NTNU
Kunnskap for en bedre verden

Anvendelser av Internett – web

Bjørn Klefstad/Helge Hafting, Institutt for datateknologi og informatikk (IDI), NTNU
Lærestoffet er utviklet for faget «IINI1009/IFUD1109 Datakommunikasjon»

Resymé: I denne leksjonen ser vi nærmere på ulike anvendelser av Internett og hvilke oppgaver som løses på applikasjonslaget. Vi skal studere nærmere ulike protokoller i forbindelse med nettsider. Vi ser også på eksempler på bruk av programmet Wireshark for å analysere datapakker.

Lenkene i leksjonen er klikkbare, hvis pdf-leseren din støtter dette.

Innhold

2 Anvendelser av Internett – web

2.1 Anvendelser av Internett

2.1.1 Hva er Internett? kap 3.1

Internett er et nettverk som spenner over hele verden. I løpet av de siste årene har bruken av Internett nærmest eksplodert. Dette henger sammen med at folk flest har tatt dette i bruk hjemme. En hjemmedatamaskin og bredbåndstilknytning til Internett er etter hvert blitt et must. Både barn og voksne har etter hvert blitt daglige brukere av Internett og de tjenestene nettet kan tilby. Mange av tjenestene som epost, chat og ehandel har etter hvert blitt svært populære.

Wikipedia har en artikkel om Internettet, (`https://no.wikipedia.org/wiki/Internett`), som gir mye interessant informasjon og en god oversikt over utviklingen.

Når du leser læreboka er det viktig å merke seg forskjellen på Internettets infrastruktur, tjenester, roller og trafikk.

2.1.2 Web / www / nettsider / verdensveven kap 3.2

I dag finnes det svært mye informasjon som ligger tilgjengelig på nettet som nettsider. Praktisk talt alle bedrifter og svært mange private personer har sine egne hjemmesider der de legger ut alle typer informasjon. Det kan være tekstbaserte sider som innholder hypertekst. Det kan være bilder, lyd, video, blogger osv. Det finnes etter hvert nesten ingen grenser for hva enkelte folk legger ut. Derfor gjelder det for oss som brukere å være kritisk til våre kilder. Mellom all relevant og faglig informasjon, er det også utrolig mye søppel. Enkelte ting kan være direkte feil. Vi må derfor være ekstra oppmerksomme på kilden når vi henter informasjon fra Internett. Informasjonsmengden har etter hvert blitt så stor at det faktisk er et problem. Uten spesielle verktøy (søkemaskiner) finner vi ikke den informasjonen vi er på jakt etter. Søkemaskiner er alfa og omega for å finne frem riktig informasjon på Internett. Her finnes det mange ulike alternativer. Kvasir er den norske søkemaskinen og ifølge dem selv det beste alternative når vi jakter på informasjon på norsk. Ellers finner det mange andre som hver har sine styrker og svakheter (Google, Alta Vista, Yahoo).

I læreboka refereres det flere ganger til ulike RFC'er. I RFC-arkivet (`https://www.rfc-archive.org/`) kan du selv slå opp og studere innholdet i den enkelte RFC.

I dette kapitlet i læreboka er det viktig å merke seg hva en webklient (nettleser) gjør kontra en webtjener. Hvem tar initiativ til kommunikasjon, og hvilken vei går det som regel mest data. I tillegg gjelder det å merke seg hvilken protokoll som brukes og hvilke egenskaper denne har. Vi skal videre i leksjonen se nærmere på en del undertemaer til Web-tjenesten.

2.1.3 Nettlesere

Alle de eksemplene som er tatt med her er grafiske, men det finnes også de som er tekstbaserte (eks. Lynx, se (https://en.wikipedia.org/wiki/Lynx_(web_browser))).

Microsoft edge er en mye brukt nettleser for tiden. På deres nettside finner du mer informasjon om blant annet funksjonalitet (https://support.microsoft.com/nb-no/hub/4337664/microsoft-edge-help).

Mozilla firefox er en del av en hel produktfamilie og du kan lese mer om dem her (http://www.firefox.no/).

Opera er en norskutviklet nettleser som har eksistert i mange år og som stadig ypper seg mot Microsoft sin Internett explorer. Den er rask og egnet for håndholdte enheter. Du kan lese mer om denne nettleseren på (https://www.opera.com/browser/).

Safari er apple sin nettleser, les mer om den på (https://www.apple.com/safari/).

Google chrome er en forholdsvis ny men populær nettleser. Du kan lese mer om den på (https://www.google.com/chrome).

Webkit er en nettlesermotor basert på åpen kildekode. Den brukes både i Apple sin Safari, og på en del mobiltelefoner som f.eks. HTC Desire. Mer informasjon: (https://webkit.org/)

2.1.4 Webtjenere

Apache (https://www.apache.org/). Denne webtjeneren er utviklet på Unix, og finnes nå også på Windows. Tjeneren kan fritt distribueres og benyttes, og er den mest benyttede webtjeneren på Internett. Apache konfigureres og driftes i motsetning til IIS ved at driftsansvarlig arbeider med tekstfiler, slik det er vanlig i Unixmiljø.

Internet Information Services, se denne nettsiden for utfyllende informasjon: (https://www.iis.net/). Dette er en webtjener for Windowsmiljø som har et driftsmiljø basert på egne grafiske Windowsprogram. Dette er en egenskap som skiller den fra Apache. Webtjeneren IIS følger med i mange Windows versjoner. Denne tjeneren inneholder en rekke automatiserte tjenester, er skalerbar for web-applikasjoner og gir lave administratorkostnader. Men vi må betale lisens for å bruke den.

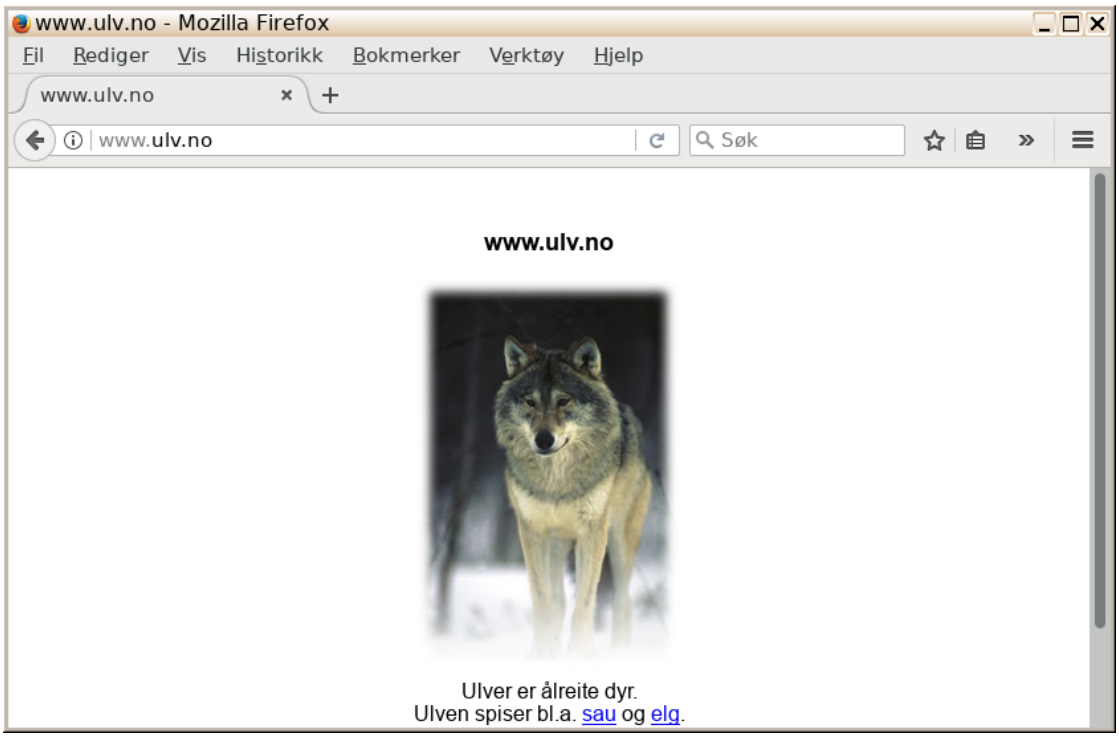
Nginx er en forholdsvis ny webtjener, som begynte å gjøre seg gjeldende i 2009. Etter eget utsagn fokuserer nginx på høy ytelse og lavt minneforbruk; i den grad dette stemmer, er den godt egnet for travle nettsteder med stor pågang. Nginx er tilgjengelig som open source fra https://nginx.org, og profesjonell support er tilgjengelig på https://nginx.com.

Det fins mange andre tjenere, men disse tre har til sammen over 80% av nettstedene. For detaljer om hvilke webtjenere som brukes for tiden, ta en titt på (`https://news.netcraft.com/archives/web_server_survey.html`).

2.1.5 HTML-kode

Se `https://no.wikipedia.org/wiki/HTML`

Vi må også ha klart for oss hvilken funksjon HTML-kode har og at en nettside bygges opp av tagger. For å bli bedre kjent med denne typen koding kan du åpne et vindu i nettleseren og be denne vise deg selve HTML-koden (Menyvalg: vis - kilde). Se om du kjenner igjen hovedprinsippene for bruk av tagger som er beskrevet i boka. Først ser vi på den grafiske utgaven av nettsiden, i figur 2.1.



Figur 2.1: Nettsiden `http://www.ulv.no`

Så ser vi på html-koden som ligger bak og genererer siden (bestemmer hvordan den skal se ut):

Listing 2.1: Html-kode for `http://www.ulv.no`

```
<html><head>
<meta http-equiv="content-type" content="text/html;␣charset=windows-1252">
```

```
<title>www.ulv.no</title>
</head>
<body bgcolor="#FFFFFF">
<center>
<br>
<br>
<font face="Arial,Helvetica">
<font size="+1"><b>www.ulv.no</b></font>
<br>
<br>
<img src="Filer_for_www.ulv.no/ulv.jpg" alt="Foto:␣Tom␣Schandy" width="200" height="303"><br>
Ulver er ålreite dyr.<br>Ulven spiser bl.a. <a href="http://www.sau.no/">sau</a> og <a href="http://www.elg.no/">elg</a>.
<br>
<br>
<br>
<br>
<font size="0">Domenet er registrert gjennom <a href="http://www.domeneshop.no/">domeneshop.no</a>.</font>
</font>
</center>


</body></html>
```

2.1.6 URL

Vi må kjenne til oppbygningen av en URL og hva som skjer dersom vi utelater deler av formatet. Prøv ut dette selv i nettleseren med for eksempel: `https://www.ntnu.no:443/idi`. Velg et annet portnummer eller ressursnavn og se hva som skjer. Prøv også å utelate deler av informasjonen og merk deg hva som skjer da.

2.1.7 Http

Vi må vi kunne skille mellom en http-forespørsel og et http-svar. Her kan du bruke Wireshark for å fange pakker selv, og sjekke at teorien presentert i læreboken virkelig stemmer. Start en pakkefangst i Wireshark. Så spør du etter en bestemt nettside i nettleseren. Du venter til siden er mottatt og avslutter pakkefangsten. Deretter setter du på et filter http og prøver å finne igjen forespørsel og svar. Får du dette til å stemme?

2.1.8 HTTP referer

Når du klikker på en lenke, sender nettleseren en forespørsel til den aktuelle tjeneren. I tillegg til å be om den aktuelle *nettsiden,* sender nettleseren også med et referer-felt som sier hvilke nettside du var på når du klikket.

De som lager nettsider har nytte av dette. Hvis de har mange sider som lenker til hverandre, kan de optimalisere designet utfra hvordan folk faktisk bruker nettstedet. Hvis veldig mange går fra side A til side B, kan de f.eks. slå sammen disse to sidene.

Mange er også interessert i å vite hvordan kunder finner nettstedet deres. Kom de via google? Eller via en reklameside? Det kan gi noen hint om hvorvidt de bør satse på reklame, eller på å komme høyere opp i google-søk.

Se også `https://en.wikipedia.org/wiki/HTTP_referer`

2.1.9 Effektivisering

For lokal mellomlagring kan vi enten ha en lokal cache på vår egen maskin eller vi kan ha en egen server (proxyserver) som tar seg av mellomlagringen.

En nettside består som regel av flere objekter. Det kan være et bilde, en tekstramme, en lydfil, en videofilm osv. Når nettleseren bruke vedvarende forbindelser overføres alle disse på den samme forbindelsen. Dersom vi ikke hadde brukt vedvarende forbindelser måtte vi ha koblet opp en forbindelse for hver av de objektene som skal overføres. Å koble opp og ned slike forbindelser tar tid og gjør at bruk av vedvarende forbindelser blir langt raskere ved mange objekter. Merk at det er en maks grense for antall objekter på en forbindelse, som settes av webtjeneren. Merk også at pipelining er koblet til vedvarende oppkobling og gir enda bedre flyt i kommunikasjonen

Parallelle forbindelser er koblet til nettleseren og ikke http-protokollen.

2.1.10 Cookies

Cookies/informasjonkapsler lagres på samme maskin som nettleresen befinner seg på. Dette betyr at vi har full kontroll på hvorvidt vi tillater slike cookies eller ikke. Nettleseren hekter på aktuelle cookies ved en forspørsel til en bestemt adresse, slik at for oss ser det ut som webtjeneren husker hva vi gjorde sist vi var på denne adressen.

Cookies brukes til mange ting. Som nevnt får de nettsider til å huske oss, slik at vi slipper å legge inn samme informasjon på nytt hver gang vi bruker siden. Vi kan f.eks. forbli innlogget på facebook. Http er en tilstandsløs protokoll, i utganspunktet har tjenermaskinen «glemt deg» fullstendig mellom hvert «klikk». Så uten cookies måtte du lagt inn innloggingsinformasjonen for *hvert eneste klikk* på facebook og lignende steder.

Cookies brukes også til å holde orden på «handlekurven» i en nettbutikk. I praksis husker ikke cookies på hele handlekurven, bare et «handlekurvnummer». Innholdet i handlekurven fins i butikkens database, handlekurvnummeret brukes for å plukke frem akkurat din handlekurv blant mange andre.

Cookies brukes også til mindre populære formål, som å overvåke oss. Det er særlig reklamebransjen som driver med dette. Hver gang du er inne på en nettside som står i forbindelse med reklameservere, noterer de seg hva du ser på. (Det er gjerne nettbutikker, sosiale nettverk, utstyrsprodusenters nettsider og reklamefinansierte sider som gjør slikt.) Reklamefinansierte sider vil sjekke dine cookies, og prøve å vise reklame for ting du har vist interesse for tidligere.

Dette gir oss mer relevant reklame, bortsett fra tilfeller som «jeg har nettopp lest mye om PCer på nett og *bestilt en*, så nå er jeg *ikke* lenger interessert i mer reklame for pc-utstyr…»

2.1.11 Cookies, tredjeparts cookies og personvern

Folk som er opptatt av personvern, ser mange problemer her. Svært mange nettsider er enten butikker, eller de er reklamefinansierte sider. Når alle disse driver overvåking, kan man finne ut nesten alt om de som surfer innom. Spesielt hvis nettstedene samarbeider. Hva er surferne interessert i, hvem er de, når er de på ulike steder? (Både nettsteder de er innom, og stedene de surfer fra.) Kort sagt, de vet hvem du er, hvor du bor, når du er på jobb, og når du er hjemme. Og alt om hva som interesserer deg.

En *tredjeparts cookie* er en cookie som settes av ett nettsted og leses av andre. Det er f.eks. mange nettsteder som bruker samme reklamefirma (doubleclick, google, adsense, …). Dermed vil alle disse stedene sette cookies avhengig av hva du ser på, og alle reklamefinansierte nettsider får hint om hva de bør vise deg. Å få «interessant» reklame kan føles bra. Men reklamefirmaet vet altså svært mye om deg, fordi de får samlet informasjon fra *alle* steder du er innom. Det kan også være avslørende om en låner bort PCen og andre ser hva slags reklamer en får opp, noen har interesser de ikke ønsker å dele med alle.

Nettlesere har lenge hatt mulighet for å skru av cookies fullstendig. Det er imidlertid en dårlig løsning, fordi mange nettsteder ikke kan fungere uten. Det har etterhvert kommet plugins for populære nettlesere som blokkerer cookies mer selektivt. Typisk tillater de nettsteder å sette/lese sine egne cookies, men ikke tredjeparts cookies satt av andre nettsteder. Dermed fungerer handlekurven i nettbutikken, men uten at reklamefirmaene får vite hva du klikker på eller kjøper. Andre web-teknologier, som javascript, håndteres tilsvarende. Interesserte kan se etter plugins som ublock origin, noscript, flashblock og adblock.

Å bruke slike personvernstillegg har noen bivirkninger:

- Surfingen går raskere, fordi det blir mindre reklame og sporingsrelatert programvare å laste ned.
- Mindre av skjermen går med til å vise reklamebilder, færre blinkende, hoppende eller glorete elementer.
- Færre overraskelser av typen «reklamefilm med kraftig lyd»
- Noen nettsider virker ikke, fordi de er for avhengige av spesielle opplegg med cookies/javascript. Men med litt ekstra innsats får man lagt inn unntak for sider som trenger det. Ublock er flink til å ikke blokkere «for mye».

En annen type sporing skjer via eksterne lenker. Det er for eksempel mange nettsteder som har facebook sine «like»-knapper. Surfer vi innom `https://www.ntnu.no`, ser vi bl.a. disse: Undersøker vi kildekoden, ser vi at ikonene eller scriptene deres ikke serveres fra NTNU sin tjener, men direkte fra twitter, facebook, instagram og youtube. Med andre ord: så snart du åpner høgskolen sin side, vet disse fire organisasjonene at du ser på NTNU sin side, fordi du nettopp lastet ned ikonene deres. De vet du kom fra ntnu.no, pga. «referer»-feltet i protokollen. Og de vet det er *deg*, pga. cookies. De logger når, og hvor ofte du er på NTNU. Det samme gjelder alle andre nettsider som har slike ikoner/knapper! Hvis du leser om en nyhetssak i nettavisen eller et bestemt produkt i nettbutikken, og det er en «like»-knapp der, er det noen som får vite akkurat hva du ser på. Selv om du ikke klikker på slike knapper. Programvare som «ublock» eller «noscript» kan hindre slikt også – da forsvinner facebook- og andre sosiale mediers knapper fra siden. En side uten sporingsprogrammer blir også ryddigere. Hvis jeg f.eks. lar ublock blokkere tredjepartinnhold, forsvinner twitter-, facebook- og instagramknappene.

2.1.12 Andre typer cookies

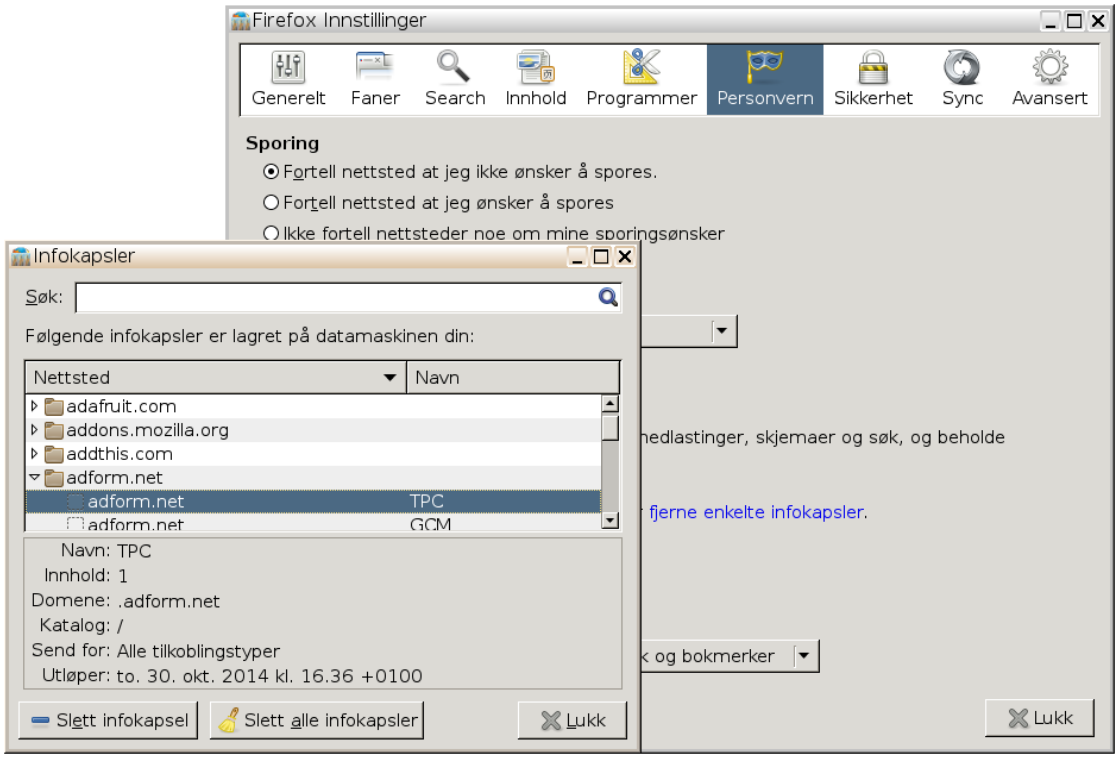
Programvare som flash og java har mulighet for å lagre egne filer på maskinen. De kan dermed oppnå samme funksjonalitet som cookies, selv om cookies er skrudd av. Men blokkeringsprogrammene nevnt i forrige avsnitt håndterer mange av disse mulighetene også.

2.1.13 Håndtere cookies

Nettlesere tilbyr oss å slette cookies, alle eller én og én. I firefox kan vi bruke «Rediger → Innstillinger → Personvern → fjerne enkelte infokapser» Se figur 2.2 på neste side. Det kan være en god idé å fjere cookies fra steder man ikke kjenner igjen. Hvis du fjerner *alle* cookies, må du logge inn på nytt på passordbeskyttede sider som ellers pleier å «huske» hvem du er.

2.2 Oppsummering

Til slutt og kanskje aller viktigst må vi merke oss hvordan alle disse begrepene henger sammen. Dette ble illustrert tidlig i kapittel 3 med en figur. Dersom du tar utgangspunkt i denne figuren vil du relativt greit kunne koble på og huske resten av innholdet i denne leksjonen. Vi gjentar figuren her, som figur 2.3 på side 10 slik at du kan forsøke å repetere innholdet assosiert med denne figuren.

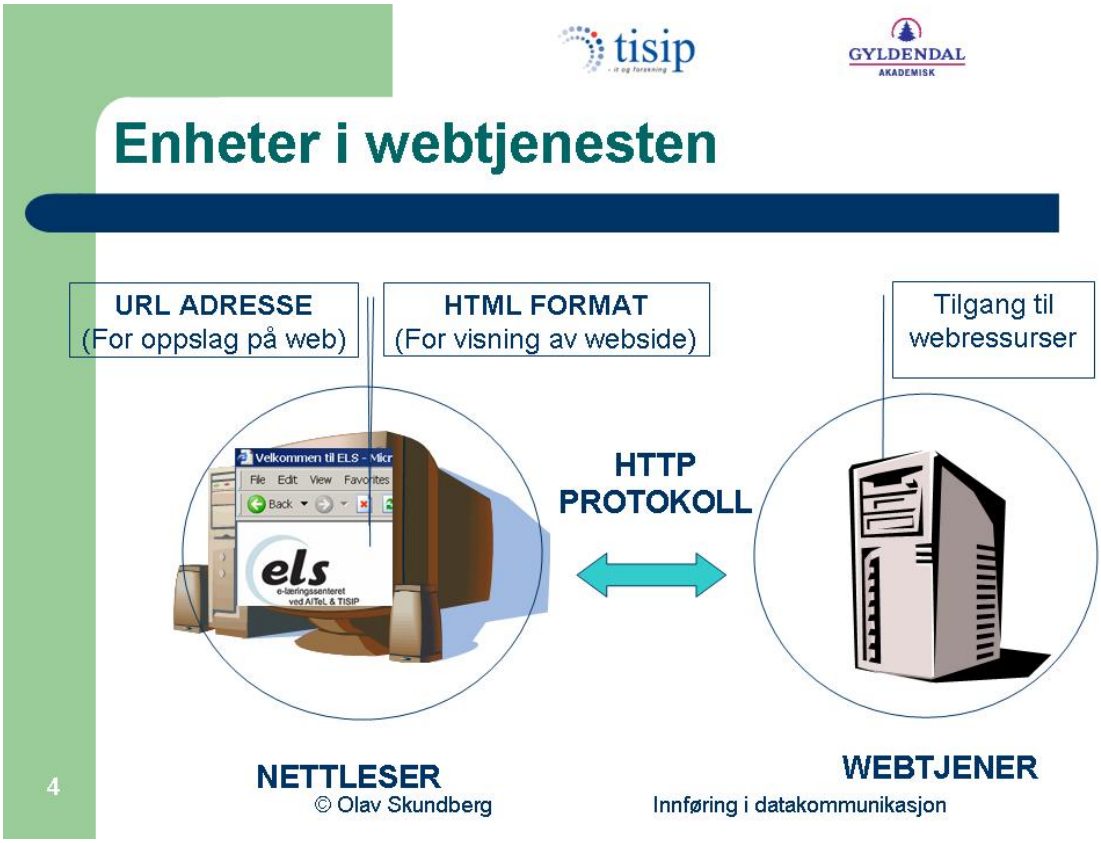


Figur 2.2: Håndtere cookies i firefox

2.3 Wireshark

For å kunne analysere datatrafikk trenger vi et verktøy både for å fange og å analysere pakker. I dette kurset skal vi bruke Wireshark som er en videreutvikling av tidligere brukte Ethereal (som er omtalt i læreboken). I Its' learning finner du en camtasiainnspilling som viser hvordan du kommer igang med å bruke verktøyet. Alternativt kan du lese «Bruke wireshark», som er mer oppdatert.

Installér Wireshark på din maskin og prøv deg frem med å fanger pakker. Dette programmet vil du få bruk for mange ganger i de påfølgende leksjonene for å analysere faktisk pakketrafikk for å sjekke at teorien stemmer med virkeligheten. De protokoller og egenskaper vi omtaler i teorien skal det være mulig å finne igjen i den faktiske datatrafikken som overføres. Dersom dere lykkes i å kjenne igjen teorien i datatrafikken vil dere oppnå en langt bedre forståelse for de prinsippene som ligger til grunn. Tiden dere bruker på å analysere datatrafikk er derfor vel anvendt tid. Prøv ut Wireshark!



Figur 2.3: Fra tjener til nettleser

2.4 Tilhørende kapitler i Innføring i Datakommunikasjon

Kapittel	Pensum	Navn
3	Ja	Anvendelser av Internett kap 3.1 og 3.2