Hypertext Transfer Protocol
og
World Wide Web

Dag Diesen
Institutt for informatikk
Universitetet i Oslo

Bakgrunn

World Wide Web (WWW) med overføring av hypertekst dokumenter over Internettet har oppnådd bred popularitet. I tillegg til å bruke WWW klienter, har mange lært seg Hypertext Markup Languge og dermed kunnet legge opp informasjon tilgjengelig for hele verden (eller i allefall hele Internettet). Men Hypertext Transfer Protocol (HTTP) for å overføre dokumentene fra en HTTP tjener til en WWW klient er ikke like kjent.

På dette seminaret vil jeg gjemmomgå Hypertext Transfer Protocol som brukes for å overføre dokumenter mellom en HTTP tjener og en WWW klient.

Innhold

Seminaret vil bli bygd opp slik:

- Rammeverket for World Wide Web vil bli presentert, og HTTP protokollen sin plass vil bli forklart.
- HTTP protokollen og framtidige utvidelser til denne blir tatt opp.
- HTTP/1.0 vil bli gjennongått i stor detalj.
- Herunder vil følgende punkter bli tatt opp:
  - Meldingstyper
  - Forespørsel
  - Svar
  - HTTP Metoder
  - Status
  - Ledelinjer
  - Data objekter
  - Bli enig algoritme

Kommunikasjon

Ideen med World Wide Web er å lage et rammeverk for bruk av forskjellige tjenester over internettet. En klient må derfor være i stand til å forstå flere protokoller (FTP, NNNTP for news, Gopher, Z39.50 for å søke i WAIS og HTTP).

Typisk mønster for kommunikasjon mellom en WEB-klient og en tjener vil være at WEB klienten sender en forespørsel om et dokument, og tjeneren sender tilbake dokumentet til klienten (hvis alt er OK).

Eksempelvis vil spøsrsmål etter et hypertekst dokument som har et bilde inkludert gi følgende kommunikasjon mellom WEB klient og HTTP tjener:

Klient Send selve hypertekst dokumentet.

Tjener Her er dokumentet.

Klient Send bildet som er inkludert i dokumentet.

Tjener Her er bildet.

URL

For å kunne finne et objekt (eller dokument), må adressen til objektet kunne identifiseres entydig. Syntaks og semantikk for dette adresserommet er definert som Uniform Resource Locator (URL) †.

En adresse angies ofte med følgende komponononenter:

- Protokoll navn.
- Internet domain name eller IP adresse.
- Port nummer (kan sløyfes hvis standard port nummer brukes).
- Vei (navnet til objektet hos tjeneren).

Følgende protokoll navn er definert:

ftp, http, gopher, wais (Z39.50), news (lokal news tjener), nntp og mailto.

Eksempler:

```
http://www.ifi.uio.no/les-meg.html
ftp://ifi.uio.no/pub/cim/README
mailto:dagdis@ifi.uio.no
news:ifi.tavle
```

† RFC 1738, Proposed Standard Protocol, Status: Elective

URI

Et generelt forslag til standard for å angi et mere generelt adresserom enn URL er kjent som URI (Internet-Draft).

URL standarden og en mere generelle identifikator kalt URN skal være spesialisering av URI standarden. URN er ment å identifisere et dokument som det kan finnes kopier av på ulike plasser (tilsvarende ISBN for bøker). Men siden det ikke finnes noe publisert utkast til URN standard har vi i dag bare URL standarden og holde oss til.

Forslag til WWW standarder

Følgende forslag til standarder er laget spesielt for World Wide Web:

Hypertext Transfer Protocol (HTTP)
Protokoll for kommunikasjon mellom en WWW klient og en HTTP tjener (Internet-draft).

Hypertext Markup Language (HTML)
Språket som brukes for å skrive et hypertekst dokument (Internet-draft).

Virtual Reality Modelling Language (VRML)
Språk for å modellere interaktiv simulering med mange deltagere av “kunstig” verden i ett nettverk (privat forslag).

Universal Resource Locator (URL)
Definisjon av adresser for å lokalisere dokumenter (Proposed Standard Protocol RFC: 1738).

Common Gatway Inteface (CGI) For kommunikasjon mellom en HTTP tjener og et eksternt program (privat forslag).

Common Client Interface (CCI) For kommunikasjon mellom WWW klient og et eksternt program (privat forslag).

HTTP tjener

En HTTP tjener tar i mot kommandoer fra en WWW klient, utfører disse kommandoene og gir svar tilbake til klienten.

Mange HTTP-tjener program er i stand til å sende forespørsler videre til andre program lokalt. Disse programmene leverer et svar direkte til HTTP tjeneren som så sender det videre til klienten. Denne mekanismen er kjent som HTTP tjener “Gatway” til andre program.

De fleste HTTP tjenere forstår kun HTTP protokollen, men det er minst et eksempel på en kombinert HTTP/Gopher tjener-program.

De to mest kjente HTTP tjener programmer og viktigste egenskapen som skiller mellom programmene:

Cern httpd 3.0 Kan brukes til å formidle spørsmål og svar mellom Internettet og installasjoner beskyttet av en brannvegg (“firewall”).

NCSA httpd 1.3 Gir mulighet for “Server Side Includes”.

Forslag til HTTP-protokollen

Her gir jeg en kort skisse av utviklingen av HTTP protokollen:

HTTP/0.9 Den første versjon av HTTP protokollen. Forespørselen er en GET kommando. Svaret er et data objekt.

HTTP/1.0 Basic Et tidlig forslag til HTTP/1.0 protokoll. Den inneholder noen metoder/kommandoer som seinere er utelatt.

HTTP/1.0 March 8, 1995 Siste Internet Draft til HTTP/1.0 protokollen.

Det er forslag til protokoll av 8. mars 1995 som omtales som HTTP/1.0 protokollen i dette seminaret.

HTTP/1.0 protokollen er en ren tekstlig protokoll. Den er basert på at en klient sender en forespørsel, som så besvares av en tjener.

Med hensyn på lagdelingen i OSI modellen ligger den på applikasjons nivået.

Framtidig utvikling av HTTP

Her er foreslåtte utvidelser til HTTP/1.0 protokollen:

Kryptert passord-kontroll Et tillegg kalt Digest Access Authenticiation er foreslått for å gjøre det mulig å sende passord kryptert.

Tredje part's kontroll Et annet tillegg kalt Mediated Digest Authentication skal gi en mulighet for å la en tredje part's tjener foreta passord-kontroll på vegne av kunde og tilbyder. Tredje part's tjener gir da en kunde et passord, og går god for kunden overfor flere informasjons leverandører.

Generell utvidelse En mekanisme som tillater at tillegg til protokollen kan defineres. Den inneholder også en generell mekanisme for å omsvøpe en forespørsel eller et svar. Omsvøpingen vil da tillate f. eks. kryptering av meldinger.

HTTP Ny Generasjon

Det er foreslått at det defineres en ny HTTP-NG protokoll. Denne protokollen oppretter (f. eks.) en TCP-forbindelse mellom klient og tjener som kan brukes for å sende mange meldinger begge veier. Denne forbindelsen deles opp i mange sesjoner. En av disse tar kontroll informasjon, og de andre tar dataobjekter. Protokollen koder alle meldinger. Dette i motsetning til HTTP/1.0 der meldingene sendes som tekst.

Noe forslag til protokollen er ikke definert †. Men en kravspesifikasjon er satt opp. Noen av disse krav er som følger:

- Effektiv overføring av data objekter.
- Støtte asynkron overføring.
- Støtte kryptering og betaling for informasjon.
- Tillate mellom-tjenere.
- Støtte obligatorisk visning (forfatter, opphavsrett, lisens-informasjon).
- Uavhengig av transport lag.

† Et meget uferdig utkast til en del av protokollen er laget.

Mindre kjent funksjonalitet

HTTP/1.0 protokollen er en temmelig ny protokoll som ennå ikke har oppnådd status som “Proposed Standard Protocol”. Den har derfor noen helt nye muligheter som venter på å bli tatt i bruk (av klient og tjener programmer):

- Mekanisme for å hente beste dokument, når flere utgaver av samme dokument er tilgjengelige. Det kan være dokument skrevet i det språk som blir best forstått av en bruker, eller et bilde som kan vises på den maskinen brukeren sitter ved.
- Metoder (eller kommandoer) som tillater en bruker å oppdatere dokumenter lagret hos en tjener, med hjelp av klient programmet (PUT, DELETE, LINK, UNLINK). Et eksempel kan være en informasjonsleverandør som selger plass til andre. De som kjøper plassen har mulighet for å sende et oppdatert dokument til en HTTP tjener med bruk av PUT metoden.

Meldingstyper

HTTP/1.0 protokollen definerer disse meldingstyper:

Full forespørsel Kommando som en WWW klient sender til en HTTP tjener.

Fullt svar Svar på en forespørsel som sendes fra en HTTP tjener til en WWW klient.

I tillegg defineres to meldingstyper som passer med HTTP/0.9 protokollen:

Enkel forespørsel GET kommando som en klient sender dersom HTTP/0.9 protokollen må brukes.

Enkelt svar Data som sendes fra en HTTP server til en klient når HTTP/0.9 protokollen må brukes.

Full forespørsel

En WWW klient sender forespørsel til en HTTP tjener. Denne forespørselen inneholder følgende deler:

- Forespørsels linje. Den må alltid være med.
- Eventuelt lede linjer.
- CR LF. Disse skilletegnene må alltid være med.
- Eventuelt et data objekt.

Lede linjene kan være av generell type, forespørsels type eller data objekt type.

En forespørsel

Her er eksemplel på en HTTP forespørsel:

```
GET /demo/postscript.ps HTTP/1.0
Date: Tue, 6 Apr 1995 19:05:03 GMT
Message-Id: <1256112.32@ifi.uio.no>
MIME-version: 1.0
Accept: application/postscript
Accept-Encoding: compress, gzip
User-Agent: Mosaic/1.5 libwww/2.17
```

Første linje er en forespørsels linje. De neste tre linjene er ledelinjer av generell type som angir tid meldingen ble laget, meldingsidentifikatoren, og MIME-protokollen som følges. Deretter kommer tre ledelinjer av forespørsels typen som angir mediatyper som aksepteres, koding som aksepteres og det WWW-klient programmet som sender forespørselen.

Siden ikke noe data objekt avslutter meldingen, må den avsluttes med en blank linje.

Fullt svar

En HTTP tjener sender svar til en WWW klient. Dette svaret inneholder følgende deler:

- Svar linje. Den må alltid være med.
- Eventuelt lede linjer.
- CR LF. Disse skilletegnene må alltid være med.
- Eventuelt et data objekt.

Lede linjene kan være av generell type, svar type eller data objekt type.

Eksempel på svar

Her er svaret på forespørselen (foil 15). Postscript koden i dette eksemplet sletter en fil med navnet demo.tmp når et ikke sikret ghostview program startes fra en WWW-klient †.

```
HTTP/1.0 200 OK
Date: Tue, 6 Apr 1995 19:05:15 GMT
Message-Id: <1212376.63@ifi.uio.no>
Server: NSCA/1.3 libwww/2.17
Content-Length: 25
Content-Type: application/postscript

%!
(demo.tmp) deletefile
```

Eksempel starter med en status linje. Den må alltid være med. Deretter kommer to ledelinjer av generell type som angir tid og meldings-identifikator. Så følger en ledelinje av svar type som gir beskjed om tjener programmet som brukes. Deretter kommer to ledelinjer av data objekt type som angir lengde og mediatypen til data objektet. Etter en blank linje kommer data objektet.

† Takk til Knut Omang som ga et eksempel på et slikt postscript program

Forespørsels linje

En forespørsels linje inneholder følgende elementer:

- Metode eller kommando som HTTP tjeneren skal utføre.
- SP
- URL. Den delen som er igjen når protokoll og tjenernavn og eventuell port nummer er fjernet.
- HTTP versjon som meldingen er kodet i.
- CR LF

Eksempler på forespørselslinjer:

```
GET /ifi/http.ps HTTP/1.0
POST /ifi/cgi-bin/motta-form HTTP/1.0
```

HTTP metoder

Følgende kommandoer eller metoder er definert i HTTP protokollen:

GET Gi meg data objektet med oppgitt URL (som angitt i forespørsels linja).

HEADER Returner status og ledelinjer for oppgitt URL, men ikke selve data objektet.

POST Ta i mot et data objekt fra klienten og kobl dette til oppgitt URL.

PUT Ta i mot et data objekt og lagr dette med den oppgitte URL.

DELETE Fjern data objektet med oppgitt URL.

LINK Opprett en link fra oppgitt URL til en annen ressurs angitt i en Link header ledelinje.

UNLINK Fjern linken med oppgitt URL.

Utvidelse Andre metoder som ikke er definert i HTTP/1.0.

Bruk av HTTP kommandoer

Her skisseres bruken av de tre vanlige kommandoene:

GET En WWW klient spør etter data fra en HTTP tjener. Dataene kan være HTML dokumenter, GIF-bilder, postscript filer og alt annet som klienten kan håndtere. Forespørselen kan også være et søk etter data. Da er søketermene angitt til slutt i URL'en etter et spørsmålstegn.

HEADER En spesiell WWW klient vil bruke denne kommandoen for å sjekke om en eller flere URL'er fortsatt kan brukes. En slik klient vil da typisk skrive ut beskjed om URL'er som det er noe galt med.

POST Denne kommandoen brukes når en WWW klient vil sende data utfylt i en form til en HTTP tjener. Slike data vil vanligvis sendes videre fra tjeneren til et program som kan behandle dataene. Mere kompliserte søkedata er også naturlig å sende med POST kommandoen.

Bruk av HTTP kommandoer

Her skisseres mulig bruk av kommandoer som foreløpig anvendes sjelden:

PUT Denne kommandoen brukes for å la en WWW klient opprette et permanent dataobjekt hos tjeneren. I de aller fleste tillfeller vil tjeneren ønske et gyldig passord fra WWW klienten før denne kommandoen aksepteres.

DELETE Denne kommandoen lar en klient fjerne et dataobjekt hos en server. Også her vil det være naturlig å kreve et passord.

LINK En klient kan med denne kommandoen opprette en link til f. eks. forrige side, en mail-adresse eller noe annet.

UNLINK En klient kan med denne fjerne en link.

Utvidelse Her kan andre ikke standard kommandoer defineres. Kommersielle tjenerprogram og klienter vil typisk bruke denne muligheten.

Status linje

En status linje inneholder følgende elementer:

- HTTP-versjon som svar meldingen er kodet i.
- SP
- Status kode.
- SP
- Tekst som følger statuskoden.
- CR LF

Eksempler på statuslinjer:

```
HTTP/1.0 200 OK
HTTP/1.0 300 Moved Permanently
HTTP/1.0 401 Unauthorized
HTTP/1.0 501 Not Implemented
```

Status koder

Statuskodene er delt i 5 grupper, avhengig av første siffer i koden:

1xx Informasjon Er tenkt brukt til å gi informasjon. *Reservert for framtidig bruk.*

2xx Suksess Aksjonen er utført, forstått eller akseptert.

3xx Ny retning Flere aksjoner må til for å fullføre forespørselen.

4xx Klient feil Noe er galt med forespørselen fra klienten.

5xx Tjener feil Tjeneren var ikke i stand til å behandle en gyldig forespørsel.

Gruppen for ny retning kan gi beskjed om at en ressurs er flyttet, om flere valg, og at en ressurs ikke er modifisert.

Gruppen for suksess kan gi beskjed om at et objekt er laget, at forespørselen er akseptert for prosessering, at informasjonen er provisorisk, at intet data objekt er returnert, eller at ganske enkelt at forespørselen er fullført.

Lede linjer

En ledelinje består av et navnefelt og et verdifelt skilt av et kolon †. Eksempel på en ledelinje:

```
Content-Language: no
```

Følgende typer av ledelinjer er definert:

Generell type Gir generell informasjon om meldingen (dato og tid, meldingsidentifikator, MIME-versjon og om meldingen er videresendt).

Forespørsels type Brukes til å gi informasjon om en forespørsel eller klienten som sender en forespørsel.

Svar type Brukes til å gi informasjon om tjeneren som svarer på en forespørsel.

Data objekt type Gir (meta-)informasjon om dataobjektet som tjeneren returnerer, eller om den ressursen som det er spurt om.

Ledelinjer av generell type og data objekt type brukes både i forespørsel og svar melding.

† Mye av syntaksen her er hentet fra MIME: RFC 1521 Draft Standard Protocol

Data objekt

Data objektet kan følge både en forespørsel og et svar. POST og PUT kommandoene vil ha et dataobjekt med seg i forespørselen fra klienten. En HTTP tjener vil returnere et data objekt som et resultat av en GET kommando.

Et dataobjekt kan være mye forskjellig, som:

- Vanlig tekst.
- HTML tekst.
- VRML kode.
- Postscript kode.
- DVI kode.
- Et GIF bilde.
- En MPEG video-sekvens.
- Kode for lyd.

En HTML tekst (eller VRML kode) får klienten til å sende forespørsel om andre data objekter som enten er “inline” bilder eller linker som brukeren aktiviserer.

Formatet til et dataobjekt

Hva er formatet til et dataobjekt? Ikke annet enn at det skal være en sekvens av 8 bits tegn.

Data-typen av et dataobjekt blir bestemt av en ordnet 3 nivå modell slik:

1. *Content-Transfer-Encoding:* Hvordan dataene blir overført.
2. *Content-Encoding:* Hvordan (og om) dataene er komprimert.
3. *Content-Type:* Dataenes mediatype.

Eventuell de-koding og prosessering av mediatype skjer i rekkefølge som angitt (f.eks dekomprimering før proseesering av postscript-filen).

Lengden av dataobjektet angies ved Content-Length, et eksplisitt merke som angir slutt på data, at tjeneren avslutter forbindelsen eller Content-Transfer-Encoding (eksperimentell mekanisme for å dele opp dataobjektet i pakker).

Forespørsel type

Følgende ledelinjer beskriver hvilke type dataobjekt som aksepteres. Dette gir grunnlag for forhandlinger mellom klient og tjener, der de samme dataene er på forskjellig form.

Accept Gir beskjed om hvilke media type som er akseptable for klienten. Type kan her være slikt som lyd, bilde eller forskjellige typer tekst. Preferanse mellom forskjellige media-typer kan angis.

Accept-Charset Kan akseptere ikke standard alfabeter.

Accept-Encoding Klienten kan akseptere kodede data.

Accept-Language De naturlige språk (brukar av) klienten foretrekker og rekkefølgen av hvilke språk som foretrekkes mest.

Eksempler på forespørsels ledelinjer

Her er eksempler på ledelinjer av forespørsel type:

```
Accept: audio/*; q=0.4,
        audio/x-aiff, mxb=10000
Accept-Charset: unicode-1-1
Accept-Encoding: compress, gzip
Accept-Language: no, dk, se, en-gb
From: dagdis@ifi.uio.no
Referer: http://sec-as/private.html
User-agent: DD-WWW/0.9
Pragma: no-cache
```

Bli enig

HTTP/1.0 protokollen gir mulighet for WWW-klient og HTTP-tjener å bli enig om hvilke dokument som WWW-klienten skal få. Klienten kan gi beskjed om:

- Hvilke mediatype som er akseptable, og graden av aksept.
- Hvilke alfabet som er akseptable.
- Hvilke kodinger som er akseptable (gzip, compress)
- Hvilke språk som foretrekkes.
- Hvor stort dokumentet kan være.

I tillegg kan tjeneren ha sin egen gradering på kvaliteten av dokumentene.

Hvis tjeneren har tilgang til flere utgaver av det etterspurte dokument, kan den velge den beste utgaven, utfra de krav og preferanser klienten setter, og den gradering som ligger inne hos tjeneren. Dersom ingen utgaver av dokumentet oppfyller klientens krav sendes en feilmeldingen om dette.

Bli enig

HTTP/1.0 algoritmen gir en nøyaktig algoritme for hvordan dette valget skal foretaes.

Anta at tjeneren har en eller flere utgaver av et dokument. Da velges det bort uakseptable eller ikke foretrukne utgaver slik:

- Ikke velg de utgaver som er kodet på uforståelig vis, ikke har et akseptabelt alfabet eller mediatype.
- Gi feilmelding dersom ingen utgaver av dokumentet er akseptable. Er det bare en utgave igjen returner denne.
- Velg den eller de utgavene av dokumentet som er skrevet i det mest foretrukne språk.

Er det flere utgaver igjen gies de resterende utgaver et kvalitetsmål etter en bestemt definert algoritme.

Bli enig

Er det flere utgaver igjen av dokumentet brukes følgende algoritme for å gi hvert dokument et kvalitetsmål:

```
hvis maks-antall-byte >= antall-byte
   eller
   maks-antall-byte ikke er gitt
så
   kvalitet :=
     tjener-mål * klient-mål
ellers
   kvalitet := 0.0
```

Både klient-mål og tjener-mål er et tall mellom 0.0 og 1.0, `antall-byte` er størrelsen på dokumentet, og `max-antall-byte` er maksimal størrelse satt av klienten.

Om det fortsatt gjenstår flere utgaver av samme dokument skal tjeneren lage et html-dokument med referanser til de utgavene som gjenstår slik at brukeren får anledning til å velge mellom de ulike formene av samme antatte kvalitet.

Generell type

Følgende ledelinjer er av generell type:

Date Tid for når meldingen ble laget.

Forwarded Hvem som videreformidlet meldingen.

Message-ID Entydig identifikator for meldingen.

Mime-version Mime protokollen som brukes for å lage meldinger.

Eksempler:

```
Date: Tue, 4 Apr 1995 16:21:46 GMT
Mime-Version: 1.0
```

Svar type

Følgende ledelinjer er av svar type:

Public Gir ei liste over ikke standar metoder støttet av tjeneren.

Retry-After Gir beskjed om hvor lenge tjeneren er utilgjengelig, dersom den returnerer status kode: 503 Service Unavailable. Verdien kan oppgies som absolutt tidspunkt eller forsinkelse etter forespørselen.

Server Inneholder informasjon om tjener programmet som behandler forespørselen.

WWW-Authenticate Beskjed om hvilke kontroll skjema for passord som skal brukes av klienten når den får status kode: 401 Unauthorized.

Eksempler:

```
Public: MGET, MHEAD, MYMETHOD
Retry-After: 240
Server: NSCA 1.3 libwww/2.17
```

Data objekt type

Følgende ledelinjer er av data objekt type og beskriver dataobjektet som sendes med eller ressursen det spørres om:

Allow De metodene som er støttet av ressursen identifisert med URL'en oppgitt i forespørselen.

Last-Modified Dato og tid for når ressursen sist ble endret.

Title Tittel på data-objektet.

Content-Encoding Hvilken kode som er brukt for dataobjektet.

Content-Language Hvilke naturlig språk brukes i dataobjektet.

Content-Length Størrelsen på dataobjektet.

Content-Transfer-Encoding Hvilke overførings-kode er brukt på dataobjektet.

Content-Type Media typen på dataobjektet.

Data objekt type

Følgende ledelinjer er av data objekt type og beskriver dataobjektet som sendes med eller ressursen det spørres om:

Version Versjons-nummeret på dataobjektet.

Deriwed-From Versjons-nummeret som et dataobjekt hadde før det ble forandret. Brukes når en PUT kommando erstatter et data-objekt med et endret data-objekt. Dette må være med når versjons-nummeret fulgte med det opprinnelig dataobjektet som klienten nå endrer.

Expires Gir tidspunktet for når et data-objekt ikke lenger skal oppbevares lokalt ("caches").

Link Beskriver en link mellom data-objektet og en annen ressurs.

URI Gir en eller flere Universal Resource Identifiers † for et data-objekt. Må følge med et svar med statuskodene: 201: Created, 301: Moved Permantly og 302: Moved Temporarily.

† Det vil være en URL som brukes.

Eksempler på data objekt ledelinjer

Her er eksempler på ledelinjer av data objekt type:

```
Allow: POST
Content-Encoding: compress
Content-Language: no, en
Content-Length: 132560
Content-Type: application/postscript
Content-Transfer-Encoding: 7bit
Version: 1.1.6
Derived-From: 1.1.5
Expires: Thu, 06 Apr 1995 23:00:00 GMT
Title: HTTP/1.0
URI: http://secure.no/moved.html
```

Litteraturliste ---

Her er litteratur om HTTP protokollen som er brukt som grunnlag for foredraget:

Barners-Lee, Fielding and
Frystyck Nielsen:
Hypertext Transfer Protocoll HTTP/1.0
HTTP-Working Group
Internet-draft: March 8, 1995

Hoestetler et. al.: A proposed exten-
sion to HTTP:
Digest Access Autentication
HTTP-Working Group
Internet-draft: March 24, 1995

Kristol: A proposed
exension mechanism for HTTP
HTTP-Working Group
Internet-draft: January 1995

Raggett: Mediated digest
authentication
HTTP-Working Group
Internet-draft: March 28, 1995

Spero: HTTP-NG Architectural Overview

URL-linker

HTTP spesifikasjonene kan nåes med følgende URL-linker:

```
<URL:http://www.ics.uci.edu/pub/
        ietf/http/>
```

og

```
<URL:http://www.w3.org/hypertext/WWW/
        Protocols/Overview.html>
```