**Aflevering 1**

**NGK**

Genafleveret: 3. april 2025

Sidst rettet: 3 april 2025

Gruppe nr. 18

|  |  |
| --- | --- |
| Studie nr. | Navn |
| 202309622 | Julia Aspoen Rasmussen |
| 202310768 | Kalja Blirup Grønning |
| 202300114 | Sophia Vagnsbjerg Bach |

Indhold

[Øvelse 3 3](#_Toc194573384)

[Øvelse 4 15](#_Toc194573385)

[Øvelse 5 28](#_Toc194573386)

[Øvelse 6 30](#_Toc194573387)

[Øvelse 7 32](#_Toc194573388)

# Øvelse 3

*Formålet med øvelsen er at anvende Wireshark til analyse af netværksfunktionalitet og at undersøge DNS-protokollens funktionalitet.*

**DNS**

**1. Undersøg vha. Wireshark hvilke DNS-telegrammer der initielt genereres når en hjemmeside (www.au.dk) tilgås fra en webbrowser (fokuser på A telegrammerne). Forklar hvad der sker?**

Først foretages en DNS-forespørgsel (Query) til en DNS-server. Da vi kun fokuserer på A telegrammerne betyder det at vi beder om en IPv4-adresse. DNS-serveren returnerer et svar som indeholder IPv4-addressen, der svarer til [www.au.dk](http://www.au.dk) (som vi bruger i denne øvelse). IP-addressen bruges til at sende en http-forespørgsel til serveren, hvor selve hjemmesiden hentes.

*DNS Query:*

* 1. **Hvilken information indeholder ”Queries” feltet i DNS query telegrammet (dokumenter med Wireshark)?**

Vi åbner ”Queries” feltet i DNS telegrammet for A www.au.dk:

Et billede, der indeholder tekst, skærmbillede, software, Webside

Indhold genereret af kunstig intelligens kan være forkert.

Figur 1 DNS query telegrammet i wireshark

Queries for www.au.dk indeholder

* Domænets navn: [www.au.dk](http://www.au.dk)
* Hvor mange symboler der er i navnet: 9 (inklusiv punktum)
* Antal dele i navnet: 3 (www, au, dk)
* Typen: A (IPv4)
* Klasse: IN = internet
  1. **Hvilken Destination Port har telegrammet (dokumenter med Wireshark)?**

Destination Port’en findes under User Datagram Protocol:

Et billede, der indeholder tekst, skærmbillede, software, Webside

Indhold genereret af kunstig intelligens kan være forkert.

Figur 2 Destination port for [www.au.dk](http://www.au.dk) i wireshark

Telegrammet har destination port 53.

*DNS Query Response:*

* 1. **Hvilken information indeholder ”Answers” feltet i DNS query response telegrammet, og hvad betyder det (dokumenter med Wireshark)?**

Vi vælger respons telegrammet:

Et billede, der indeholder tekst, software, Webside, Website

Indhold genereret af kunstig intelligens kan være forkert.

Figur 3 - DNS respons telegram

Answers indeholder

* Navn på domænet: www.au.dk
* Typen: A (IPv4)
* Klasse: IN = Internet
* Den tidsinterval informationen lagres på siden før den glemmes/opdateres.
* Data længde: 4
* IP-adressen for [www.au.dk](http://www.au.dk), som er 185.45.20.48
  1. **Hvilken Source Port har telegrammet (dokumenter med Wireshark)?**

Et billede, der indeholder tekst, skærmbillede, software, Webside

Indhold genereret af kunstig intelligens kan være forkert.

Figur 4 Source Port for [www.au.dk](http://www.au.dk) DNS telegram

Telegrammet har source port 56161.

* 1. **Generes der yderligere DNS-opslag fra selve hjemmesiden efter det initielle DNS-opslag (indsæt oversigt over DNS-telegrammer – max et screen dump fra Wireshark)?**

Vi filtrerer, så vi kun ser DNS telegrammer:

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, nummer/tal

Indhold genereret af kunstig intelligens kan være forkert.

Figur 5 - Samtlige DNS telegrammer

Ja, au.dk fremgår på adskillige linjer efterfølgende.

**Delay**

**2. Mål den tid der går fra en ping kommando startes i H1 til ping-respons fra H2 modtages i H1.**

Et billede, der indeholder tekst, elektronik, skærmbillede, software

Automatisk genereret beskrivelse

Figur 6 - Terminal med ping kommando

|  |  |  |
| --- | --- | --- |
| 5 ping: | 1 ping: | 19 ping: |
| Min = 0,854 ms  Avg = 1,685 ms  Max = 2,921 ms  Mdev = 0,699 ms | Min = 1,219 ms  Avg = 1,219 ms  Max = 1,219 ms  Mdev = 0,000 ms | Min = 0,421 ms  Avg = 1,135 ms  Max = 2,214 ms  Mdev = 0,495 ms |

**3. Mål minimum-/maksimum-/gennemsnits-forsinkelsestider og standardafvigelsen for 10 på hinanden følgende ping-kommandoer, udført som i punkt 2.**

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Figur 7 - 10 ping vist I terminalen

Min = 0,732 ms

Avg = 1,499 ms

Max = 2,377 ms

Mdev = 0,600 ms

*I de følgende punkter skal der måles den tidsforsinkelse der opstår, når den virtuelle maskine via eth0 kommunikerer med en server, som er placeret et sted på internettet.*

**4. Mål den tid der går fra kommandoen ping –c 1 www.google.dk startes i H1 til ping-respons fra web-serveren www.google.dk modtages i H1.**

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede

Automatisk genereret beskrivelse

Figur 8 - enkelt ping på [www.google.dk](http://www.google.dk) i terminalen

Min = 20,549 ms

Avg = 20,549 ms

Max = 20,549 ms

Mdev = 0,000 ms

**5. Mål minimum-/maksimum-/gennemsnits-forsinkelsestider og standardafvigelsen for 10 på hinanden følgende ping-kommandoer, der tester** [**www.google.dk**](http://www.google.dk)

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, software

Automatisk genereret beskrivelse

Figur 9 - 10 ping på www.google.dk

Min = 20,845 ms

Avg = 22,077 ms

Max = 23,331 ms

Mdev = 0,913 ms

**6. Anvend Wireshark til måling af tidsforsinkelsen til www.au.dk (Bemærk: www.au.dk understøtter ping respons, men måling af responstiden SKAL foregå med Wireshark i dette punkt ved at analysere SYN / SYN-ACK). Anvend din Web Browser i H1 som klient.**

Vi beregner RTT ved at se på tidsforsinkelsen mellem hvornår en SYN-pakke bliver sendt fra klienten og serveren svarer med en SYN-ACK-pakke:

Et billede, der indeholder tekst, nummer/tal, software, Font/skrifttype

Indhold genereret af kunstig intelligens kan være forkert.

Figur 10 - TCP pakker i wireshark med fokus på SYN / SYN-ACK

Tiden for den første SYN-pakke: 0,053973776 sekunder

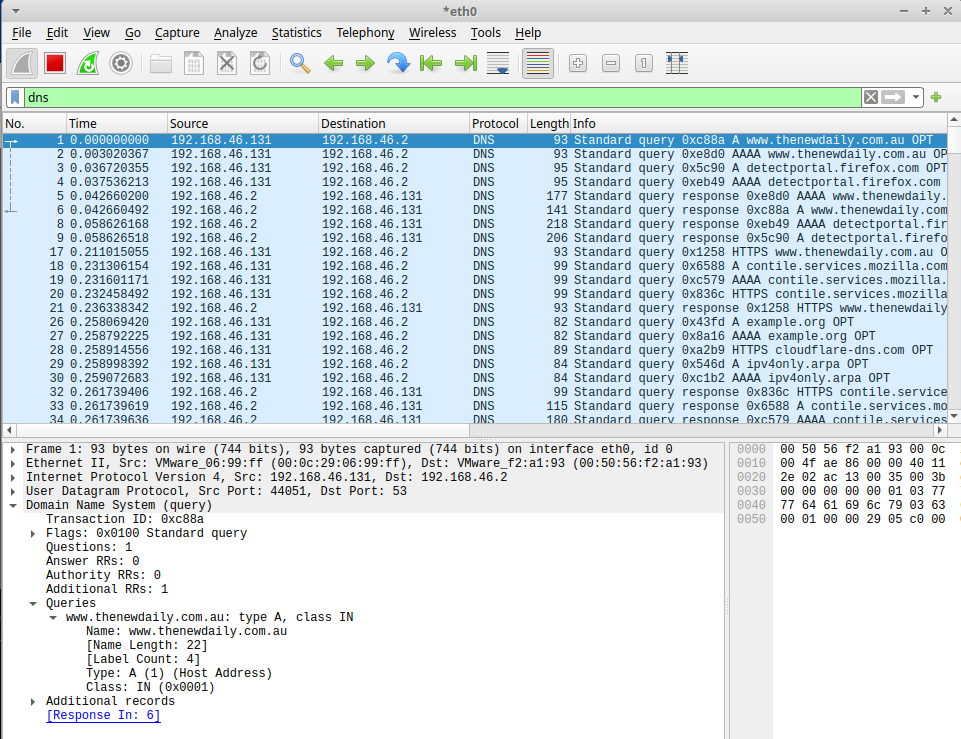
Tiden for SYN-ACK pakken: 0,080810247 sekunder

Vi beregner RTT:

**7. Udvælg en australsk web-server og undersøg vha. Wireshark DNS-opslag for denne server. Dokumenter med Wireshark.**

Vi har valgt [www.thenewdaily.com.au](http://www.thenewdaily.com.au)

**7.1 Hvilken information indeholder ”Queries” feltet i DNS query telegrammet (dokumenter med Wireshark)?**



Figur - DNS pakker for australsk hjemmeside

* Hjemmesidens navn: [www.thenewdaily.com.au](http://www.thenewdaily.com.au)
* Hvor mange symboler der er i navnet: 22 (inklusiv punktum)
* Antal dele i navnet: 4 (www, thenewdaily, com, au)
* Typen: A (IPv4)
* Klasse: IN = internet
  1. **Hvilken information indeholder ”Answers” feltet i DNS query response telegrammet, og hvad betyder det (dokumenter med Wireshark)?**

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, linje/række

Automatisk genereret beskrivelse

Figur 12 - Answer felt i DNS query response telegrammet

Answers indeholder samtlige IP-adresser for domænet.

* 1. **Hvilken ip-adresse har den australske server (dokumenter med Wireshark)?**

IP-adresserne fremgår i forrige screenshot.

IP-adresserne er:

104.26.6.91

104.26.7.91

172.67.69.42

**8. Anvend Wireshark til måling af responstiden til den australske server fra pkt. 7. Anvend din Web Browser i H1 som klient. Måling af responstiden SKAL foregå med Wireshark i dette punkt ved at analysere SYN / SYN-ACK.**

Vi skal igen beregne RTT ved at se på tidsforsinkelsen mellem hvornår en SYN-pakke bliver sendt fra klienten og serveren svarer med en SYN-ACK-pakke:

Et billede, der indeholder tekst, software, Webside, Website

Indhold genereret af kunstig intelligens kan være forkert.

Figur - TCP pakker for den australske hjemmeside

Tiden for den første SYN-pakke: 1,284921258 sekunder

Tiden for SYN-ACK pakken: 1,300303060 sekunder

Vi beregner RTT:

**9. Analyser målingerne i punkt 6 og punkt 8.**

Man ville forvente at der er en større tidsforsinkelse til den australske hjemmeside, men umiddelbart er det ikke det man ser.

* 1. **Beregn forskellen mellem responstiderne for de 2 servere?**
  2. **Hvad kan årsagen til denne forskel være?**

Selvom man ville forvente at responstiden er længere for en australsk server, kunne forespørgslen måske blive håndteret af en server i Europa eller hurtige udbydere, som ikke påvirker distancen.

# Øvelse 4

**HTTP Client/Server**

*Formålet med øvelsen er at anvende Wireshark til analyse af netværksfunktionalitet og at undersøge http-protokollens funktionalitet.*

**4.1 Undersøg vha. Wireshark hvad der sker, når denne webside hentes vha. en web-browser:**

**http://iha-i4prj2.uni.au.dk**

*DNS-opslag og DNS-respons:*

**Hvilken relevant information indeholder DNS opslaget?**

Et billede, der indeholder tekst, skærmbillede, software, nummer/tal

Indhold genereret af kunstig intelligens kan være forkert.

Figur - DNS opslag ved indlæsning af http://iha-i4prj2.uni.au.dk

DNS opslaget indeholder:

* Transaction ID: 0x6da0
* Flags: 0x0100 Standard Query - Det er en standard forespørgsel
* Questions: 1 - Der er en enkelt DNS forespørgsel i pakken
* Queries - Navn på domænet, type etc.

**Hvilken relevant information indeholder DNS responset?**

Et billede, der indeholder tekst, software, nummer/tal, Font/skrifttype

Indhold genereret af kunstig intelligens kan være forkert.

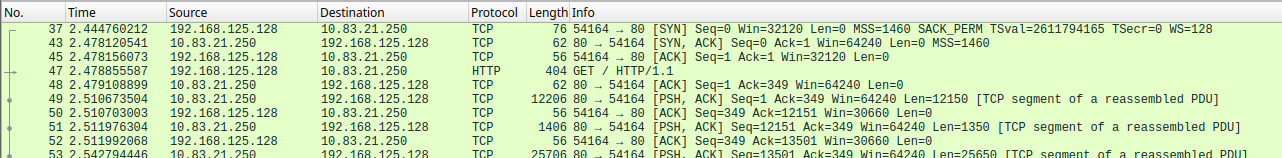
Figur - DNS respons ved indlæsning af http://iha-i4prj2.uni.au.dk

DNS-responset indeholder:

* Transaction ID: 0x6da0 - skal matche med opslaget
* Flags: 0x8180 Standard query response, no error
* Questions: 1 - Der bliver svaret på en enkelt forespørgsel i pakken
* Queries - Navn på domænet, type etc.

*Tidsforsinkelse til web siden:*

**Mål Round Trip Time (RTT) til server ved at måle på SYN / SYN-ACK telegrammerne?**



Figur - SYN / SYN-ACK telegrammer til udregning af RTT

Tiden for den første SYN-pakke: 2,444760212 sekunder

Tiden for SYN-ACK pakken: 2,478120541 sekunder

Vi beregner RTT:

*http request header:*

**Vis indhold af http request header telegram der sendes til server.**

Et billede, der indeholder tekst, Font/skrifttype, nummer/tal, linje/række

Indhold genereret af kunstig intelligens kan være forkert.

Figur 17 - Indhold af http request header

Indhold af request header:

* GET / http/1.1
* Host: iha-i4prj2.uni.au.dk - Værtsnavnet på serveren
* User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:136.0) Gecko/20100101 Firefox/136.0 - Fortæller at vi har at gøre med en Firefox brower på Ubuntu i Linux
* Accept: text/html, application/xhtml+xml, application/xml;q=0.9, \* /\*;q=0.8 - Fortæller at klienten accepterer HTML-sider, XHTML og XML som foretrukne formater
* Accept-Language: en-US, en;q=0.5 - Klienten foretrækker engelsk
* Accept-Encoding: gzip, deflate
* Connection: keep-alive
* Osv.

*http respons header:*

**Vis indhold af http respons header telegram der modtages fra server.**

**Et billede, der indeholder tekst, software, nummer/tal, Font/skrifttype

Indhold genereret af kunstig intelligens kan være forkert.**

Figur 18 - Indhold af http respons header

Indhold af respons header:

* HTTP/1.1 200 OK - Andmodningen var en succes
* Content-Type: text/html
* Accept-Ranges: bytes - Serveren understøtter range requests
* ETag: "349159968078db1” - Hvis klienten har ETag i sin cache, kan den undlade at downloade filen igen
* Server: Microsoft-IIS/10.0

**Hvilken webserver type anvendes, og hvilken version har den?**

I forrige spørgsmål kunne vi aflæse i respons headeren at der anvendes Microsoft IIS versions 10.0.

**Hvor mange linjer data (text/html) modtages der i http respons?**

Et billede, der indeholder tekst, software, Webside, nummer/tal

Indhold genereret af kunstig intelligens kan være forkert.

Figur 19 - Line-based text data i respons header

Der modtages 5538 linjer.

**Vis indhold af de første linjer (text/html) i http respons telegram der modtages fra server.**

De første linjer ses nedenfor:

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Indhold genereret af kunstig intelligens kan være forkert.

Figur - Første linjer i http telegrammet

**Hvor lang tid går der fra der er afsendt en http GET fra klienten til der modtages en http 200 OK fra serveren?**

GET ses på linje 47 og 200 OK er på linje 75:

Et billede, der indeholder tekst, skærmbillede, nummer/tal, Font/skrifttype

Indhold genereret af kunstig intelligens kan være forkert.

Figur 21 screenshot af wireshark

Tid ved GET: 2,478855587

Tid ved 200 OK: 2,642413681

Tid:

**4.2. Test http 1.1 protokollen vha. telnet med fokus på:**

* **Oprettelse/nedlukning af TCP-connection**
* **Persistent/non-persistent HTTP-kommunikation vha. HTTP-protokollen (uden/med pipelining).**

Dette testes og ses nedenfor i terminalen:

Et billede, der indeholder tekst, elektronik, skærmbillede, computer

Indhold genereret af kunstig intelligens kan være forkert.

Figur 22 - 1.1 protokol vha. telnet

*Bemærkning: 4.3 er lavet forinden denne test…*

**Hvor lang tid går der inden en TCP-forbindelsen lukkes når HTTP 1.1 anvendes?**

Der kan ses i nedenstående figur, at tiden mellem den sidste ACK og FIN er cirka 5.027 sekunde:



Figur 23 - Tid for TCP forbindelse

Yderligere dokumentation der viser GET-request og 200 OK telegrammerne:

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, nummer/tal

Indhold genereret af kunstig intelligens kan være forkert.

Figur 24 - GET-request og 200 OK telegrammer i samme screenshot

**Hvad er fordelen ved at nedlukningen af TCP-forbindelsen udskydes?**

I stedet for at lukke med det samme holder den forbindelsen åben indtil alt er hentet, før forbindelsen lukkes.

**Er det web-server eller web-client, der starter nedlukning af TCP-forbindelsen?**



Figur 25 - Nedlukning af TCP-forbindelse

Vi kan se ovenfor i Wireshark at nedlukning kommer fra web-serveren, altså 10.0.0.1.

**Fremgår version af Apache serveren af http respons, og I givet fald hvor?**

Den fremgår som Apache version 2.4.58 under 200 OK telegrammet:

Et billede, der indeholder tekst, skærmbillede, software, nummer/tal

Indhold genereret af kunstig intelligens kan være forkert.

Figur - Apache server version

**4.3. Anvend Firefox web-browser som web-client i H2 sammen med apache-server i H1**

Her ses et screenshot af web-browseren:

Et billede, der indeholder banan, frugt, skærmbillede, Saba-banan

Indhold genereret af kunstig intelligens kan være forkert.

Figur 27 screenshot af web-browser

**Analyser vha. Wireshark relevante hændelser på LAN-segmentet. Herunder: dokumenter den anvendte:**

1. **HTTP-version - Wireshark**

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, nummer/tal

Indhold genereret af kunstig intelligens kan være forkert.

Figur 28 http-version

HTTP/1.1

1. **Request header – Wireshark /beskriv indhold**

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede

Indhold genereret af kunstig intelligens kan være forkert.

Figur 29 - Request header

Request headeren indeholder:

* HTTP/1.1 bruges som protocol.
* Host’en er 10.0.01.
* Browseren identificeres som Firefox.
* Samtlige “Accept”, der beskriver ting som understøttede billedformater og sprog.
* ”Connection: keep-alive” fortæller at forbindelsen skal holdes åben for flere forespørgsler og ikke lukke efter én request.
* Signal til serveren om, at klienten foretrækker en HTTPS forbindelse.
* Angivelse af prioritet af forespørgslen.

1. **Response header – Wireshark/beskriv indhold**

Et billede, der indeholder tekst, skærmbillede, software, Webside

Indhold genereret af kunstig intelligens kan være forkert.

Figur 30 screenshot af wireshark

Respons headeren indeholder:

* HTTP/1.1 200 OK indikerer at anmodningen blev behandlet som den skulle og serveren returnerede det ønskede indhold.
* Dato for tidspunkt af serverens respons, i dette tilfælde mandag d. 10 februar 2025.
* Serveren kører på en ubuntu server.
* Tidspunkt for hvornår den sidst returnerede ressource blev ændret, i dette tilfælde er det også mandag d. 10 februar 2025.
* ”Accept-Ranges: bytes” fortæller at serveren understøtter delvise downloads.
* ”Keep-Alive: timeout=5, max=100” fortæller at forbindelsen holdes åben i 5 sekunder, og at der kan foretages op til 100 anmodninger, før forbindelsen lukkes.
* ”Connection: Keep alive” fortæller at forbindelsen ikke lukkes efter responsen.
* Respons indholdet er en HTML-side.

1. **Persistent/non-persistent connection – der indsættes en oversigt fra Wireshark over den samlede kommunikation mellem klient og server. Bemærk at hvis de 3 billeder overføres i en tcp-connection er det en persistent connection. Hvis de 3 billeder hver kræver en connection er det en nonpersistent connection. Hvilken connection der anvendes dokumenteres med Wireshark, ved at observere hvordan billederne overføres.**

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, dokument

Indhold genereret af kunstig intelligens kan være forkert.

Figur - TCP telegrammer

Vi ser at der er en persistent connection fordi FIN først opstår sidst, efter billederne er hentet.

# Øvelse 5

**DNS client**

**Afprøv kommandoerne host og nslookup på forskellige web-sites. Anvend også options til kommandoerne, så detaljeret information bliver udskrevet (hint til at finde detaljeret information om en kommando: man kommando, for host’s vedkommende kan du også prøve at skrive host <enter> )**

Her kan der ses hvad der forekommer når kommandoen køres på www.au.dk:

Et billede, der indeholder tekst, software, Multimediesoftware, Webside

Automatisk genereret beskrivelse

Figur - kommandoer køres på [www.au.dk](http://www.au.dk)

Et billede, der indeholder tekst, software, Font/skrifttype, Multimediesoftware

Automatisk genereret beskrivelse

Figur 33 - yderligere kommando køres på [www.au.dk](http://www.au.dk)

Her kan der ses hvad der forekommer når kommandoen køres på www.google.dk:

Et billede, der indeholder tekst, elektronik, skærmbillede, software

Automatisk genereret beskrivelse

Figur 34 - kommandoer køres på [www.google.dk](http://www.google.dk)

Her kan der ses hvad der forekommer når kommandoen køres på www.facebook.com:

Et billede, der indeholder tekst, skærmbillede, software, Font/skrifttype

Automatisk genereret beskrivelse

Figur 35 - kommandoer køres på www.facebook.dk

# Øvelse 6

Se filer i mappe

Accepttest:

|  |  |
| --- | --- |
| **Krav:** | **Passed** |
|  |  |
| 1. Serveren er baseret på ”Server” template i ”Exercise6-template.zip” filen. | ✓ |
| 2. Klienten er baseret på ”Client” template i ”Exercise6-template.zip” filen. | ✓ |
| 3. Serveren er på port 9000. | ✓ |
|  |  |
| Normal scenarie: – Hent billede fra Server |  |
| 1. Server startes fra en terminal (H1) med kommandoen:  *./file\_server* | ✓ |
| 2. Serveren udskriver status på opstartssekvens | ✓ |
| 3. Serveren udskriver at den venter på ”Accept” | ✓ |
| 4. Klienten startes fra en terminal (H2) med kommandoen:  *./file\_client* 10.0.0.1 <filnavn> (<filnavn> er navn på billede der skal hentes fra server) | ✓ |
| 5. Klienten udskriver <filnavn>. | ✓ |
| 6. Serveren udskriver navn på filen modtaget fra klienten. | ✓ |
| 7. Serveren udskriver størrelsen på filen | ✓ |
| 8. Klienten udskriver størrelsen på filen | ✓ |
| 9. Server og klient udskriver størrelse på segmenter der sendes indtil alle segmenter, er overført (max størrelse på segmenter er 1000 bytes) | ✓ |
| 10. Serveren afslutter og er klar til ny Accept | ✓ |
| 11. Klienten afslutter | ✓ |
| 12. Den overførte fil <filnavn> verificeres på klienten | ✓ |
|  |  |
|  |  |
| Fejl scenarie: <filnavn> findes ikke på server Pre condition: Serveren er startet. |  |
| 1. Klienten startes fra en terminal (H2) med kommandoen:  *./file\_client* 10.0.0.1 <filnavn> (<filnavn> er navn på fil der ikke findes på server) | ✓ |
| 2. Klienten udskriver filnavn (navn på billede som skal hentes fra serveren.) | ✓ |
| 3. Serveren udskriver navnet på filen modtaget fra klienten. | ✓ |
| 4. Serveren udskriver størrelsen (=0) på filen | ✓ |
| 5. Klienten udskriver en fejlbesked om at filen ikke findes på serveren | ✓ |
| 6. Serveren afslutter og er klar til ny Accept | ✓ |
| 7. Klienten afslutter | ✓ |
|  |  |

Tabel - Accepttest af TCP server

# Øvelse 7

Se filer i mappe

Accepttest:

|  |  |
| --- | --- |
| **Krav:** | **Passed** |
|  |  |
| **”Normalt scenarie:” - Server** |  |
| 1. Serveren er på port 9000 | **Check** |
| 2. Server kan startes fra en terminal med kommandoen:    *./measurement\_server* | **Check** |
| 3. Serveren udskriver ved kommandoer beskrevet under *”Normal scenarie” - klient* information om kommandoer modtaget fra klienten | **Check** |
|  |  |
| **”Normalt scenarie:” - Klient** |  |
| 1. *./get\_measurement* *10.0.0.1 u*  medfører at klienten udskriver hele indholdet af filen **uptime** modtaget fra server. | **Check** |
| 2. *./get\_measurement* *10.0.0.1 U*  medfører at klienten udskriver hele indholdet af filen **uptime** modtaget fra server. | **Check** |
| 3. *./get\_measurement* *10.0.0.1 l*  medfører at klienten udskriver hele indholdet af filen **loadavg** modtaget fra server. | **Check** |
| 4. *./get\_measurement* *10.0.0.1 L*  medfører at klienten udskriver hele indholdet af filen **loadavg** modtaget fra server. | **Check** |
| 5. *./get\_measurement* *10.0.0.1 e*  medfører at klienten udskriver fejlbesked på klienten | **Check** |
|  |  |

Tabel - Accepttest af UDP server