# **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.

1. 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

Dette headerfeltet ber tjeneren om å holde forbindelsen åpen etter at den aktuelle transaksjonen er ferdig, slik at den kan brukes til ytterligere HTTP-forespørsler uten å måtte opprette en ny forbindelse.

* Tjener svarer:

Connection: Keep-Alive

Keep-Alive: timeout=5, max=100

Tjeneren bekrefter at den vil holde forbindelsen åpen for flere forespørsler fra klienten.

timeout=5: Dette indikerer hvor mange sekunder tjeneren vil vente på neste forespørsel fra klienten før den lukker forbindelsen. I dette tilfellet, hvis tjeneren ikke mottar en ny forespørsel fra klienten innen 5 sekunder etter den siste responsen ble sendt, vil tjeneren lukke forbindelsen.

max=100: Dette definerer det maksimale antall forespørsler som klienten kan sende over en enkelt forbindelse før tjeneren lukker den. Etter 100 forespørsler, selv om timeouten ikke er nådd, vil tjeneren lukke forbindelsen.

Dette er med på å forbedre effektiviteten av nettverkskommunikasjonen ved å gjenbruke den samme TCP-forbindelsen for flere HTTP-forespørsler og svar, noe som reduserer overhead ved å åpne og lukke forbindelser. Det bidrar til raskere lasting av nettsider, ettersom tid som ellers ville blitt brukt på å opprette nye forbindelser, blir spart. Samtidig setter det grenser for å forhindre at forbindelser forblir åpne for alltid, noe som kunne ført til unødvendig ressursbruk på tjeneren.

1. 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??

Last-Modified: Wed, 23 Sep 2020 13:02:56 GMT

Lagring ved første nedlasting: Når klienten først laster ned bildet, lagrer den datoen gitt av Last-Modified-headeren sammen med bildet i sin cache.

Betinget GET-forespørsel: Ved etterfølgende forespørsler for det samme bildet, sender klienten en GET-forespørsel som inkluderer headeren If-Modified-Since med verdien satt til datoen den fikk sist.

Serverens respons:

Hvis bildet ikke er endret: Hvis bildet ikke har blitt endret siden datoen angitt i If-Modified-Since, vil tjeneren svare med en 304 Not Modified statuskode, og bildet blir ikke sendt på nytt. Klienten kan fortsette å bruke den cachete versjonen av bildet.

Hvis bildet er endret: Hvis bildet har blitt endret siden den angitte datoen, vil tjeneren svare med en 200 OK statuskode og sende det oppdaterte bildet.

Ved å bruke denne metoden unngår klienten å laste ned hele bildet på nytt hvis det ikke har endret seg, noe som sparer båndbredde og reduserer lastetiden.

1. Hva er stikkord for HTTPS; HTTP/2 og HTTP/3?

* HTTPS
* **Sikkerhet:** HTTPS er en sikker versjon av HTTP. Den bruker SSL/TLS-protokollen for å kryptere data som sendes mellom klienten og tjeneren.
* **Kryptering:** Beskytter dataene som overføres fra å bli lest eller endret.
* **Autentisering:** Serveren autentiseres av klienten ved hjelp av SSL/TLS-sertifikater for å bekrefte identiteten.
* **Integritet:** Sikrer at dataene som sendes, ikke er endret underveis.
* Port 443: Standard nettverksport for HTTPS-forespørsler.
* HTTP/2
* **Ytelse:** Forbedrer ytelsen til HTTP ved å tillate flere samtidige forespørsler over en enkelt forbindelse.
* **Binær:** HTTP/2 bruker en binær protokoll, i motsetning til den tekstbaserte HTTP/1.x.
* **Multiplexing:** Tillater flere forespørsler og svar å bli sendt samtidig over samme forbindelse.
* **Server Push:** Tjenere kan sende ressurser proaktivt til klienten før klienten ber om dem.
* **Header Compression:** Reduserer overhead ved å komprimere headerinformasjon.
* HTTP/3
* **QUIC:** HTTP/3 bruker QUIC-protokollen, som kjører over UDP, i stedet for TCP.
* **Lav latens:** Reduserer forbindelses- og transportlatens, spesielt i mobilmiljøer.
* **Forbindelse migrasjon:** Tillater at en forbindelse kan vedlikeholdes selv når underliggende nettverksendringer skjer, som når man bytter fra WiFi til mobilnett.
* **Strømmer:** Har forbedret håndtering av individuelle strømmer for å unngå problemet med "head-of-line blocking" som kan oppstå i HTTP/2.
* **Sikkerhet:** Til tross for at den kjører over UDP, tilbyr QUIC innebygd kryptering som ligner på den som tilbys av TLS/SSL i HTTPS.

## **Navnetjenesten/DNS**

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

1. Hvilke to typer ressursrecords spørres det automatisk etter?

Et bilde som inneholder tekst, Font, skjermbilde

Automatisk generert beskrivelse

Vi kan se forespørsler for A (Address Record) som representerer en IPv4-adresse

Det er primært A og AAAA rekorder som forespørres for å løse domenenavnet til henholdsvis IPv4 og IPv6-adresser.

1. Hva er time-to-live for disse svarene? Hvorfor er det hensiktsmessig å sette en utløpsverdi?
2. En annen type ressursrecord er MX for eposttjener. Hva viser svaret *for nslookup -type=MX ntnu.no*?
3. Finn IPv4-adressen til denne eposttjeneren. Hva er adressen, og kan den nås fra Internett?

# **Transportlaget**

## **TCP**

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

1. 3-Way handshake. Hva er resultatet av de tre første pakkene i pakkefangsten? Hvilke flagg i pakkene benyttes for dette?

1. 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 |  |  |  | - | - | - |
| 8 | ACK | - | - | - |  |  |  |
| 9 | TCP | - | - | - |  |  |  |
| 10 | TCP | - | - | - |  |  |  |
| 11 | TCP | - | - | - |  |  |  |
| 12 | TCP | - | - | - |  |  |  |
| 13 | HTTP | - | - | - |  |  |  |
| 14 | ACK |  |  |  | - | - | - |

Kontroll:

1. Skriv opp og regn ut summen av sekvensnummer og nyttelast i pakke 10 og sammenlikne denne med sekvensnummer i pakke 11.
2. Sammenlikne summen av TCP nyttelast fra tjener og siste kvittering fra klient.

1. Hva er det da sekvensnummer og kvitteringsnummer forteller oss?

## 

## **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».

1. Hva heter den suiten som tjeneren velger?

1. Hvilke to metoder inneholder denne suiten for kryptering og hashing?



## **Digitale sertifikater**

1. Digitalt sertifikat
   1. NTNU-web bruker et digitalt sertifikat for autentisering og sikker kommunikasjon. Hvilken *signaturalgoritme* er brukt i dette sertifikatet for å lage sertifikatets fingeravtrykk?

* 1. Hvordan brukes signaturalgoritmen for å lage sertifikatets fingeravtrykk?

* 1. Hvordan kan mottaker av sertifikatet kontrollere at sertifikatet er ekte, dvs ikke forfalsket?

# Nettverkslaget

**Undersøkelser av konfigurering på egen PC på campus**

1. Kjør en ipconfig /all og ta et skjermklipp. Hva er *nettverksadressen* til eget IPv4-Subnett?

1. Er IPv4-adressen til DNS, DHCP og default Gateway del av eget IP-subnett?   
   Begrunn svaret.

1. Finn din egen offentlige IPv4-adresse ved f.eks. oppslag på <https://whatismyipaddress.com/> . Hvilken konklusjon kan vi trekke av dette?

1. 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)**

1. 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.
2. List opp følgende opplysninger for det spesifiserte /28-nettverket gitt at det opprinnelige /25-subnettet har nettverksadresse 192.168.0.0

Nettverksadresse :

Laveste nodeadresse :

Høyeste nodeadresse :

Kringkastingsadresse :

1. List opp de samme opplysningene for /26-nettverket

Nettverksadresse :

Laveste nodeadresse :

Høyeste nodeadresse :

Kringkastingsadresse :

# 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.

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

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

## TRACEROUTE

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

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