

# TK1100

## FORELESNING 0x06 WWW

### 1. WWW og historie

- a) Forklar forskjellen mellom Internett og WWW (World Wide Web)

Internettet er et stort nettverk av maskiner som er koblet sammen og kommuniserer med hverandre. Internettet kjører på TCP/IP suiten, og bruker mange protokoller.

World Wide Web er en applikasjon som kjører på internettet, og er ansvarlig for du aksesserer via en browser/nettleser. Det finnes også andre typer Web, som holder på annen type informasjon og nettsider, eller har andre sikkerhetsnivåer. Eksempler på dette er Deep Web og Dark Web.

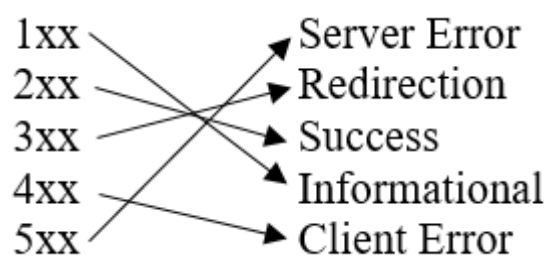
- b) Gi noen eksempler på når du ville brukt uttrykket «De Facto».

- «Man skal *de facto* alltid levere dokumenter i PDF format, om annet ikke er spesifisert»
- «Å bruke QWERTY oppsettet på keyboardet er en *de facto* standard. (Nesten alle gjør det)»
- «Når man skal koble til en skjerm (i nyere tid), vil man *de facto* bruke HDMI, Display Port eller VGA kabel.»
- «Når man får en presang, skal man *de facto* alltid si takk»

- c) Hva står RFC for?

Svar: Request for Comment

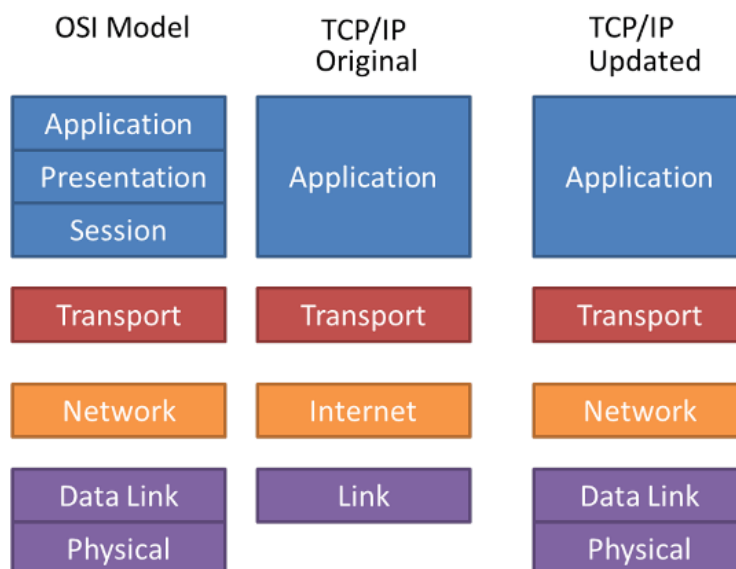
- d) Sett strek mellom de riktige statuskodene.



e) Nevn noen forskjeller mellom OSI og TCP/IP modellene.

OSI modellen har 7 lag (Application, Presentation, Session, Transport, Network, Data Link og Physical), mens TCP/IP bare (tidligere) hadde 4 (Application, Transport, Internet og Network Interface).

Likevel er det viktig å huske på at når man snakker om TCP/IP i dag, vil det være en kombinasjon av de to tidligere modellene, med 5 forskjellige lag. (Applikasjon, Transport, Nettverk, Datalink og fysisk)



Likevel er det forskjeller i hvordan disse lagene blir utført og brukt. TCP/IP headeren er blant annet 20bytes stor, mens OSI headeren bare er på 5bytes.

f) List opp de viktigste hendelsene i internettets historikk.

- Kleinrocks **Pakkeswitching**-prinsipp i 1961.
- Første **ARPAnet** node operativ i 1969.
- **Norge** blir en del av ARPAnet i 1972.
- **SMTP** og begynnelsen på email i 1982.
- **TCP** som standardprotokoll for ARPAnet i 1983.
- TCP **metningskontroll**, **IPv4** og **UDP** i 1988.
- Berner-Lee oppfantom **HTML** og **HTTP** i 1991.

## 2. Protokoller og andre begreper

a) Sant/Usant: Protokoller definerer hvordan kommunikasjon over internett skal fungere.

Sant. Dette kommer vi til å gå grundigere gjennom i senere forelesninger.

- b) Hva er forskjellen mellom en Klient-tjener-modell og en Peer-peer modell, tegn og vis.

En Klient Tjener modell er en modell der to parter kommuniserer med hverandre. Den ene av disse partene er en Tjener, ofte en server, som sitter på informasjon om for eksempel en nettside og dens innhold. Den andre parten er en klient, ofte en bruker av nettsiden, som vil sende en forespørsel til tjeneren om å for eksempel få informasjon, sende filer, logge inn eller se på en video.

Peer to Peer modellen har i motsetning ikke noe forskjell på medlemmene av den. Alle brukere kan fungere som både «klienter» og «tjenere», og den baserer seg på at noen brukere laster opp innhold, mens andre laster ned. Dette brukes ofte når man skal dele store filer over store mengder folk. For eksempel er det enkelte spillskaper som velger å bruke dette, blant annet Blizzard, som har det i StarCraft 2, World of Warcraft m.m.

- c) Hvilken port kobler en TCP-forbindelse seg mot for en hjemmeside.

Hvis nettsiden benytter http, går TCP over port 80, men hvis siden benytter https, går TCP over port 443.

- d) Bruk tracert (Windows) eller traceroute (OSX/LINUX) i terminalen mot [www.vg.no](http://www.vg.no), hvor mange stop gjør ruten?

Eksempel på output, i dette tilfelle er det 6 stop fra maskinen frem til [vg.no](http://vg.no).

```
C:\Users>tracert vg.no

Tracing route to vg.no [195.88.55.16]
over a maximum of 30 hops:

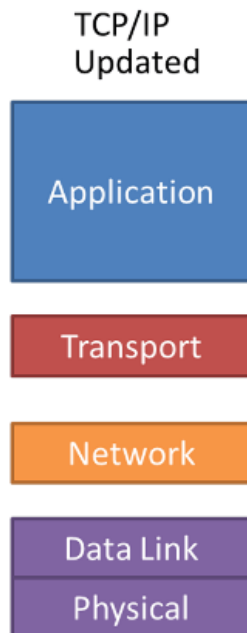
  1    3 ms    2 ms    3 ms    172.26.0.1
  2    5 ms    2 ms    3 ms    cB152584D.static.as2116.net [77.88.82.177]
  3    6 ms    2 ms    3 ms    193.90.113.51
  4    3 ms    9 ms   10 ms    ae21.cr2.oslosda310.as2116.net [193.75.1.76]
  5    4 ms    4 ms    3 ms    he5-2-0.ar2.ulv89.as2116.net [193.90.113.91]
  6    3 ms    2 ms    3 ms    www.vg.no [195.88.55.16]

Trace complete.
```

- e) Hvorfor deler man opp internett i ulike lag?

Dette er av flere grunner, der en av dem er struktur. Siden internettet er så stort, og det er så mange protokoller, har man funnet en måte å strukturere dette på. En annen av dem er abstraksjon og sikkerhet. For å sende en pakke over internett, må den først pakkes ned i mange ulike lag med kode, før den blir sendt, og senere pakket opp igjen gjennom alle lagene. De forskjellige lagene fikser på forskjellige problemer som kan ha oppstått i prosessen med de andre lagene. På et vis, kan man nesten også si at grunnen for at vi har så mange lag rett og slett er fordi internettet ikke var bygget for den mengden brukere vi er i dag, og dermed har man måttet fikse problemer underveis, ergo ble alle protokollene og lagene opprettet.

- f) I forelesning snakket vi om organisering av internett: Tegn opp den oppdaterte TCP/IP modellen.



- g) Hvorfor er internett usikkert oppbygget?

Den største grunnen er fordi at når internett ble laget var det ikke tenkt at det skulle bli så svært som det er i dag og derfor lagdes det ikke med innebygd sikkerhet. Man har med tiden skjønnet at det finnes mange sikkerhetsfeil i internett og når man først har oppdaget en feil, har man kun løst *det* problemet, ofte med å lage en protokoll. Derfor er internettet som vi kjenner i dag en stor pakke med mange ulike protokoller.

- h) Hva kan gå galt når data overføres over internett?

De tre hovedtingene som kan gå galt når informasjon transporteres over internett er tap av data, båndbredde og timing:

**Tap av data:** Avhengig av hva som brukes for å transportere informasjonen, er det mulig med tap av data, her er det noen applikasjoner som tåler litt tap av data (Feks. Streamingtjenester (Youtube) og enkelte spill), mens noe andre må ha 100% pålitelig transport av data (Email, dokument-delning).

**Båndbredde:** noen applikasjoner krever et minimum båndbredde (dataoverførings-kapasitet) for å fungere, for eksempel audio og video, mens noe kan bruke båndbredden dynamisk og «rette seg etter kapasitet», for eksempel filoverføring.

**Timing:** Om det er tidsforsinkelser så kan opplevelsen av ytelsen minskes, dette er vanlig i spill eller sanntidsprosesser.

i) For å se på nettverkstrafikk i Chrome:

1. Åpne Chrome
2. Høyreklikk og trykk på «Inspiser»
3. Velg fanen «Network» lengt opp i inspiser-vindu
4. Skriv in valgfri URL i browsern og trykk enter
5. Orienter deg siden litt rundt i inspiser-vindu for å se hva som skjer når du åpner en hjemmeside, klikk også rundt litt på hjemmesiden og se hva som skjer.
6. Gjør siden det samme med [www.ikke.no](http://www.ikke.no)

### 3. HTTP

a) Sant/Usant: HTTP oppbevarer informasjon om tidligere forespørsler.

Usant, HTTP protokollen er det man kaller for «Stateless».

b) Sant/Usant: En URL er en type URI

Sant. URI (Uniform Resource Identifier) er en streng som inneholder karakterer for å identifisere en ressurs, URL (Uniform Resource Locator) er en type av URI og brukes for å spesifisere en lokasjon av en ressurs på et nettverk og har mulighet å hente in ressursen.

c) <https://www.kristiania.no/sok?phrase=Digital%20Teknologi>

Identifiser de ulike delene av HTTP url-en, host, port, path, query, fragment. Alle er ikke nødvendigvis tilstede. Vet du også hva %20 betyr her?

<https://www.kristiania.no/sok?phrase=Digital%20Teknologi>

**Host :** <https://www.kristiania.no>

**Path :** /sok

**Query:** phrase=Digital%20Teknologi

%20 henter ut Ascii tegnet som tilsvarer hexadecimal 20. Dette er tenget for «mellomrom» / [SPACE].

d) I HTTP 1.0 GET-request, hvordan spesifiseres språk?

The diagram illustrates the structure of an HTTP 1.0 GET request. It shows the following components:

- request line (GET, POST, HEAD commands):** Indicated by a blue arrow pointing to the first line of the request: `GET /somedir/page.html HTTP/1.0`
- header lines:** Indicated by a blue arrow pointing to the subsequent lines: `User-agent: Mozilla/4.0`, `Accept: text/html, image/gif, image/jpeg`, and `Accept-language: fr`. The last line is highlighted with an orange box.
- Carriage return, line feed:** Indicated by a blue arrow pointing to the end of the header lines, with a note: (extra carriage return, line feed) indicates end of message.

- e) Hvilken type metoder har HTTP 1.1 som ikke HTTP 1.0 har?

**HTTP 1.0:** GET, POST, HEAD

**HTTP 1.1:** GET, POST, HEAD, PUT, DELETE, OPTION, TRACE

- f) Gi tre eksempler på statuskoder og hva de betyr.

**200 OK**

- spørring vellykket, objektet kommer senere i meldingen

**301 Moved Permanently**

- etterspurt objekt flyttet, ny adresse senere i meldingen

**400 Bad Request**

- spørring ikke forstått av tjeneren

**404 Not Found**

- etterspurt dokument/fil ikke funnet på denne tjeneren

**505 HTTP Version Not Supported**

## 4. DNS

- a) Hva brukes DNS til?

Når du skal inn på en hjemmeside, skriver du inn navnet på siden i browseren, her er det ikke enkelt å finne hvilken hjemmeside som skal lastes inn i din browser hvis ikke man vet hvor informasjonen skal hentes, og det er det DNS brukes til. Se på det som et personnummer eller et ID-nummer, [www.vg.no](http://www.vg.no) har en adresse og det er denne adressen som trengs for å finne ut hvor informasjonen skal hentes fra, og denne adressen kommer fra DNS. DNS-en inneholder altså en type register hvor en IP-adresse er knyttet til en hjemmeside, fungerer på samme måte som en telefonbok hvor det går an å søke opp på navn for å finne telefonnummeret.

- b) Hvordan er navne-tjenere fordelt?

DNS er hele systemet hvor det finnes flere navne-tjenere, her er det oppdelt for at det blant annet skal bli raskere å få tak i informasjonen man ser etter, men også for at det skal bli mer sikkert. Fordelingen er gjort hierarkisk og når din maskin spør etter en adresse så spør den først den nærmeste navne-tjeneren, hvis den ikke har informasjonen du ser etter, spør den neste navne-tjener i leddet. Totalt finnes det 13 stk. hoved navne-tjenere i verden og disse blir da kontaktet av lokale navne-tjenere ved behov.

- c) Bruk `nslookup` mot `www.vg.no` i terminalen, hvilken informasjon vises?

```
C:\Users>nslookup www.vg.no
Server: OSLDC102.EGMS.no
Address: 172.25.4.11

Non-authoritative answer:
Name: www.vg.no
Addresses: 2001:67c:21e0::16
           195.88.55.16
           195.88.54.16
```

- d) Prøv deg frem og endre Resource Records når du skal kjøre `nslookup`, hvordan ser de ulike svarene ut? Er det annerledes fra når du kjørte det første gang?

Eksempel på hvordan det ser ut med type **A** og **NS**:

```
C:\Users>nslookup
Default Server: OSLDC102.EGMS.no
Address: 172.25.4.11

> set type=A
> www.vg.no
Server: OSLDC102.EGMS.no
Address: 172.25.4.11

Non-authoritative answer:
Name: www.vg.no
Addresses: 195.88.54.16
           195.88.55.16
```

```
> set type=NS
> www.vg.no
Server: OSLDC102.EGMS.no
Address: 172.25.4.11

vg.no
      primary name server = fw.hmg9.vgnett.no
      responsible mail addr = hostmaster.ns.vg.no
      serial = 2021011402
      refresh = 300 (5 mins)
      retry = 150 (2 mins 30 secs)
      expire = 2419200 (28 days)
      default TTL = 300 (5 mins)
>
```