CDIO Opgave 01

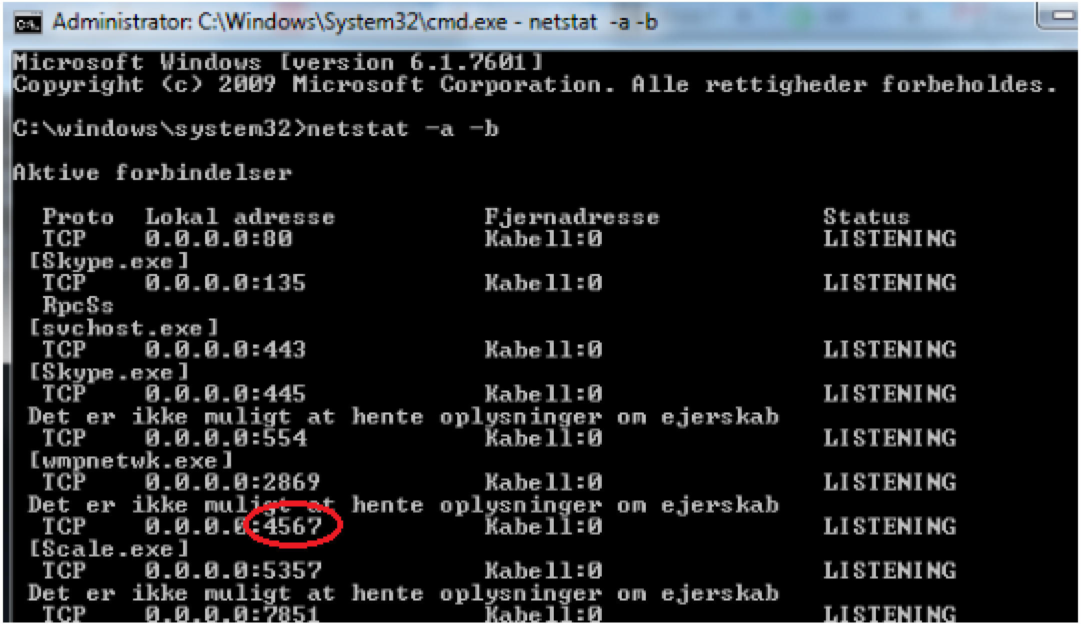
Magnus Brandt Sløgedal s103185
Christian Budtz s134000
Nielsen, Martin s123064,
Rúni Egholm Vørmadal s134004,
Morten Hesselbjerg s017704,
Rikke Christina Hansen s120359
Eirik Oterholm Nielsen s123006

24/02/2014

1 Answers to Questions / Actions

Question 01

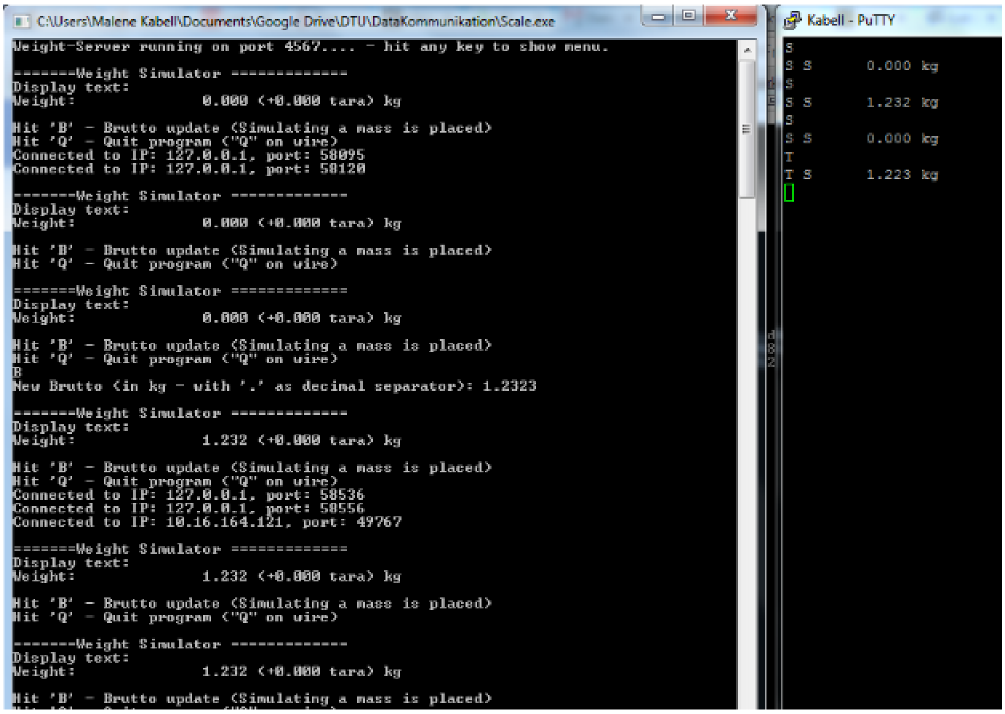
The Scale.exe process listens on Port 4567. This can be discovered using a port sniffer or the built in windows tool netstat: **netstat -a -b**



```
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle rettigheder forbeholdes.

C:\windows\system32>netstat -a -b

Aktive forbindelser

  Proto  Lokal adresse          Fjernadresse           Status
  TCP    0.0.0.0:80             Kabell:0               LISTENING
 [Skype.exe]
  TCP    0.0.0.0:135            Kabell:0               LISTENING
  RpcSs
 [svchost.exe]
  TCP    0.0.0.0:443            Kabell:0               LISTENING
 [Skype.exe]
  TCP    0.0.0.0:445            Kabell:0               LISTENING
 Det er ikke muligt at hente oplysninger om ejerskab
  TCP    0.0.0.0:554            Kabell:0               LISTENING
 [wmpnetwk.exe]
  TCP    0.0.0.0:2869           Kabell:0               LISTENING
 Det er ikke muligt at hente oplysninger om ejerskab
  TCP    0.0.0.0:4567           Kabell:0               LISTENING
 [Scale.exe]
  TCP    0.0.0.0:5357           Kabell:0               LISTENING
 Det er ikke muligt at hente oplysninger om ejerskab
  TCP    0.0.0.0:7851           Kabell:0               LISTENING
```

Question 02

Using Putty we connected several users and tested the Scale. Apart from the first command being rejected the program worked to specification.



C:\Users\Malene Kabell\Documents\Google Drive\DTU\DataKommunikation\Scale.exe
Kabell - PuTTY

```
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: 127.0.0.1, port: 58095
Connected to IP: 127.0.0.1, port: 58120

=======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)

=======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): 1.2323

=======Weight Simulator =============
Display text:
Weight:               1.232 (+0.000 tara) kg

Hit 'B' - Brutto update (Simulating a mass is placed)
Hit 'Q' - Quit program ("Q" on wire)
Connected to IP: 127.0.0.1, port: 58536
Connected to IP: 127.0.0.1, port: 58556
Connected to IP: 10.16.164.121, port: 49767

=======Weight Simulator =============
Display text:
Weight:               1.232 (+0.000 tara) kg

Hit 'B' - Brutto update (Simulating a mass is placed)
Hit 'Q' - Quit program ("Q" on wire)

=======Weight Simulator =============
Display text:
Weight:               1.232 (+0.000 tara) kg

Hit 'B' - Brutto update (Simulating a mass is placed)
```

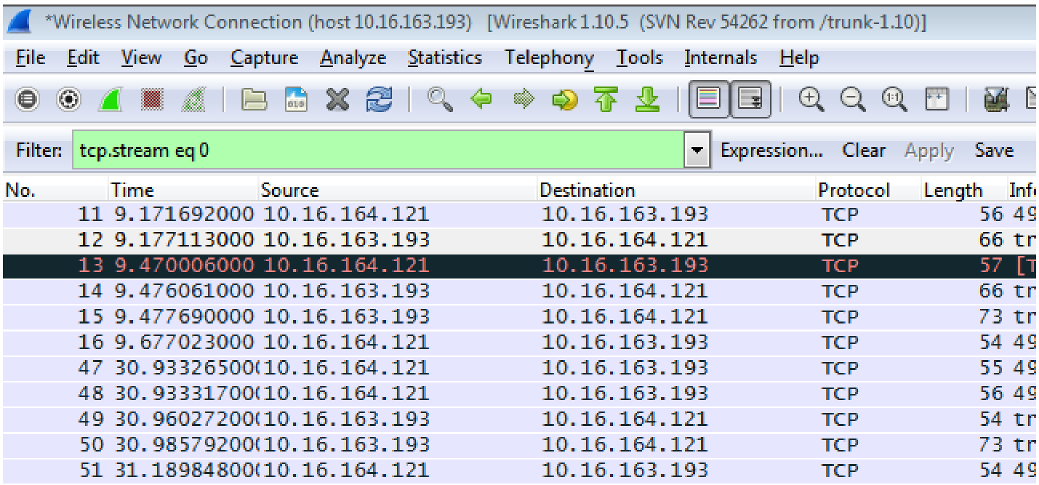
```
S
S S       0.000 kg
S
S S       1.232 kg
S
S S       0.000 kg
T
T S       1.223 kg
```

Question 03

We have made a test-class "TestMultipleConnections" that has a method that takes an int numberOfConnections and tries making that number of connections using a for loop. We were able to make up to 98 connections to the weight, and all attempts past 99 failed. We have searched the manual for the connection limit on the physical weight, but we were not able to find anything about it, but we expect that it takes a few connections at least, so that 1 person cannot block the weight for all the other clients, and one port dedicated to keep an activitylog for the weight.

Question 04

We used wireshark to sniff the packets sent between two computers. Below is a screenshot documenting the capture.



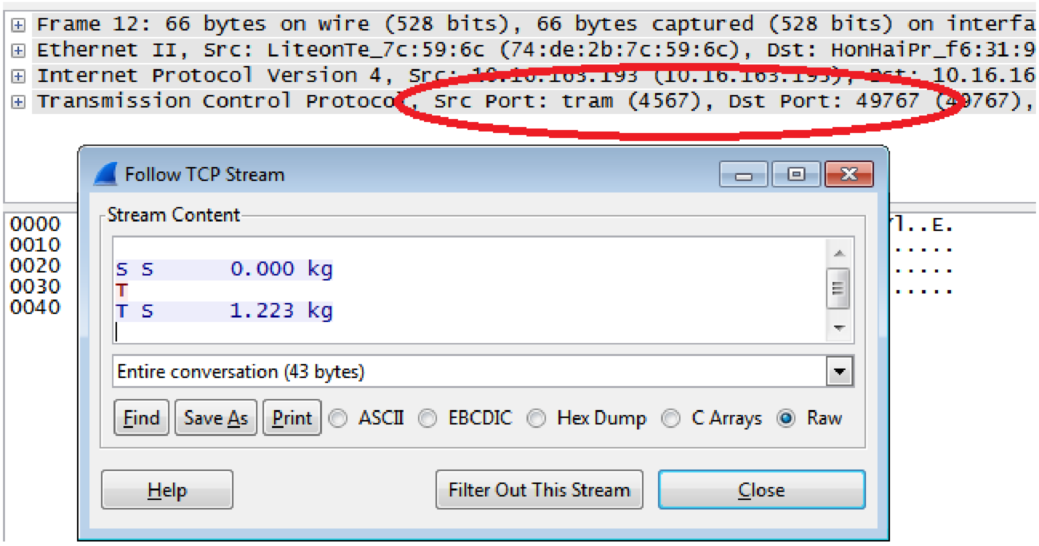
*Wireless Network Connection (host 10.16.163.193) [Wireshark 1.10.5 (SVN Rev 54262 from /trunk-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: tcp.stream eq 0

Expression... Clear Apply Save

```
No.  Time             Source         Destination     Protocol  Length  Inf
  11 9.171692000 10.16.164.121  10.16.163.193   TCP           56 49
  12 9.177113000 10.16.163.193  10.16.164.121   TCP           66 tr
  13 9.470006000 10.16.164.121  10.16.163.193   TCP           57 [T
  14 9.476061000 10.16.163.193  10.16.164.121   TCP           66 tr
  15 9.477690000 10.16.163.193  10.16.164.121   TCP           73 tr
  16 9.677023000 10.16.164.121  10.16.163.193   TCP           54 49
  47 30.93326500(10.16.164.121  10.16.163.193   TCP           55 49
  48 30.93331700(10.16.164.121  10.16.163.193   TCP           56 49
  49 30.96027200(10.16.163.193  10.16.164.121   TCP           54 tr
  50 30.98579200(10.16.163.193  10.16.164.121   TCP           73 tr
  51 31.18984800(10.16.164.121  10.16.163.193   TCP           54 49
```



```
⊞ Frame 12: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interfa
⊞ Ethernet II, Src: LiteonTe_7c:59:6c (74:de:2b:7c:59:6c), Dst: HonHaiPr_f6:31:9
⊞ Internet Protocol Version 4, Src: 10.16.163.193 (10.16.163.193), Dst: 10.16.16
⊞ Transmission Control Protocol, Src Port: tram (4567), Dst Port: 49767 (49767),
```

```
0000                                                    l..E.
0010                                                    .....
0020                                                    .....
0030                                                    .....
0040
```

Follow TCP Stream

Stream Content

```
S S      0.000 kg
T
T S      1.223 kg
```

Entire conversation (43 bytes)

Question 05

The destination and source IP are in the column with same name, in the picture above. Port numbers are highlighted with a red circle, where the src (source) port is the destination of the weight and the dst (destination) port is where the commands were sent from.

Question 06

See enclosed java code.

Question 07

Pro's of using the built-in webserver would be to make it possible to use html to design the user interface and interaction with the weight - which makes Java unnecessary, but necessitates a browser. Since http uses port 80, which is open by default, there is no problem when accessing the weight over a network where other ports are closed (connecting over the internet e.g.).
Using the LAN interface it is necessary to create your own sockets and connections - which in the web browser is handled for you. You can on the other hand use a port, which is blocked by default, thus preventing people from accessing the weight from the internet. The requests to the weight are shorter using the LAN interface, which makes transfer time faster - this is not an issue though - probably not even if we implemented real time reading of the scale.

Question 08

The solution we created implements a GUI instead of a command line tool. This was done, because we felt that we might use the GUI in a later submission too.

Suggestions:
- Scale.exe should implement some form of security - for instance a password needed to connect
- The returned string should be a fixed length, and values should begin/end at a fixed index. This would make reading the values easier. We had to use regex to read the values, because the position was not fixed.
- The bug that causes the first command sent to the weight ought to be fixed.