

APLIKACIJE: SVETSKA RAČUNARSKA MREŽA (HTTP)

HTTP

Uvod

- ◆ Jedna od najpopularnijih aplikacija je podrška za svetsku računarsku mrežu (www, World Wide Web).
- ◆ Ovde su obrađeni:
- ◆ (1) osnovni protokol za prenos web stranica do web pregledača (browser).
- ◆ (2) baferovanje ("keširanje") web stranica.

Istorijat i značaj Web-a

- ◆ Na početku razvoja Internet-a, 1/3 saobraćaja je poticala od upotrebe FTP-a.
- ◆ Web je uveden početkom 90-tih.
- ◆ 1995.g. Web saobraćaj je prevazišao FTP saobraćaj.
- ◆ Do 2000.g. Web je postao dominantan oblik upotrebe Interneta, i danas većina korisnika poistovećuje ta dva pojma.

Arhitektura

- ◆ Web se sastoji od velikog skupa *hipermedijumskih* dokumenata, *Web stranica*, dostupnih putem Interneta.
- ◆ Sufiks *medijumski*, označava da dokument može da sadrži informaciju u različitim oblicima (tekst, slike, govor, zvuk, itd.).
- ◆ Prefiks hiper, označava da dokument sadrži *veze (links)* do drugih delova istog ili srodnih dokumenata.
- ◆ Arhitektura Web-a ima dve komponente:
 - ◆ (1) *Web pregledač*, program klijenta, i
 - ◆ (2) *Web server*, koji upravlja stranicama, i dužan je da klijentu obezbedi primerak tražene stranice.

HTML (HyperText Markup Language)

- ◆ Je jezik u kom se pišu Web stranice.
- ◆ Često se sadržaj Web stranica naziva *hipertekst*.
- ◆ Same stranice predstavljaju mešavinu teksta (ASCII) i drugih sadržaja, koji se predstavljaju u odgovarajućim standardima.
- ◆ Npr. slike se mogu predstaviti u GIF (Graphics Interchange Format) ili JPEG (Joint Picture Encoding Group) obliku.
- ◆ Pored sadržaja, hipertekst sadrži i uputstva za njihovo prikazivanje, koja se nazivaju oznakama (tags).

Hipertekst oznake (tags)

- ◆ Oznaka se piše između znakova "<" i ">".
- ◆ Neke oznake se pojavljuju u parovima, i odnose se na hipertekst između uparenih oznaka.
- ◆ Npr. pregledač će tekst između oznaka <CENTER> i </CENTER> prikazati na sredini prozora (tj. izvršiće "centriranje").

Jednoobrazni lokator resursa (Uniform Resource Locator, URL)

- ◆ Je jedinstvena adresa Web stranice.
- ◆ Predstavlja specifičan slučaj jednoobraznog identifikatora resursa (Uniform Resource Identifier, URI).
- ◆ URL specificira šemu (tj. protokol) pristupa resursu.
- ◆ Format URL-a zavisi od šeme, odnosno protokola.
- ◆ Za *http*, URL ima sledeći oblik:
- ◆ *http://racunar[:prolaz]/putanja[;parametri][?upit]*
- ◆ Uglaste zagrade označavaju neobavezne stavke.
- ◆ *racunar* je ime domena ili IP adresa Web servera.

Objašnjenje http URL-a

- ◆ *:prolaz* se navodi samo ako se ne koristi dobro poznati prolaz broj 80.
- ◆ *putanja* specificira položaj stranice u sistemu za rukovanje informacijom (file system) Web servera.
- ◆ *parametri* su neobavezni dodatni parametri, a *upit* je neobavezno pitanje.
- ◆ Npr. <http://www.google.com/> je URL popularnog Web pretraživača.
- ◆ Ukoliko je Web server implicitno poznat, moguće je koristiti *relativni* URL, koji se dobija od gore opisanog *apsolutnog* izostavljanjem stavke *racunar*.

Pristup Web-u

- ◆ Počinje zadavanjem URL-a.
- ◆ Pregledač ga analizira i dobavlja traženu stranicu.
- ◆ Za svaku hipervezu dokument sadrži par vrednosti:
 - ◆ (a) tekst koji se prikazuje na ekranu, i
 - ◆ (b) URL koji se prati ako korisnik izabere tu stavku.
- ◆ Sidro je par oznaka `<A ...>` i ``. Ono služi za definisanje hiperveze.
- ◆ Unutar prve oznake navodi se URL (umesto ...), a tekst koji pregledač prikazuje se navodi između para oznaka.

Primer jednostavne HTML stranice

<HTML>

Jedan od najpopularnijih pretraživača danas je

Google.

<HTML>

Na ekranu se ova stranica vidi ovako:

Jedan od najpopularnijih pretraživača danas je Google.

Pregledač podvlači reč *Google* da bi pokazao da ona odgovara hipervezi koja se može dalje pratiti.

HTTP (HyperText Transport Protocol) (1/3)

- ◆ Je protokol za prenos HTML stranica od Web servera do Web pregledača. On ima sledeće osobine:
- ◆ (1) Radi na **aplikacionom** nivou. Deklariše pouzdanu isporuku podataka, i oslanja se na protokole nižeg nivoa u tom pogledu.
- ◆ (2) Jedna strana (obično pregledač) šalje **zahtev**, na koji druga strana šalje **odgovor**.
- ◆ (3) HTTP radi **kombinaciono**, odnosno nema unutrašnjih stanja. Drugim rečima, ne čuva se nikakav istorijat prethodnih zahteva.

HTTP (HyperText Transport Protocol) (2/3)

- ◆ (4) HTTP obezbeđuje **prenos** informacije u **oba smjera**. Najčešće se stranice prenose od servera ka pregledaču. Moguć je prenos i u drugom smeru (npr. kad korisnik šalje obrazac, tzv. "formu").
- ◆ (5) Postoji mogućnost **pregovaranja** o detaljima komunikacije (npr. skup znakova koji se koristi u prenosu).
- ◆ (6) U cilju poboljšanja vremena odziva, pregledač **baferuje** primerak svake stranice. Kad korisnik ponovo zatraži istu stranicu, pregledač preko HTTP može da proveri da li se ta stranica na serveru u međuvremenu promenila.

HTTP (HyperText Transport Protocol) (3/3)

- ◆ (7) HTTP dozvoljava da mašina na putanji između pregledača i servera dobije ulogu zastupnika (proxy server), koji baferuje stranice i odgovara na zahtev pregledača šaljući svoje lokalne kopije.

Vrste HTTP zahteva

- ◆ GET – pribavljanje web stranice
- ◆ HEAD – pribavljanje zaglavlja web stranice
- ◆ POST – dopunjavanje web stranice
- ◆ PUT – smeštanje web stranice na poslužioca
- ◆ DELETE – uklanjanje web stranice
- ◆ TRACE – zahtev za eho od servera
- ◆ CONNECT – povezivanje preko zastupnika
- ◆ OPTIONS – upit o tome koje se metode i zaglavlja mogu koristiti sa datom stranicom

HTTP zahtev (GET)

- ◆ Pregledač može da zahteva Web stranicu od servera putem HTTP komande GET. To je jedan red teksta sledećeg oblika:
- ◆ GET *url*
- ◆ Gde je *url* URL stranice koja se zahteva. Napomene:
- ◆ (1) Pregledač prethodno koristi DNS da bi dobio IP adresu servera, a zatim sa njim uspostavlja TCP vezu, na prolazu 80 (ili drugom ako je zadat).
- ◆ (2) U prvoj GET komandi pregledač mora da zada apsolutni URL, a u narednim može koristiti relativni (navodi se samo putanja, bez imena računara).

HEAD metod

- ◆ Može se koristiti za pribavljanje informacija o stranici za potrebe indeksiranja ili proveru validnosti URL

POST metod

- ◆ Slanje podataka na server (sadržaj forme ili parametri poziva RPC procedure)

PUT metod

- ◆ Smeštanje stranice na server (inverzna operacija od GET)

DELETE metod

- ◆ Uklanjanje stranice sa servera

TRACE metod

- ◆ Koristi se za otkrivanje grešaka (debugging)
- ◆ Instrukcija serveru da uradi eho zahteva koji je dobio od klijenta

CONNECT metod

- ◆ Povezivanje klijenta sa serverom preko zastupnika

OPTIONS metod

- ◆ Upit serveru koje metode i zaglavlja podržava data stranica

Poruke o greškama

- ◆ U slučaju da server primi pogrešan zahtev, on pregledaču šalje odgovarajuću HTML stranicu. Npr.:

```
<HTML>
```

```
  <HEAD><TITLE>400 Greška</TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    <H1>Greška</H1>Primljen je neispravan zahtev.
```

```
  </BODY>
```

```
</HTML>
```

Izgled ove stranice na ekranu pregledača je sledeći:

Izgled stranice sa porukom o grešci:

Greška

Primljen je neispravan zahtev.

Kodovi grešaka (primeri u zagradi)

- ◆ 1xx – informacioni – (100 zahtev primljen na obradu)
- ◆ 2xx – uspešno izvršena operacija (200 zahtev izvršen)
- ◆ 3xx – preusmerenje (301 – stranica premeštena, 304 – keširana stranica i dalje validna)
- ◆ 4xx – greška klijenta (na primer neispravan zahtev)
- ◆ 5xx – greška servera (500 interna greška servera, 503 – pokušaj kasnije)

Trajne veze

- ◆ Prve verzije HTTP su, po ugledu na FTP, pretpostavljale uspostavu nove TCP veze za svaku Web stranicu.
- ◆ Klijent je otvarao TCP vezu i slao komandu GET, zatim je čitao podatke do oznake EOF (end of file), i na kraju zatvarao TCP vezu.
- ◆ HTTP verzija 1.1 se pojavila u obliku RFC-a u junu 1999, i uvela pojam *trajne* HTTP veze.
- ◆ Trajna veza je TCP veza koja je otvorena tokom više zahteva i odgovora.
- ◆ Trajne veze su pre svega smanjile dodatno opterećenje.

Prednosti trajnih veza

- ◆ (1) manji broj TCP veza
- ◆ (2) manje opterećenje mreže
- ◆ (3) manji resursi procesora (memorija i vreme)
- ◆ (4) pregledač može da pošalje više zahteva bez čekanja odgovora, slično kao kod klizajućeg prozora
- ◆ (5) tehnika višestrukih istovremenih zahteva je posebno korisna kad na stranici ima više slika, a pristup Internetu ima dovoljan kapacitet.

Nedostatak trajnih veza

- ◆ Nedostatak trajnih veza je što se mora znati početak i kraj svake stavke koja se šalje preko te veze.
- ◆ Rešenja sa zaglavljem i/ili začeljem ne dolaze u obzir pošto je potrebno prenositi bilo koji niz okteta.
- ◆ Iz tog razloga je usvojeno rešenje da svakom nizu okteta koji treba preneti prethodi podatak o dužini tog niza.
- ◆ Međutim, u skladu sa *zajedničkom spregom konvertora* (*Common Gateway Interface*, CGI), server može i dinamički da pravi Web stranice.
- ◆ U slučaju dinamičkih Web stranica, server u opštem slučaju ne zna unapred dužinu stranice.

Prenos dinamičkih Web stranica

- ◆ Čuvanje podataka u datoteci pre njihovog slanja ima 2 nedostatka:
 - ◆ (1) nepotrebno se troše resursi servera
 - ◆ (2) nepotrebno se odlaže njihov prenos.
- ◆ Radi podrške dinamičkim Web stranicama, HTTP omogućava da kada server ne zna dužinu stavke:
 - ◆ (a) o tome obavesti klijenta,
 - ◆ (b) zatim mu pošalje stavku, i
 - ◆ (c) na kraju raskida vezu.

Mehanizmi za generisanje dinamičkih web stranica

- ◆ CGI, PHP, JSP, ASPX
- ◆ CGI (Common Gateway Interface) je sprega koja omogućuje web serveru da razmenjuje informacije sa back-end programima (na primer upiti na baze podataka)
- ◆ Web server pokreće izvršenje programa u cgi-bin direktorijumu i prosleđuje mu sadržaj zahteva kao ulazni parametar. Izlaz programa se šalje web klijentu u obliku web stranice.

PHP, JSP, ASPX

- ◆ PHP (PHP: Hypertext Processor) primenjuje pristup u kome se skript ugrađuje u web stranice i izvršava na web serveru. Rezultat se prosledjuje klijentu u obliku web stranice. Ekstenzija datoteke sa PHP skriptom je .php.
- ◆ Sličan pristup primenjuje i JSP (JavaServer Pages), samo što koristi Java programski jezik umesto PHP. U tom slučaju ekstenzija datoteke je .jsp.
- ◆ ASP.NET (Active Server Pages .NET) je Microsoft verzija tog pristupa. U tom slučaju ekstenzija datoteke je .aspx.

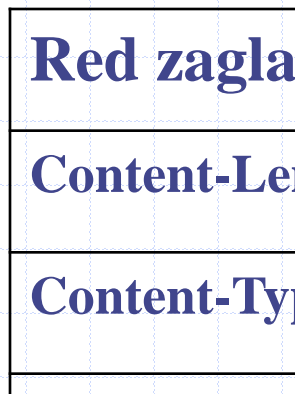
Oblik HTTP poruke

- ◆ HTTP koristi osnovni format elektronske pošte, format 822 i MIME proširenja.
- ◆ U skladu sa 822, svaka HTTP poruka sadrži zaglavlje, prazan red i stavku koja se šalje.
- ◆ Pored toga, svaki red zaglavlja sadrži:
 - ◆ (a) ključnu reč,
 - ◆ (b) znak dvotačke (":"), i
 - ◆ (c) vrednost.
- ◆ Neki mogući redovi zaglavlja i njihova značenja su dati u narednoj tabeli.

Primeri redova u zaglavlju HTTP poruke

(Atributi Content-Type i Content-Encoding potiču iz MIME.)

Red zaglavlja	Značenje
Accept	Content-types koji su prihvatljivi u odgovoru.
Accept-Charset	Karakter set koji je prihvatljiv za klijenta.
Accept-Encoding	Način kompresije prihvatljiv za klijenta.
Accept-Language	Prirodni jezik (ili jezici) sadržaja koji su prihvatljivi u odgovoru.



Red zaglavljac	Značenje
Content-Length	Dužina tela stavke u oktetima.
Content-Type	Tip sadržaja (MIME type).
Content-Encoding	Način kodiranja sadržaja.
Content-Language	Jezik (ili jezici) sadržaja.

Red zaglavlja	Značenje
Cache-Control	Uputstva u vezi keširanja za sve uređaje koji obavljaju keširanje na putanji između klijenta i servera.
Cookie	HTTP cookie koji je prethodno postavio server korišćenjem Set-Cookie zaglavlja
Max-Forwards	Maksimalan broj prosleđivanja preko servera zastupnika ili konvertora protokola.
Referer	Adresa web stranice sa koje je korisnik “došao” na trenutnu, prateći link

Red zaglavlja	Značenje
Via	Adresa proksija preko kog je zahtev došao do servera, odnosno odgovor do klijenta.
Age	Vreme koje je stranica “provela” u kešu nekog zastupnika
WWW-Authenticate	Specifikacija autentikacione šeme koja se mora koristiti za pristup nekom objektu.
Proxy-Authorization	Kredencijali koje šalje klijent za pristup zastupniku.

Red zaglavlja	Značenje
Proxy-Authenticate	Šalje zastupnik kada zahteva autentikaciju od strane klijenta.
Set-Cookie	Šalje server i sadrži HTTP cookie
Connection	Način korišćenja TCP veze.
Authorization	Kredencijali za HTTP autentikaciju, šalje klijent.

Primer prenosa HTML dokumenta

(Kod trajnih veza, atribut Content-Length je obavezan.)

Content-Length: 40

Content-Language: en

Content-encoding: ascii

<HTML> Ovo je jedan jednostavan dokument. </HTML>

Drugi tipovi redova u zaglavlju

- ◆ Pored već navedenih, postoje i drugi redovi zaglavlja koji služe za prenos meta informacije.
- ◆ Npr., kao što je već pomenuto, u slučaju da server ne zna unapred dužinu stavke, on na to upozorava klijenta, tako što umesto reda *Content-Lenght* navodi red:
- ◆ *Connection:close*
- ◆ Nakon prijema ovog upozorenja, pregledač ne sme da šalje dodatne zahteve.

Pregovaranje klijenta i servera

- ◆ Zaglavlje služi i za pregovaranje o mogućnostima klijenta i servera. Skup mogućnosti čine:
 - ◆ (1) osobine veze (npr. da li se koristi autentifikacija)
 - ◆ (2) oblici predstavljanja (npr. koji tipovi kompresije dolaze u obzir)
 - ◆ (3) sadržaji (npr. na kom jeziku moraju biti tekstualne datoteke), i
 - ◆ (4) kontrole (npr. vreme važenja stranice).
- ◆ Pregovaranje može biti pod kontrolom servera ili pod kontrolom pregledača.

Pregovaranje pod kontrolom servera

- ◆ Započinje zahtevom pregledača.
- ◆ Zahtev, uz URL, sadrži i spisak preferenci.
- ◆ Server bira svoje mogućnosti koje najviše odgovaraju preferencama klijenta.
- ◆ Npr. ako se neki dokument čuva na više jezika, a zahtev daje prednost engleskom, server šalje englesku verziju.

Pregovaranje pod kontrolom klijenta

- ◆ Ovde pregledač bira mogućnosti u dva koraka:
- ◆ (1) pregledač postavlja pitanje šta je raspoloživo
- ◆ (2) server vraća spisak, iz kog pregledač bira željenu stavku i šalje drugi zahtev da bi je obezbedio.
- ◆ Prednost: pregledač ima punu kontrolu nad biranjem.
- ◆ Pomoću atributa *Accept*, pregledač navodi prihvatljive oblike info. (medijumi) i načina njihovog predstavljanja, npr.:
- ◆ Accept: text/html,text/plain;q=0.5;text/x-dvi;q=0.8
- ◆ Napomena: parametar q je nivo prvenstva opcije.

Nivo prvenstva opcije (q)

- ◆ Ako se ne navede podrazumeva se $q=1$.
- ◆ Vrednost $q=0$ znači da taj tip nije prihvatljiv.
- ◆ Kod medijuma gde parametar kvaliteta, q , ima smisla (npr. zvuk), vrednost q se može tumačiti kao mera dozvoljenog smanjenja kvaliteta.
- ◆ Dalje, koristeći redove: *Accept-Encoding*, *Accept-Charset*, *Accept-Language*, pregledač može da navede koji način kodiranja, skup znakova, i jezik, je spreman da prihvati.

Uslovni zahtevi

- ◆ HTTP omogućava uslovne zahteve.
- ◆ Unutar zahteva navodi se uslov pod kojim zahtev treba ispuniti.
- ◆ Tako uslov *If-Modified-Since* omogućava izbegavanje prenosa stavke koja nije menjana nakon zadatog datuma, npr. zaglavlje:
 - ◆ If-Modified-Since; Sat, 01 Jan 2000 05:00:01 GMT
 - ◆ uz zahtev GET dovodi do prenosa stavke samo ako je ona menjana nakon 1.1.2000, 05:00:01, srednje vreme po Griniču.

Serveri zastupnici (proxy)

- ◆ Serveri zastupnici su važan deo Web arhitekture, pošto skraćuju vreme čekanja pregledača, kao i opterećenje servera.
- ◆ Ovi serveri nisu transparentni, već:
 - ◆ (a) pregledač uspostavlja vezu sa lokalnim serverom zastupnikom (proxy), a ne sa pravim Web serverom, i
 - ◆ (b) server zastupnik se konfiguriše tako da baferuje stranice Web servera.
- ◆ Npr. organizacija može da ima svoj lokalni server zastupnik, koji zadovoljava zahteve iz organizacije za istim Web stranicama.

HTTP podrška za servere zastupnike

- ◆ HTTP poseduje eksplicitnu podršku za ove servere:
- ◆ (a) kako zastupnik obrađuje svaki zahtev,
- ◆ (b) kako zastupnik tumači zaglavlja,
- ◆ (c) kako pregledač pregovara sa zastupnikom, i
- ◆ (d) kako zastupnik pregovara sa Web serverom.
- ◆ Pored toga, postoje redovi zaglavlja posebno namenjeni zastupnicima, kao npr.:
 - ◆ (a) red za potvrdu autentičnosti zastupnika,
 - ◆ (b) red u koji zastupnik unosi svoju identifikaciju, tako da pregledač dobija spisak svih svojih posrednika.

Ograničavanje broja posrednika

- ◆ HTTP omogućava serveru da kontroliše kako posrednici obrađuju pojedine stranice.
- ◆ Pomoću reda u zaglavlju *Max-Forwards*, server može da ograniči broj zastupnika, npr. na samo jednog zastupnika sledećim redom zaglavlja:
 - ◆ Max-Forwards: 1
 - ◆ Ili npr. može u potpunosti da zabrani zastupnike sa:
 - ◆ Max-Forwards: 0

Baferovanje

- ◆ Svrha baferovanja je da se poveća efikasnost.
- ◆ Nakon prvog pristupa stranici, ona se zapisuje na disk pretraživača i/ili zastupnika.
- ◆ Naknadni zahteve za istom stranicom moguće je skratiti uzimanjem njene kopije sa diska.
- ◆ Osnovno pitanje je: Koliko dugo treba čuvati kopiju?
- ◆ Prekratko vreme dovodi do neefikasnosti, a predugo do zastarevanja stranica.
- ◆ HTTP omogućava da se baferovanje kontroliše na dva načina.

Kontrola baferovanja

- ◆ (1) Server u odgovoru navodi detalje baferovanja (da li se stranica upošte može baferovati, da li zastupnik sme da je baferuje, s kim se baferovana stranica sme deliti, vreme zastarevanja, i ograničenja dozvoljenih transformacija).
- ◆ (2) Pregledač može ponovo zahtevati originalnu stranicu, tako što šalje zahtev za stranicom, sa zaglavljem u kom se zahteva da maksimalna "starost" ne bude veća od 0, što ni jedan zastupnik ne može da zadovolji. Web server ispunjava ovaj zahtev, a usputni zastupnici dobijaju svežu kopiju te stranice.