29	Ack=24	win=17496	Len=2			
49331	128.8820250	10.16.165.73	10.16.163.140	TCP	54	tram	>	54363	[ACK]	Seq=24	Ack=31	win=17408	Len=0			
49360	128.8964060	10.16.165.73	10.16.163.140	TCP	73	tram	>	54363	[PSH, ACK]	Seq=24	Ack=31	win=17408	Len=19			
49472	129.0992070	10.16.163.140	10.16.165.73	TCP	54	54363	>	tram	[ACK]	Seq=31	Ack=43	win=17476	Len=0			

Figur 13 - "samtalerne" mellem puTTY og scale

5

Which IP-addresses & port-numbers were used?

IP og port-numrene er forskellige hver gang der forbindes. Dette skyldes at hver gang der forbindes til internettet tildeles man en ny IP. I nedenstående billede er to computere forbundet til scale gennem puTTY. IP og port-numrene kan ses når de er forbundet.

```
Weight-Server running on port 4567.... - hit any key to show menu.
=====Weight Simulator =====
Display text:
Weight:          0.000 <+0.000 tara> kg
Hit 'B' - Brutto update <Simulating a mass is placed>
Hit 'Q' - Quit program <"Q" on wire>
Connected to IP: 10.16.163.140, port: 54363
Connected to IP: 10.16.168.184, port: 61015
```

Figur 14 - to computere tilsluttet simulatoren

Første persons IP: 10.16.163.140 port: 54363.

Anden persons IP: 10.16.168.184 port: 61015.

6

Write a console-based client that communicates with the Scale/simulator.

Der er blevet udviklet en konsol, der umiddelbart overholder 3-lags-principperne. Konsollen består af de tre lag – grænseflade (Dialog.java), funktionalitet (IScale.java og Scale.java) og datalaget (IScaleConnection.java og ScaleConnection.java.) Der vil ikke blive givet dokumentation for Swing-komponenterne.

Grænsefladen taler kun med interface fra

funktionalitetslaget (IScale), mens funktionalitetslaget kun

```
12 private IScaleConnection con;
13 private IScale scale;
14 private Dialog dialog;
15
16 public Main() {
17     // Set layout manager
18     setLayout(new FlowLayout());
19
20     // Instantiate objects
21     con = new ScaleConnection();
22     scale = new Scale(con);
23     dialog = new Dialog(scale);
24
25     add(dialog);
```

Figur 15 – Konsollens tre lag instantieres

```
public Dialog(IScale scale) {
    this.scale = scale;

    // General setup
    setLayout(new BorderLayout());
```

Figur 16 – Dialog.java

```
private IScaleConnection con;

public Scale(IScaleConnection con) {
    this.con = con;
```

Figur 17 – Scale.java

```
public ScaleConnection() {
    this.host = "";
    this.port = 0;
}
```

Figur 18 – ScaleConnection.java

taler med interface fra datalaget (IScaleConnection.)

Feltet "scale" i Dialog.java bliver erklæret som typen "IScale", mens felterne "host" og "port" er String-typer.

Når konsollen åbnes, kan der i "host"-feltet indtastes en ip, som bliver sendt til funktionalitetslaget, og videre over i "ScaleConnection", som vil forsøge, via oprettelse af en socket, at forbinde til en server, der

```
public boolean scaleConnect(String host, String port) {
    try {
        if((this.scaleSocket = new Socket(this.host, this.port)) != null) {
            this.scaleOutput = new PrintWriter(this.scaleSocket.getOutputStream(), true);
            this.scaleInput = new BufferedReader(new InputStreamReader(this.scaleSocket.getInputStream()));
            return true;
        }
    }
    catch (IOException e) {
        e.printStackTrace();
    }
    return false;
}
```

Figur 19 – Socket oprettes med host (IPv4-adresse) og port (String input fra JTextArea er blevet parsed som Integer)

lytter på den indtastet port (der i øvrigt også bliver indtastet i konsolvinduet – se Figur 19.)

Efter oprettet forbindelsen, kalder diverse knapper en fælles funktion, "getScaleResponse", med String-parameteren "output". Denne parameter kræver en String, som er kommandoen til serveren – f.eks. "S" vil sende "S\r\n" til serveren, og der vil blive givet et respons, som bliver sendt tilbage gennem lagene til grænsefladen, hvor respons bliver tilføjet i en læselig facon i JTextArea (se Figur 20). Knappen "Read" kalder i Dialog.java metoden "scaleRead()". Metoden "getDate()" returnerer en formateret dato, Figur 21 – Funktionalitetslagets metode "scaleRead" kalder getScaleResponse, med parameteren "S". getDigit(str) og getUnit(str) er formateringsmetoder, som gør responset læseligt og Figur 22 – scaleOutput og -Input er forbundet med den tidligere oprettet socket. readLine() returnerer serverens respons.)

```
private void scaleRead() {
    textArea.append("[ " + getDate() + "]\t" + scale.scaleRead() + "\n");
}
```

Figur 20 – Knappen "Read" kalder i Dialog.java metoden "scaleRead()". Metoden "getDate()" returnerer en formateret dato

```
public String scaleRead() {
    String str = con.getScaleResponse("S");

    return "Current scale weight: " + getDigit(str) + " " + getUnit(str);
}
```

Figur 21 – Funktionalitetslagets metode "scaleRead" kalder getScaleResponse, med parameteren "S". getDigit(str) og getUnit(str) er formateringsmetoder, som gør responset læseligt

```
public String getScaleResponse(String output) {
    try {
        this.scaleOutput.println(output);

        return this.scaleInput.readLine();
    }
    catch(IOException e) {
        e.printStackTrace();

        return e.getMessage();
    }
}
```

Figur 22 – scaleOutput og -Input er forbundet med den tidligere oprettet socket. readLine() returnerer serverens respons

Når bruger er færdig med sin session, kan der med fordel trykkes "Disconnect" i grænsefladen – selvom "Windows-krydset" også umiddelbart vil lukke (.close()) den oprettede socket (se Figur 23 for manuel luk). Der kan herefter genetableres forbindelse til en anden vægt.

```
public void scaleDisconnect() {
    if(this.scaleSocket != null) {
        try {
            this.scaleOutput.flush();
            this.scaleOutput.close();
            this.scaleInput.close();
            this.scaleSocket.close();
        }
        catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Figur 23 – Disconnectmetoden sørger for, at ingen bytes i bufferen (scaleOutput) bliver siddende, men tvinger dem til at blive udskrevet (.flush()). Begge buffers (scaleOutput og -input) samt socket bliver derefter lukket, for at undgå memory leaks.

7

The scale could also have used the built-in webserver for readouts. What is the Pro's and Con's of this compared to the LAN-interface you have used?

Scale er en server og den kan også indbygges i en webserver, som returnerer data til enhver der kalder den.

Fordelen ved scale'en er, at den giver et præcist svar tilbage til dem som kalder. Såsom, hvis der bliver kaldt en bestemt kommando, vil den returnere kommandoen samt data.

F.eks. Man kan bruge scale til at opsætte en vægt, og man kan kalde den med en bestemt kommando for at returnere data fra den. I dette tilfælde bruger man kommandoen "S" for at kommunikere med scale, og den udskriver kommando "S" og selve kommando samt vægten der er sat til den. Det er også letlæseligt for programmører eller personer, som har lidt større baggrundsviden inden for kodesproget. De vil have det nemt med at få en forståelse til at bruge scale. Det vil også spare en del tid for programmørerne, til at udvikle programmet.

I forhold til brugeren, som ikke har nogen baggrundsviden inden for kodesproget, vil det være svært at forstå, da scale returnerer med et kommando svar.

F.eks. brugeren skal mindst have en lille forståelse inde for kodesproget, da man skal anvende kommandoen for at tilkalde et kald til scale, og derefter vil den som regel returnere selve kommandoen samt data, som kalder det man har bedt om. Det er en af ulemperne hvis man ikke har forstand på kodesproget.

8

Document the above

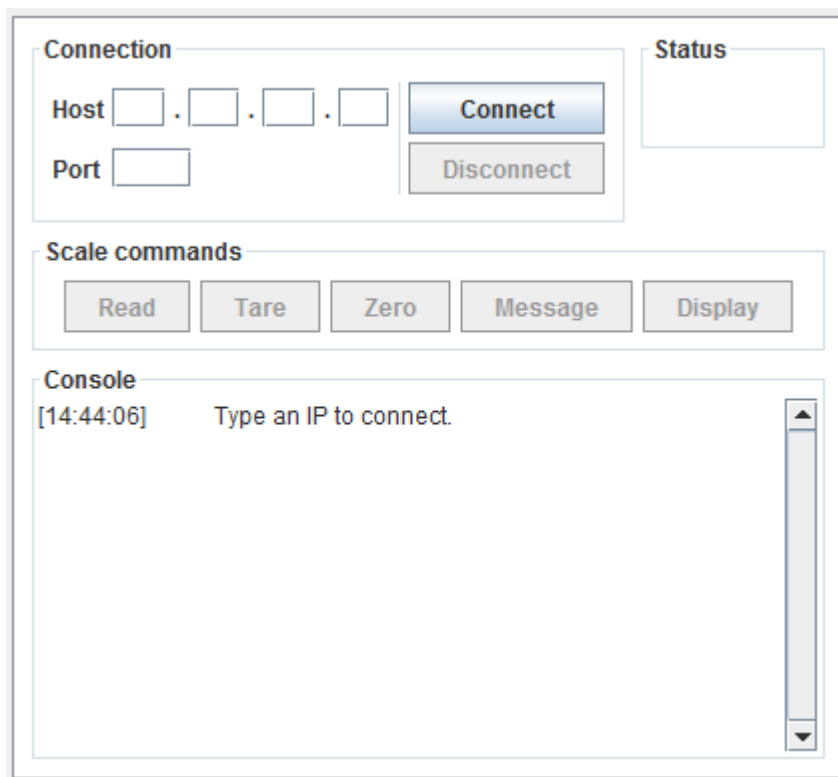
Test af konsol

Da der er blevet udviklet et helt interface til brug af simulatoren (se Figur 1), er test blevet lavet manuelt, hvor hver knap bliver testet og vist.

Forbind

På Figur 1 ses programmet når det er startet. Der kan i og for sig forbindes til hvad som helst, der tillader det. I vores tilfælde, er vi interesseret i at forbinde til vores simulator (se Figur 24.)

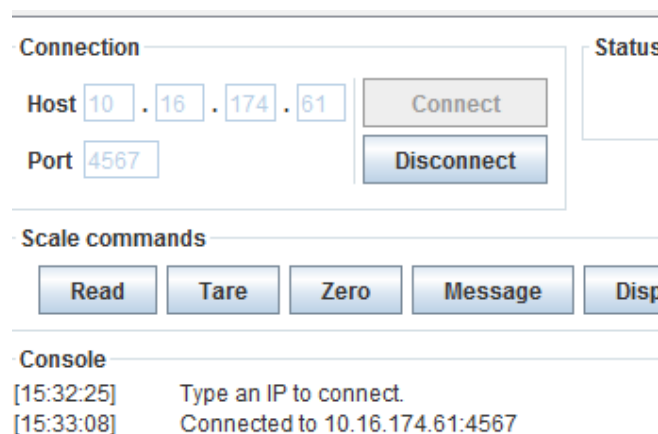
På Figur 25 ses nu, at siden vi nu er forbundet, kan de tidligere "grayed" knapper bruges, og der bliver givet besked om, at vi nu er forbundet til den indtastede ip og port.



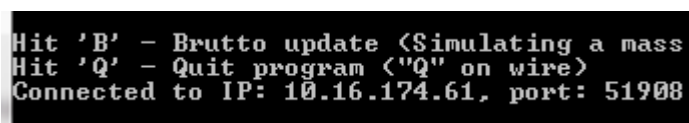
Figur 24 – "Scale Console"

De fem knapper, "Read", "Tare", "Zero", "Message" og "Display" vil alle blive testet, samt afsluttende "Disconnect".

Der er ligeledes fra serverens (simulatoren) side (Figur 26), blevet accepteret en forbindelse mellem de to enheder. Klienten (vores program) og hvilken port der forbindes igennem bliver vist.



Figur 25 – Der er indtastet en gyldig ip og port, og forbundet til serveren



Figur 26 - Serveren (simulatoren) har accepteret forbindelsen

Read

Der skal nu forevises, at vægten kan læses. Ved at trykke på "Read"-knappen, vil kommandoen "S" bliver sendt til serveren. Respons skulle gerne være S S [vægt] [unit]. [unit] er altid i kg ved simulatoren, men ved den "rigtige" vægt, vil det være en indstillingsmulighed på selve vægten.

På Figur 27 ses klienten udskrive den nuværende vægt på en læselig form, "Current scale weight: 0.000 kg". På Figur 29 indstiller vi simulatoren til at have en vægt på 42 kg. Figur 28 viser, at den nye vægt bliver læst, og igen udskrevet i et læseligt "sprog".



Console

```
[15:32:25] Type an IP to connect.
[15:33:08] Connected to 10.16.174.61:4567
[15:33:49] Current scale weight: 0.000 kg
```

Figur 27 – Konsolvinduets udskrift af nuværende vægt

Scale commands



Console

```
[15:32:25] Type an IP to connect.
[15:33:08] Connected to 10.16.174.61:4567
[15:33:49] Current scale weight: 0.000 kg
[15:34:44] Current scale weight: 42.000 kg
```

Figur 28 – Den nye vægt genkendes

```
b
New Brutto <in kg - with '.' as decimal separator>: 42
=====Weight Simulator =====
Display text:
Weight:          42.000 <+0.000 tara> kg
Hit 'B' - Brutto update <Simulating a mass is placed>
Hit 'Q' - Quit program <"Q" on wire>
```

Figur 29 – Ny vægt indskrives

Tare

Ved at trykke på "Tare", bliver vægten nulstillet i forhold til nuværende, stabile vægt. På Figur 30 ses "Tare"-knappen trykket, hvor der bliver givet respons "Scale tared, tara reads: 42.000 kg", hvilket er klart mere læseligt end f.eks. Serverens originale respons "T S [Tara vægt] [unit]".

På Figur 31 ses simulatorens reaktion på, at der i konsollen er blevet trykket "Tare".

Read	Tare	Zero	Message
Console			
[15:32:25]			Type an IP to connect.
[15:33:08]			Connected to 10.16.174.61:4567
[15:33:49]			Current scale weight: 0.000 kg
[15:34:44]			Current scale weight: 42.000 kg
[15:36:56]			Scale tared, tara reads: 42.000 kg

Figur 30 – Vægt er blevet "tared"

Kaldes "Read"-funktionen nu, skulle respons gerne være 0.00 kg, hvilket ses på Figur 32.

```
Display text:
Weight:          0.000 (<+42.000 tara> kg
Hit 'B' - Brutto update <Simulating a mass i
Hit 'Q' - Quit program <"Q" on wire>
```

Figur 31 – Simulatoren har accepteret kommandoen og nulstillet vægten, mens 42.000 kg er "gemt" som tara

```
5:37:44] Current scale weight: 42.000 kg
5:36:56] Scale tared, tara reads: 42.000 kg
5:37:44] Current scale weight: 0.000 kg
```

Figur 32 – Programmet læser nu 0.000 kg efter at være blevet "tared"

Zero

Vægten kan ligeledes nulstilles vha. "Zero"-knappen. Dette nulstiller hele vægten. På Figur 33 ses beskeden når "Zero"-knappen trykkes, mens Figur 34 viser simulatorens reaktion.

```
5:37:44] Current scale weight: 0.00
5:38:21] Scale has been zeroed.

[15:38:21] Scale has been zeroed.
[15:39:18] Current scale weight: 0.000 kg
[15:39:19] Scale tared, tara reads: 0.000 kg
```

Figur 33 – Besked i konsollen ved nulstilling af vægt, samt en ny "Read"- og "Tare"-funktion kaldt (nu nulstillet og derved 0.000)

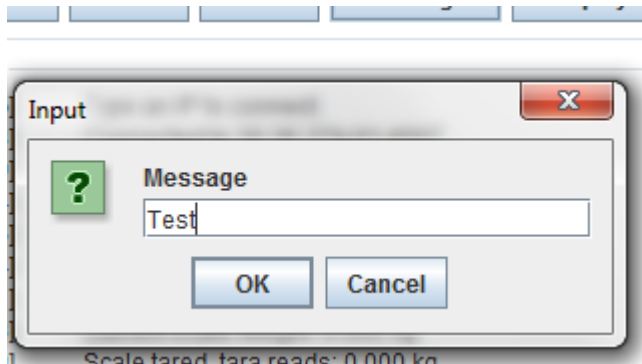
```
Display text:
Weight:          0.000 (<+42.000 tara> kg
Hit 'B' - Brutto update <Simulating a mass i
Hit 'Q' - Quit program <"Q" on wire>

=====Weight Simulator =====
Display text:
Weight:          0.000 (<+0.000 tara> kg
Hit 'B' - Brutto update <Simulating a mass i
Hit 'Q' - Quit program <"Q" on wire>
```

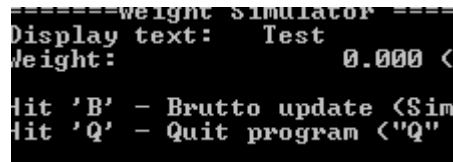
Figur 34 – Simulatoren reaktion på nulstilling

Message

Bedre kendt som "Display text", kan bruges til at sætte en midlertidig "overskrift". På Figur 35 ses input prompt, mens Figur 37 og Figur 36 viser hhv. klienten og serverens reaktion på input.



Figur 35 – Input prompt ved "Message"-tryk



Figur 37

```
[15:39:19] Scale tared, tara reads: 0.000 kg
[15:40:32] Balance display set with "Test".
```

Figur 36 – Simulatoren gengiver den indtastede tekst

Disconnect

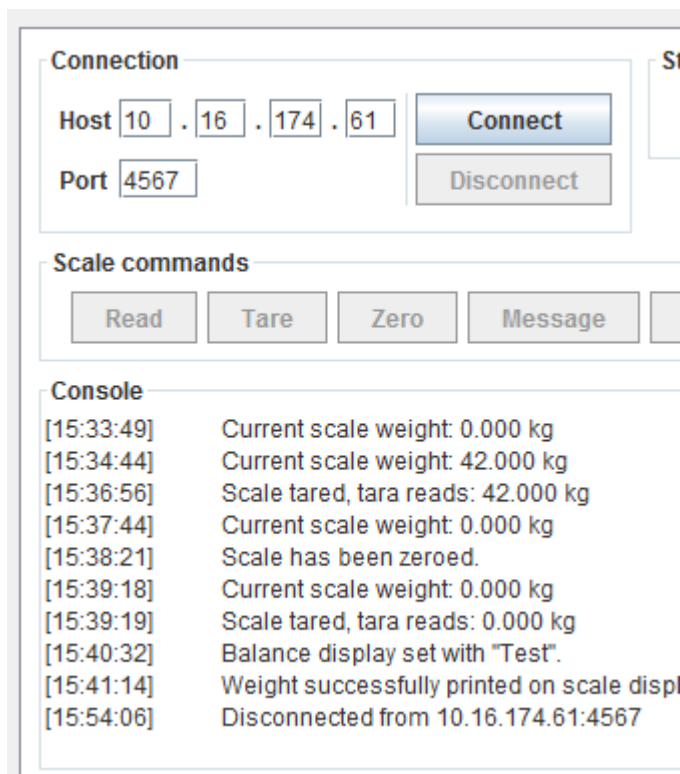
Slutteligt kan der frakobles. Dette vil ødelægge den nuværende connection, og dermed kalde "close" på både input, output og socket. Figur 39, Figur 38 og Figur 40 ses beskeder samt simulatorens reaktion.

```
[15:54:06] Disconnected from 10.16.174.61:4567
```

Figur 39 - Der gives besked om, at der nu er frakoblet fra ip:port

```
Hit 'Q' - Quit program ('Q' on wire)
Remote socket disconnected or in error
```

Figur 38 – Simulatorens reaktion



Figur 40 – Knapperne er igen "grayed" og ikke tilgængelige, før en ny forbindelse er oprettet

Suggestions for improvements on the simulator

- a. Angive hvilken scale den simulere (version). På den måde er det nemmere at finde ud af hvilke kald man kan lave, endvidere udvikle det bedst mulige program til at læse vægten med.
- b. Gem display teksten, indtil at der bliver sendt en ny display tekst i scale-simulatoren. Denne fejl sker under DW (display weight) kommandoen.

Konklusion

Formålet med opgaven var at besvare de 8 spørgsmål, under hvert spørgsmål er der givet et svar. Her til sidst vil vi dog lige lave en konklusion på vores færdige projekt. Vi fik lavet en brugergrænseflade til Scale-simulatoren, her benyttede vi sockets til at forbinde computere til serveren. Derudover brugte vi wireshark til at opfange trafikken der kører på internettet, på den måde kunne vi observere "samtalen" mellem puTTY og Scale-simulatoren.