# SW3NGK

# Besvarelse af opgave

## 1.modul  - Øvelse 8-10

**Afleveret: [08-11-21**]

**Afleveret af**: Rasmus Baunvig Aagaard
**Gruppe: 22**

**Deltagere i afleveringen**

| Studienummer | AU-id | Navn | Studieretning |
|---|---|---|---|
| 201510642 | AU532459 | Rasmus Baunvig Aagaard | SW |

# Indhold

# Øvelse 8 - Transport Layer – UDP Client/Server, TCP Client/Server

## Indledning

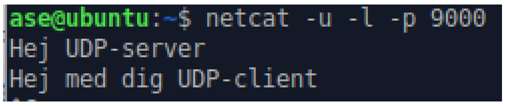
Formålet med denne øvelse er at analysere funktionalitet af UDP- og TCP-protokollerne med udgangspunkt i transportlaget.

## Opgave 1 - UDP-Client/Server:

Der skal i denne opgave anvendes en UDP-server og UDP-client, som repræsenteres på to virtuelle maskiner H1 for server og H2 for client. Til dette vil der blive benyttet netcat.

Clienten skal sende tekst strengen ”Hej UDP-server” og efterfølgende skal strengen ”Hej med dig UDP-client sendes fra serveren.

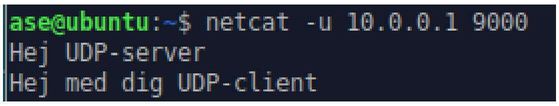
I Figur 1 vises serveren, hvor netcat kommandoen benyttes med -u (UDP) -l (listen) og -p (portnummer).



```
ase@ubuntu:~$ netcat -u -l -p 9000
Hej UDP-server
Hej med dig UDP-client
```

*Figur 1 - netcat UDP server*

I Figur 2 vises clienten, hvor netcat kommandoen benyttes med -u(UDP) 10.0.0.1(IP-adress H1) og portnummer.



```
ase@ubuntu:~$ netcat -u 10.0.0.1 9000
Hej UDP-server
Hej med dig UDP-client
```

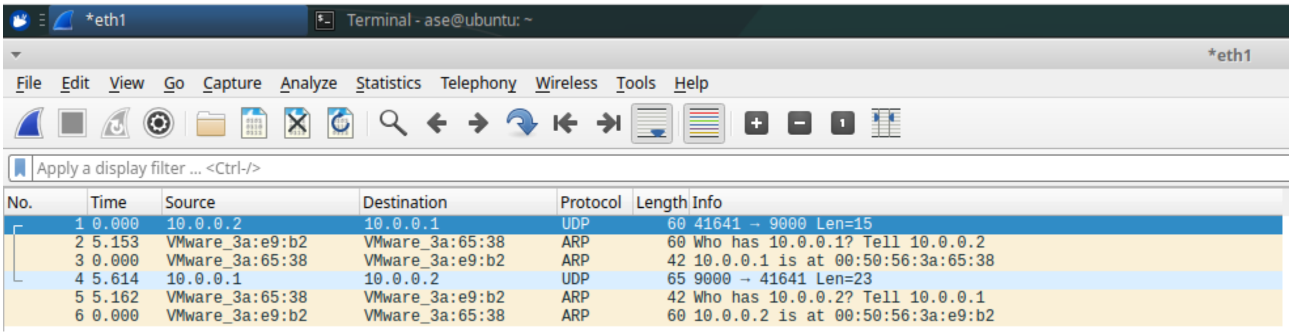
*Figur 2 - netcat UDP client*

## Konklusion på delopgave

Ved brug af netcat er det lykkedes at lave UDP client og server på de 2 virtuelle maskiner. I opgaven er det vist at clienten kan sende en tekststreng til serveren, som modtages, og at serveren kan sende en tekststreng tilbage til client, som ligeledes modtages.

## Opgave 2 - UDP Wireshark analyse

Kommunikationen mellem Client og server analyseres vha. Wireshark med fokus på hændelser i transportlaget. Nedenfor i Figur 3 ses capture fra Wireshark.



```
No.  Time   Source             Destination        Protocol  Length Info
   1 0.000  10.0.0.2           10.0.0.1           UDP       60 41641 → 9000 Len=15
   2 5.153  VMware_3a:e9:b2    VMware_3a:65:38    ARP       60 Who has 10.0.0.1? Tell 10.0.0.2
   3 0.000  VMware_3a:65:38    VMware_3a:e9:b2    ARP       42 10.0.0.1 is at 00:50:56:3a:65:38
   4 5.614  10.0.0.1           10.0.0.2           UDP       65 9000 → 41641 Len=23
   5 5.162  VMware_3a:65:38    VMware_3a:e9:b2    ARP       42 Who has 10.0.0.2? Tell 10.0.0.1
   6 0.000  VMware_3a:e9:b2    VMware_3a:65:38    ARP       60 10.0.0.2 is at 00:50:56:3a:e9:b2
```

*Figur 3 - Wireshark capture*

Der fokuseres på disse hændelser i transportlaget (UDP):

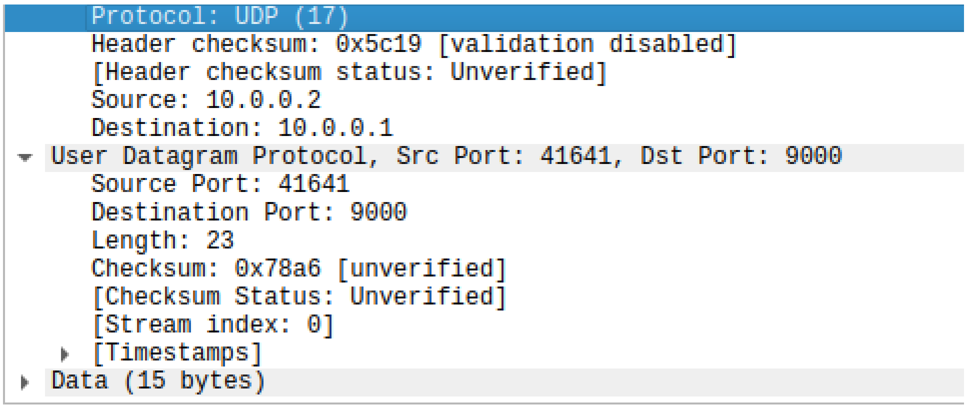
**-Eventuel connection i starten af kommunikationsforløbet i stil med en TCP-connection?**

En UDP forbindelse er connectionless, hvilket betyder der ikke er et handshake som det ses ved TCP.

**-UDP-header indhold– hver plads i headeren analyseres:**

UDP header segmentet indeholder source port#, destination port#, length(i bytes), checksum og data(payload).
I Figur 4 vises UDP header segmentet for ift. den første tekst streng som blev sendt fra clienten til server.

Der fremgår at:
Porten for client (Source Port) er 41641
Porten for serveren(Destination Port) er 9000 - hvilken blev valgt i netcat kommandoen
Længden inklusiv header(Length) er 23 bytes
Checksum er 0x78a6, men status er ikke verificerede.

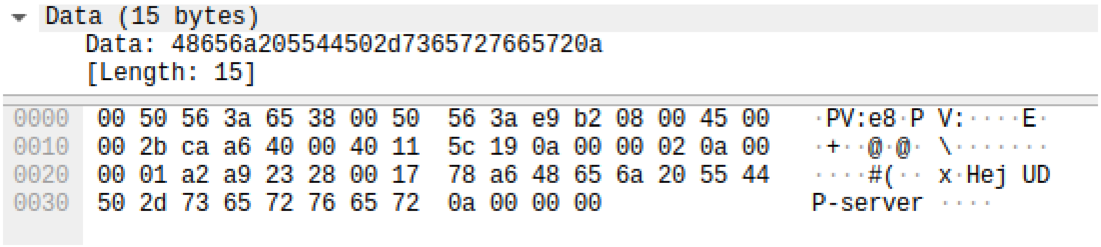


```
    Protocol: UDP (17)
    Header checksum: 0x5c19 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.0.0.2
    Destination: 10.0.0.1
▾ User Datagram Protocol, Src Port: 41641, Dst Port: 9000
    Source Port: 41641
    Destination Port: 9000
    Length: 23
    Checksum: 0x78a6 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  ▸ [Timestamps]
▸ Data (15 bytes)
```

*Figur 4 - UDP header segment*

**-UDP-payload indhold– Indholdet af payload analyseres – er indholdet krypteret, komprimeret eller er indholdet bare ”plain text”?**
**i** Figur 5 ses payload indholdet for tekst strengen ”Hej UDP-server”.

Som det fremgår nedenfor er data (Payload) 15 bytes, og vist med ASCII står beskeden som blev send i ”plain text”.



```
▾ Data (15 bytes)
     Data: 48656a205544502d7365727665720a
     [Length: 15]

0000  00 50 56 3a 65 38 00 50  56 3a e9 b2 08 00 45 00   ·PV:e8·P V:····E·
0010  00 2b ca a6 40 00 40 11  5c 19 0a 00 00 02 0a 00   ·+··@·@· \·······
0020  00 01 a2 a9 23 28 00 17  78 a6 48 65 6a 20 55 44   ····#(·· x·Hej UD
0030  50 2d 73 65 72 76 65 72  0a 00 00 00              P-server ····
```

*Figur 5 - UDP payload*

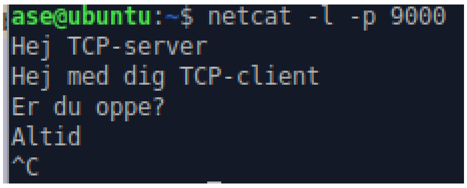
**-Eventuel nedlukning af connection i slutningen af kommunikationsforløbet i stil med en TCP-connection?**

UDP er connectionless og forbindelsen mellem client og server ophører når beskeden er sendt.

## Opgave 3 - TCP-Client/Server:

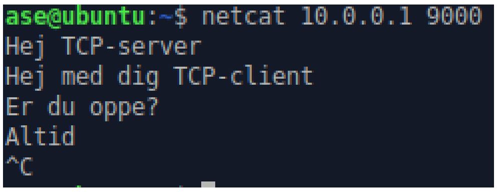
Der skal i denne opgave laves en TCP-client/server forbindelse i stedet for UDP som i tidligere opgaver. Netcat banyttes fortsat, hvor den virtuelle maskine H1 vil fungere som Server og den virtuelle maskine H2 fungere som Client.
i Figur 6 og Figur 7 ses netcat kommandoen til at oprette TCP Server og TCP Client, og kommunikationen i mellem dem.

-Fra client til server: ”Hej TCP-server”
-Fra server til client: ”Hej med dig TCP-client”
-Fra client til server: ”Er du oppe?”
-Fra server til client: ”Altid”



```
ase@ubuntu:~$ netcat -l -p 9000
Hej TCP-server
Hej med dig TCP-client
Er du oppe?
Altid
^C
```

*Figur 6 - TCP server*

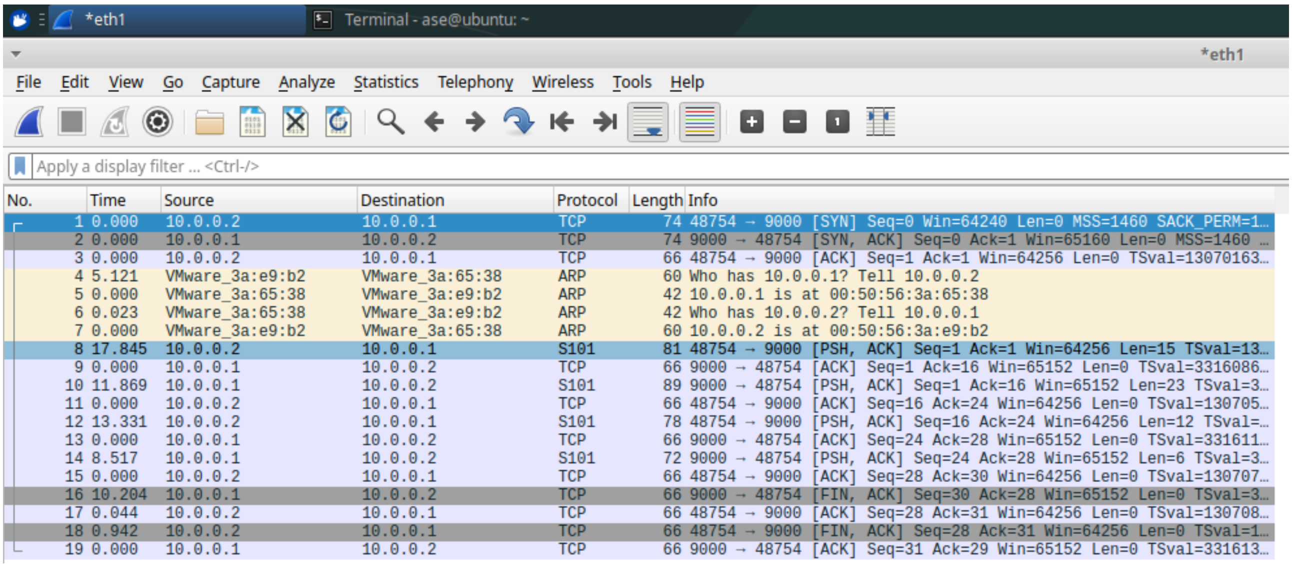


```
ase@ubuntu:~$ netcat 10.0.0.1 9000
Hej TCP-server
Hej med dig TCP-client
Er du oppe?
Altid
^C
```

*Figur 7 - TCP Client*

## Opgave 4 - TCP Wireshark analyse

Wireshark benyttes ligeledes til analyse af TCP, og nedenfor fokuseres på specifikke hændelser i transportlaget (TCP):

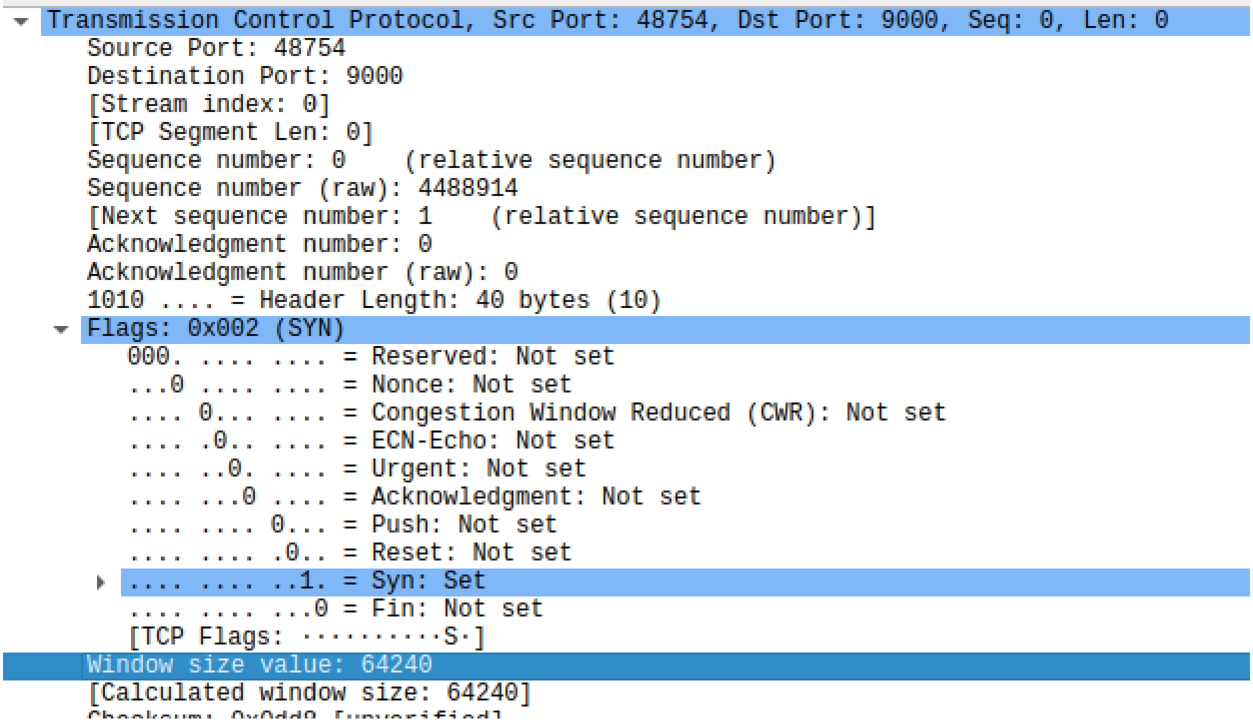


| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1 | 0.000 | 10.0.0.2 | 10.0.0.1 | TCP | 74 | 48754 → 9000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1… |
| 2 | 0.000 | 10.0.0.1 | 10.0.0.2 | TCP | 74 | 9000 → 48754 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 … |
| 3 | 0.000 | 10.0.0.2 | 10.0.0.1 | TCP | 66 | 48754 → 9000 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=13070163… |
| 4 | 5.121 | VMware_3a:e9:b2 | VMware_3a:65:38 | ARP | 60 | Who has 10.0.0.1? Tell 10.0.0.2 |
| 5 | 0.000 | VMware_3a:65:38 | VMware_3a:e9:b2 | ARP | 42 | 10.0.0.1 is at 00:50:56:3a:65:38 |
| 6 | 0.023 | VMware_3a:65:38 | VMware_3a:e9:b2 | ARP | 42 | Who has 10.0.0.2? Tell 10.0.0.1 |
| 7 | 0.000 | VMware_3a:e9:b2 | VMware_3a:65:38 | ARP | 60 | 10.0.0.2 is at 00:50:56:3a:e9:b2 |
| 8 | 17.845 | 10.0.0.2 | 10.0.0.1 | S101 | 81 | 48754 → 9000 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=15 TSval=13… |
| 9 | 0.000 | 10.0.0.1 | 10.0.0.2 | TCP | 66 | 9000 → 48754 [ACK] Seq=1 Ack=16 Win=65152 Len=0 TSval=3316086… |
| 10 | 11.869 | 10.0.0.1 | 10.0.0.2 | S101 | 89 | 9000 → 48754 [PSH, ACK] Seq=1 Ack=16 Win=65152 Len=23 TSval=3… |
| 11 | 0.000 | 10.0.0.2 | 10.0.0.1 | TCP | 66 | 48754 → 9000 [ACK] Seq=16 Ack=24 Win=64256 Len=0 TSval=130705… |
| 12 | 13.331 | 10.0.0.2 | 10.0.0.1 | S101 | 78 | 48754 → 9000 [PSH, ACK] Seq=16 Ack=24 Win=64256 Len=12 TSval=… |
| 13 | 0.000 | 10.0.0.1 | 10.0.0.2 | TCP | 66 | 9000 → 48754 [ACK] Seq=24 Ack=28 Win=65152 Len=0 TSval=331611… |
| 14 | 8.517 | 10.0.0.1 | 10.0.0.2 | S101 | 72 | 9000 → 48754 [PSH, ACK] Seq=24 Ack=28 Win=65152 Len=6 TSval=3… |
| 15 | 0.000 | 10.0.0.2 | 10.0.0.1 | TCP | 66 | 48754 → 9000 [ACK] Seq=28 Ack=30 Win=64256 Len=0 TSval=130707… |
| 16 | 10.204 | 10.0.0.1 | 10.0.0.2 | TCP | 66 | 9000 → 48754 [FIN, ACK] Seq=30 Ack=28 Win=65152 Len=0 TSval=3… |
| 17 | 0.044 | 10.0.0.2 | 10.0.0.1 | TCP | 66 | 48754 → 9000 [ACK] Seq=28 Ack=31 Win=64256 Len=0 TSval=130708… |
| 18 | 0.942 | 10.0.0.2 | 10.0.0.1 | TCP | 66 | 48754 → 9000 [FIN, ACK] Seq=28 Ack=31 Win=64256 Len=0 TSval=1… |
| 19 | 0.000 | 10.0.0.1 | 10.0.0.2 | TCP | 66 | 9000 → 48754 [ACK] Seq=31 Ack=29 Win=65152 Len=0 TSval=331613… |

*Figur 8 - TCP Wireshark Capture*

**-Oprettelse af connection: Flags, Sequence Counter, Acknowledge Counter og Window Size undersøges. Er det client eller server, der tager initiativ til oprettelsen af TCP-connection?**

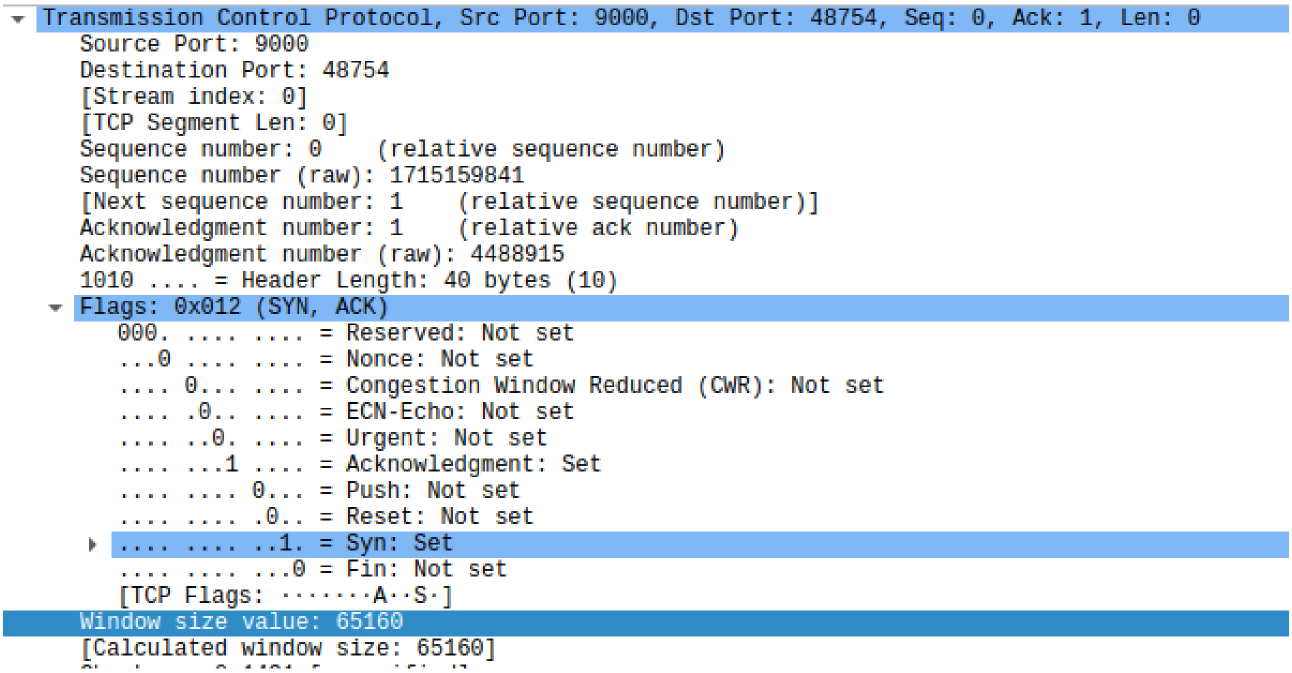
Handshake 1: Client tager kontakt til Server



```
Transmission Control Protocol, Src Port: 48754, Dst Port: 9000, Seq: 0, Len: 0
    Source Port: 48754
    Destination Port: 9000
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Sequence number (raw): 4488914
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 0
    Acknowledgment number (raw): 0
    1010 .... = Header Length: 40 bytes (10)
  Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ··········S·]
    Window size value: 64240
    [Calculated window size: 64240]
    Checksum: 0x0dd8 [unverified]
```

*Figur 9 - Handshake 1/3*

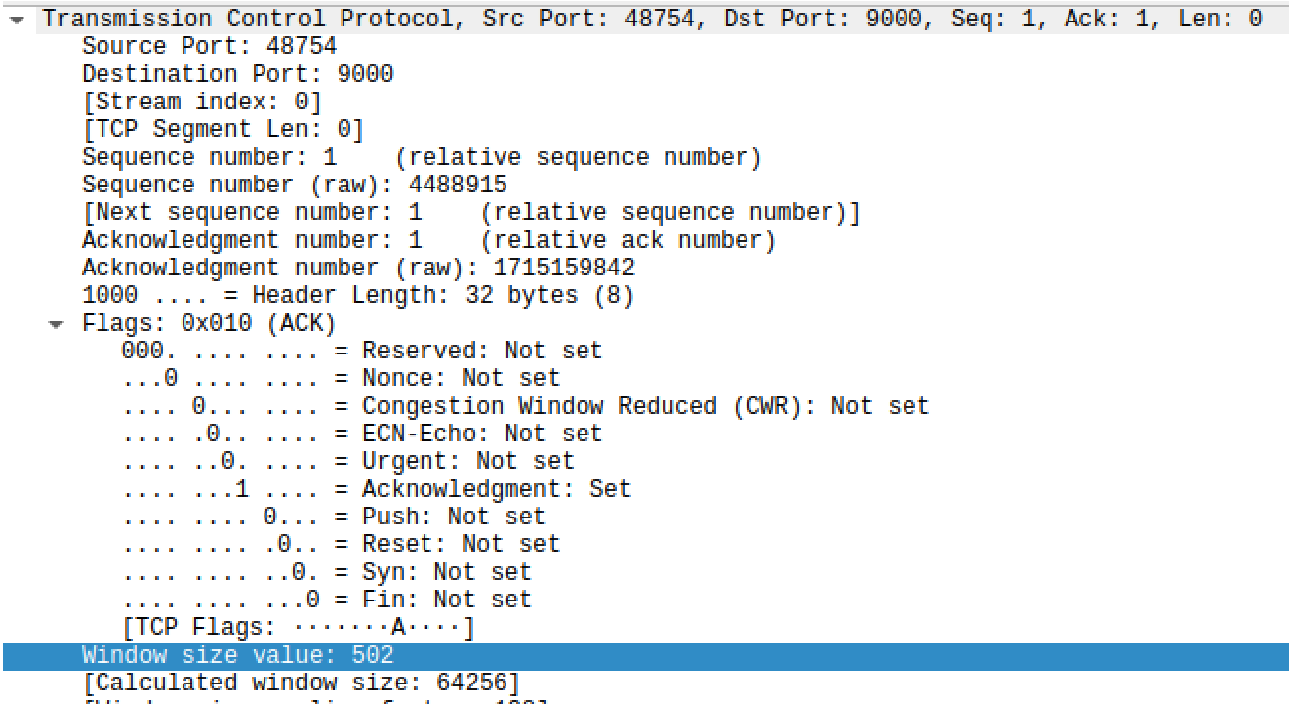
Handshake 2: Server responderer til Client



```
▾ Transmission Control Protocol, Src Port: 9000, Dst Port: 48754, Seq: 0, Ack: 1, Len: 0
    Source Port: 9000
    Destination Port: 48754
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Sequence number (raw): 1715159841
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Acknowledgment number (raw): 4488915
    1010 .... = Header Length: 40 bytes (10)
  ▾ Flags: 0x012 (SYN, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    ▸ .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·······A··S·]
    Window size value: 65160
    [Calculated window size: 65160]
```

*Figur 10 - Handshake 2/3*

Handshake 3: Client fuldender three-way handshake med kontakt til Server

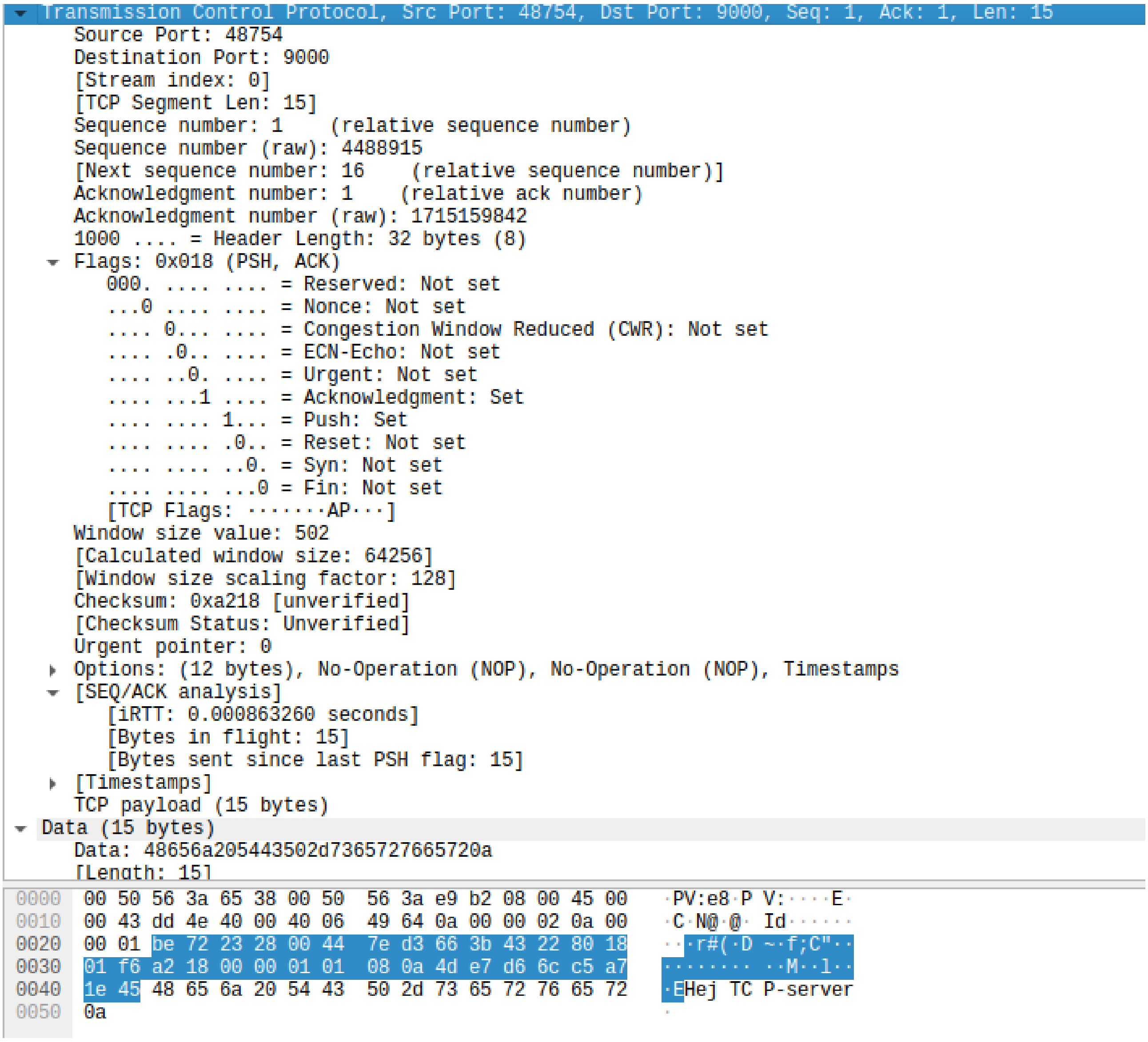


```
▾ Transmission Control Protocol, Src Port: 48754, Dst Port: 9000, Seq: 1, Ack: 1, Len: 0
    Source Port: 48754
    Destination Port: 9000
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    Sequence number (raw): 4488915
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Acknowledgment number (raw): 1715159842
    1000 .... = Header Length: 32 bytes (8)
  ▾ Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·······A····]
    Window size value: 502
    [Calculated window size: 64256]
```

*Figur 11 - Handshake 3/3*

**-Overførsel af applikationslags-data: Flags, Sequence Counter, Acknowledge Counter og Window Size undersøges.**

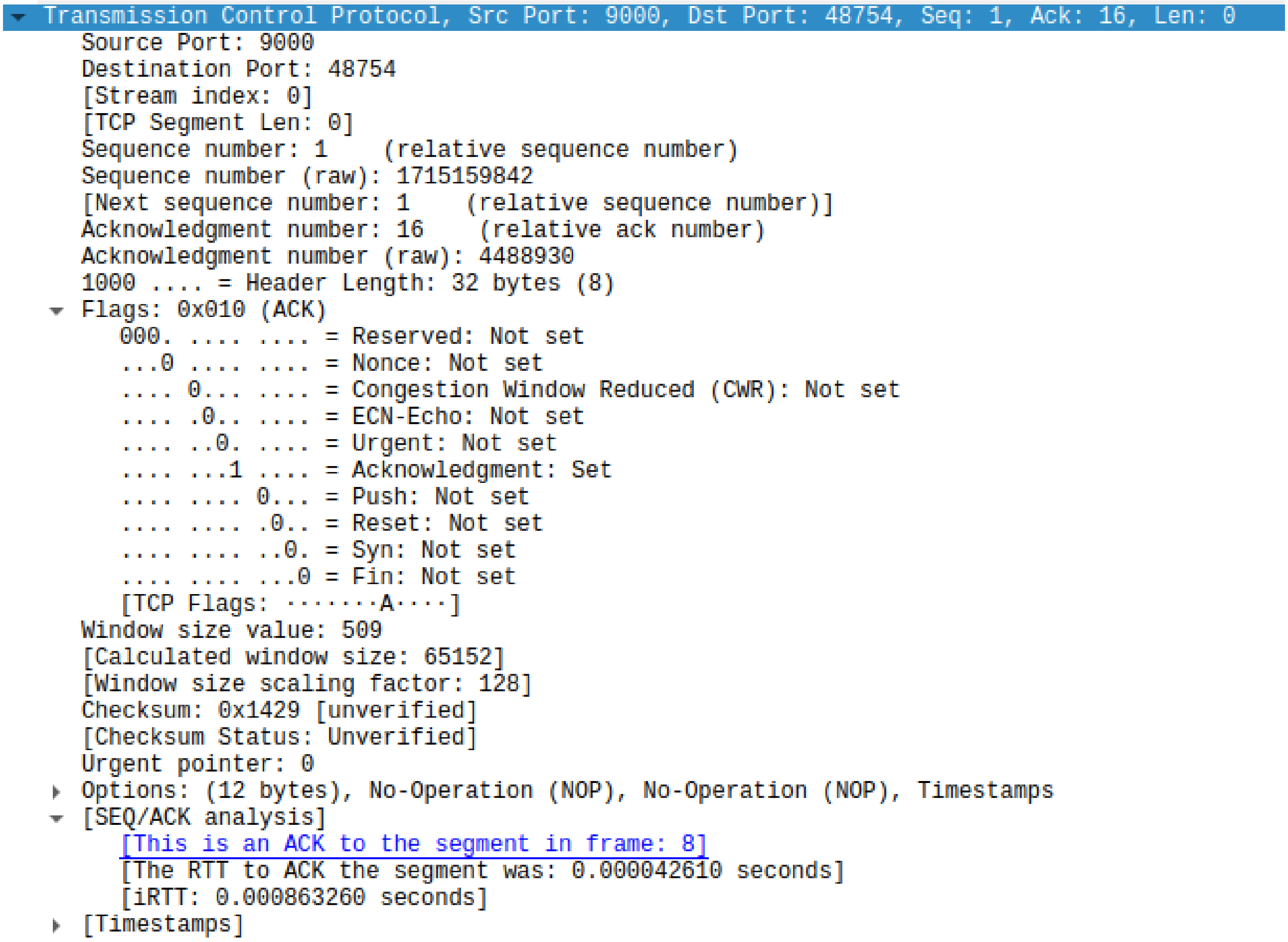
Besked sendes far Client til Server [PSH, ACK] - ”Hej TCP-server”



```
▾ Transmission Control Protocol, Src Port: 48754, Dst Port: 9000, Seq: 1, Ack: 1, Len: 15
      Source Port: 48754
      Destination Port: 9000
      [Stream index: 0]
      [TCP Segment Len: 15]
      Sequence number: 1    (relative sequence number)
      Sequence number (raw): 4488915
      [Next sequence number: 16    (relative sequence number)]
      Acknowledgment number: 1    (relative ack number)
      Acknowledgment number (raw): 1715159842
      1000 .... = Header Length: 32 bytes (8)
    ▾ Flags: 0x018 (PSH, ACK)
         000. .... .... = Reserved: Not set
         ...0 .... .... = Nonce: Not set
         .... 0... .... = Congestion Window Reduced (CWR): Not set
         .... .0.. .... = ECN-Echo: Not set
         .... ..0. .... = Urgent: Not set
         .... ...1 .... = Acknowledgment: Set
         .... .... 1... = Push: Set
         .... .... .0.. = Reset: Not set
         .... .... ..0. = Syn: Not set
         .... .... ...0 = Fin: Not set
         [TCP Flags: ·······AP···]
      Window size value: 502
      [Calculated window size: 64256]
      [Window size scaling factor: 128]
      Checksum: 0xa218 [unverified]
      [Checksum Status: Unverified]
      Urgent pointer: 0
    ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    ▾ [SEQ/ACK analysis]
         [iRTT: 0.000863260 seconds]
         [Bytes in flight: 15]
         [Bytes sent since last PSH flag: 15]
    ▸ [Timestamps]
      TCP payload (15 bytes)
▾ Data (15 bytes)
      Data: 48656a205443502d7365727665720a
      [Length: 15]
0000  00 50 56 3a 65 38 00 50  56 3a e9 b2 08 00 45 00   ·PV:e8·P V:····E·
0010  00 43 dd 4e 40 00 40 06  49 64 0a 00 00 02 0a 00   ·C·N@·@· Id······
0020  00 01 be 72 23 28 00 44  7e d3 66 3b 43 22 80 18   ···r#(·D ~·f;C"··
0030  01 f6 a2 18 00 00 01 01  08 0a 4d e7 d6 6c c5 a7   ········ ··M··l··
0040  1e 45 48 65 6a 20 54 43  50 2d 73 65 72 76 65 72   ·EHej TC P-server
0050  0a                                                 ·
```

*Figur 12 - Client "Hej TCP-server"*

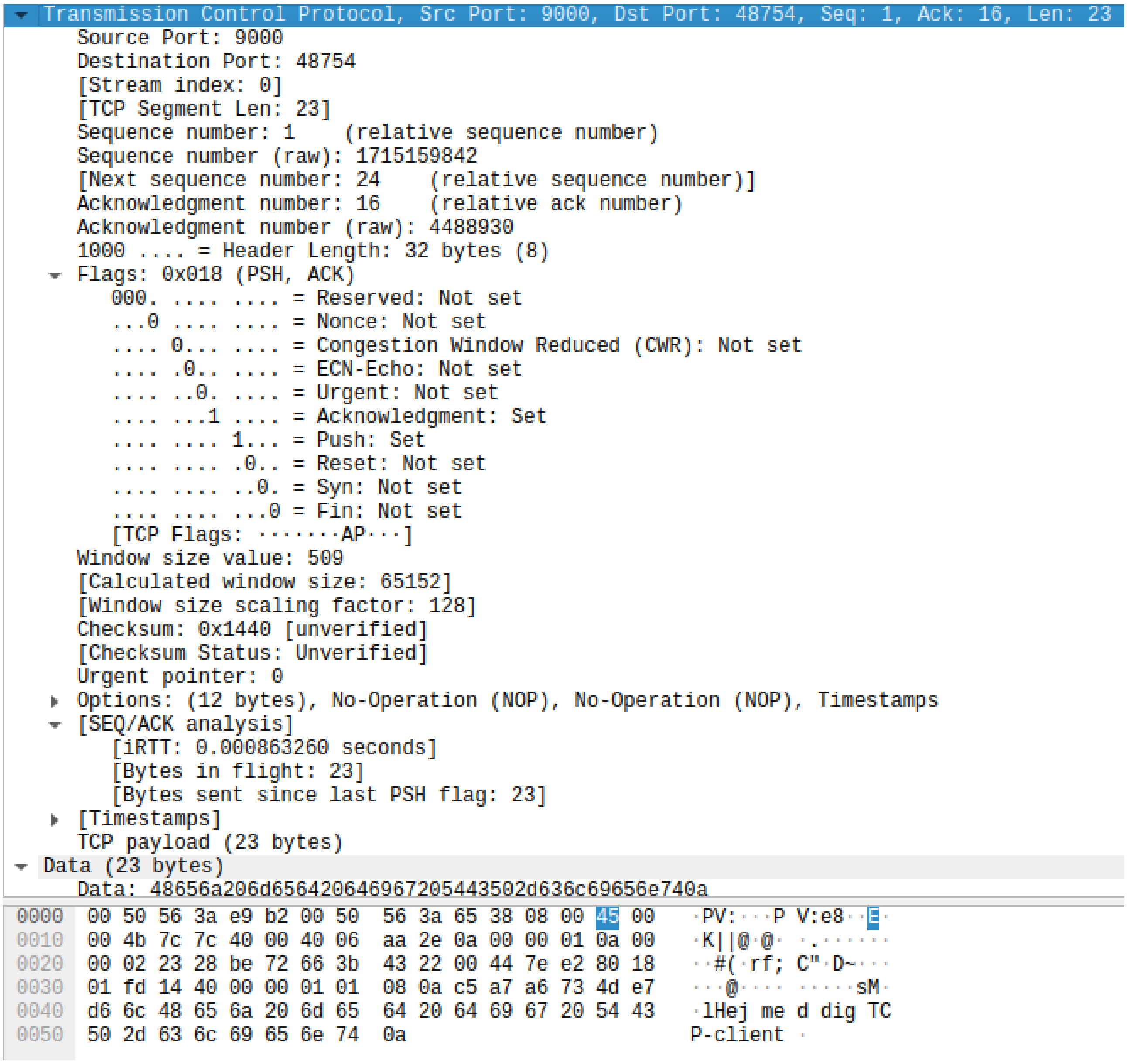
Bekræftelse af beskeden sendes til Client fra Server. [ACK]



```
▾ Transmission Control Protocol, Src Port: 9000, Dst Port: 48754, Seq: 1, Ack: 16, Len: 0
      Source Port: 9000
      Destination Port: 48754
      [Stream index: 0]
      [TCP Segment Len: 0]
      Sequence number: 1    (relative sequence number)
      Sequence number (raw): 1715159842
      [Next sequence number: 1    (relative sequence number)]
      Acknowledgment number: 16    (relative ack number)
      Acknowledgment number (raw): 4488930
      1000 .... = Header Length: 32 bytes (8)
   ▾ Flags: 0x010 (ACK)
         000. .... .... = Reserved: Not set
         ...0 .... .... = Nonce: Not set
         .... 0... .... = Congestion Window Reduced (CWR): Not set
         .... .0.. .... = ECN-Echo: Not set
         .... ..0. .... = Urgent: Not set
         .... ...1 .... = Acknowledgment: Set
         .... .... 0... = Push: Not set
         .... .... .0.. = Reset: Not set
         .... .... ..0. = Syn: Not set
         .... .... ...0 = Fin: Not set
         [TCP Flags: ·······A····]
      Window size value: 509
      [Calculated window size: 65152]
      [Window size scaling factor: 128]
      Checksum: 0x1429 [unverified]
      [Checksum Status: Unverified]
      Urgent pointer: 0
   ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
   ▾ [SEQ/ACK analysis]
         [This is an ACK to the segment in frame: 8]
         [The RTT to ACK the segment was: 0.000042610 seconds]
         [iRTT: 0.000863260 seconds]
   ▸ [Timestamps]
```

*Figur 13 - "Hej TCP-server" [ACK]*

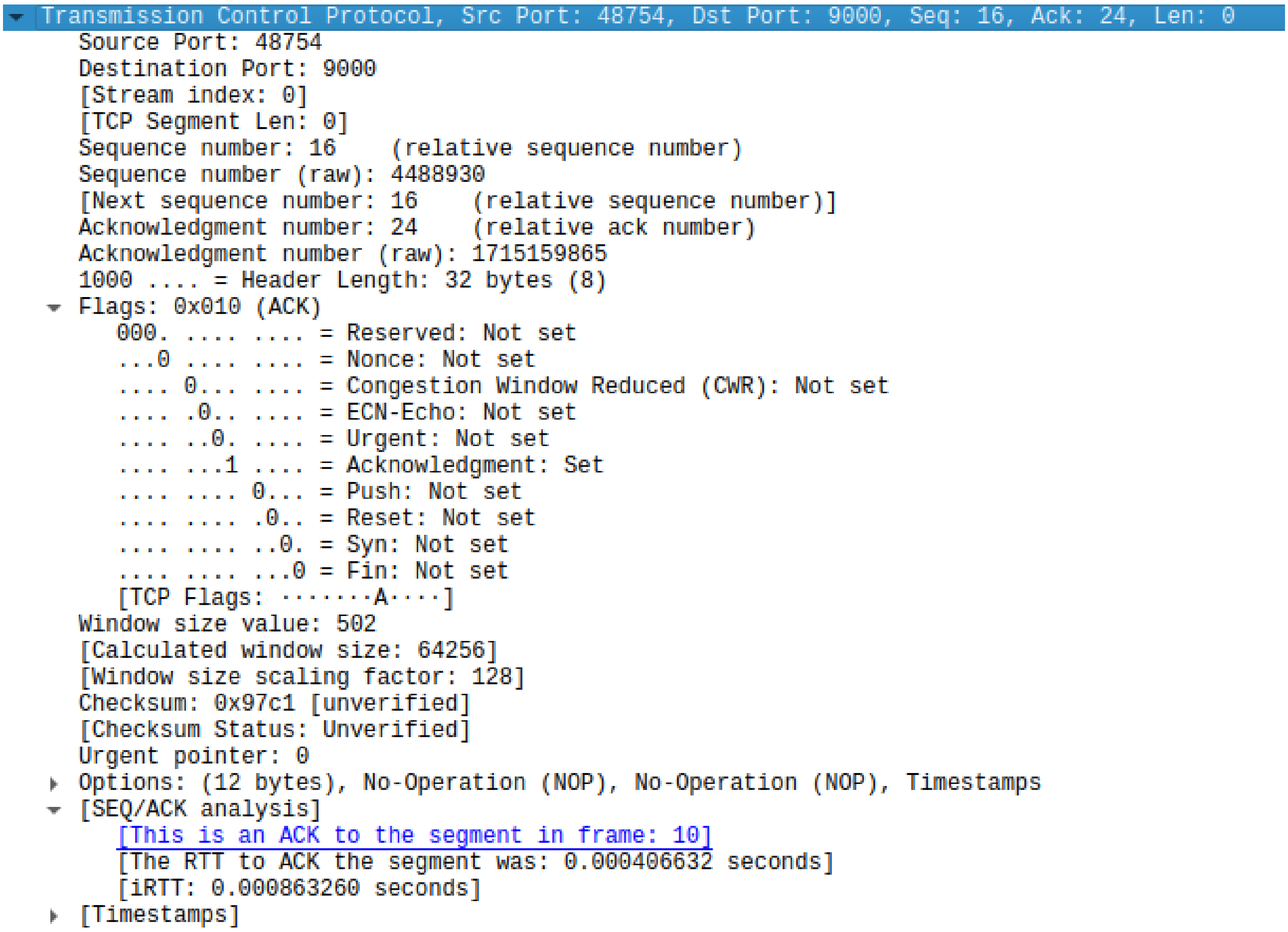
Besked fra Server til Client [PSH, ACK] - ”Hej med dig TCP-client”



```
▾ Transmission Control Protocol, Src Port: 9000, Dst Port: 48754, Seq: 1, Ack: 16, Len: 23
      Source Port: 9000
      Destination Port: 48754
      [Stream index: 0]
      [TCP Segment Len: 23]
      Sequence number: 1    (relative sequence number)
      Sequence number (raw): 1715159842
      [Next sequence number: 24    (relative sequence number)]
      Acknowledgment number: 16    (relative ack number)
      Acknowledgment number (raw): 4488930
      1000 .... = Header Length: 32 bytes (8)
   ▾ Flags: 0x018 (PSH, ACK)
         000. .... .... = Reserved: Not set
         ...0 .... .... = Nonce: Not set
         .... 0... .... = Congestion Window Reduced (CWR): Not set
         .... .0.. .... = ECN-Echo: Not set
         .... ..0. .... = Urgent: Not set
         .... ...1 .... = Acknowledgment: Set
         .... .... 1... = Push: Set
         .... .... .0.. = Reset: Not set
         .... .... ..0. = Syn: Not set
         .... .... ...0 = Fin: Not set
         [TCP Flags: ·······AP···]
      Window size value: 509
      [Calculated window size: 65152]
      [Window size scaling factor: 128]
      Checksum: 0x1440 [unverified]
      [Checksum Status: Unverified]
      Urgent pointer: 0
   ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
   ▾ [SEQ/ACK analysis]
         [iRTT: 0.000863260 seconds]
         [Bytes in flight: 23]
         [Bytes sent since last PSH flag: 23]
   ▸ [Timestamps]
      TCP payload (23 bytes)
▾ Data (23 bytes)
      Data: 48656a206d656420646967205443502d636c69656e740a

0000  00 50 56 3a e9 b2 00 50  56 3a 65 38 08 00 45 00   ·PV:···P V:e8··E·
0010  00 4b 7c 7c 40 00 40 06  aa 2e 0a 00 00 01 0a 00   ·K||@·@· ·.······
0020  00 02 23 28 be 72 66 3b  43 22 00 44 7e e2 80 18   ··#(·rf; C"·D~···
0030  01 fd 14 40 00 00 01 01  08 0a c5 a7 a6 73 4d e7   ···@···· ·····sM·
0040  d6 6c 48 65 6a 20 6d 65  64 20 64 69 67 20 54 43   ·lHej me d dig TC
0050  50 2d 63 6c 69 65 6e 74  0a                        P-client ·
```

*Figur 14 - Server "Hej med dig TCP-client"*

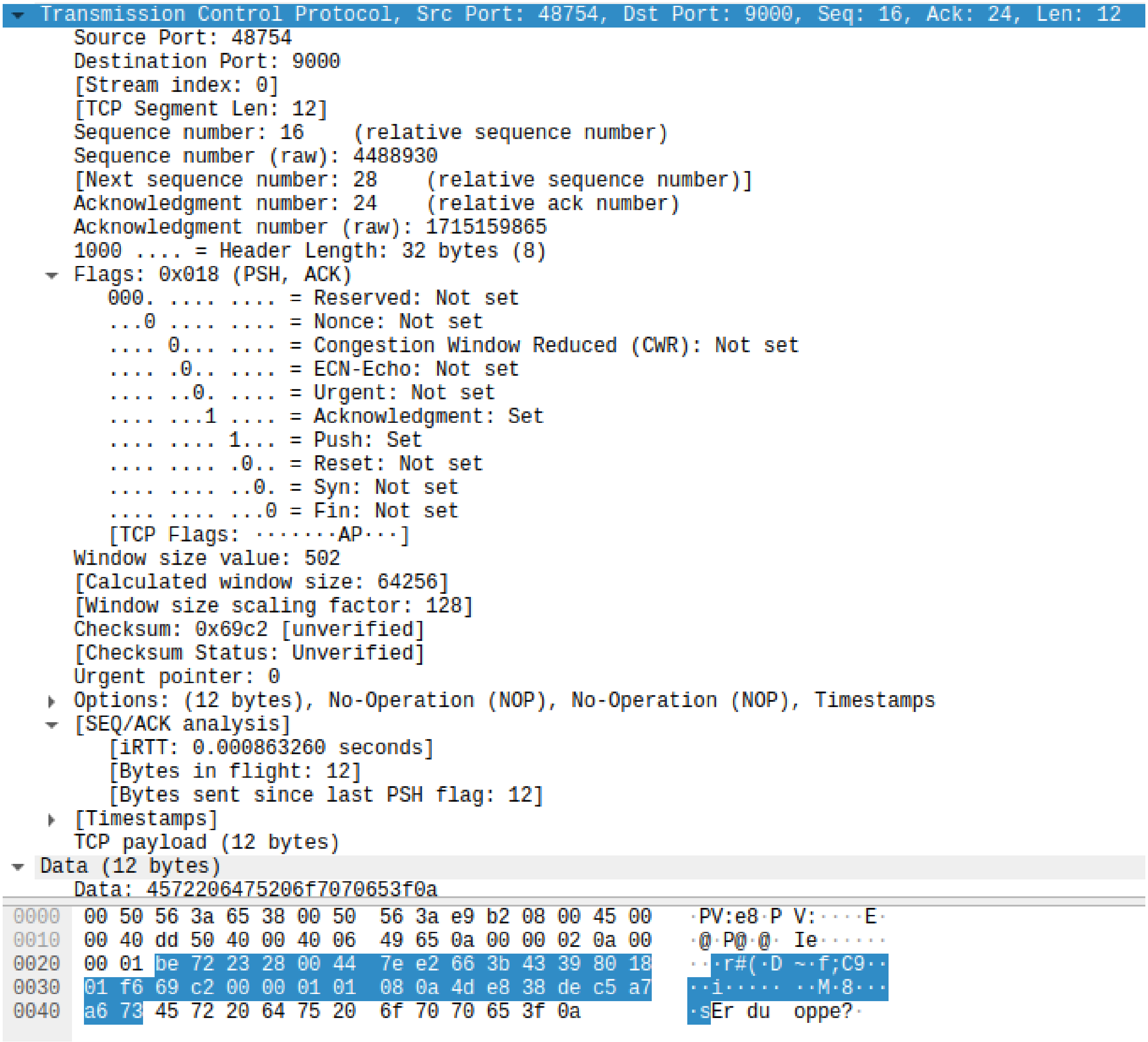
Bekræftelse af beskeden sendes fra Client til Server. [ACK]



```
Transmission Control Protocol, Src Port: 48754, Dst Port: 9000, Seq: 16, Ack: 24, Len: 0
    Source Port: 48754
    Destination Port: 9000
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 16    (relative sequence number)
    Sequence number (raw): 4488930
    [Next sequence number: 16    (relative sequence number)]
    Acknowledgment number: 24    (relative ack number)
    Acknowledgment number (raw): 1715159865
    1000 .... = Header Length: 32 bytes (8)
    Flags: 0x010 (ACK)
        000. .... .... = Reserved: Not set
        ...0 .... .... = Nonce: Not set
        .... 0... .... = Congestion Window Reduced (CWR): Not set
        .... .0.. .... = ECN-Echo: Not set
        .... ..0. .... = Urgent: Not set
        .... ...1 .... = Acknowledgment: Set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
        .... .... ..0. = Syn: Not set
        .... .... ...0 = Fin: Not set
        [TCP Flags: ·······A····]
    Window size value: 502
    [Calculated window size: 64256]
    [Window size scaling factor: 128]
    Checksum: 0x97c1 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
    Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    [SEQ/ACK analysis]
        [This is an ACK to the segment in frame: 10]
        [The RTT to ACK the segment was: 0.000406632 seconds]
        [iRTT: 0.000863260 seconds]
    [Timestamps]
```

*Figur 15 - "Hej med dig TCP-client" [ACK]*

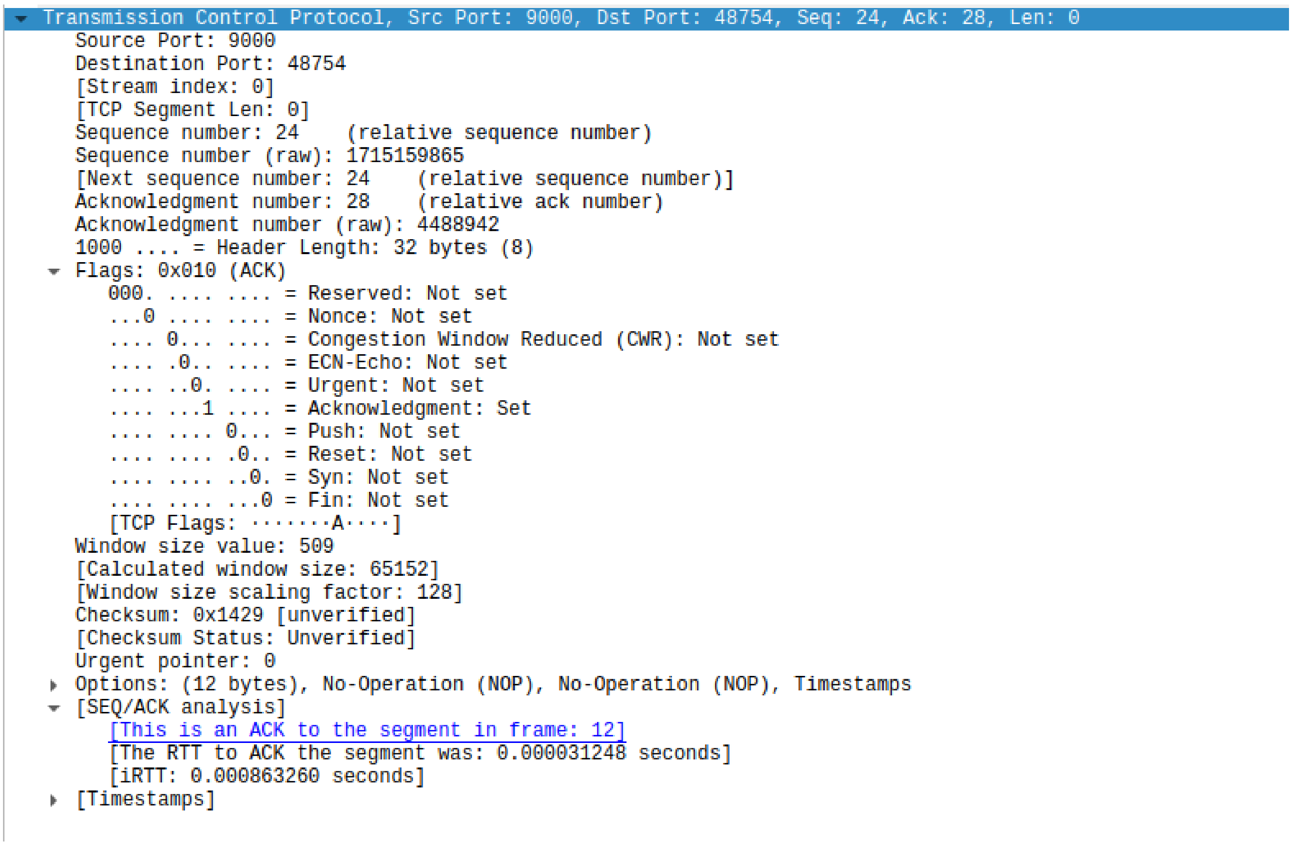
Besked til Server fra Client [PSH, ACK] - ”Er du oppe?”



```
▾ Transmission Control Protocol, Src Port: 48754, Dst Port: 9000, Seq: 16, Ack: 24, Len: 12
      Source Port: 48754
      Destination Port: 9000
      [Stream index: 0]
      [TCP Segment Len: 12]
      Sequence number: 16    (relative sequence number)
      Sequence number (raw): 4488930
      [Next sequence number: 28    (relative sequence number)]
      Acknowledgment number: 24    (relative ack number)
      Acknowledgment number (raw): 1715159865
      1000 .... = Header Length: 32 bytes (8)
    ▾ Flags: 0x018 (PSH, ACK)
         000. .... .... = Reserved: Not set
         ...0 .... .... = Nonce: Not set
         .... 0... .... = Congestion Window Reduced (CWR): Not set
         .... .0.. .... = ECN-Echo: Not set
         .... ..0. .... = Urgent: Not set
         .... ...1 .... = Acknowledgment: Set
         .... .... 1... = Push: Set
         .... .... .0.. = Reset: Not set
         .... .... ..0. = Syn: Not set
         .... .... ...0 = Fin: Not set
         [TCP Flags: ·······AP···]
      Window size value: 502
      [Calculated window size: 64256]
      [Window size scaling factor: 128]
      Checksum: 0x69c2 [unverified]
      [Checksum Status: Unverified]
      Urgent pointer: 0
    ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    ▾ [SEQ/ACK analysis]
         [iRTT: 0.000863260 seconds]
         [Bytes in flight: 12]
         [Bytes sent since last PSH flag: 12]
    ▸ [Timestamps]
      TCP payload (12 bytes)
▾ Data (12 bytes)
      Data: 4572206475206f7070653f0a
0000  00 50 56 3a 65 38 00 50  56 3a e9 b2 08 00 45 00   ·PV:e8·P V:····E·
0010  00 40 dd 50 40 00 40 06  49 65 0a 00 00 02 0a 00   ·@·P@·@· Ie······
0020  00 01 be 72 23 28 00 44  7e e2 66 3b 43 39 80 18   ···r#(·D ~·f;C9··
0030  01 f6 69 c2 00 00 01 01  08 0a 4d e8 38 de c5 a7   ··i····· ··M·8···
0040  a6 73 45 72 20 64 75 20  6f 70 70 65 3f 0a         ·sEr du  oppe?·
```

*Figur 16 - Client "Er du oppe?"*

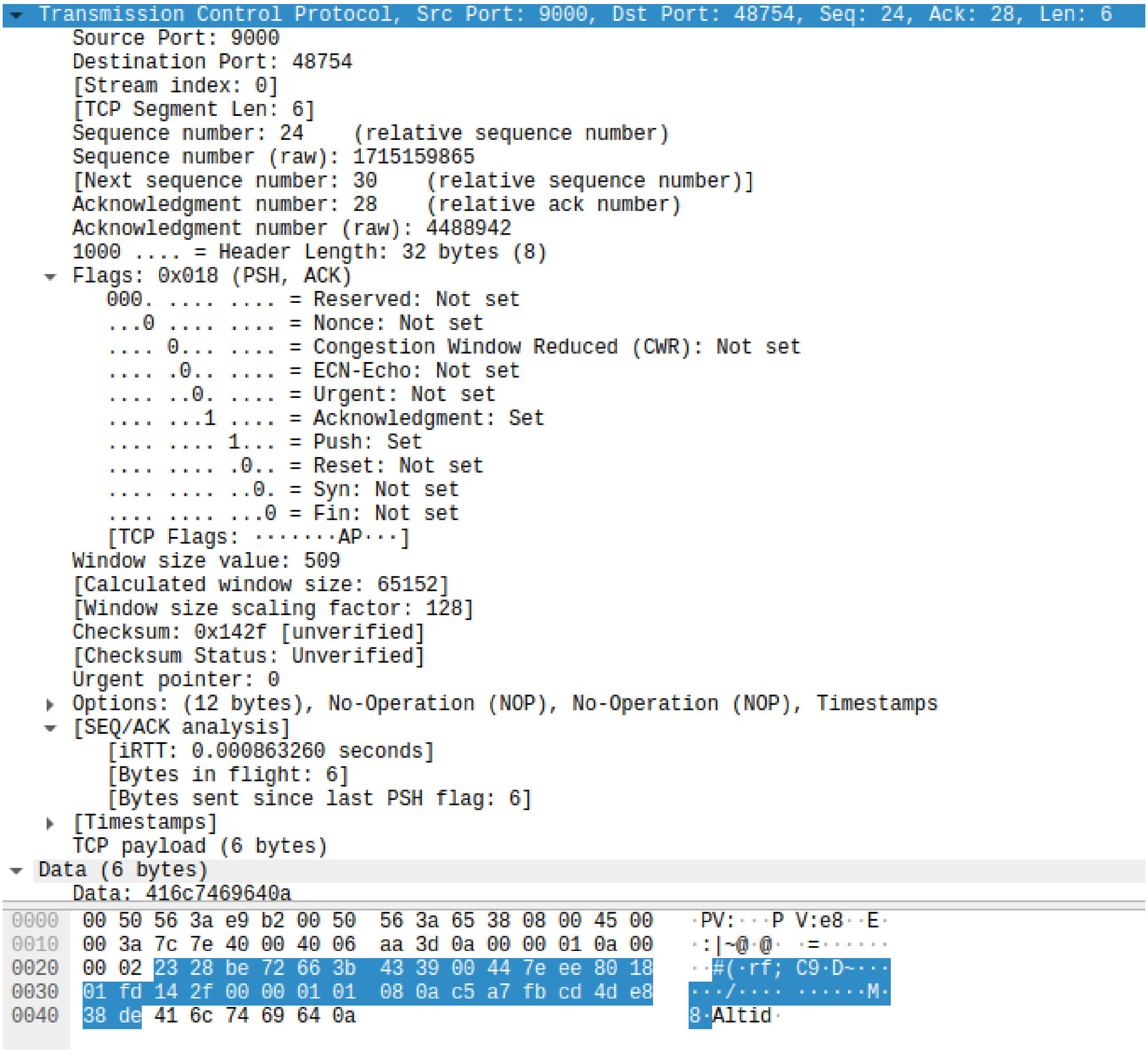
Bekræftelse af beskeden sendes til Client fra Server. [ACK]



```
▾ Transmission Control Protocol, Src Port: 9000, Dst Port: 48754, Seq: 24, Ack: 28, Len: 0
      Source Port: 9000
      Destination Port: 48754
      [Stream index: 0]
      [TCP Segment Len: 0]
      Sequence number: 24    (relative sequence number)
      Sequence number (raw): 1715159865
      [Next sequence number: 24    (relative sequence number)]
      Acknowledgment number: 28    (relative ack number)
      Acknowledgment number (raw): 4488942
      1000 .... = Header Length: 32 bytes (8)
    ▾ Flags: 0x010 (ACK)
         000. .... .... = Reserved: Not set
         ...0 .... .... = Nonce: Not set
         .... 0... .... = Congestion Window Reduced (CWR): Not set
         .... .0.. .... = ECN-Echo: Not set
         .... ..0. .... = Urgent: Not set
         .... ...1 .... = Acknowledgment: Set
         .... .... 0... = Push: Not set
         .... .... .0.. = Reset: Not set
         .... .... ..0. = Syn: Not set
         .... .... ...0 = Fin: Not set
         [TCP Flags: ·······A····]
      Window size value: 509
      [Calculated window size: 65152]
      [Window size scaling factor: 128]
      Checksum: 0x1429 [unverified]
      [Checksum Status: Unverified]
      Urgent pointer: 0
    ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    ▾ [SEQ/ACK analysis]
         [This is an ACK to the segment in frame: 12]
         [The RTT to ACK the segment was: 0.000031248 seconds]
         [iRTT: 0.000863260 seconds]
    ▸ [Timestamps]
```

*Figur 17 - "Er du oppe?" [ACK]*

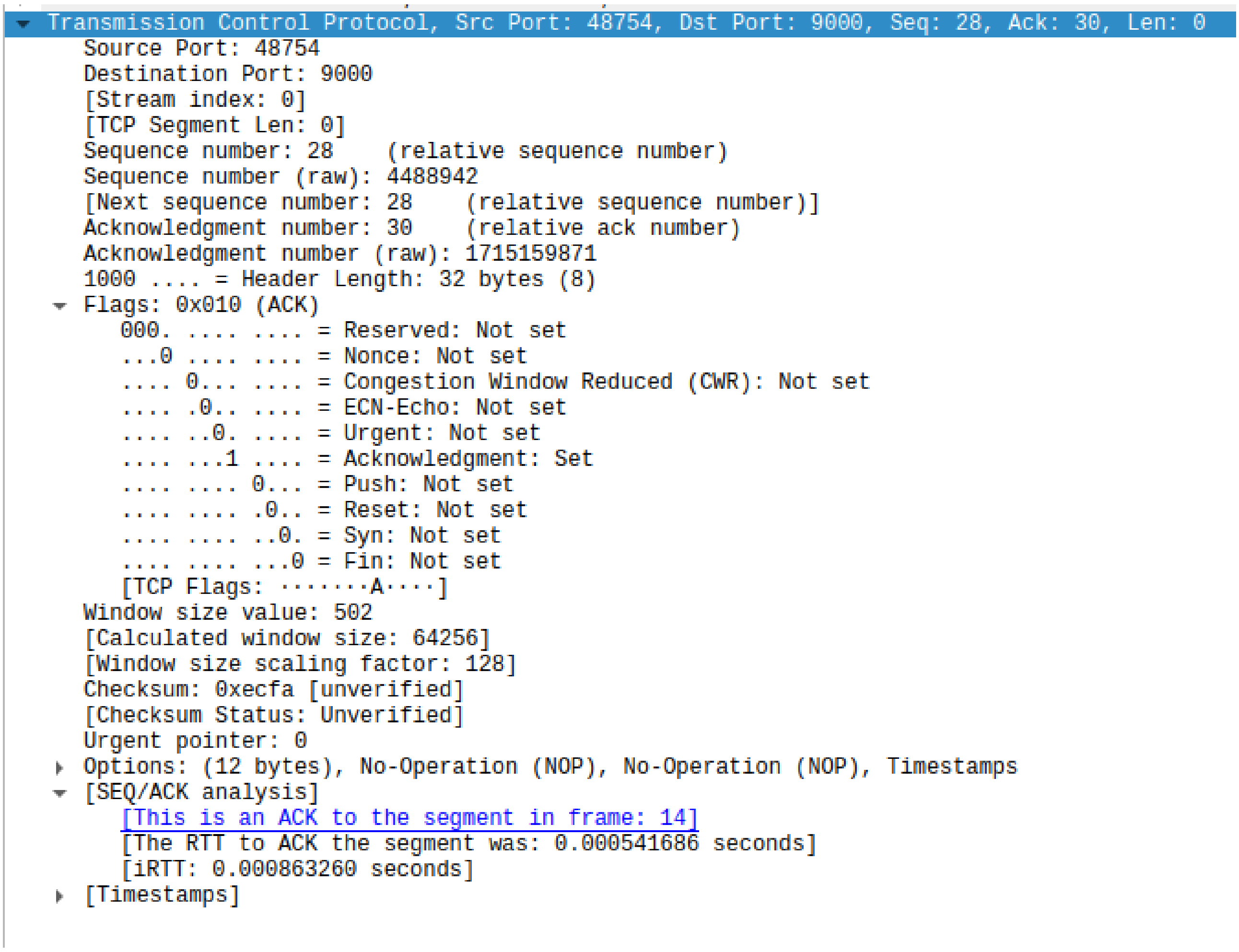
Besked fra Server til Client [PSH, ACK] - ”Altid”



```
▾ Transmission Control Protocol, Src Port: 9000, Dst Port: 48754, Seq: 24, Ack: 28, Len: 6
      Source Port: 9000
      Destination Port: 48754
      [Stream index: 0]
      [TCP Segment Len: 6]
      Sequence number: 24    (relative sequence number)
      Sequence number (raw): 1715159865
      [Next sequence number: 30    (relative sequence number)]
      Acknowledgment number: 28    (relative ack number)
      Acknowledgment number (raw): 4488942
      1000 .... = Header Length: 32 bytes (8)
   ▾ Flags: 0x018 (PSH, ACK)
         000. .... .... = Reserved: Not set
         ...0 .... .... = Nonce: Not set
         .... 0... .... = Congestion Window Reduced (CWR): Not set
         .... .0.. .... = ECN-Echo: Not set
         .... ..0. .... = Urgent: Not set
         .... ...1 .... = Acknowledgment: Set
         .... .... 1... = Push: Set
         .... .... .0.. = Reset: Not set
         .... .... ..0. = Syn: Not set
         .... .... ...0 = Fin: Not set
         [TCP Flags: ·······AP···]
      Window size value: 509
      [Calculated window size: 65152]
      [Window size scaling factor: 128]
      Checksum: 0x142f [unverified]
      [Checksum Status: Unverified]
      Urgent pointer: 0
   ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
   ▾ [SEQ/ACK analysis]
         [iRTT: 0.000863260 seconds]
         [Bytes in flight: 6]
         [Bytes sent since last PSH flag: 6]
   ▸ [Timestamps]
      TCP payload (6 bytes)
▾ Data (6 bytes)
      Data: 416c7469640a

0000  00 50 56 3a e9 b2 00 50  56 3a 65 38 08 00 45 00   ·PV:···P V:e8··E·
0010  00 3a 7c 7e 40 00 40 06  aa 3d 0a 00 00 01 0a 00   ·:|~@·@· ·=······
0020  00 02 23 28 be 72 66 3b  43 39 00 44 7e ee 80 18   ··#(·rf; C9·D~···
0030  01 fd 14 2f 00 00 01 01  08 0a c5 a7 fb cd 4d e8   ···/···· ······M·
0040  38 de 41 6c 74 69 64 0a                            8·Altid·
```

*Figur 18 - Server "Altid"*

Bekræftelse af beskeden sendes fra Client til Server. [ACK]

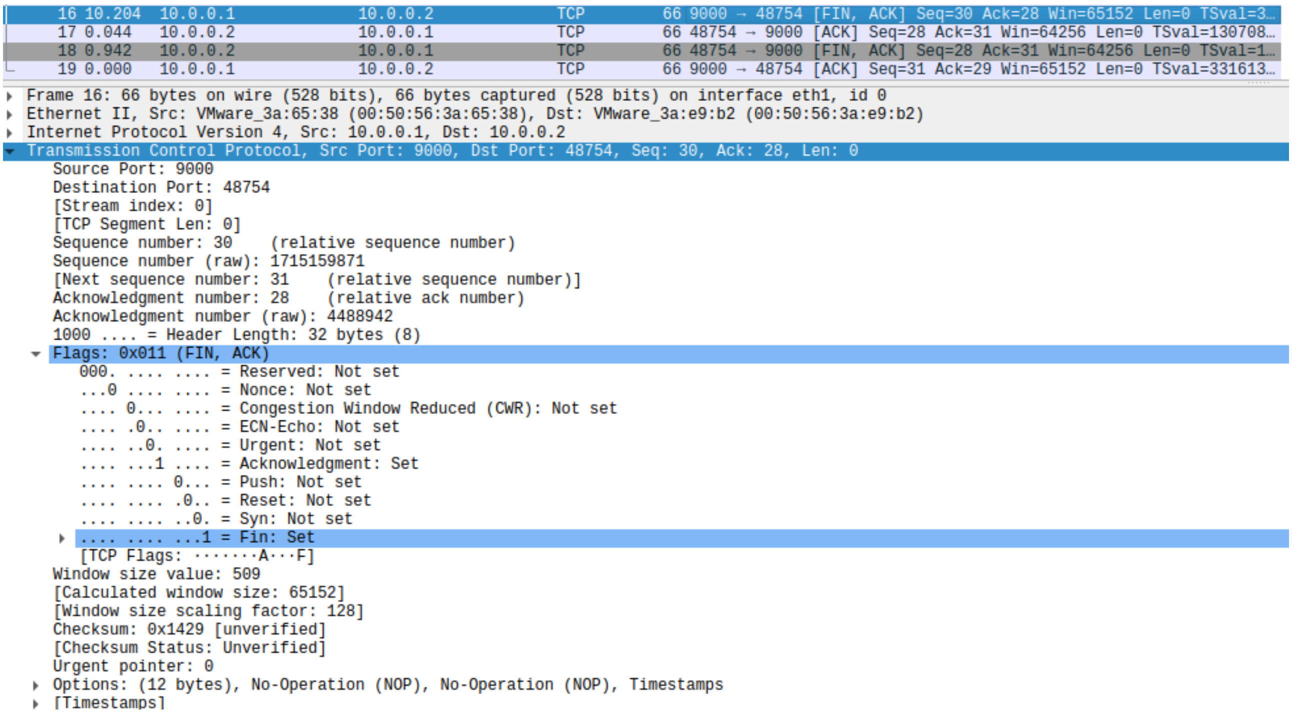


```
▾ Transmission Control Protocol, Src Port: 48754, Dst Port: 9000, Seq: 28, Ack: 30, Len: 0
      Source Port: 48754
      Destination Port: 9000
      [Stream index: 0]
      [TCP Segment Len: 0]
      Sequence number: 28    (relative sequence number)
      Sequence number (raw): 4488942
      [Next sequence number: 28    (relative sequence number)]
      Acknowledgment number: 30    (relative ack number)
      Acknowledgment number (raw): 1715159871
      1000 .... = Header Length: 32 bytes (8)
    ▾ Flags: 0x010 (ACK)
         000. .... .... = Reserved: Not set
         ...0 .... .... = Nonce: Not set
         .... 0... .... = Congestion Window Reduced (CWR): Not set
         .... .0.. .... = ECN-Echo: Not set
         .... ..0. .... = Urgent: Not set
         .... ...1 .... = Acknowledgment: Set
         .... .... 0... = Push: Not set
         .... .... .0.. = Reset: Not set
         .... .... ..0. = Syn: Not set
         .... .... ...0 = Fin: Not set
         [TCP Flags: ·······A····]
      Window size value: 502
      [Calculated window size: 64256]
      [Window size scaling factor: 128]
      Checksum: 0xecfa [unverified]
      [Checksum Status: Unverified]
      Urgent pointer: 0
    ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    ▾ [SEQ/ACK analysis]
         [This is an ACK to the segment in frame: 14]
         [The RTT to ACK the segment was: 0.000541686 seconds]
         [iRTT: 0.000863260 seconds]
    ▸ [Timestamps]
```

*Figur 19 - "Altid" [ACK]*

**-Nedlukning af connection: Flags, Sequence Counter, Acknowledge Counter og Window Size undersøges.**

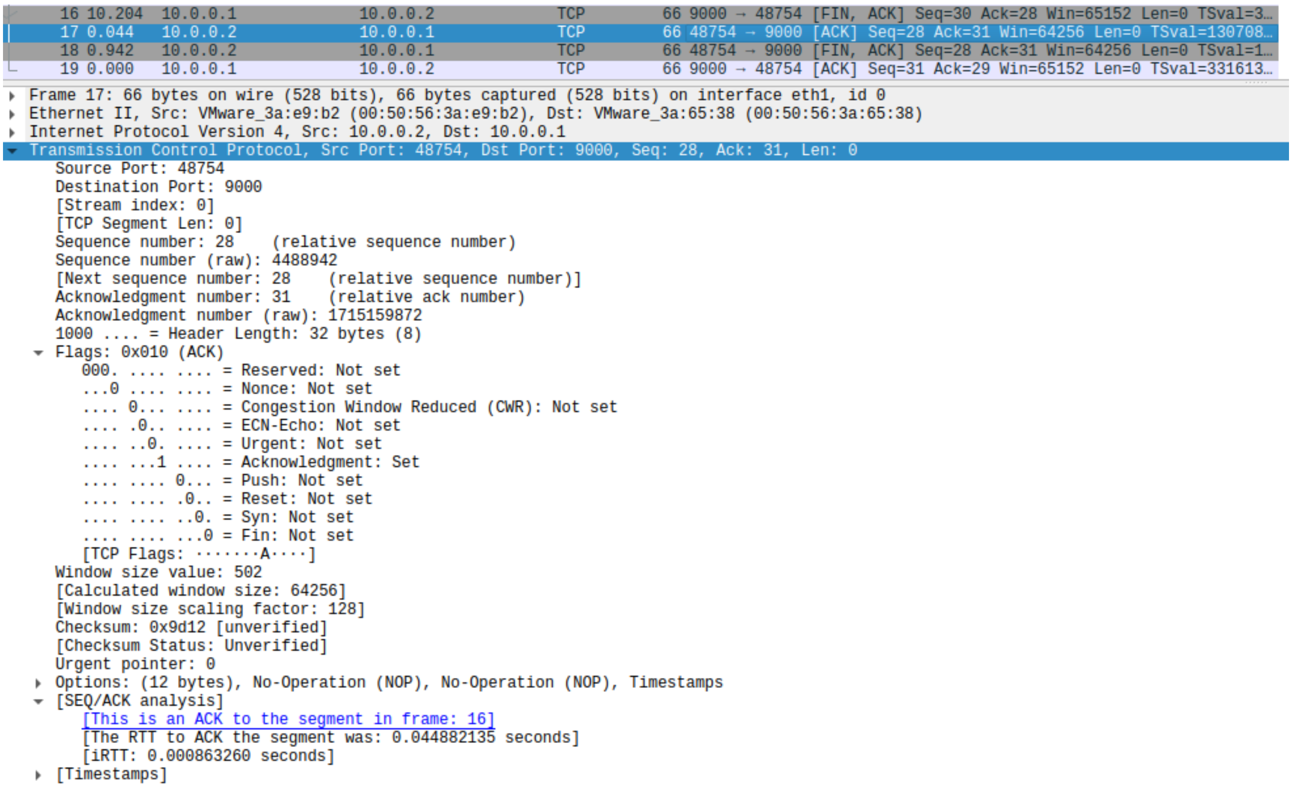
Nedlukning af server.[FIN, ACK]



```
  16 10.204  10.0.0.1             10.0.0.2             TCP           66 9000 → 48754 [FIN, ACK] Seq=30 Ack=28 Win=65152 Len=0 TSval=3…
  17 0.044   10.0.0.2             10.0.0.1             TCP           66 48754 → 9000 [ACK] Seq=28 Ack=31 Win=64256 Len=0 TSval=130708…
  18 0.942   10.0.0.2             10.0.0.1             TCP           66 48754 → 9000 [FIN, ACK] Seq=28 Ack=31 Win=64256 Len=0 TSval=1…
  19 0.000   10.0.0.1             10.0.0.2             TCP           66 9000 → 48754 [ACK] Seq=31 Ack=29 Win=65152 Len=0 TSval=331613…

▸ Frame 16: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth1, id 0
▸ Ethernet II, Src: VMware_3a:65:38 (00:50:56:3a:65:38), Dst: VMware_3a:e9:b2 (00:50:56:3a:e9:b2)
▸ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2
▾ Transmission Control Protocol, Src Port: 9000, Dst Port: 48754, Seq: 30, Ack: 28, Len: 0
     Source Port: 9000
     Destination Port: 48754
     [Stream index: 0]
     [TCP Segment Len: 0]
     Sequence number: 30    (relative sequence number)
     Sequence number (raw): 1715159871
     [Next sequence number: 31    (relative sequence number)]
     Acknowledgment number: 28    (relative ack number)
     Acknowledgment number (raw): 4488942
     1000 .... = Header Length: 32 bytes (8)
   ▾ Flags: 0x011 (FIN, ACK)
        000. .... .... = Reserved: Not set
        ...0 .... .... = Nonce: Not set
        .... 0... .... = Congestion Window Reduced (CWR): Not set
        .... .0.. .... = ECN-Echo: Not set
        .... ..0. .... = Urgent: Not set
        .... ...1 .... = Acknowledgment: Set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
        .... .... ..0. = Syn: Not set
      ▸ .... .... ...1 = Fin: Set
        [TCP Flags: ·······A···F]
     Window size value: 509
     [Calculated window size: 65152]
     [Window size scaling factor: 128]
     Checksum: 0x1429 [unverified]
     [Checksum Status: Unverified]
     Urgent pointer: 0
   ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
   ▸ [Timestamps]
```

*Figur 20 - Server nedlukning*

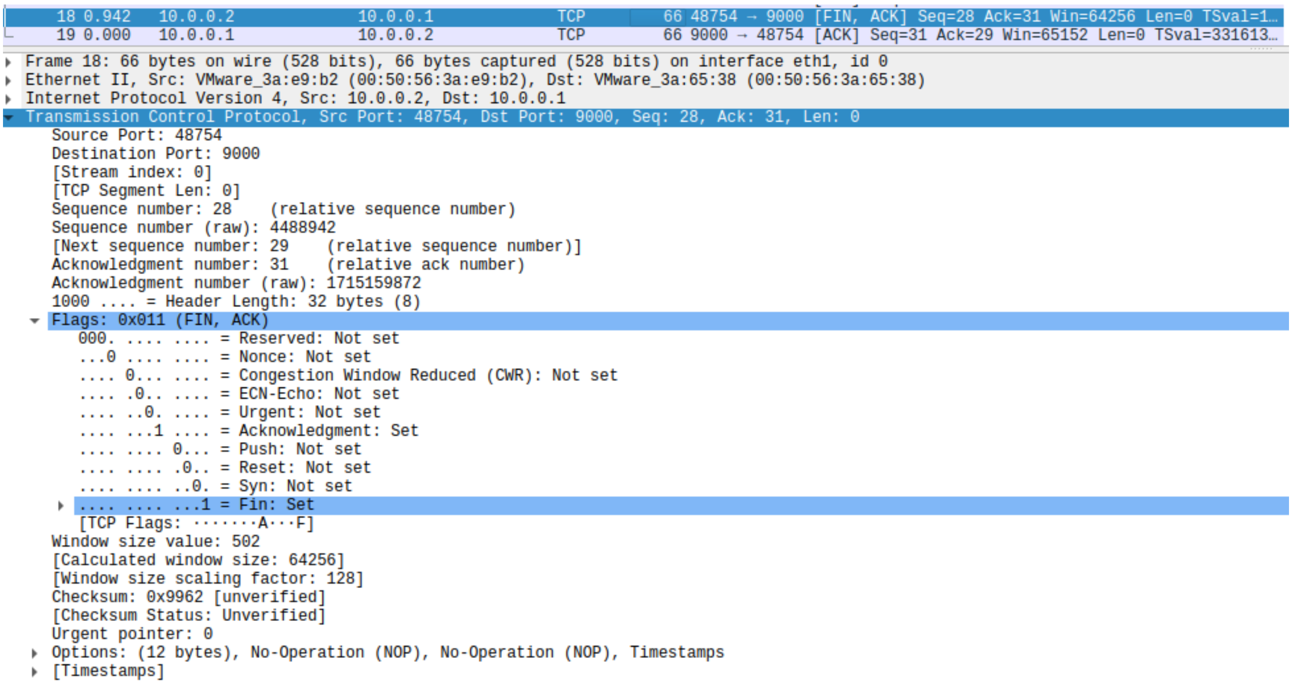
Bekræftelse fra Client



```
  16 10.204  10.0.0.1             10.0.0.2             TCP           66 9000 → 48754 [FIN, ACK] Seq=30 Ack=28 Win=65152 Len=0 TSval=3…
  17 0.044   10.0.0.2             10.0.0.1             TCP           66 48754 → 9000 [ACK] Seq=28 Ack=31 Win=64256 Len=0 TSval=130708…
  18 0.942   10.0.0.2             10.0.0.1             TCP           66 48754 → 9000 [FIN, ACK] Seq=28 Ack=31 Win=64256 Len=0 TSval=1…
  19 0.000   10.0.0.1             10.0.0.2             TCP           66 9000 → 48754 [ACK] Seq=31 Ack=29 Win=65152 Len=0 TSval=331613…

▸ Frame 17: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth1, id 0
▸ Ethernet II, Src: VMware_3a:e9:b2 (00:50:56:3a:e9:b2), Dst: VMware_3a:65:38 (00:50:56:3a:65:38)
▸ Internet Protocol Version 4, Src: 10.0.0.2, Dst: 10.0.0.1
▾ Transmission Control Protocol, Src Port: 48754, Dst Port: 9000, Seq: 28, Ack: 31, Len: 0
     Source Port: 48754
     Destination Port: 9000
     [Stream index: 0]
     [TCP Segment Len: 0]
     Sequence number: 28    (relative sequence number)
     Sequence number (raw): 4488942
     [Next sequence number: 28    (relative sequence number)]
     Acknowledgment number: 31    (relative ack number)
     Acknowledgment number (raw): 1715159872
     1000 .... = Header Length: 32 bytes (8)
   ▾ Flags: 0x010 (ACK)
        000. .... .... = Reserved: Not set
        ...0 .... .... = Nonce: Not set
        .... 0... .... = Congestion Window Reduced (CWR): Not set
        .... .0.. .... = ECN-Echo: Not set
        .... ..0. .... = Urgent: Not set
        .... ...1 .... = Acknowledgment: Set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
        .... .... ..0. = Syn: Not set
        .... .... ...0 = Fin: Not set
        [TCP Flags: ·······A····]
     Window size value: 502
     [Calculated window size: 64256]
     [Window size scaling factor: 128]
     Checksum: 0x9d12 [unverified]
     [Checksum Status: Unverified]
     Urgent pointer: 0
   ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
   ▾ [SEQ/ACK analysis]
        [This is an ACK to the segment in frame: 16]
        [The RTT to ACK the segment was: 0.044882135 seconds]
        [iRTT: 0.000863260 seconds]
   ▸ [Timestamps]
```

*Figur 21 - Server nedlukning [ACK]*

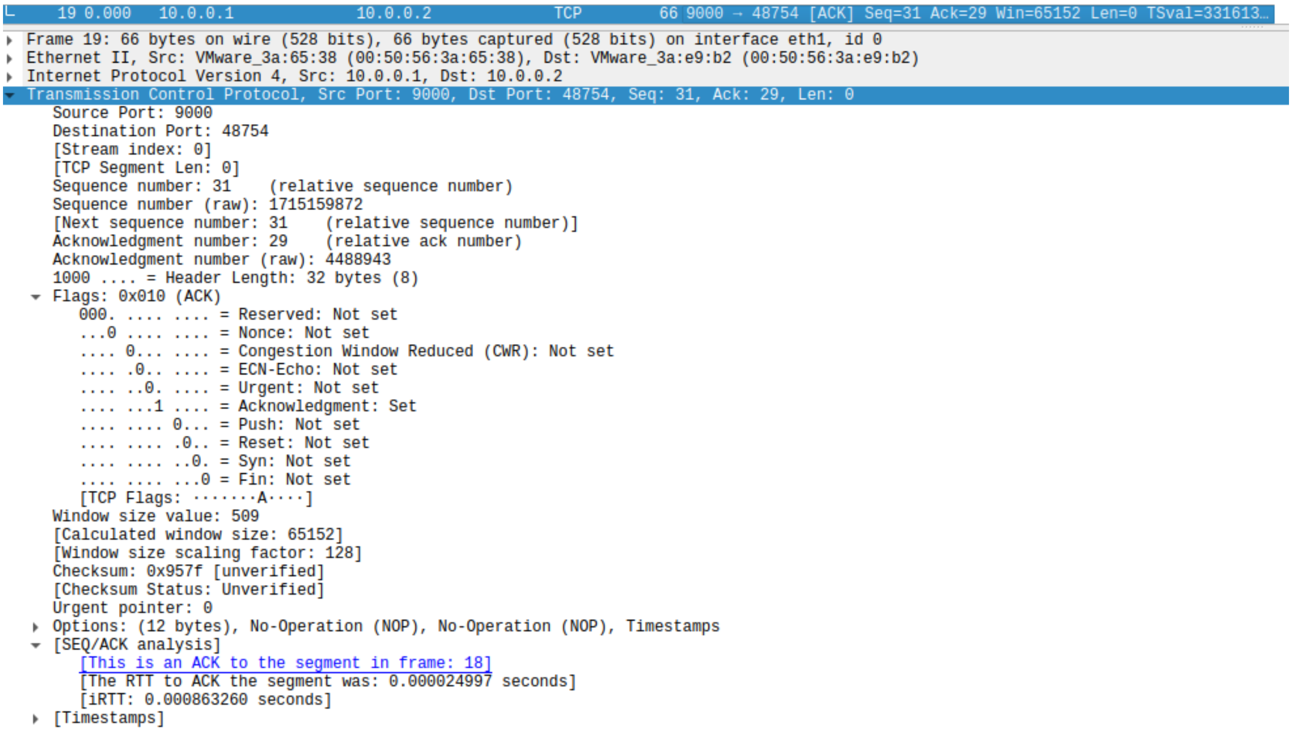
## Nedlukning af Client. [FIN, ACK]



```
   18 0.942   10.0.0.2             10.0.0.1             TCP         66 48754 → 9000 [FIN, ACK] Seq=28 Ack=31 Win=64256 Len=0 TSval=1…
   19 0.000   10.0.0.1             10.0.0.2             TCP         66 9000 → 48754 [ACK] Seq=31 Ack=29 Win=65152 Len=0 TSval=331613…
▸ Frame 18: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth1, id 0
▸ Ethernet II, Src: VMware_3a:e9:b2 (00:50:56:3a:e9:b2), Dst: VMware_3a:65:38 (00:50:56:3a:65:38)
▸ Internet Protocol Version 4, Src: 10.0.0.2, Dst: 10.0.0.1
▾ Transmission Control Protocol, Src Port: 48754, Dst Port: 9000, Seq: 28, Ack: 31, Len: 0
    Source Port: 48754
    Destination Port: 9000
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 28    (relative sequence number)
    Sequence number (raw): 4488942
    [Next sequence number: 29    (relative sequence number)]
    Acknowledgment number: 31    (relative ack number)
    Acknowledgment number (raw): 1715159872
    1000 .... = Header Length: 32 bytes (8)
  ▾ Flags: 0x011 (FIN, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
    ▸ .... .... ...1 = Fin: Set
      [TCP Flags: ·······A···F]
    Window size value: 502
    [Calculated window size: 64256]
    [Window size scaling factor: 128]
    Checksum: 0x9962 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▸ [Timestamps]
```

*Figur 22 - Nedlukning af client*

## Bekræftelse fra Server [ACK]



```
   19 0.000   10.0.0.1             10.0.0.2             TCP         66 9000 → 48754 [ACK] Seq=31 Ack=29 Win=65152 Len=0 TSval=331613…
▸ Frame 19: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth1, id 0
▸ Ethernet II, Src: VMware_3a:65:38 (00:50:56:3a:65:38), Dst: VMware_3a:e9:b2 (00:50:56:3a:e9:b2)
▸ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2
▾ Transmission Control Protocol, Src Port: 9000, Dst Port: 48754, Seq: 31, Ack: 29, Len: 0
    Source Port: 9000
    Destination Port: 48754
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 31    (relative sequence number)
    Sequence number (raw): 1715159872
    [Next sequence number: 31    (relative sequence number)]
    Acknowledgment number: 29    (relative ack number)
    Acknowledgment number (raw): 4488943
    1000 .... = Header Length: 32 bytes (8)
  ▾ Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·······A····]
    Window size value: 509
    [Calculated window size: 65152]
    [Window size scaling factor: 128]
    Checksum: 0x957f [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▾ [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 18]
      [The RTT to ACK the segment was: 0.000024997 seconds]
      [iRTT: 0.000863260 seconds]
  ▸ [Timestamps]
```

*Figur 23 - Nedlukning af client [ACK]*

I dette eksempel blev serveren lukket ned i H1, hvorefter ovenstående flow gennemføres. Ligeledes kunne det være Client siden der blev lukket ned først og rækkefølgen vil så være modsat. Et normalt brugsscenarie kunne være  at clienten tager initiativ til at nedlukke connection, hvorefter serveren ville connect med en ny client.

## Opgave 5 - Oversigt over Counters

Oversigt over H1 Sequence Counter og Acknowledge Counter, og H2 Sequence Counter og Acknowledge Counter. Oversigten er opdelt I Client og server og nummereringen foran er benyttet til at fortælle rækkefølgen. I Figur 24 og Figur 25, benyttes wireshark til at illustrere sequence counter for client og server.

Client

**Handshake**

1) Len 0, Seq 0, Ack 0
3) Len 0, Seq 1, Ack 1

**Beskeder**

4) Len 15, Seq 1 , Ack 1
7) Len 0, Seq 16, Ack 24
8) Len 12, Seq 16, Ack 24
11)Len 0, Seq 28, Ack 30

**Nedlukning**

13) Len 0, Seq 28, Ack 31
14) Len 0, Seq 28, Ack 31

Server

**Handshake**

2) Len 0, Seq 0, Ack 1

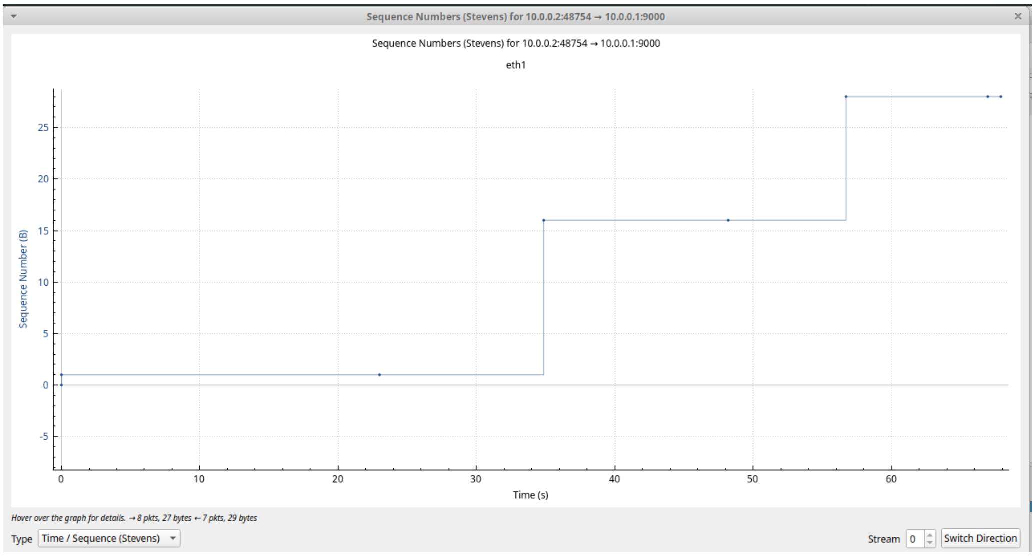
**Beskeder**

5) Len 0, Seq 1, Ack 16
6) Len 23, Seq 1, Ack 16
9 )Len 0, Seq 24, Ack 28
10) Len 6, Seq 24, Ack 28

**Nedlukning**

12) Len 0, Seq 30, Ack 28
15) Len 0, Seq 31, Ack 28

**Client to server**



Sequence Numbers (Stevens) for 10.0.0.2:48754 → 10.0.0.1:9000

Sequence Numbers (Stevens) for 10.0.0.2:48754 → 10.0.0.1:9000

eth1

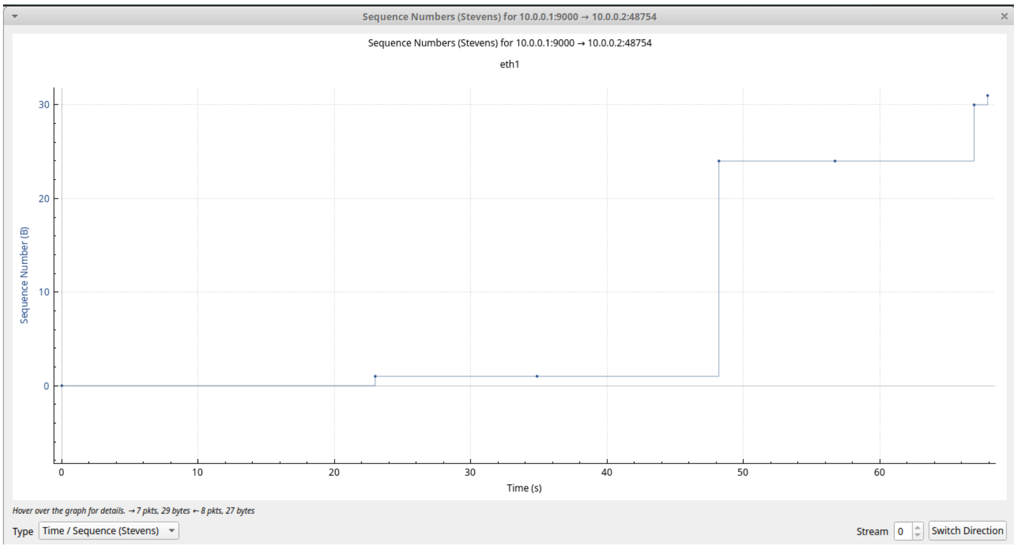
*Hover over the graph for details. → 8 pkts, 27 bytes ← 7 pkts, 29 bytes*

Type Time / Sequence (Stevens)

Stream 0

*Figur 24 - Client Sequence numbers*

**Server to Client**



Sequence Numbers (Stevens) for 10.0.0.1:9000 → 10.0.0.2:48754

Sequence Numbers (Stevens) for 10.0.0.1:9000 → 10.0.0.2:48754

eth1

*Hover over the graph for details. → 7 pkts, 29 bytes ← 8 pkts, 27 bytes*

Type Time / Sequence (Stevens)

Stream 0

*Figur 25 - Server Sequence numbers*

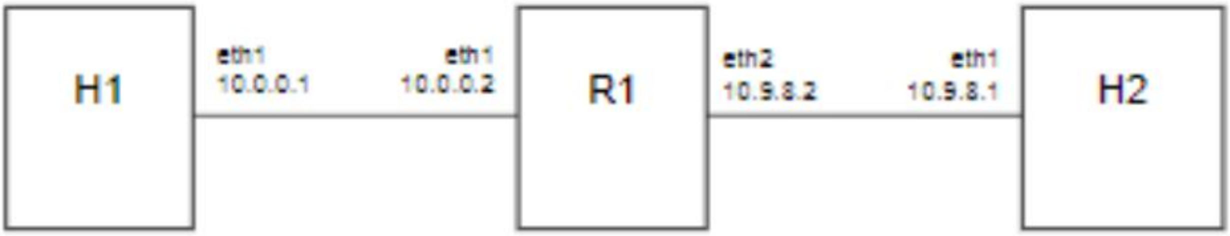
## Konklusion på øvelsen

Med udgangspunkt I transportlaget er der i denne øvelse undersøgt både TCP og UDP protokollerne. Wireshark er blevet benyttet til at undersøge hændelser i form af handshake(kun for TCP) flags og counters når der sendes en tekststreng mellem client og server.

# Øvelse 9 - Opsætning af en simpel router (forwarder)

## Indledning

Formålet med denne øvelse er at opsætte en router(forwarder)R1, som gør det muligt for den virtuelle maskine H1 på en LAN-segment at kommunikere med den virtuelle maskine H2 på et andet LAN-segment.

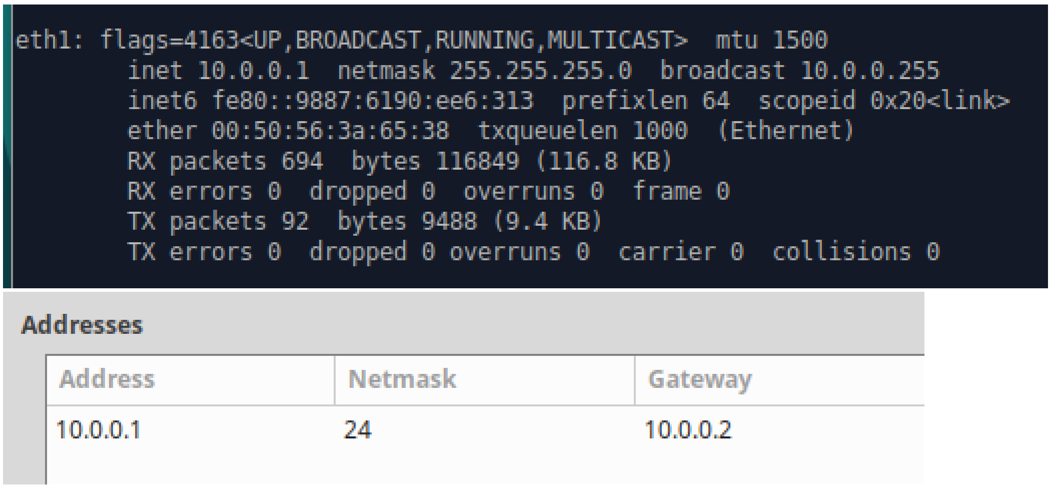


*Figur 26 - Opsætning af router*

## Opgave 2 - Konfigurering

De tre virtuelle maskiner konfigureres efter Figur 26

H1 - konfiguration



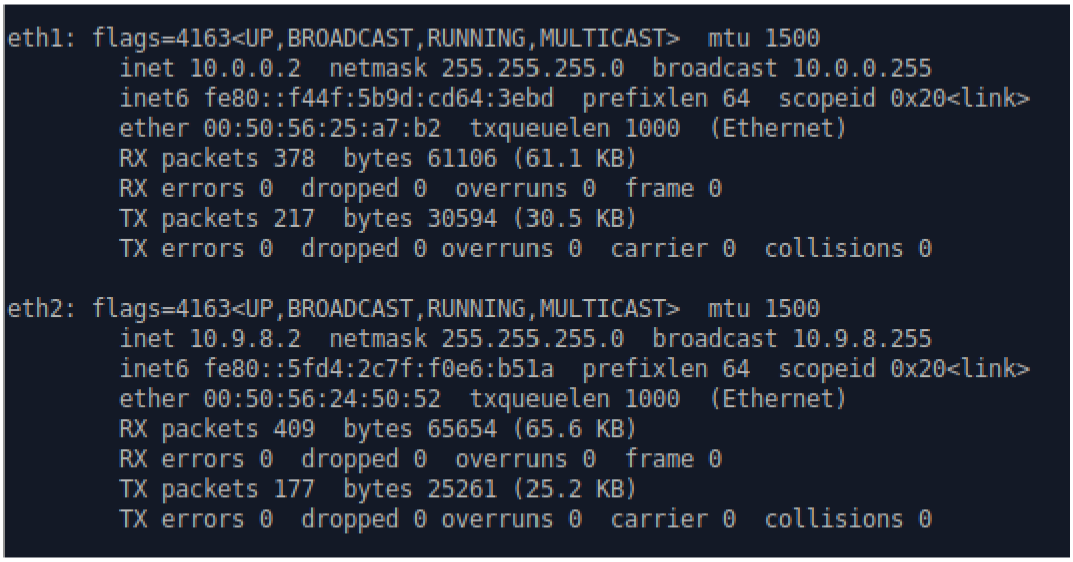
```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.1  netmask 255.255.255.0  broadcast 10.0.0.255
        inet6 fe80::9887:6190:ee6:313  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:3a:65:38  txqueuelen 1000  (Ethernet)
        RX packets 694  bytes 116849 (116.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 92  bytes 9488 (9.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

**Addresses**

| Address | Netmask | Gateway |
|---|---|---|
| 10.0.0.1 | 24 | 10.0.0.2 |

*Figur 27 - H1 Address and Gateway*

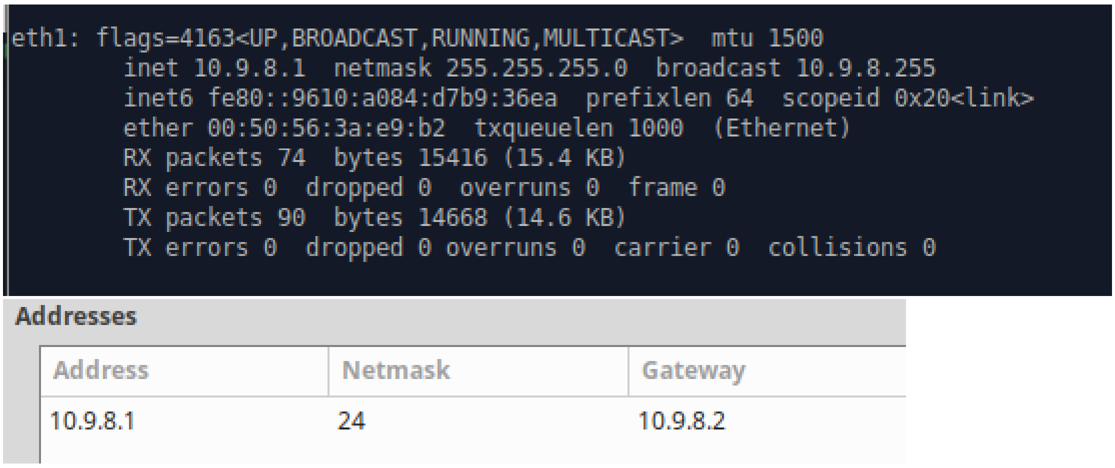
R1 - konfiguration



```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.2  netmask 255.255.255.0  broadcast 10.0.0.255
        inet6 fe80::f44f:5b9d:cd64:3ebd  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:25:a7:b2  txqueuelen 1000  (Ethernet)
        RX packets 378  bytes 61106 (61.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 217  bytes 30594 (30.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.8.2  netmask 255.255.255.0  broadcast 10.9.8.255
        inet6 fe80::5fd4:2c7f:f0e6:b51a  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:24:50:52  txqueuelen 1000  (Ethernet)
        RX packets 409  bytes 65654 (65.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 177  bytes 25261 (25.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

*Figur 28 - R1 opsætning*

H2 - konfiguration



```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.8.1  netmask 255.255.255.0  broadcast 10.9.8.255
        inet6 fe80::9610:a084:d7b9:36ea  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:3a:e9:b2  txqueuelen 1000  (Ethernet)
        RX packets 74  bytes 15416 (15.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 90  bytes 14668 (14.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Addresses

| Address | Netmask | Gateway |
|---|---|---|
| 10.9.8.1 | 24 | 10.9.8.2 |

*Figur 29 - H2 Address and Gateway*

## Konklusion på delopgave

En del af konfigureringen er lavet i VMware hvor R1 oprettes og network adapters sættes til bridged connection. På de virtuelle maskiner benyttes R1 efterfølgende som gateway som vist i Figur 27 og Figur 29. Denne opsætning medfører at der er 2 LAN-segmenter:
For at H1 kan kommunikere med H2 benyttes R1 gateway 10.0.0.2
H1 (10.0.0.1) <------> R1 (10.0.02)

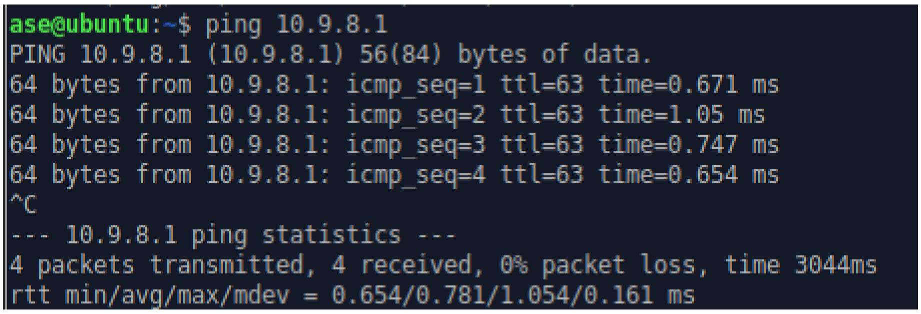
For at H2 kan kommunikere med H1 benyttes R1 gateway 10.9.8.2
R1 (10.9.8.2) <------> H2 (10.9.8.1)

I VMware sikres desuden af alle virtuelle maskiner har unikke MAC adresser.

## Opgave 3 - Test af konfigurering

Verificer at de 2 hosts H1 og H2, som er tilsluttet hvert sit LAN-segment, kan kommunikere indbyrdes vha. Linux-kommandoen ping

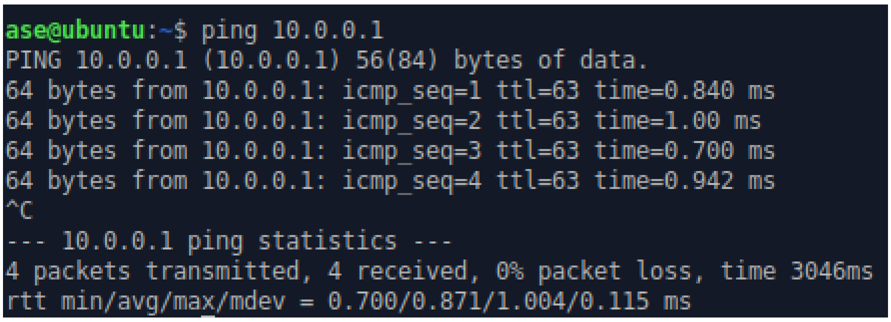
Ping fra H1 til H2 med R1 som Router



```
ase@ubuntu:~$ ping 10.9.8.1
PING 10.9.8.1 (10.9.8.1) 56(84) bytes of data.
64 bytes from 10.9.8.1: icmp_seq=1 ttl=63 time=0.671 ms
64 bytes from 10.9.8.1: icmp_seq=2 ttl=63 time=1.05 ms
64 bytes from 10.9.8.1: icmp_seq=3 ttl=63 time=0.747 ms
64 bytes from 10.9.8.1: icmp_seq=4 ttl=63 time=0.654 ms
^C
--- 10.9.8.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3044ms
rtt min/avg/max/mdev = 0.654/0.781/1.054/0.161 ms
```

*Figur 30 - Ping H1 to H2 via R1*

Ping fra H2 til H1 med R1 som Router



```
ase@ubuntu:~$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=63 time=0.840 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=63 time=1.00 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=63 time=0.700 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=63 time=0.942 ms
^C
--- 10.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3046ms
rtt min/avg/max/mdev = 0.700/0.871/1.004/0.115 ms
```

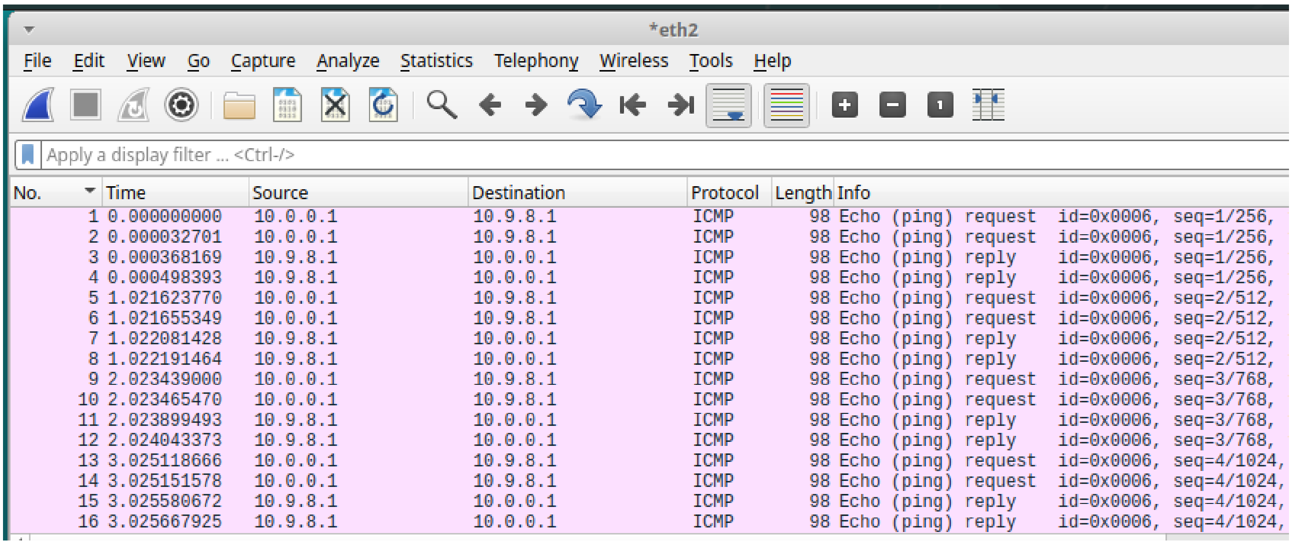
*Figur 31 - Ping H2 to H1 via R1*

## Konklusion på delopgave

Det er muligt at ping fra H1 til H2 og ligeledes modsat, hvilket indikere at R1(Router forward) fungere som ønsket og opsætning er korrekt.

## Opgave 4 - Wireshark

Analyser relevante hændelser på de to LAN-segmenter vha. Wireshark.



```
No.   Time         Source     Destination  Protocol Length Info
    1 0.000000000  10.0.0.1   10.9.8.1     ICMP       98 Echo (ping) request  id=0x0006, seq=1/256,
    2 0.000032701  10.0.0.1   10.9.8.1     ICMP       98 Echo (ping) request  id=0x0006, seq=1/256,
    3 0.000368169  10.9.8.1   10.0.0.1     ICMP       98 Echo (ping) reply    id=0x0006, seq=1/256,
    4 0.000498393  10.9.8.1   10.0.0.1     ICMP       98 Echo (ping) reply    id=0x0006, seq=1/256,
    5 1.021623770  10.0.0.1   10.9.8.1     ICMP       98 Echo (ping) request  id=0x0006, seq=2/512,
    6 1.021655349  10.0.0.1   10.9.8.1     ICMP       98 Echo (ping) request  id=0x0006, seq=2/512,
    7 1.022081428  10.9.8.1   10.0.0.1     ICMP       98 Echo (ping) reply    id=0x0006, seq=2/512,
    8 1.022191464  10.9.8.1   10.0.0.1     ICMP       98 Echo (ping) reply    id=0x0006, seq=2/512,
    9 2.023439000  10.0.0.1   10.9.8.1     ICMP       98 Echo (ping) request  id=0x0006, seq=3/768,
   10 2.023465470  10.0.0.1   10.9.8.1     ICMP       98 Echo (ping) request  id=0x0006, seq=3/768,
   11 2.023899493  10.9.8.1   10.0.0.1     ICMP       98 Echo (ping) reply    id=0x0006, seq=3/768,
   12 2.024043373  10.9.8.1   10.0.0.1     ICMP       98 Echo (ping) reply    id=0x0006, seq=3/768,
   13 3.025118666  10.0.0.1   10.9.8.1     ICMP       98 Echo (ping) request  id=0x0006, seq=4/1024,
   14 3.025151578  10.0.0.1   10.9.8.1     ICMP       98 Echo (ping) request  id=0x0006, seq=4/1024,
   15 3.025580672  10.9.8.1   10.0.0.1     ICMP       98 Echo (ping) reply    id=0x0006, seq=4/1024,
   16 3.025667925  10.9.8.1   10.0.0.1     ICMP       98 Echo (ping) reply    id=0x0006, seq=4/1024,
```

*Figur 32 - R1 wireshark capture*

For hver ping noteres at der kommer 2 request og 2 reply. Dette er grundet de 2 LAN-segmenter i vores opsætning, da H1 og H2 bliver nødt til at kommunikere gennem R1(gateway).

Ping fra H1 til H2 har derfor følgende flow.
H1 -> R1 og R1 -> H2 // Request
H2 -> R1 og R1 -> H1 // Reply

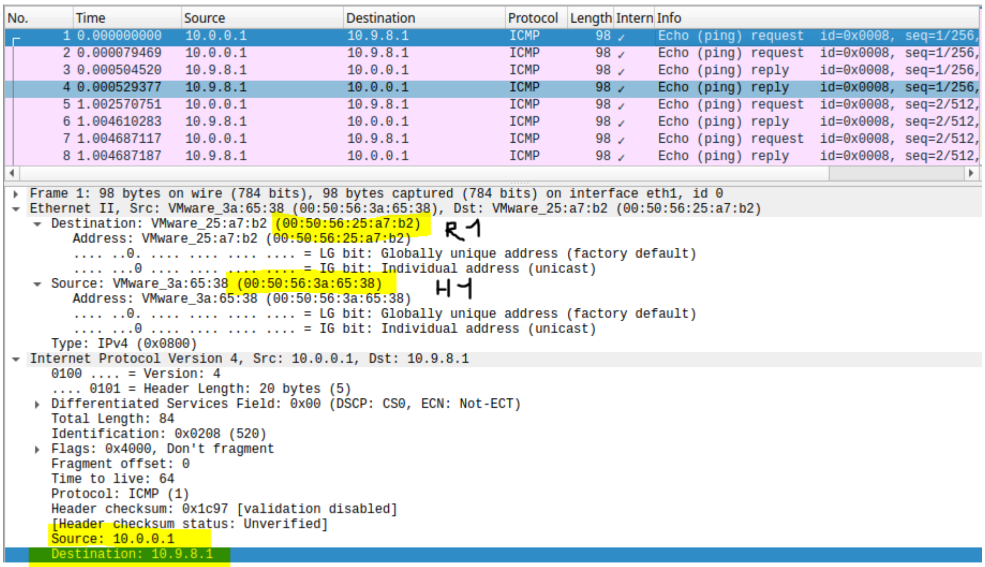
I næste opgave vil hændelser og indholdet i pakkerne undersøges yderligere.

## Opgave 5 - Undersøgelse af IP-adresserne

Er IP-adresserne konstante på netværkslaget, eller udskiftes de undervejs, når en IP-pakke sendes fra H1 til H2 (IP-adresserne undersøges i forhold til IP-pakkerne)?

Wireshark benyttes til at analysere kommunikation når der pinges fra H1 til H2. Wireshark køres i R1 således at kommunikation til både H1 og H2 kan undersøges.

## Ping request 1

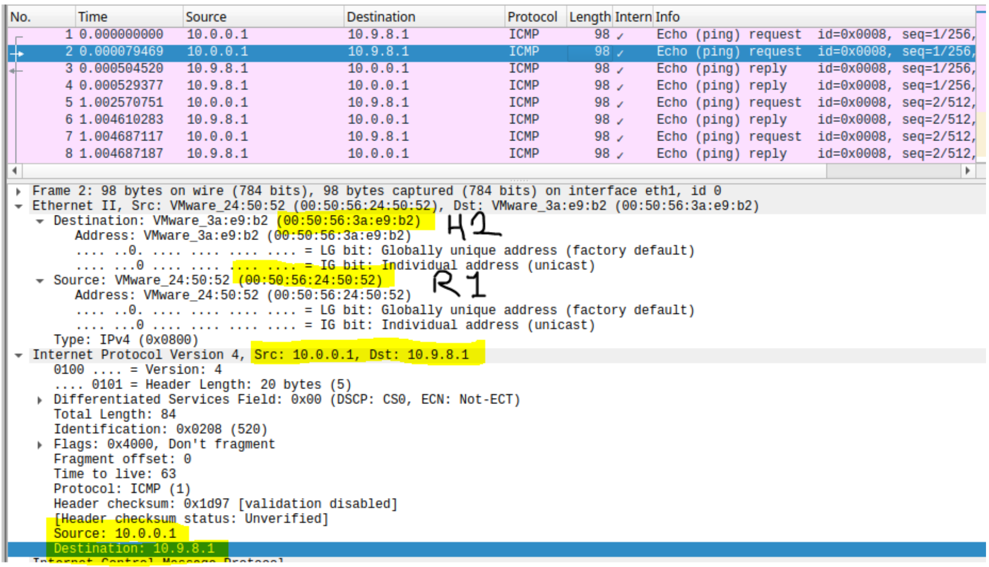


```
No.  Time         Source     Destination  Protocol  Length Intern Info
   1 0.000000000  10.0.0.1   10.9.8.1     ICMP        98 ✓  Echo (ping) request  id=0x0008, seq=1/256,
   2 0.000079469  10.0.0.1   10.9.8.1     ICMP        98 ✓  Echo (ping) request  id=0x0008, seq=1/256,
   3 0.000504520  10.9.8.1   10.0.0.1     ICMP        98 ✓  Echo (ping) reply    id=0x0008, seq=1/256,
   4 0.000529377  10.9.8.1   10.0.0.1     ICMP        98 ✓  Echo (ping) reply    id=0x0008, seq=1/256,
   5 1.002570751  10.0.0.1   10.9.8.1     ICMP        98 ✓  Echo (ping) request  id=0x0008, seq=2/512,
   6 1.004610283  10.9.8.1   10.0.0.1     ICMP        98 ✓  Echo (ping) reply    id=0x0008, seq=2/512,
   7 1.004687117  10.0.0.1   10.9.8.1     ICMP        98 ✓  Echo (ping) request  id=0x0008, seq=2/512,
   8 1.004687187  10.9.8.1   10.0.0.1     ICMP        98 ✓  Echo (ping) reply    id=0x0008, seq=2/512,
```

```
▸ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth1, id 0
▾ Ethernet II, Src: VMware_3a:65:38 (00:50:56:3a:65:38), Dst: VMware_25:a7:b2 (00:50:56:25:a7:b2)
  ▾ Destination: VMware_25:a7:b2 (00:50:56:25:a7:b2)
      Address: VMware_25:a7:b2 (00:50:56:25:a7:b2)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: VMware_3a:65:38 (00:50:56:3a:65:38)
      Address: VMware_3a:65:38 (00:50:56:3a:65:38)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▾ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.9.8.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x0208 (520)
  ▸ Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0x1c97 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.0.0.1
    Destination: 10.9.8.1
```

*Figur 33 - H1 to R1*

## Ping request 2

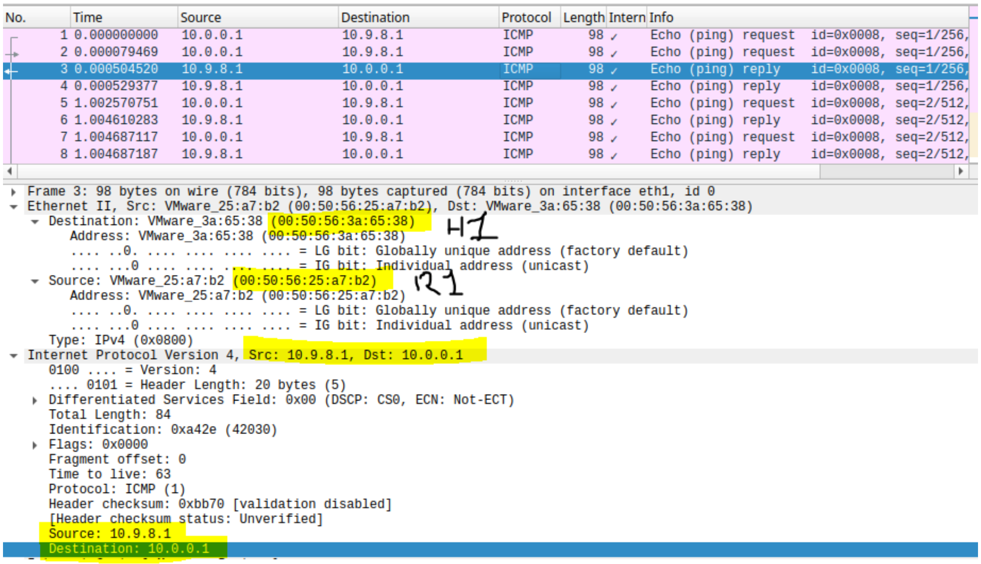


```
No.  Time         Source     Destination  Protocol  Length Intern Info
   1 0.000000000  10.0.0.1   10.9.8.1     ICMP        98 ✓  Echo (ping) request  id=0x0008, seq=1/256,
   2 0.000079469  10.0.0.1   10.9.8.1     ICMP        98 ✓  Echo (ping) request  id=0x0008, seq=1/256,
   3 0.000504520  10.9.8.1   10.0.0.1     ICMP        98 ✓  Echo (ping) reply    id=0x0008, seq=1/256,
   4 0.000529377  10.9.8.1   10.0.0.1     ICMP        98 ✓  Echo (ping) reply    id=0x0008, seq=1/256,
   5 1.002570751  10.0.0.1   10.9.8.1     ICMP        98 ✓  Echo (ping) request  id=0x0008, seq=2/512,
   6 1.004610283  10.9.8.1   10.0.0.1     ICMP        98 ✓  Echo (ping) reply    id=0x0008, seq=2/512,
   7 1.004687117  10.0.0.1   10.9.8.1     ICMP        98 ✓  Echo (ping) request  id=0x0008, seq=2/512,
   8 1.004687187  10.9.8.1   10.0.0.1     ICMP        98 ✓  Echo (ping) reply    id=0x0008, seq=2/512,
```

```
▸ Frame 2: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth1, id 0
▾ Ethernet II, Src: VMware_24:50:52 (00:50:56:24:50:52), Dst: VMware_3a:e9:b2 (00:50:56:3a:e9:b2)
  ▾ Destination: VMware_3a:e9:b2 (00:50:56:3a:e9:b2)
      Address: VMware_3a:e9:b2 (00:50:56:3a:e9:b2)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: VMware_24:50:52 (00:50:56:24:50:52)
      Address: VMware_24:50:52 (00:50:56:24:50:52)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▾ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.9.8.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x0208 (520)
  ▸ Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 63
    Protocol: ICMP (1)
    Header checksum: 0x1d97 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.0.0.1
    Destination: 10.9.8.1
  Internet Control Message Protocol
```

*Figur 34 - R1 to H2*

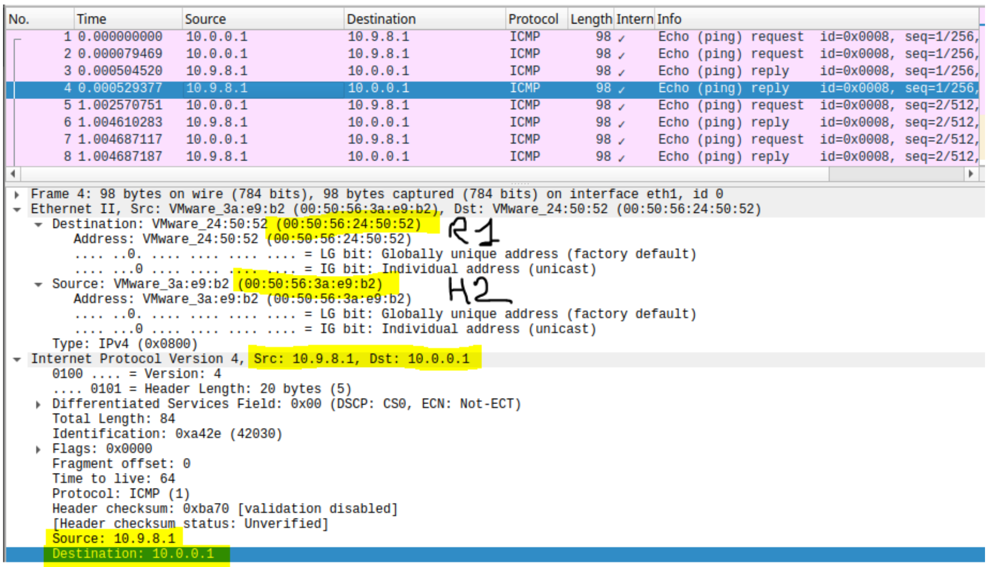
Reply 1



```
No.   Time         Source      Destination  Protocol Length Intern Info
    1 0.000000000  10.0.0.1    10.9.8.1     ICMP        98 ✓   Echo (ping) request  id=0x0008, seq=1/256,
    2 0.000079469  10.0.0.1    10.9.8.1     ICMP        98 ✓   Echo (ping) request  id=0x0008, seq=1/256,
    3 0.000504520  10.9.8.1    10.0.0.1     ICMP        98 ✓   Echo (ping) reply    id=0x0008, seq=1/256,
    4 0.000529377  10.9.8.1    10.0.0.1     ICMP        98 ✓   Echo (ping) reply    id=0x0008, seq=1/256,
    5 1.002570751  10.0.0.1    10.9.8.1     ICMP        98 ✓   Echo (ping) request  id=0x0008, seq=2/512,
    6 1.004610283  10.9.8.1    10.0.0.1     ICMP        98 ✓   Echo (ping) reply    id=0x0008, seq=2/512,
    7 1.004687117  10.0.0.1    10.9.8.1     ICMP        98 ✓   Echo (ping) request  id=0x0008, seq=2/512,
    8 1.004687187  10.9.8.1    10.0.0.1     ICMP        98 ✓   Echo (ping) reply    id=0x0008, seq=2/512,

▸ Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth1, id 0
▾ Ethernet II, Src: VMware_25:a7:b2 (00:50:56:25:a7:b2), Dst: VMware_3a:65:38 (00:50:56:3a:65:38)
  ▾ Destination: VMware_3a:65:38 (00:50:56:3a:65:38)
      Address: VMware_3a:65:38 (00:50:56:3a:65:38)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: VMware_25:a7:b2 (00:50:56:25:a7:b2)
      Address: VMware_25:a7:b2 (00:50:56:25:a7:b2)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▾ Internet Protocol Version 4, Src: 10.9.8.1, Dst: 10.0.0.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0xa42e (42030)
  ▸ Flags: 0x0000
    Fragment offset: 0
    Time to live: 63
    Protocol: ICMP (1)
    Header checksum: 0xbb70 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.9.8.1
    Destination: 10.0.0.1
```

*Figur 35 - R1 to H1*

Reply 2



```
No.   Time         Source      Destination  Protocol Length Intern Info
    1 0.000000000  10.0.0.1    10.9.8.1     ICMP        98 ✓   Echo (ping) request  id=0x0008, seq=1/256,
    2 0.000079469  10.0.0.1    10.9.8.1     ICMP        98 ✓   Echo (ping) request  id=0x0008, seq=1/256,
    3 0.000504520  10.9.8.1    10.0.0.1     ICMP        98 ✓   Echo (ping) reply    id=0x0008, seq=1/256,
    4 0.000529377  10.9.8.1    10.0.0.1     ICMP        98 ✓   Echo (ping) reply    id=0x0008, seq=1/256,
    5 1.002570751  10.0.0.1    10.9.8.1     ICMP        98 ✓   Echo (ping) request  id=0x0008, seq=2/512,
    6 1.004610283  10.9.8.1    10.0.0.1     ICMP        98 ✓   Echo (ping) reply    id=0x0008, seq=2/512,
    7 1.004687117  10.0.0.1    10.9.8.1     ICMP        98 ✓   Echo (ping) request  id=0x0008, seq=2/512,
    8 1.004687187  10.9.8.1    10.0.0.1     ICMP        98 ✓   Echo (ping) reply    id=0x0008, seq=2/512,

▸ Frame 4: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth1, id 0
▾ Ethernet II, Src: VMware_3a:e9:b2 (00:50:56:3a:e9:b2), Dst: VMware_24:50:52 (00:50:56:24:50:52)
  ▾ Destination: VMware_24:50:52 (00:50:56:24:50:52)
      Address: VMware_24:50:52 (00:50:56:24:50:52)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: VMware_3a:e9:b2 (00:50:56:3a:e9:b2)
      Address: VMware_3a:e9:b2 (00:50:56:3a:e9:b2)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▾ Internet Protocol Version 4, Src: 10.9.8.1, Dst: 10.0.0.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0xa42e (42030)
  ▸ Flags: 0x0000
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0xba70 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.9.8.1
    Destination: 10.0.0.1
```

*Figur 36 - H2 to R1*

## Konklusion på øvelsen

Der observeres at IP-adresserne for source/destination er uændret, men at MAC-adresserne ændres undervejs.

Figurerne 33, 34, 35, 36 benyttes ligeledes til Øvelse 10 til undersøgelse af ændring af MAC adresserne.

# Øvelse 10 - netværksanalyse af ARP-protokollens funktionalitet

## Indledning

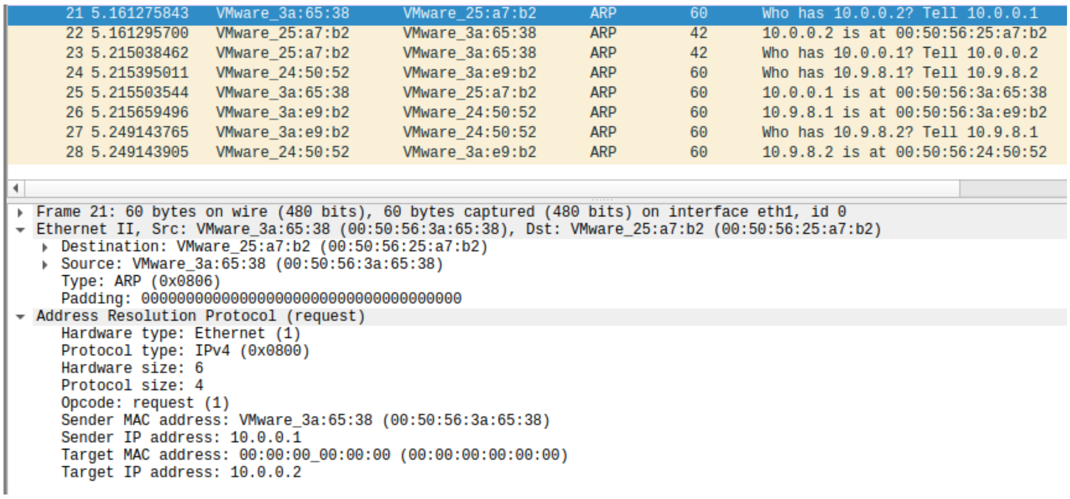
Formålet med denne øvelse er at fortsætte fra tidligere øvelse (Router forward) og nu undersøge ARP protokollens funktionalitet.

## Opgave 1 - Analyser relevante hændelser for linklaget på de to LAN-segmenter

H1 Request:” Who has IP address 10.0.0.2”

Target IP address: 10.0.0.2

Target MAC address: 00:00:00:00:00:00



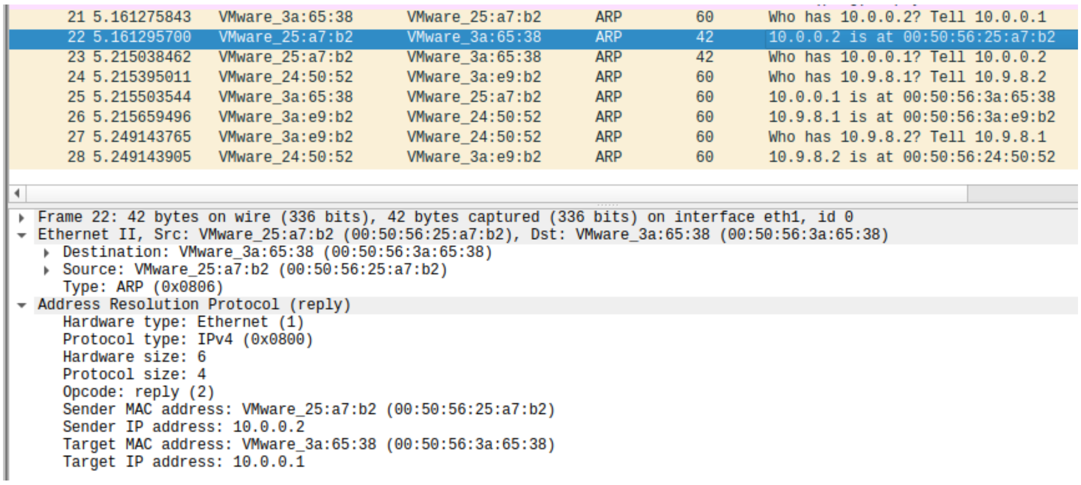
```
21 5.161275843  VMware_3a:65:38   VMware_25:a7:b2   ARP   60   Who has 10.0.0.2? Tell 10.0.0.1
22 5.161295700  VMware_25:a7:b2   VMware_3a:65:38   ARP   42   10.0.0.2 is at 00:50:56:25:a7:b2
23 5.215038462  VMware_25:a7:b2   VMware_3a:65:38   ARP   42   Who has 10.0.0.1? Tell 10.0.0.2
24 5.215395011  VMware_24:50:52   VMware_3a:e9:b2   ARP   60   Who has 10.9.8.1? Tell 10.9.8.2
25 5.215503544  VMware_3a:65:38   VMware_25:a7:b2   ARP   60   10.0.0.1 is at 00:50:56:3a:65:38
26 5.215659496  VMware_3a:e9:b2   VMware_24:50:52   ARP   60   10.9.8.1 is at 00:50:56:3a:e9:b2
27 5.249143765  VMware_3a:e9:b2   VMware_24:50:52   ARP   60   Who has 10.9.8.2? Tell 10.9.8.1
28 5.249143905  VMware_24:50:52   VMware_3a:e9:b2   ARP   60   10.9.8.2 is at 00:50:56:24:50:52

▸ Frame 21: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth1, id 0
▾ Ethernet II, Src: VMware_3a:65:38 (00:50:56:3a:65:38), Dst: VMware_25:a7:b2 (00:50:56:25:a7:b2)
  ▸ Destination: VMware_25:a7:b2 (00:50:56:25:a7:b2)
  ▸ Source: VMware_3a:65:38 (00:50:56:3a:65:38)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
▾ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: VMware_3a:65:38 (00:50:56:3a:65:38)
    Sender IP address: 10.0.0.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.0.0.2
```

*Figur 37 - ARP- Who has IP 10.0.0.2*

R1 Reply: “10.0.0.2 is at ….”

Sender IP address: 10.0.0.2

Sender MAC address: 00:50:56:25:a7:b2 (R1)



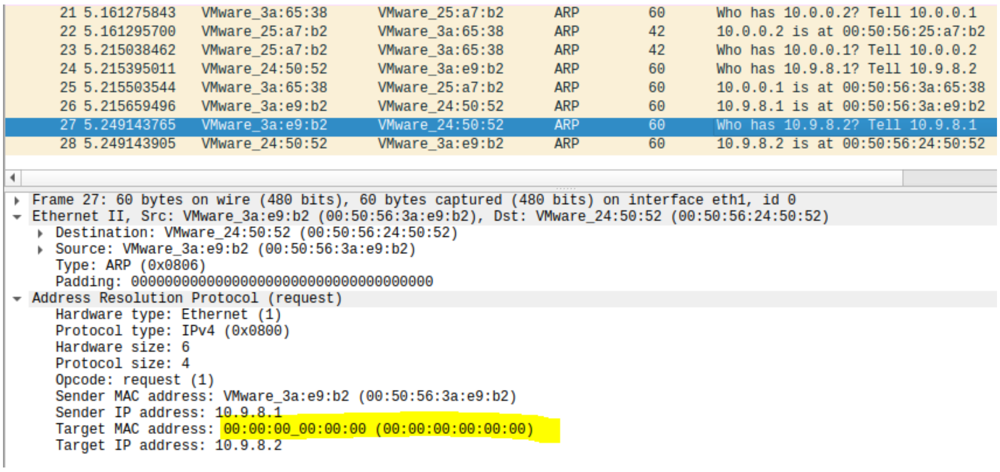
```
21 5.161275843  VMware_3a:65:38   VMware_25:a7:b2   ARP   60   Who has 10.0.0.2? Tell 10.0.0.1
22 5.161295700  VMware_25:a7:b2   VMware_3a:65:38   ARP   42   10.0.0.2 is at 00:50:56:25:a7:b2
23 5.215038462  VMware_25:a7:b2   VMware_3a:65:38   ARP   42   Who has 10.0.0.1? Tell 10.0.0.2
24 5.215395011  VMware_24:50:52   VMware_3a:e9:b2   ARP   60   Who has 10.9.8.1? Tell 10.9.8.2
25 5.215503544  VMware_3a:65:38   VMware_25:a7:b2   ARP   60   10.0.0.1 is at 00:50:56:3a:65:38
26 5.215659496  VMware_3a:e9:b2   VMware_24:50:52   ARP   60   10.9.8.1 is at 00:50:56:3a:e9:b2
27 5.249143765  VMware_3a:e9:b2   VMware_24:50:52   ARP   60   Who has 10.9.8.2? Tell 10.9.8.1
28 5.249143905  VMware_24:50:52   VMware_3a:e9:b2   ARP   60   10.9.8.2 is at 00:50:56:24:50:52

▸ Frame 22: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth1, id 0
▾ Ethernet II, Src: VMware_25:a7:b2 (00:50:56:25:a7:b2), Dst: VMware_3a:65:38 (00:50:56:3a:65:38)
  ▸ Destination: VMware_3a:65:38 (00:50:56:3a:65:38)
  ▸ Source: VMware_25:a7:b2 (00:50:56:25:a7:b2)
    Type: ARP (0x0806)
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: VMware_25:a7:b2 (00:50:56:25:a7:b2)
    Sender IP address: 10.0.0.2
    Target MAC address: VMware_3a:65:38 (00:50:56:3a:65:38)
    Target IP address: 10.0.0.1
```

*Figur 38 - ARP MAC address Reply*

## Opgave 2 - Undersøgelse af MAC-adresserne konstante på linklaget

Undersøgelse af ARP protokollen:

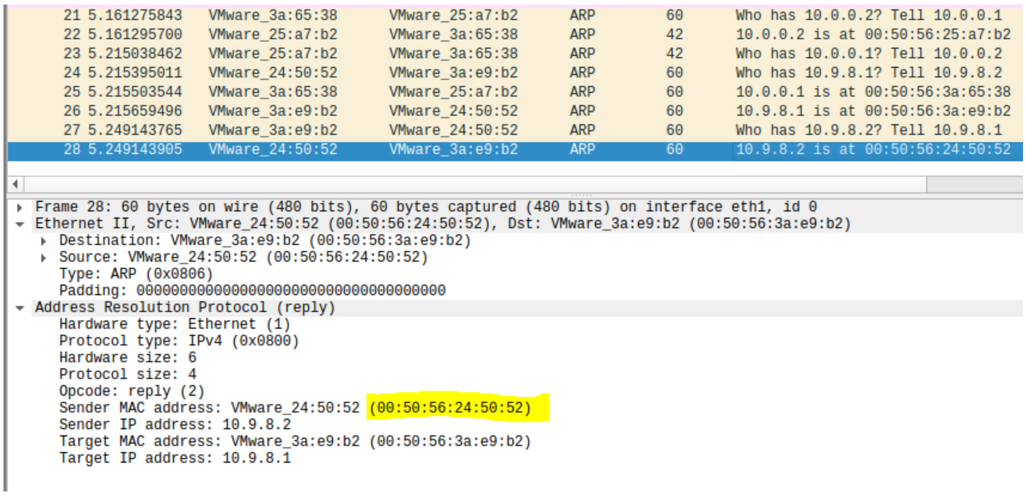
”Who has 10.9.8.2? Tell 10.9.8.1”



```
21 5.161275843   VMware_3a:65:38    VMware_25:a7:b2    ARP     60     Who has 10.0.0.2? Tell 10.0.0.1
22 5.161295700   VMware_25:a7:b2    VMware_3a:65:38    ARP     42     10.0.0.2 is at 00:50:56:25:a7:b2
23 5.215038462   VMware_25:a7:b2    VMware_3a:65:38    ARP     42     Who has 10.0.0.1? Tell 10.0.0.2
24 5.215395011   VMware_24:50:52    VMware_3a:e9:b2    ARP     60     Who has 10.9.8.1? Tell 10.9.8.2
25 5.215503544   VMware_3a:65:38    VMware_25:a7:b2    ARP     60     10.0.0.1 is at 00:50:56:3a:65:38
26 5.215659496   VMware_3a:e9:b2    VMware_24:50:52    ARP     60     10.9.8.1 is at 00:50:56:3a:e9:b2
27 5.249143765   VMware_3a:e9:b2    VMware_24:50:52    ARP     60     Who has 10.9.8.2? Tell 10.9.8.1
28 5.249143905   VMware_24:50:52    VMware_3a:e9:b2    ARP     60     10.9.8.2 is at 00:50:56:24:50:52

Frame 27: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth1, id 0
Ethernet II, Src: VMware_3a:e9:b2 (00:50:56:3a:e9:b2), Dst: VMware_24:50:52 (00:50:56:24:50:52)
   Destination: VMware_24:50:52 (00:50:56:24:50:52)
   Source: VMware_3a:e9:b2 (00:50:56:3a:e9:b2)
   Type: ARP (0x0806)
   Padding: 000000000000000000000000000000000000
Address Resolution Protocol (request)
   Hardware type: Ethernet (1)
   Protocol type: IPv4 (0x0800)
   Hardware size: 6
   Protocol size: 4
   Opcode: request (1)
   Sender MAC address: VMware_3a:e9:b2 (00:50:56:3a:e9:b2)
   Sender IP address: 10.9.8.1
   Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
   Target IP address: 10.9.8.2
```

*Figur 39 - ARP Who has? Tell*

”Reply 10.9.8.2 is at ….”



```
21 5.161275843   VMware_3a:65:38    VMware_25:a7:b2    ARP     60     Who has 10.0.0.2? Tell 10.0.0.1
22 5.161295700   VMware_25:a7:b2    VMware_3a:65:38    ARP     42     10.0.0.2 is at 00:50:56:25:a7:b2
23 5.215038462   VMware_25:a7:b2    VMware_3a:65:38    ARP     42     Who has 10.0.0.1? Tell 10.0.0.2
24 5.215395011   VMware_24:50:52    VMware_3a:e9:b2    ARP     60     Who has 10.9.8.1? Tell 10.9.8.2
25 5.215503544   VMware_3a:65:38    VMware_25:a7:b2    ARP     60     10.0.0.1 is at 00:50:56:3a:65:38
26 5.215659496   VMware_3a:e9:b2    VMware_24:50:52    ARP     60     10.9.8.1 is at 00:50:56:3a:e9:b2
27 5.249143765   VMware_3a:e9:b2    VMware_24:50:52    ARP     60     Who has 10.9.8.2? Tell 10.9.8.1
28 5.249143905   VMware_24:50:52    VMware_3a:e9:b2    ARP     60     10.9.8.2 is at 00:50:56:24:50:52

Frame 28: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth1, id 0
Ethernet II, Src: VMware_24:50:52 (00:50:56:24:50:52), Dst: VMware_3a:e9:b2 (00:50:56:3a:e9:b2)
   Destination: VMware_3a:e9:b2 (00:50:56:3a:e9:b2)
   Source: VMware_24:50:52 (00:50:56:24:50:52)
   Type: ARP (0x0806)
   Padding: 000000000000000000000000000000000000
Address Resolution Protocol (reply)
   Hardware type: Ethernet (1)
   Protocol type: IPv4 (0x0800)
   Hardware size: 6
   Protocol size: 4
   Opcode: reply (2)
   Sender MAC address: VMware_24:50:52 (00:50:56:24:50:52)
   Sender IP address: 10.9.8.2
   Target MAC address: VMware_3a:e9:b2 (00:50:56:3a:e9:b2)
   Target IP address: 10.9.8.1
```

*Figur 40 - ARP reply*

Herudover henvises til **figurerne 33, 34, 35, 36** fra øvelse 9, hvor der blev observeret at IP-adresserne forbliver uændret, men at netop MAC adresserne ændres.

## Konklusion for øvelsen

Der er observeret som i øvelse 9 at IP-adresserne forbliver uændret, men at netop MAC adresserne ændres.

10.0.0.1 til 10.9.8.1 for IP-adresserne.
H1 til R1 og R1 til H2 for MAC adresserne. (Ping fra H1 til H2 med R1 som gateway)

ARP (Address Resolution Protocol) fungere ved at hver enkelt IP-node, her H1,H2 og R1 har et ARP table hvor IP adresserne matches med MAC adresserne.

I øvelsen ses der at H1 spørger efer MAC adressen der matcher IP adressen 10.0.0.2, hvilket R1 kender og sender tilbage til H1.

Dette resulterer i at hvor der i Requestet fra H1 var Target MAC address: 00:00:00:00:00:00, altså ukendt, ved H1 efterfølgende at IP adressen 10.0.0.2 har MAC adressen  00:50:56:25:a7:b2.

Det samme bliver gentaget for de andre noder som der er vist i Figur 39 og Figur 40.