IDATT2104 - Oblig 1

Applikasjonslaget

WEB/HTTP

Denne oppgaven bruker en ferdig Wireshark pakkefangst til analysen. Fila heter *WS-HTTP-Capture* og er publisert sammen med oppgaven.

Velg linje 5 i pakkefangsten, høyreklikk og velg «Follow/TCP stream». Dette viser kommunikasjonen for den første av de samtidige forbindelsene som ble etablert mot webtjener.

a) Undersøk vedvarende forbindelser.
Linje 5 fra klient og linje 13 fra tjener viser headerlinjene for å holde en vedvarende forbindelse. Hva sender klient og hva svarer tjener på dette?

- Klient sender: Connection: keep-alive
- Tjener svarer:
    - Keep-Alive: timeout=5, max=100
    - Connection: Keep-Alive

b) Når ble bildet fra webtjener sist endret? Hvordan kan denne informasjonen brukes til å spare senere nedlasting ved oppslag på samme domenenavn, og hva spør klienten om da i sin forespørsel?

- Hvis «bildet» referere til faktiske bilder, som i .png, .jpg, .gif, er det flere alternativer, siden klienten spør om alle sammen:
    - /bitmaps/topp_datakommunikasjon.gif - Last-Modified: Fri, 26 Aug 2005 12:58:24 GMT
    - /bitmaps/bok_datakommunikasjon.jpg - Last-Modified: Wed, 06 Jul 2005 11:22:38 GMT
    - /bitmaps/bunn_datakommunikasjon.gif - Last-Modified: Thu, 07 Jul 2005 07:17:54 GMT
    - /forfatterne.JPG - Last-Modified: Thu, 07 Jul 2005 10:55:04 GMT
- Hvis «bildet» refererer til et bilde (image) av nettsiden, finner vi Last-Modified: Wed, 23 Sep 2020 13:02:56 GMT i stream 0 i requesten fra tjeneren.

c) Hva er stikkord for HTTPS; HTTP/2 og HTTP/3?

- HTTPS
    - Sikkerhet. S står for «Secure». Bruker TLS for å kryptere kommunikasjon.
    - Integritet. Beskytter mot man-in-the-middle angrep.
    - Autentisering. Sertifikater som autentiserer og bekrefter identiteten til tjener og klient.
    - Konfidensialitet. Sikrer at tilkoblingen er privat mellom kun de tiltenkte mottakerne.
- HTTP/2
    - I stor grad reduksjon av latency.

    - Header-kompressjon. Komprimerer informasjonen som sendes, hvilket reduserer latency.
    - Multiplexing. Mange forespørsler og svar kan sendes over samme forbindelse, som "ett signal", samtidig.
    - Server push. Server kan sende ressurser til klient før klienten ber om dem.
- HTTP/3
    - QUICK. Multiplexing via UDP.
    - Lavere latency, raskere loading, mye pga. QUICK.
    - Bedre håndtering av pakketap.

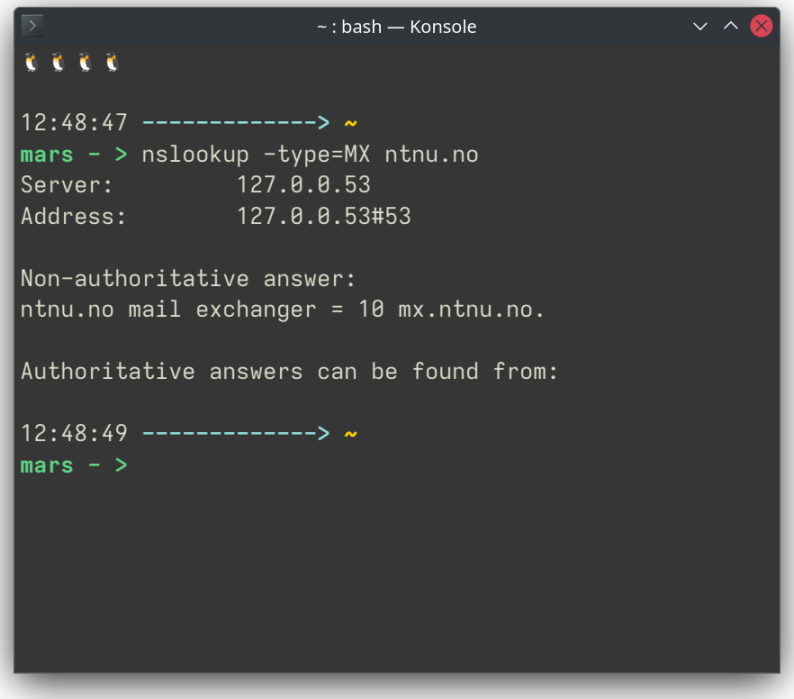
Navnetjenesten/DNS

Start Wireshark og gjør et oppslag *>nslookup ntnu.no*. Sett display-filter til: *dns.qry.name==ntnu.no*

a) Hvilke to typer ressursrecords spørres det automatisk etter?
    - Type: A (Host Address) (1)
    - Type: AAAA (IPv6 Address) (28)

b) Hva er time-to-live for disse svarene? Hvorfor er det hensiktsmessig å sette en utløpsverdi?
    - Time to live: 224 (3 minutes, 44 seconds)
    - TTL er hvor lenge en pakke kan eksistere på et nettverk før den blir forkastet av routeren. Dette gjør at pakker ikke har anledning til å sirkulere "uendelig" på nettverket og tar opp ressurser.

c) En annen type ressursrecord er MX for eposttjener. Hva viser svaret *for nslookup -type=MX ntnu.no*?
    - MX er «mail exchanger»



d) Finn IPv4-adressen til denne eposttjeneren. Hva er adressen, og kan den nås fra Internett?
   - Slår opp med *nslookup mx.ntnu.no*
     - Name: mx.ntnu.no
     - Address: 129.241.56.67

Transportlaget

TCP

Pakkefangst med Wireshark skal også brukes til å undersøke egenskaper ved TCP. Gjør samme filtrering som i oppgave 1

a) 3-Way handshake. Hva er resultatet av de tre første pakkene i pakkefangsten? Hvilke flagg i pakkene benyttes for dette?
   - SYN
   - SYN, ACK
   - ACK

b) Pålitelig overføring

I pakke nr 5 ber klienten om indeks-filen for domenenavnet *datakom.no*. I det påfølgende kommer det en serie pakker for overføring av denne. Det skal undersøkes sammenhengen mellom TCP sekvensnummer, kvitteringsnummer og nyttelast i disse pakkene frem til og med pakke 14.
OBS: sekvensnummer og kvitteringsnummer er 32 bit tall, men du skal benytte relativt nummer som vises i Wireshark. Fyll inn følgende tabell (grå farge fra klient og gul farge fra tjener)

Nr.	Innhold	KLIENT			TJENER		
		Sekv.nr	Kvitt.nr	TCP Length/ Payload	Sekv.nr	Kvitt.nr	TCP Length/ Payload
5	HTTP	1	1	619 bytes	-	-	-
8	ACK	-	-	-	1	620	N/A
9	TCP	-	-	-	1	620	1460 bytes
10	TCP	-	-	-	1461	620	1460 bytes
11	TCP	-	-	-	2921	620	1460 bytes
12	TCP	-	-	-	4381	620	1460 bytes
13	HTTP	-	-	-	5841	620	1117 bytes
14	ACK	620	6958	N/A	-	-	-

På nummer 8 og nummer 14 fant jeg ingen TCP Payload Length. Det eneste som sto der var Header Length, som var på 20 bytes i begge tilfeller.

Kontroll:

1) Skriv opp og regn ut summen av sekvensnummer og nyttelast i pakke 10 og sammenlikne denne med sekvensnummer i pakke 11.
   - Pakke 10 = 2921.
   - Dette er identisk med sekvensnummeret i pakke 11.

2) Sammenlikne summen av TCP nyttelast fra tjener og siste kvittering fra klient.
   - Summen av nyttelast = 6957 bytes.
   - Siste kvittering fra klient = 6958.

c) Hva er det da sekvensnummer og kvitteringsnummer forteller oss?
   - Sekvensnummeret forteller om det man sender, og kvitteringsnummeret forteller noe om det man har mottatt.
   - Sekvensnummer er «starten» på dataene i en TCP-pakke og teller hver enkelt byte som overføres. Sekvensnummeret er da nummeret til første byte av nyttelasten i pakken.
   - Kvitteringsnummer indikerer det neste bytenummeret man forventer å motta.

TLS

HTTP/1.1 benytter TLS for å etablere en sikker forbindelse. Start Wireshark og gjør et oppslag på NTNU. Bruk display-filter *tls.handshake.extensions_server_name==ntnu.no*
Finn den første «Client Hello» fra NTNU. Denne pakken tilbyr et sett av krypteringssuiter for kommunikasjonen. Tjener velger en av disse og sender beskjed i «Server Hello».

a) Hva heter den suiten som tjeneren velger?
   - *Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)*

b) Hvilke to metoder inneholder denne suiten for kryptering og hashing?
   - TLS, AES og RSA for kryptering
   - SHA256

Digitale sertifikater

a) Digitalt sertifikat
   a. NTNU-web bruker et digitalt sertifikat for autentisering og sikker kommunikasjon. Hvilken *signaturalgoritme* er brukt i dette sertifikatet for å lage sertifikatets fingeravtrykk?
      - Signature Algorithm: SHA-256 with RSA Encryption

   b. Hvordan brukes signaturalgoritmen for å lage sertifikatets fingeravtrykk?
      - Algoritmen brukes til å signere det digitale sertifikatet, hvilket autentiserer utsteder og sikrer integriteten til sertifikatet.
      - For å generere fingeravtrykket til sertifikatet brukes hash-funksjonen, her SHA256, for å lage en unik hash-verdi som blir sertifikatets fingeravtrykk.
      - Dette fingeravtrykket brukes til å bekrefte autentisitet og integritet.
      - RSA brukes her til å kryptere sertifikatets fingeravtrykk med utsteders private nøkkel. Deretter kan mottaker bruke den offentlige nøkkelen til å dekryptere og verifisere signaturen.

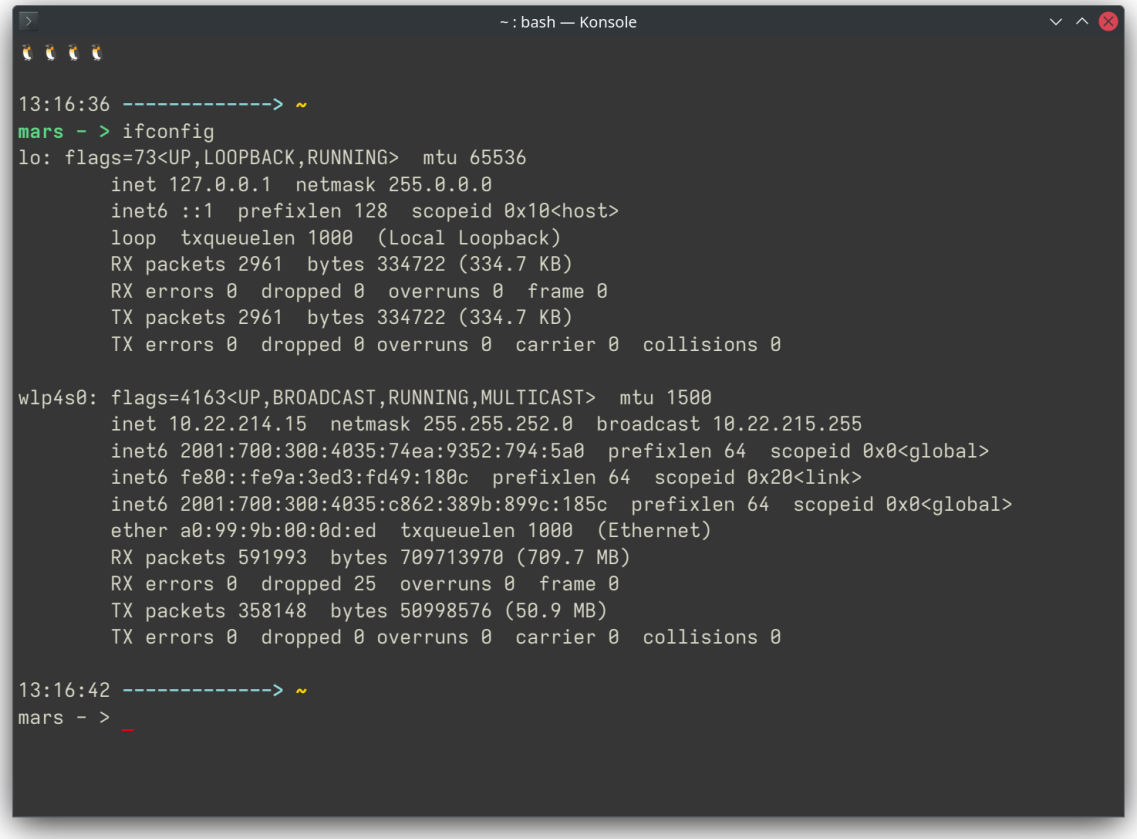
   c. Hvordan kan mottaker av sertifikatet kontrollere at sertifikatet er ekte, dvs ikke forfalsket?
      - Verifisere den digitale signaturen ved å bruke utsteders offentlige nøkkel til å dekryptere signaturen som følger med sertifikatet. Mottaker får da tilgang til hash-verdien som ble generert fra sertifikatet før signering.
      - Mottaker kjører samme hash-funksjon (f.eks SHA256) på det dekrypterte sertifikatet, hvilket genererer en hash-verdi.
      - Denne verdien sammenlignes med den dekrypterte hash-verdien fra den digitale signaturen.

- Hvis disse to stemmer overens, bekrefter dette at sertifikatet ikke har blitt endret siden det ble signert av utstedet, og dermed er sertifikatets integritet intakt.
- Man kan også kontrollere hele sertifikatkjeden, opp til sertifiseringsinstansen (Certificate Authority). Man sjekker da at de som utsteder sertifikatene faktisk er pålitelige og betrodde aktører.
- Utløpsdato og utstedelsesdato bør sjekkes.
- Sertifikater kan bli tilbakekalt og annullert av utstederen. Certificate Revocation Lists (CRL) og Online Certificate Status Protocol (OCSP).

Nettverkslaget

Undersøkelser av konfigurering på egen PC <u>på campus</u>

a) Kjør en ipconfig /all og ta et skjermklipp. Hva er n*ettverksadressen* til eget IPv4-Subnett?



IPv4-adressen er 10.22.214.15

Nettmasken er 255.255.252.0

Nettverksadressen er dermed 10.22.212.0

b) Er IPv4-adressen til DNS, DHCP og default Gateway del av eget IP-subnett? Begrunn svaret.
   - Ja, IPv4-adressene til DNS, DHCP, og default gateway er mest sannsynlig en del av samme IP-subnett som min IP-adresse.

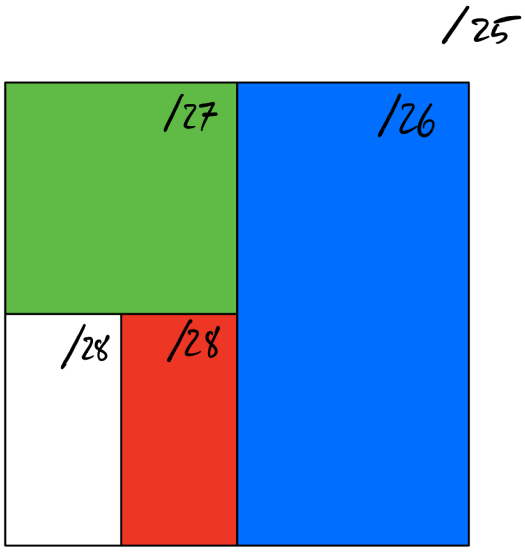
- Denne konfigurasjonen gir et /22-nettverk som strekker seg fra 10.22.212.0 til 10.22.215.255, som inkluderer både nettverks-, host-, og kringkastingsadresser.
- Ved å ha alle disse tjenestene på samme subnett som klientene blir kommunikasjonen over det lokale nettverket mer effektiv, og man slipper ruting over flere subnett. Dette optimaliserer nettverksytelsen og forenkler konfigurasjonen.

c) Finn din egen offentlige IPv4-adresse ved f.eks. oppslag på https://whatismyipaddress.com/ . Hvilken konklusjon kan vi trekke av dette?
   - IPv4: 129.241.236.115
   - Denne adressen er den man får tildelt av internettleverandøren for å kommunisere utad mot internett.
   - Adressen man har på det lokale nettverket er ikke det samme som den man har offentlig/utad.
   - Man ser her skillet mellom den lokale og den "globale" IP-adressen til enheten.

d) Frivillig undersøkelse *på Mobil*: Finn mobilens offentlige IPv4-adresse mens du er koplet på Eduroam. Slå deretter at wi-fi og prøv på nytt. Hvilken konklusjon kan du trekke av dette?

Subnetting med variable lengde på subnettmasken (VLSM)

a) Bruk «firkantmodellen» fra forelesning til å illustrere hvordan et /25-nettverk kan deles i et /26-nettverk og et /28 nettverk som følger kravet at /28-nettverket skal ha det laveste adresserommet. Det skal klart gå frem av illustrasjonen hvilke subnett som finnes totalt.



b) List opp følgende opplysninger for det spesifiserte /28-nettverket gitt at det opprinnelige /25-subnettet har nettverksadresse 192.168.0.0

Nettverksadresse : 192.168.0.0

Laveste nodeadresse : 192.168.0.1

Høyeste nodeadresse : 192.168.0.14

Kringkastingsadresse : 192.168.0.15

c) List opp de samme opplysningene for /26-nettverket

Nettverksadresse : 192.168.0.0

Laveste nodeadresse : 192.168.0.1

Høyeste nodeadresse : 192.168.0.62

Kringkastingsadresse : 192.168.0.63

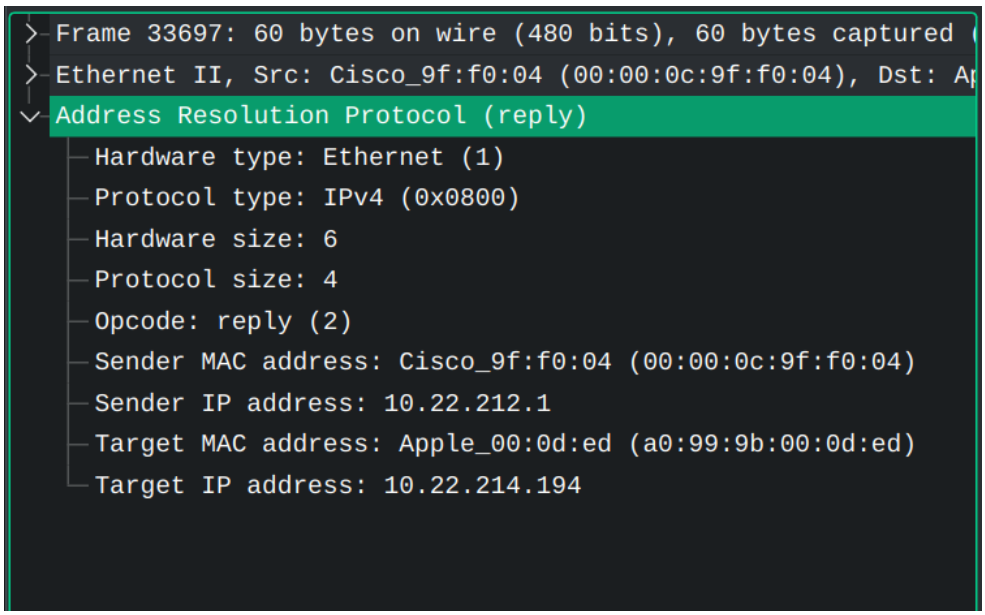
Funksjoner

ARP

ARP (Address resolution Protocol) brukes til å finne kopling fra IPv4-adresser til MAC-adresser på noder innenfor et subnett. Dette lagres dynamisk i en arp-tabell på PC.

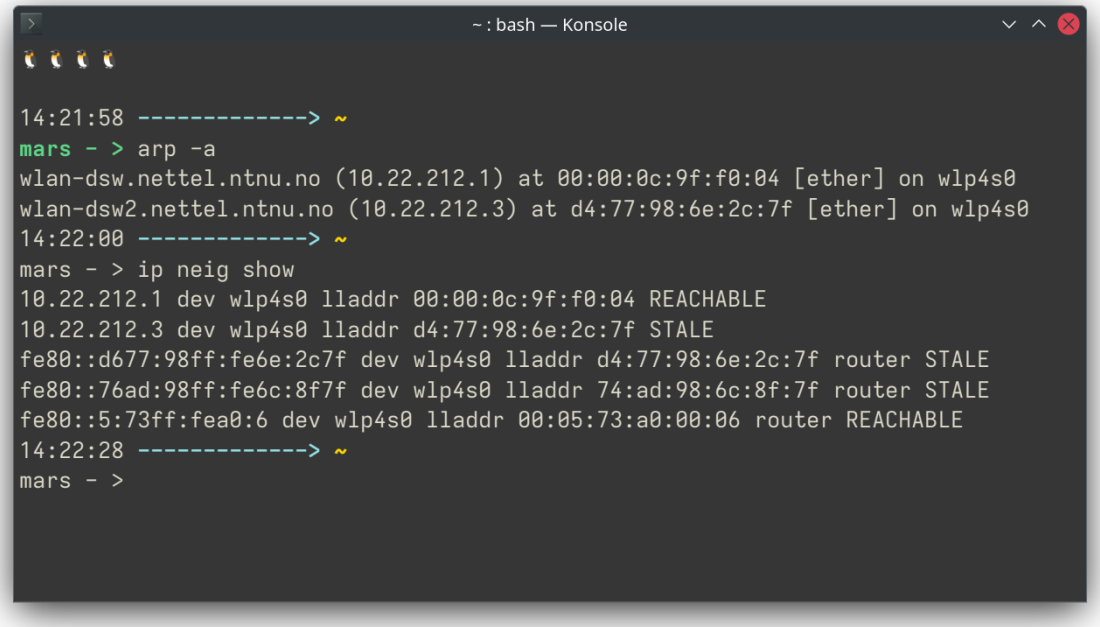
a) Beskriv hva ARP i hovedtrekk gjør og vis dette med et skjermklipp i Wireshark.

- ARP er en protokoll som brukes i IPv4-nettverk for å finne ut hvilken maskinvareadresse (MAC-adresse) som tilhører en gitt IP-adresse innenfor samme subnett.
- Når en PC skal kommunisere med en annen enhet på det lokale nettverket, men kun kjenner IP-adressen til den enheten vi vil kommunisere med, sender den en ARP-forespørsel ut på nettverket.
- Denne forespørselen kan ses på som at maskinen/enhenten sender ut en melding "hvem har IP-adresse XYZ, svar meg med din MAC-adresse".
- Enheten som har den etterspurte IP-adressen, svarer med sin MAC-adresse, slik at avsenderen kan sende data direkte til mottakerens fysiske nettverksadresse.
- Denne informasjonen lagres i avsenderens ARP-tabell for en viss periode for å redusere antallet nødvendige ARP-forespørsler for fremtidig kommunikasjon.



b) List ut din lokale IPv4 arp-tabell (>arp -a) og identifiser minst en kjent IP-adresse listet som dynamisk.

- wlan-dsw.nettel.ntnu.no (10.22.212.1) med MAC-adressen 00:00:0c:9f:f0:04
- wlan-dsw2.nettel.ntnu.no (10.22.212.3) med MAC-adressen d4:77:98:6e:2c:7f
- Dynamiske oppføringer i ARP-tabellen er adresser som ikke er "faste", dvs lagt inn av brukeren selv, men de oppføringene som maskinen "selv" legger inn og fjerner.
- Det er ikke snakk i "dynamisk *tildelte* adresser", slik som man har hos ISP, via DHCP på nettverk, osv., men "dynamiske *oppføringer i tabellen*".
- Altså at de har blitt lagt til pga en ARP-forespørsel, og vil fjernes når de utløper (normal etter noen minutter).
- Forøvrig tilsier informasjonen jeg finner om ARP, at oppføringene i tabellen i de fleste tilfeller vil være dynamiske, siden de må legges inn manuelt om de skal kunne være statiske.

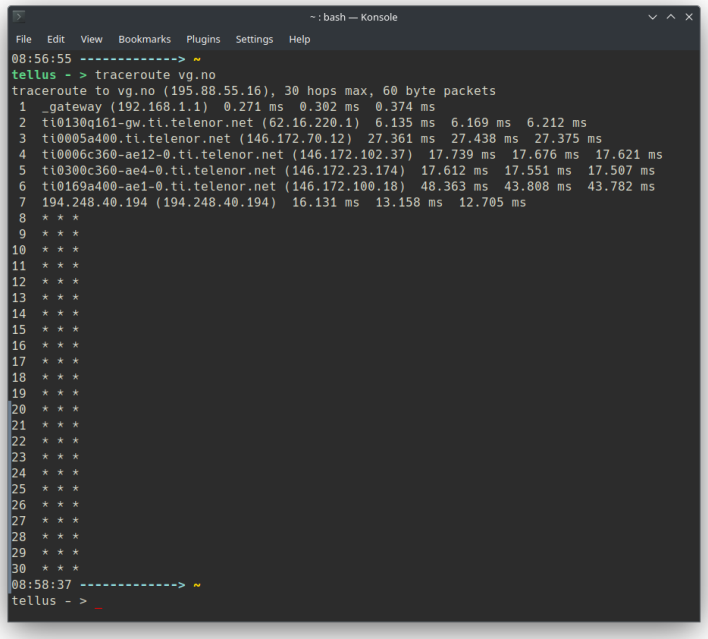


TRACEROUTE

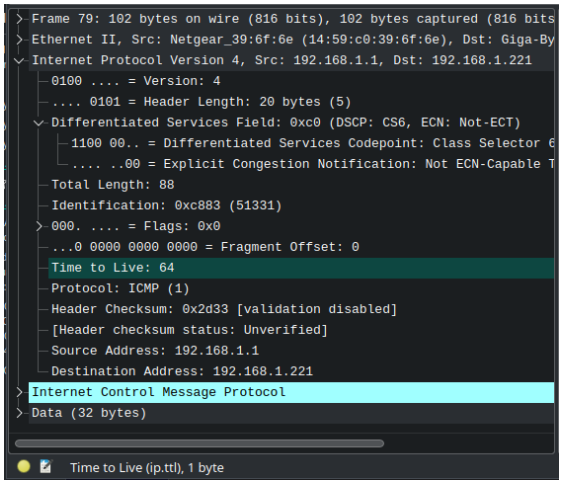
Bruk Traceroute til vg.no *(>tracert vg.no*) mens Wireshark kjører. Sett displayfilter ICMP eller ICMPv6.

a) Hva er verdi på TTL/Hop Limit i svaret fra første ruter (default gateway) og siste ruter?

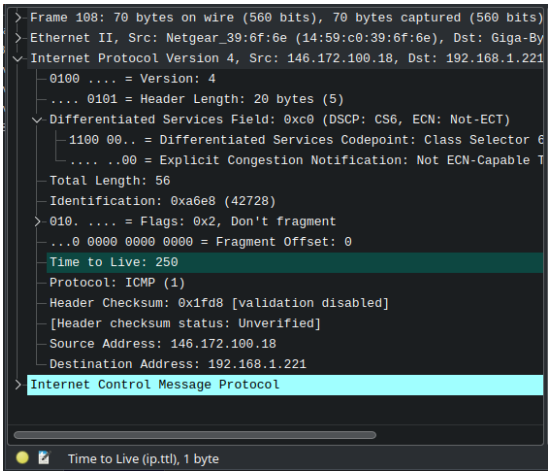
- ICMP:
  - TTL start: 64
  -
  - TTL slutt: 58
- ICMPv6:
  - Hop Limit start: 255
  - Hop Limit slutt: 255



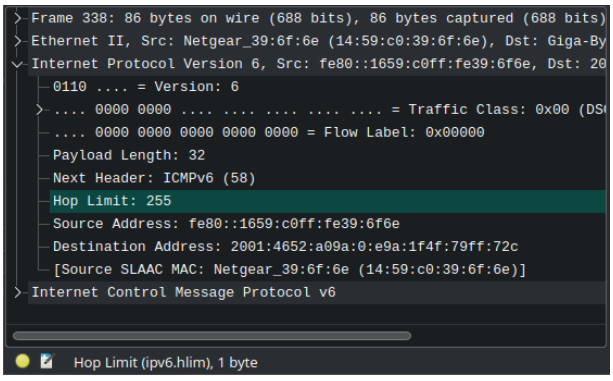
traceroute vg.no i Konsole (Linux CLI)



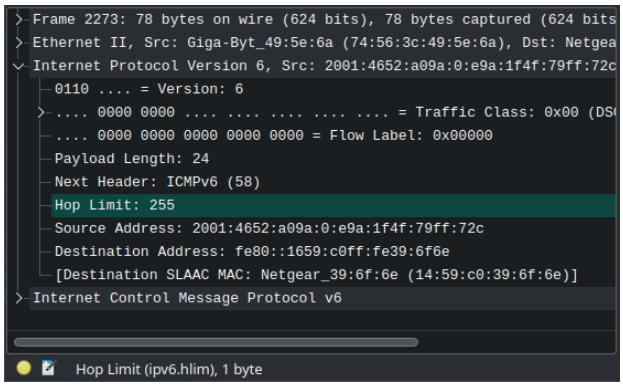
Time to Live 1



Time to Live 2



Hop Limit 1



Hop Limit 2