

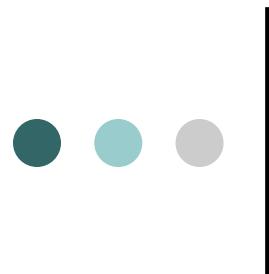


Programiranje korisničkih interfejsa:

Lekcija 4: Web sistem (II deo)

zima 2019/2020

Branimir M. Trenkić



Web sistem

HTTP-

*HyperText Transfer Protocol
(II deo)*

HTTP formati poruka

- HTTP predviđa **dva generalna tipa poruka** koje se razmenjuju između klijenta i servera:



format zahteva

3



format odgovora

- Struktura slična e-mail porukama
- HTTP poruke **nisu za ljudsko** korišćenje (**čitanje**)
- Dovoljno **izražajne** kako bi upravljale serverima, pretraživačima proxy-ima
- MIME** funkcionalnost

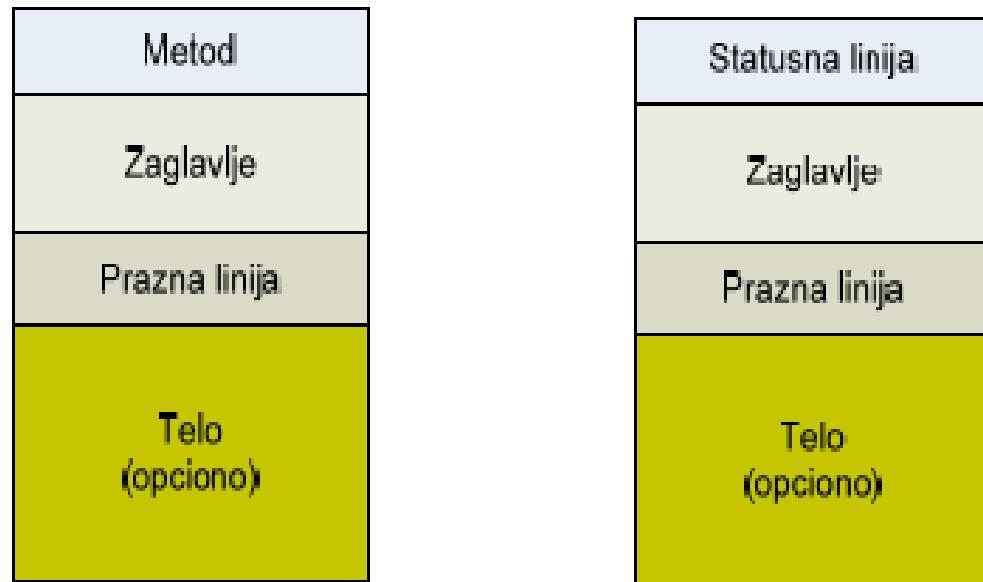
HTTP formati poruka

- **HTTP poruke** se sastoje od jedne ili više *linija* NVT ASCII *teksta*
- **Prva linija zahteva** sadrži metod ili *tip zahteva*
- **Prva linija odgovora** sadrži *informaciju o statusu odgovora*
- **Oba tipa** poruke **sadrže zaglavljे** koje od tela poruke odvojeno jednom praznom linijom



HTTP formati poruka

- **Telo poruke** je u oba slučaja **opciono** (ne mora da postoji) i koristi se **za prenos sadržaja poruke**
- Počev od verzije HTTP/1.1 **podržava MIME kodiranje sadržaja**



HTTP formati poruka

- **HTTP** je zamišljen opštije od prostog protokola za komunikaciju **tipa zahtev/odgovor**
- **HTTP** predviđa više različitih tipova zahteva koje klijent može da uputi serveru
- U terminologiji HTTP standarda **tipovi zahteva** se zovu metode

```
METHOD /path-to-resource HTTP/version-number  
Header-Name-1: value  
Header-Name-2: value  
  
[optional request body]
```

HTTP formati poruka

- Na primer, ako je zahtevani URL

http://www.mywebsite.com/sj/index.html

- **HTTP zahtev** može biti oblika:

```
GET /sj/index.html HTTP/1.1
```

```
Host: www.mywebsite.com
```

HTTP formati poruka

- Nakon projema zahteva server generiše **odgovor**
- **HTTP odgovor** je sledeće strukture:

```
HTTP/version-number status-code explanation
Header-Name-1: value
Header-Name-2: value

[response body]
```

- **Statusna linija** sadrži:
 - oznaku **HTTP verzije**,
 - tro-cifreni **statusni kod** i
 - kratko čitljivo **objašnjenje statusnog koda**



HTTP formati poruka

- Sledi uprošćena ***forma jednog HTTP odgovora***

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 9934
...
<HTML>
<HEAD>
<TITLE>SJ's Web Page</TITLE>
</HEAD>
<BODY bgcolor="#ffffff">
<H2 align="center">Welcome to Sviergn Jiernsen's Home Page</H2>
...
</H2>
</BODY>
</HTML>
```

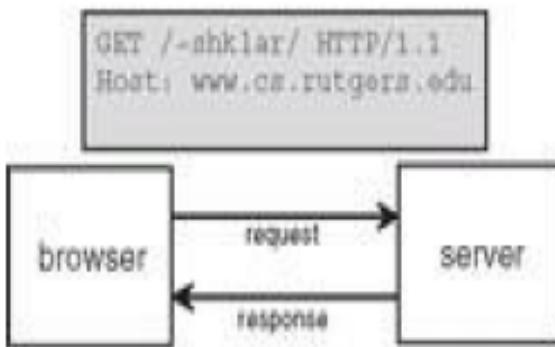


HTTP formati poruka

- Na primer, neka klijent zahteva resurs
<http://www.cs.rutgers.edu/~shklar/>
- Pretraživač šalje **GET zahtev** ka serveru, koji vraća odgovor
- Pretraživač prikazuje rezultat ove komunikacije
- Dalje, pretraživač formira **novi GET zahtev** za sliku i šalje ga serveru
- Server šalje sliku klijentu – pretraživač je prikazuje na monitoru

HTTP formati poruka

- 1. korak – inicijalni zahtev <http://www.cs.rutgers.edu/~shkclar/>



```
GET /~shkclar/ HTTP/1.1  
Host: www.cs.rutgers.edu  
  
HTTP/1.1 200 OK  
Content-Type: text/html  
  
<HTML>  
<HEAD> ... </HEAD>  
<BODY>  
...  
<IMG src="./images/photo.gif">  
...  
</BODY>  
</HTML>
```

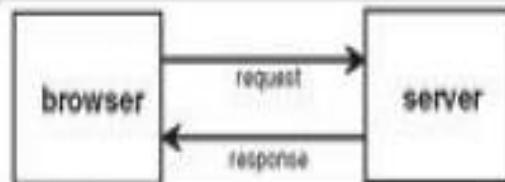


HTTP formati poruka

- 2. korak – sledeći zahtev za

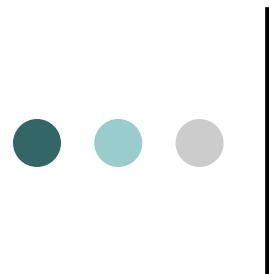
"<http://www.cs.rutgers.edu/~shkclar/images/photo.gif>"

```
GET /~shkclar/images/photo.gif HTTP/1.1
Host: www.cs.rutgers.edu
```



```
HTTP/1.1 200 OK
Content-Type: image/gif
```





HTTP metode

- Osim osnovnog ***metoda GET*** koji služi za **pribavljanje Web stranice** - HTTP predviđa još ***nekoliko dodatnih metoda***

HTTP metode

HTTP metode:

osnovne

Metod	Opis
GET	Zahtev za čitanjem Web stranice
HEAD	Zahtev za čitanjem zaglavlja Web stranice
PUT	Zahtev za prenos Web stranice na server
POST	Dodavanje sadržaja imenovanom resursu (npr. Web stranici)
DELETE	Zahtev za brisanjem Web stranice na serveru
TRACE	Echo poslatog zahteva
CONNECT	Rezervisano za neku buduću namenu
OPTIONS	Upit koji se odnosi na neke parametre servera



HTTP metode

Metod GET

- Zahteva od servera da **pošalje traženu stranicu**, tj. fajl
 - Pod stranicom se misli na **objekat** koji može biti HTML **stranica**, **slika**, **Java applet** i sl.
- **Stanica koja se vraća ne mora** da sadrži samo **ASCII tekst**, a **za identifikaciju tipa** sadržaja **koristi se MIME**, slično kao kod e-mail protokola
- Najveći broj HTTP zahteva koji se javljaju na Web-u su upravo tipa GET



HTTP metode

Metod GET

- **Uobičajeni oblik** metoda GET je:

GET /putanja HTTP/1.1

gde je,

putanja putanja iz URL-a resursa (fajla) koji se
traži, a

1.1 verzija HTTP protokola koji se koristi

- **GET zahtev ne sadrži telo**



HTTP metode

Metod HEAD

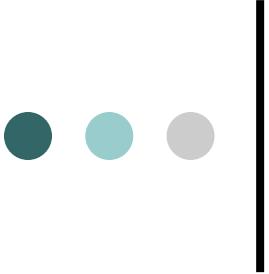
- Zahteva od servera isti sadržaj (kao GET), ali bez tela odgovora, **već samo njeno zaglavljje**
- Ovaj metod se može koristiti za:
 - pribavljane **informacije** o datumu i vremenu kada je stranica **poslednji put modifikovana**,
 - prikupljanje **drugih informacija** o stranici i njenom sadržaju ili
 - **testiranje validnosti URL-a** (da li je na datom URL-u prisutna stranica)



HTTP metode

Metod **PUT**

- **Suprotan metodu *GET*** - umesto čitanja, ovaj metod upisuje stranicu na Web server
- Uz pomoć ovog metoda moguće je **postaviti Web stranicu na udaljeni Web server**
- Stranica je sadržana u telu PUT zahteva
- **Stanica ne mora biti tekst**, **MIME tip** naveden je u polju **Content-Type zaglavlja** zahteva
- Zaglavljje zahteva sadrži i podatke za autorizaciju kojima klijent dokazuje da ima pravo da izvrši zahtevanu operaciju



HTTP metode

Metod POST

- Sličan metodu PUT
- Za razliku od **PUT**, kojim se na server **postavlja nova stranica** - **metod POST** se koristi **da se u resurs** (u najopštijem smislu) koji se nalazi na datom URL-u, **upišu (dodaju) novi podaci**
- **Postavljanje nove poruke na nekom Web forumu**, primer je ovog tipa operacije
- U praksi metod POST se mnogo više koristi nego PUT metod



HTTP metode

Metod **DELETE**

- Služi **za brisanje (uklanjanje) stranice** sa Web servera
- Kao i kod PUT, u zaglavljima zahteva tipa DELETE moraju postojati **podaci za autorizaciju**

Metod **OPTIONS**

- Omogućava klijentu da **postavi upit serveru koji se odnosi na izvesne parametre rada servera** ili parametre nekog **konkretnog fajla**



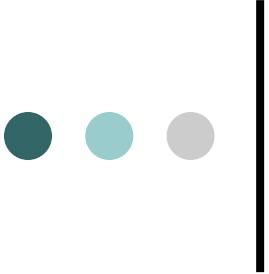
HTTP metode

Metod TRACE

- Koristi se za **testiranje veze** sa serverom
- Ovaj metod **nalaže serveru** da nazad **vrati primljenu poruku** zahteva
- Koristan je **u slučajevima** kada se **zahtevi ne obrađuju korektno**, a klijent želi da sazna da li zahtevi uopšte dolaze do servera

Metod CONNECT

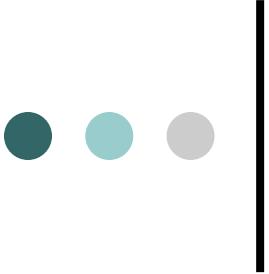
- **Nema definisanu namenu**, već je rezervisan za neku eventualnu buduću primenu



HTTP statusi

- U prvoj liniji svaki odgovor kojeg server vraća klijentu sadržan je ***trocifreni kôd statusa***
- Klijentu treba da ukaže da li je njegov zahtev **uspešno opslužen** ili **nije** i ako nije **zašto nije**

Kôd	Značenje	Primer
1xx	Informativno	100 - server pristaje da obradi klijentov zahtev
2xx	Uspešno	200 - zahtev je prihvaćen i obrađen; 204 - odgovor ne sadrži telo
3xx	Preusmeravanje	301 - stranica je premeštena na drugu URL lokaciju; 304 - keširana stranica je još uvek validna.
4xx	Greška klijenta	403 - zabranjen pristup stranici; 404 - stranica nije pronađena.
5xx	Greška servera	500 - interna greška servera; 503 - kasnije pokušaj ponovo



HTTP statusi

- **Prva cifra** statusnog kôda
- Služi za podelu odziva na **pet glavnih grupa**
- **Kôd 1xx** (x - proizvoljna cifra) – ***retko se koristi*** u praksi (informativno značenje)
- **Kôd 2xx** - zahtev je **uspešno obrađen** i da poruka sadrži traženi sadržaj
- **Kôd 3xx** - govori klijentu da izvrši **dodatne akcije**
 - Traženu stranicu ***potraži na nekom drugom URL-u*** ili u svom kešu – ***status redirekcija***



HTTP statusi

- **Prva cifra** statusnog kôda
- **Kôd 4xx - zahtev nije opslužen, problemi na strani klijenta** bilo zato što je **u** samom **zahtevu uočena greška** ili zato što je klijent zatražio **nepostojeću stranicu**
- **Kôd 5xx - na strani servera postoje problemi**, bilo zbog **greške u programu** Web servera bilo zato što je server **privremeno preopterećen**



HTTP zaglavlja

- Nakon linije metoda u poruci zahteva sledi jedna ili više linija zaglavlja koje sadrže dodatne informacije o zahtevu
- Za **svaku liniju** ove sekcije se kaže da predstavlja **jedno zaglavlje** zahteva
- Poruka odziva (odgovora) takođe može **sadržati zaglavlja** (jedno ili više)
- Neka zaglavlja se koriste i kod zahteva i kod odgovora



HTTP zaglavljia

- Čine ***metapodatke*** za sadržaj poruke
- Pametnom ***upotrebom zaglavija*** moguće je ***konstruisati složene aplikacije***, koje:
- ***Podržavaju sesiju*** komunikacije
- Uspostavljaju ***način keširanja***
- Upravljuju ***autentifikacijom i autorizacijom***
- Implementiraju ***poslovnu logiku***



HTTP zaglavljia

- Postoje sledeći oblici zaglavljia:
 - 1) opšta zaglavlja,
 - 2) zaglavlja zahteva,
 - 3) zaglavlja odgovora i
 - 4) zaglavlja entiteta

HTTP zaglavljia

- **Opšta zaglavija.** Primenjuju se i u zahtevima i u odgovorima
 - **Ne opisuju telo poruke**

```
Date: Tue, 29 Apr 2008 22:28:31 GMT
```

```
Connection: close ←
```

```
Warning: Danger, Will Robinson!
```

kompatibilnost unazad

- **Zaglavja zahteva.** Omogućuju klijentima da pošalju **dodatne informacije serveru** o sebi ili samom zahtevu

```
User-Agent: Mozilla/5.0 (Windows; U; Windows XP; en-US; rv:1.8.0.11)
```

```
Host: www.neurozen.com
```

```
Referer: http://www.cs.rutgers.edu/~shklar/index.html
```

```
Authorization: Basic [encoded-credentials]
```

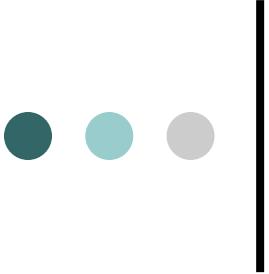
HTTP zaglavljia

- **Zaglavljia odgovora.** Omogućuju serverima da pošalju **dodatne informacije** o odgovoru koje **ne mogu da se zaključe iz statusne linije**

```
Location: http://www.mywebsite.com/relocatedPage.html
```

```
WWW-Authenticate: Basic realm="KremlinFiles"
```

```
Server: Apache/2.2.4
```



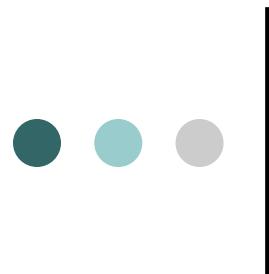
HTTP zaglavljia

- **Zaglavljia entiteta.** Opisuju ili *tela poruka* ili (u slučaju poruka zahteva) *ciljane resurse*

```
Content-Type: mime-type/mime-subtype  
Content-Length: xxxx  
Last-Modified: Tue, 29 Apr 2008 22:28:31 GMT
```

HTTP zaglavljia

Zaglavje	Tip	Sadržaj
User-Agent	Zahtev	Informacija o pretraživaču i platformi
Accept	Zahtev	Tip stranica koje klijent može da procesira
Accept-Charset	Zahtev	Skup karaktera koji je prihvatljiv za klijenta
Accept-Encoding	Zahtev	Kodiranje stranice koje klijent može da procesira
Accept-Language	Zahtev	Prirodni jezik koji klijent može da procesira
Host	Zahtev	DNS ime servera
Authorization	Zahtev	Podaci za autorizaciju klijenta
Cookie	Zahtev	Sadrži <i>cookie</i> kojeg je server prethodno poslao klijentu
Date	Zahtev/Odgovor	Datum i vreme slanja poruke
Server	Odgovor	Opšte informacije o serveru
Content-Encoding	Odgovor	Način kodiranja stranice
Content-Language	Odgovor	Prirodni jezik korišćen na stranici
Content-Length	Odgovor	Veličina stranice u bajtovima
Content-Type	Odgovor	MIME tip stranice
Last-Modified	Odgovor	Datum i vreme poslednje promene stranice
Location	Odgovor	Instrukcija klijentu da zahtev pošalje na neko drugo mesto
Set-Cookie	Odgovor	Sadrži <i>cookie</i> kojeg server šalje klijentu



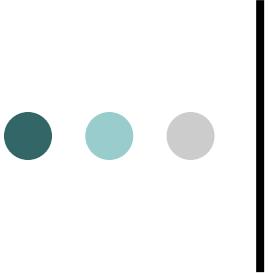
HTTP zaglavlja

- Zaglavje **User-Agent** (korisnički agent) omogućava da klijent obavesti server o
 - *tipu* pretraživača koji koristi,
 - ***operativnom sistemu*** i
 - drugim osobinama
- Četiri zaglavlja koja počinju sa **Accept** obaveštavaju server **kakav sadržaj je klijent spremam da prihvata**
 - **Prvo** od ovih zaglavlja (Accept) **navodi MIME tip stranice** koji klijentov pretraživač može da obradi (npr. ***text/html***)



HTTP zaglavljia

- **Drugo** (*Accept-Charset*) definiše **skup karaktera** (npr. ISO-8859-1 ili Unicode-1-1) **koji klijent prepoznaće**
- **Treće** (*Accept-Encoding*) definiše **metod kompresije** koji klijent podržava (npr. gzip)
- **Četvrto** (*Accept-Language*) ukazuje na prirodni **jezik** (npr. Srpski) **koji korisnik razume**



HTTP zaglavljia

Pregovaranje oko sadržaja (*content negotiation*)

- Ako server ima mogućnost izbora stranice (npr. postoji više varijanti iste stranice) - on će izabrati i vratiti klijentu onu **koja se uklapa u postavljene zahteve**
- Ako server nije u mogućnosti da udovolji zahtevima klijenta - **vratice odgovor** sa postavljenim odgovarajućim kodom **greške**



HTTP zaglavlja

- Zaglavlje **Host** sadrži **domensko ime servera**, preuzeto iz URL-a (opciono – TCP port „osluškivanja“ servera)
- Ovo zaglavlje je **obavezno** jer se može desiti da za istu IP adresu budu vezana **više DNS imena**
- Ispitivanjem sadržaja ovog polja, **sever proverava da li se zahtev odnosi na njega**
- Zaglavlje **Authorization** je neophodno za **stranice koje su zaštićene** i za koje je klijent u obavezi da dokaže da ima pravo da vidi stranicu



HTTP zaglavljia

- **Cookie**-ima su posvećena **dva zaglavlja**
- Preko zaglavla **Cookie klijent vraća serveru sadržaj cookie-a** kojeg je ranije poslat klijentu od strane neke mašine **iz domena servera**
- **Server šalje cookie klijentu** u obliku sadržaja zaglavla **Set-Cookie**
- Kao što znamo, klijent je u obavezi da **zapamti cookie na svoj hard disk**, i vrati ga serveru pri svakom narednom obraćanju (zahtevu)



HTTP zaglavljia

- Zaglavje Date (datum) se može koristiti **u oba smera** i sadrži **vreme i datum** kada je poruka poslata
- Sledi zaglavljia koja se javljaju isključivu u odgovorima koje server šalje klijentu
- Zaglavje Server omogućava serveru da **saopšti svoj identitet klijentu** (aplikaciju koja se koristi u obradi zahteva)



HTTP zaglavljia

- Sledеćа **četiri zaglavlja**, koja počinju sa **Content-** (sadržaj)

Type

Charset

Encoding

Language

- omogućavaju serveru da **opiše osobine stranice** koje šalje
 - **Značenje ovih zaglavljia je analogno odgovarajućim Accept- zaglavljima**



HTTP zaglavljia

- Zaglavljje **Last-Modified** sadrži **datum i vreme** kada je stranica poslednji put **modifikovana**
- Ovo zaglavljia ima bitnu **ulogu u keširanju stranica**



HTTP zaglavljia

- **Server koristi** zaglavje **Location** kada želi da **obavesti klijenta** da bi trebalo da pokuša **da potraži** zahtevanu **stranicu na nekom drugom URL-u** – t.j. na delu je **redirekcija**
- Ova mogućnost se koristi:
 - ako je **stranica premeštena** na drugu lokaciju
 - ako **više od jednog URL-a** ukazuje **na istu stranicu**



HTTP zaglavljа

- Zaglavlje ***Location.***
- ***Na primer,*** neka ***internacionalna kompanija*** može nakon prijema zahteva za njenu ***glavnu stranicu*** na ***.com*** domenu, da preusmeri klijenta, na osnovu njegove IP adrese, na jednu od svojih ***nacionalnih ili regionalnih stranica***



HTTP zaglavlja

- Zaglavlje **Location**.
- Interesantan **primer redirekcije** će se desiti ako klijent formira URL koji specificira direktorijum ali **izostavi završni sleš (/)**
- Ako se želi pristup Petrovićevom web- sajtu na VISER-u, potrebno je uneti URL:
<http://www.viser.edu.rs/~petrovic> u pretraživač
- To će rezultovati u HTTP zahtev:

```
GET /~petrovic HTTP/1.1
Host: www.viser.edu.rs
```



HTTP zaglavljа

- Zaglavlje ***Location.***
- U tom slučaju, server je projektovan da vrati ili sadržaj datoteke indeks.html ili ako ovaj fajl ne postoji – listing sadržaja direktorijuma
- Ali, pre svega, ***redirekciju*** sa
<http://www.viser.edu.rs/~petrovic>
na
<http://www.viser.edu.rs/~petrovic/>
- Kako bi pretraživač formirao pravilan zahtev

HTTP zaglavlja

- Zaglavljje ***Location.***

```
HTTP/1.1 301 Moved Permanently
Location: http://www.viser.edu.rs/~petrovic/
Content-Type: text/html

...
<HTML>
<HEAD><TITLE>301 Moved Permanently</TITLE></HEAD>
<BODY id="c03-body-0005">
<H1>301 Moved Permanently</H1>
The document has moved
<A href="http://www.cs.rutgers.edu/~shklar/">
here</A>.
</BODY>
</HTML>
```

Primer HTTP odgovora

HTTP/1.1 200 OK

status odgovora (pozitivan)

Date: Wed, 08 May 2002 22:54:22 GMT

Server: Apache/1.3.20 (Unix) mod_ssl/2.8.4 OpenSSL/0.9.5a

Last-Modified: Mon, 11 Sep 2000 13:56:29 GMT

ETag: "2a79d-c8b-39bce48d"

veći broj zaglavja
Zaglavlj je ETag

Accept-Ranges: bytes

Content-Length: 3211

Content-Type: text/html

prazna linija

<html>

<head>

<title>IETF RFC Page</title>

<script language="javascript">

function url() {

var x = document.form1.number.value

if (x.length == 1) {x = "000" + x }

if (x.length == 2) {x = "00" + x }

if (x.length == 3) {x = "0" + x }

document.form1.action = "/rfc/rfc" + x + ".txt"

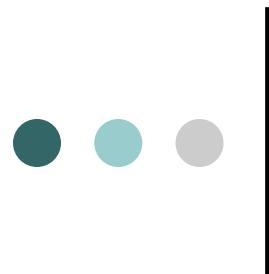
document.form1.submit

}

</script>

</head>

Web stranica

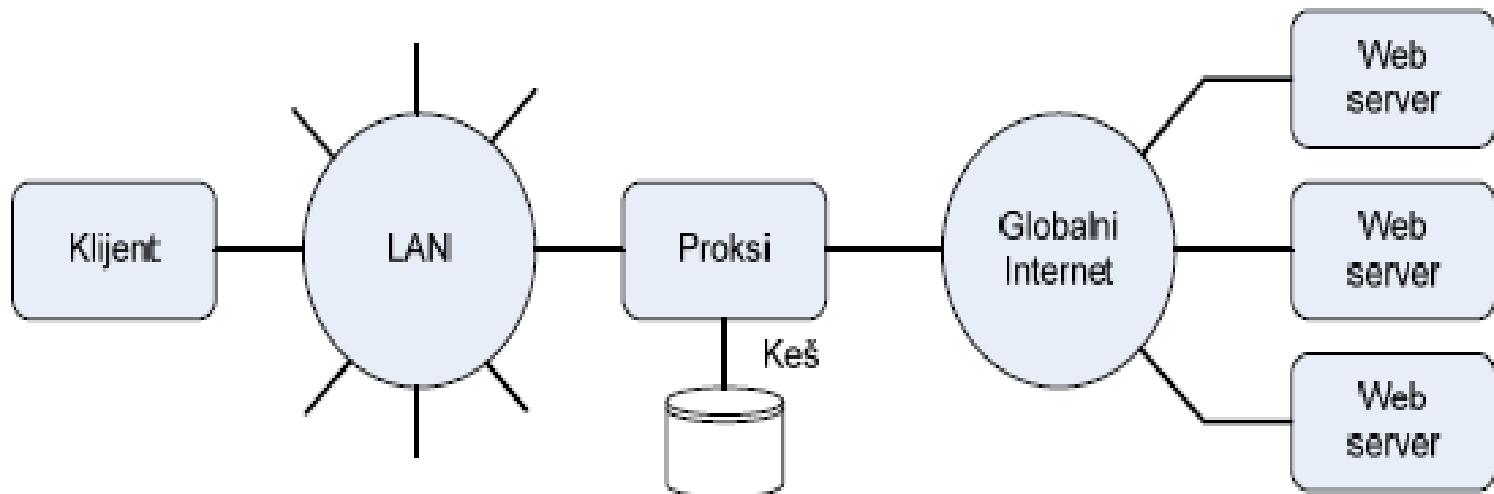


Proksi serveri i keširanje

- U dosadašnjem izlaganju o Web-u *prepostavljali smo* da ***klijent i server direktno komuniciraju*** razmenom HTTP poruka preko Interneta
- Međutim, u izvesnim slučajevima ***komunikacija klijent-server ne mora biti direktna***
- Može se ostvarivati **posredstvom** jednog ili više **među-servera**
- *Dva tipa* među-servera su: ***firewall*** i ***proxy server***

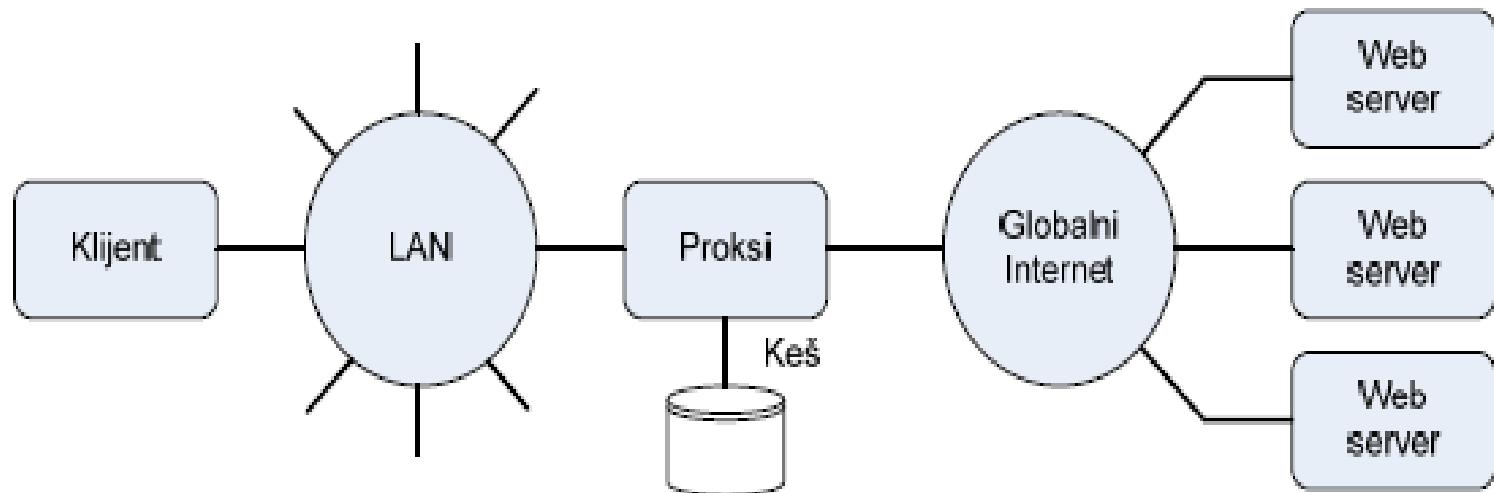
Proksi serveri i keširanje

- **Proxy server** je **posrednik** između **lokalnih (intranet) korisnika** i **Web servera**
- Omogućava **optimizaciju** kojom se **smanjuje čekanje** klijenata na pribavljanje zahtevanih Web stranica



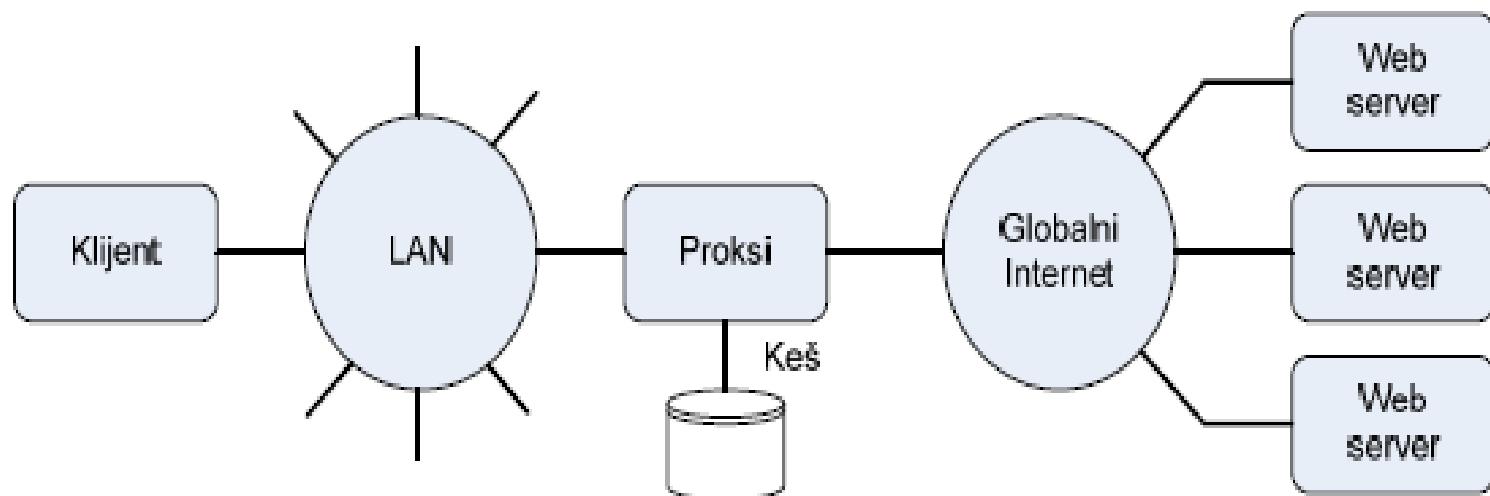
Proksi serveri i keširanje

- **Web pretraživači** u mreži koja koristi proxy server su konfigurisani tako da svoje HTTP zahteve ne upućuju direktno udaljenim Web serverima već ih **šalju lokalnom proxy serveru** koji u njihovo ime obavlja zahtevanu **transakciju**



Proksi serveri i keširanje

- Kada **prvi korisnik** u firmi (LAN-u) pristupi određenoj Web stranici, **proxy** mora da **pribavi kopiju od servera** na kome se stranica nalazi
- **Proxy ostavlja kopiju u svom kešu** i vraća traženu stranicu kao odgovor na zahtev





Proksi serveri i keširanje

- Kada **sledeći put** neki **korisnik želi da pristupi istoj stranici**, **proxy uzima podatke iz svog keša** i ne šalje zahtev preko Interneta
- Proxy serveri su **bitan deo arhitekture Web sistema**
- Osim što efektivno **skraćuju vreme** pribavljanja Web stranica, proxy serveri značajno
 - a) **redukuju saobraćaj na Internetu** i
 - b) **smanjuju opterećenje** Web servera

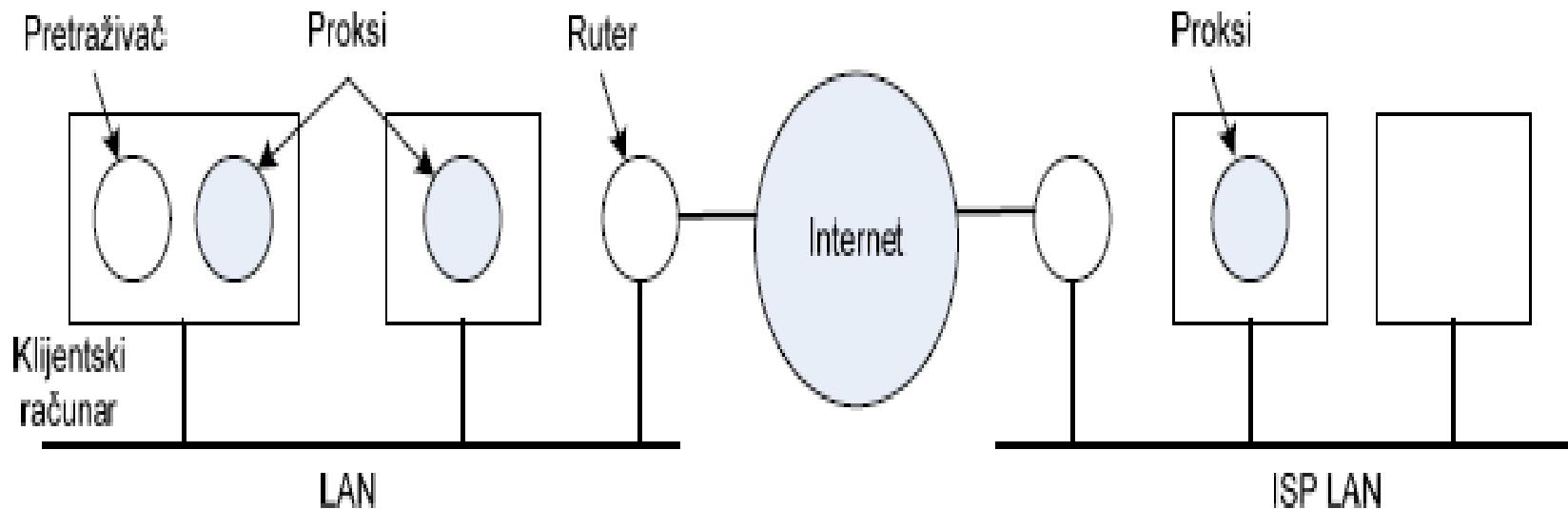


Proksi serveri i keširanje

- **Koncept proxy** servera i **keširanja** nije ograničena samo na **korporacijski intranet**, već se primenjuje i u drugim kontekstima
- Na primer, **proxy server** ne mora biti računar na lokalnoj mreži, već može biti i proces pokrenut na PC računaru (klijentu)
- Takođe, **mnogi provajderi** internet usluga (ISP) **poseduju proxy server** sa ciljem **da svojim korisnicima ubrzaju pristup Web-u** (i u isto vreme smanje protok podataka prema nadređenom ISP-u)
Šifra

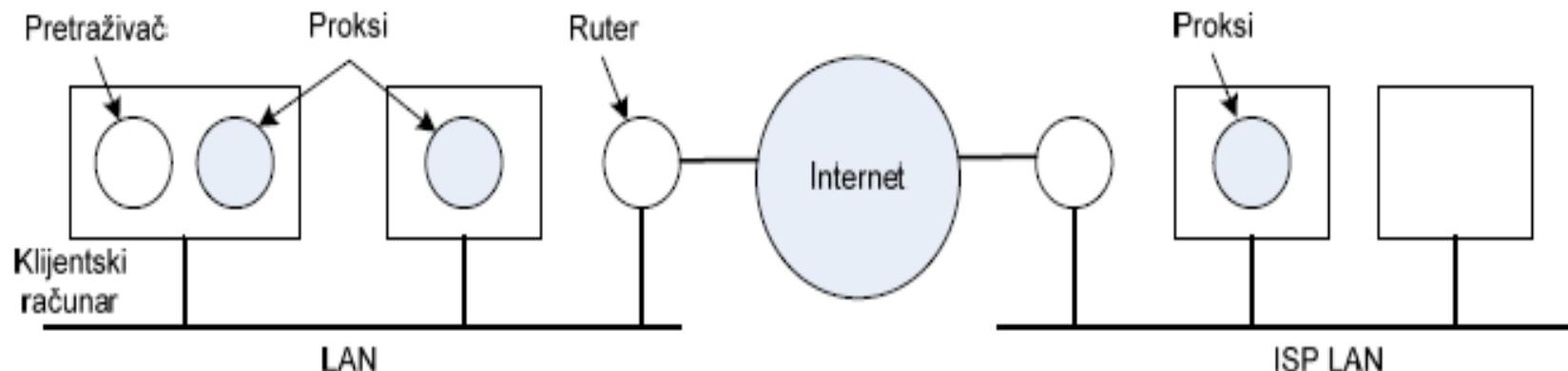
Proksi serveri i keširanje

postoji **hijerarhija proxy servera:**



• • • | Proksi serveri i keširanje

- **Zahtev** se **najpre** šalje **lokalnom proxy serveru**, koji, ako nije u stanju da opsluži zahtev, zahtev prosleđuje npr. **korporacijskom proxy serveru**, a ovaj **proxy serveru provajdera** internet usluga
- ... i tako redom sve **dok se u nekom kešu ne pronađe tražena stranica**





Proksi serveri i keširanje

- Tek ***ako stranica ne postoji u kešu proxy servera na vrhu hijerarhije*** –
ona se ***direktно tražи od Web servera***, a onda
prosleđуje nazad do pretraživačа koji je
uputio zahtev i
pri tome ***pamti u keševima svih posrednih proxy servera!***



Proksi server i HTTP

- **HTTP** sadrži eksplicitnu podršku za proxy servere
- **Protokol** tačno **određuje način:**
 - na koji proksi **obrađuje** svaki **zahtev**,
 - kako proksi treba da **tumači zaglavlje**,
 - kako pretraživač **pregovara sa proksijem** i
 - kako proksi **pregovara sa serverom**



Proksi server i HTTP

- *HTTP* sadrži eksplicitnu podršku za proxy servere
- Nekoliko *HTTP zaglavlja* je posebno *namenjeno za proksije*
- Na primer, HTTP *omogućava serveru* da *kontroliše kako proksiji obrađuju* svaku Web stranicu



Proksi server i HTTP

- **HTTP** sadrži eksplicitnu podršku za proxy servere
- Server može u odgovoru da uključi zaglavje ***Max_Forwards*** i tako **ograniči broj proksija koji obrađuju** stavku pre nego što se ona isporuči pretraživaču
- Ako server odredi samo jedan, ***Max_Forwards:*** 1, na putanji od servera do pretraživača ***dozvoljen je samo jedan proksi***
- Nula znači da je zabranjeno da proksi obrađuju stavku



Keširanje

- Eliminisanjem nepotrebnih prenosa, *proxy* keš smanjuje i vreme čekanja i mrežni saobraćaj
- **Glavni aspekt** keširanja jeste **privremeno čuvanje stranica**, a
- **Glavno pitanje** tiče se vremena čuvanja stranice, tj. **koliko dugo treba čuvati stavku u kešu**
- Ako se kopija predugo čuva u kešu - ona **može da zastari**, što se dešava ako je original u međuvremenu, nakon što je kopija uneta u keš, promenjen



Keširanje

- Ako se kopija ne čuva dovoljno dugo, dolazi do **smanjenja efikasnosti keširanja** zato što sledeći zahtev mora nepotrebno da ide do servera
- Pojedine Web stranice su **podložne čestim promenama** (npr. stranica sa rezultatima fudbalskih utakmica)
- Druge mogu ostati **neizmenjene u dužem vremenskom intervalu** (npr. stranica posvećena Grčkoj mitologiji)



Keširanje

- Sklonost stranice promenama može da **varira u vremenu**
- Neke stranice se ni u kom slučaju ne mogu keširati
 - Dinamičke Web stranice koje se generišu na strani servera na osnovu postavljenog upita
- HTTP dozvoljava da **server (proxy) kontroliše keširanje na različite načine**



Keširanje

Prvi način

- Oslanja se na informaciju ***iz zaglavlja odgovora Last-Modified*** kada određuje ***koliko dugo će stranica biti čuvana u kešu***
- Ako je stranica koja se upravo stavlja u keš promenjena pre jednog sata - ona će biti čuvana u kešu jedan sat
- Ako je promenjena pre dva meseca - u kešu će ostati dva meseca



Keširanje

Prvi način

- Ovaj pristup često ***dobro radi u praksi***,
- Zasnovan na predviđanjima
- Zato **ne garantuje** da će pretraživaču uvek biti vraćena **ažurna kopija stranice**



Keširanje

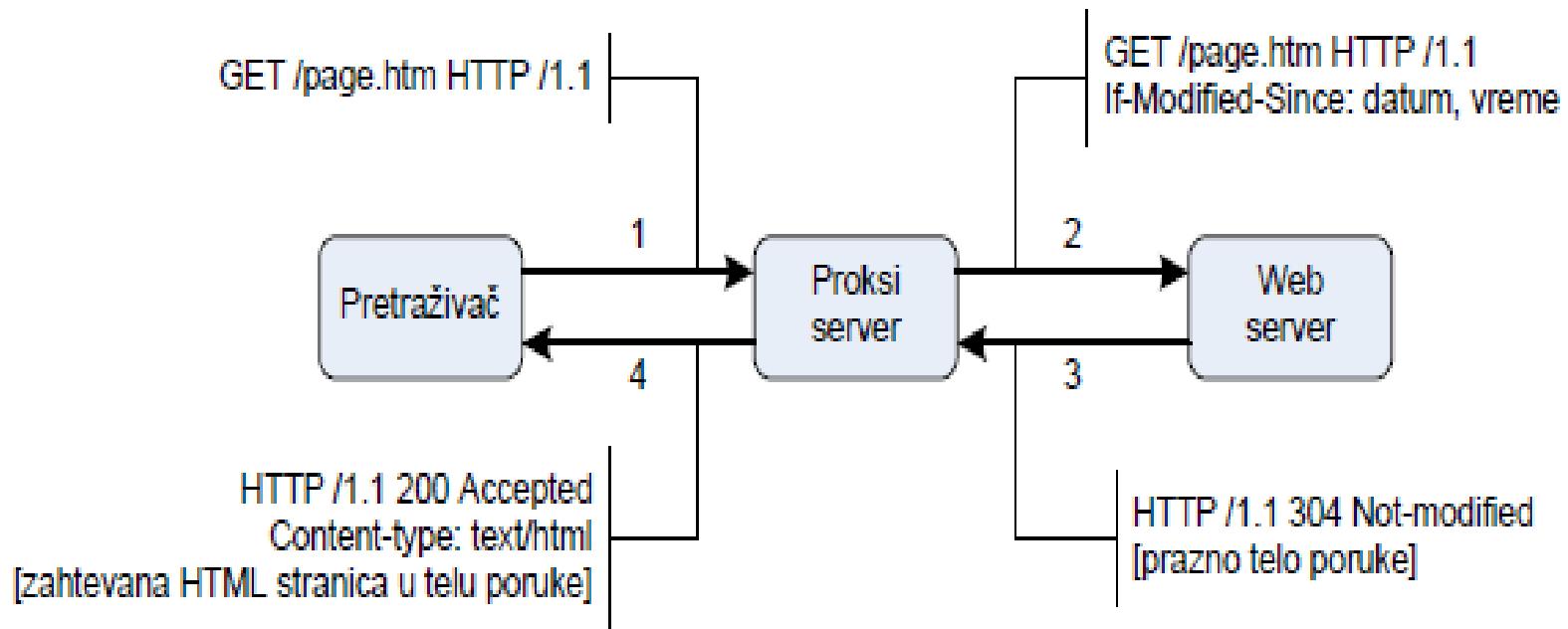
Drugi način

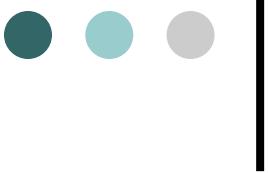
- **Eliminiše mogućnost** da pretraživač dobije zastarelu strancu
- **Na račun** izvesnog povećanja saobraćaja i vremena čekanja na pribavljanje stranice
- Pristup se oslanja na tzv. uslovni GET zahtev, koji proxy može da pošalje serveru kako bi proverio ažurnost keširane stranice

Keširanje

Drugi način

- **Uslovni GET zahtev** je HTTP poruka koja sadrži **zaglavljje If-Modified-Since** ("ako je modifikovana posle ...").





Keširanje

- **Prva dva pristupa** za kontrolu vremena keširanja se lako **mogu kombinovati**
- Na primer, **prvih T sekundi** nakon pribavljanja stranice proxy vraća pretraživačima keširanu kopiju bez postavljanja pitanja serveru
- **Po isteku T sekundi**, proxy **koristi uslovnu GET poruku** za proveru ažurnosti kopije



Keširanje

Keširanje pod kontrolom izvornog servera

- **Web server** ili serverska strana aplikacije su **najboljoj poziciji** da **donose odluku** da li neki sadržaj treba keširati ili ne
- Koriste se **namenska zaglavlja odgovora**
- **Cache-Control** (od HTTP/1.1)
- Može imati sledeće vrednosti:
 - **public** (dozvoljeno keširanje odgovora)
 - **private** (nije dozvoljeno keširanje)
 - **no-cache** (nije dozvoljeno keširanje odgovora)

Keširanje

Keširanje pod kontrolom izvornog servera

- **Pragma** zaglavlj (HTTP/1.0)
 - **Pragma: no-cache** (default- vrednost)
- Kompatibilnost rešenja – kombinovanje oba zaglavlja:

```
HTTP/1.1 200 OK
Date: Wed, 30 Apr 2008 03:26:18 GMT
Server: Apache/2.2.4
Last-Modified: Wed, 30 Apr 2008 03:25:36 GMT
Cache-Control: private
Pragma: no-cache
Content-Length: 2255
```

```
Content-Type: text/html
```

```
<HTML>
```

```
...
```

```
</HTML>
```



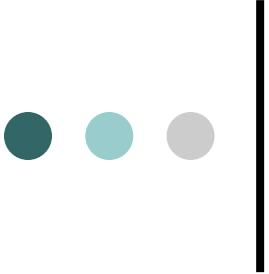
Keširanje

- *Još jedan način* za *poboljšanje performansi keširanja* naziva se *proaktivnim keširanjem*
- Kada *proksi* pribavi stranicu od servera, on je analizira i *izdvaja sadržane hiperuze*
- Nakon toga, *proksi može da pribavi i smesti u svoj keš stranice na koje ukazuju izdvojene hiperuze*, za slučaj da korisnik naknadno zatraži neku od njih



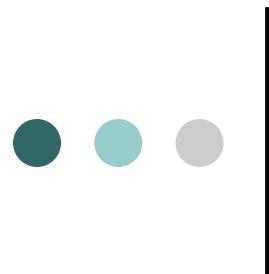
Keširanje

- Na ovaj način ***moguće je skratiti vreme pristupa za buduće zahteve*** za slučaj da korisnik izabere neki od linkova na upravo učitanoj stranici
- Međutim, pribavljajući i stranice koje nikada neće biti potrebne, ***ova tehnika*** ne smanjuje već ***povećava saobraćaj na Internetu***



Firewall

- Danas se na većini lokalnih, **korporacijskih mreža** koristi **isti skup protokola** kao i na Internetu (**TCP/IP**) – takve mreže se nazivaju **intranetima**
- Korišćenje Internet protokola na intranet mreži ima dvojako opravdanje
 - A) Olakšan je **pristup Web-u** od strane računara povezanih na intranet
 - lokalni računari na isti način komuniciraju međusobno kao i sa udaljenim serverima



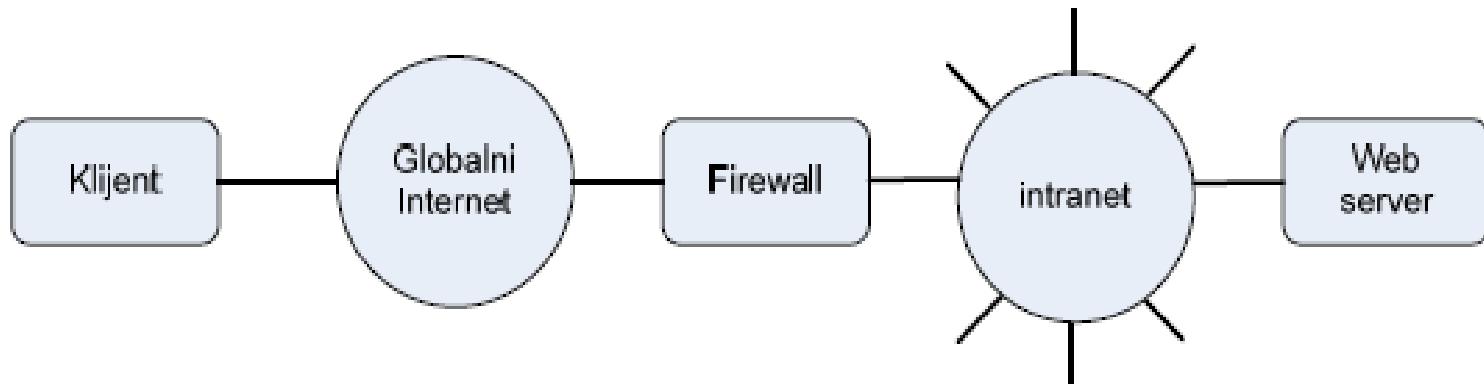
Firewall

B) Omogućeno je **spoljnim** Internet **korisnicima** da **pristupaju informacijama** i servisima dostupnim **na korporacijskim serverima**

- korporacijski Web server
- **Iz sigurnosnih razloga**, spoljnim korisnicima obično **nije dozvoljen direktni pristup** korporacijskim serverima
- Pristup se ostvaruje **posredstvom specijalizovanog servera**, tzv. **firewall** ili sigurnosni gateway, koji **nadgleda i filtrira** mrežni **saobraćaj**

Firewall

- **Firewall** kontroliše protok informacija u oba smera (kao iz tako i u intranet):

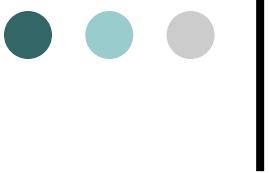


- **Firewall** presreće i filtrira pakete koji su sa Interneta upućeni lokalnim serverima, kao i sve zahteve koji se iz intraneta šalju prema Internetu



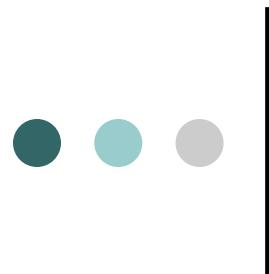
Firewall

- Filtriranje se obavlja ***na osnovu:***
 - izvornih i odredišnih ***IP adresa i brojeva portova*** sadržanim u TCP/UDP paketima ili
 - nekih ***drugih kriterijuma***
- ***Firewall*** može biti ***konfigurisan*** tako da ***prosleđuje ka intranetu***
 - samo pakete koji su upućeni ***na određene lokalne IP adrese*** i/ili
 - samo ako su paketi poslati ***iz nekog određenog domena***, a da sve ostale poništava



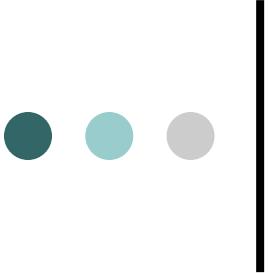
Firewall

- **Firewall** može da uvede **restrikcije** koje se **odnose na brojeve portova**
- **Na primer**, firewall može biti tako podešen da poništava sve **dolazne TCP pakete** sa brojem odredišnog porta 20 ili 21 i da na taj način **onemogući pristup** bilo kom **lokalnom FTP serveru**
- Slične restrikcije mogu biti uvedene i za pakete koji se iz intraneta šalju ka Internetu



Firewall

- **Na primer**, administrator intraneta može da konfiguriše *firewall* tako da prema Internetu **propušta smo TCP pakete** sa odredišnim **brojem porta 80** i da tako **lokalnim korisnicima omogući korišćenje globalnog Web-a**, ali u isto vreme i zabrani pristup ostalim servisima dostupnim na Internetu



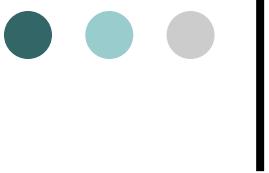
Cookie

- Web sistem **ne podržava sesije**
- **Pretraživač** šalje **zahtev**, a **server** vraća nazad **odgovor** (kopiju traženog resursa)
- Nakon obavljenе transakcije, **server** **zaboravlja** da je ikada komunicirao sa tim konkretnim klijentom – sve **interakcije** (zahtev/odgovor) su **međusobno nezavisne**
- Kaže se da je Web sistem **stateless** (bez stanja) ili bez memorije



Cookie

- ***U početnom periodu razvoja*** Web sistema
- Isključivo se koristio za ***pribavljanje javno dostupnih dokumenata*** - ovakav (***stateless***) ***način rada*** bio je u potpunosti ***zadovoljavajući***
- Međutim, sa ***naglim širenjem Web-*** prostora i porastom njegove popularnosti, ***model zasnovan na nezavisnim transakcijama*** više ***nije mogao da zadovolji*** sve potrebe novih primena



Cookie

- **Primer** - neki Web sajтови **zahtevaju od korisnika da se registruju** pre nego što im se dozvoli **korišćenje** dostupnih **usluga sajta**
- Postavlja se pitanje: **kako će Web server znati da li zahtev potiče od registrovanog ili neregistrovanog korisnika**, pa da na osnovu toga dozvoli ili zabrani pristup nekim svojim stranicama



Cookie

- ***Sličan problem*** se javlja kod Web sajtova **elektronskih (Web-) prodavnica**
- Omogućavaju korisniku da:
 - ***pregleda*** ponuđene articke,
 - ***izabere*** one koje želi da kupi i
 - ***stavi ih u*** elektronsku ***korpu*** i
 - na kraju kupovine ***plati račun*** kreditnom karticom



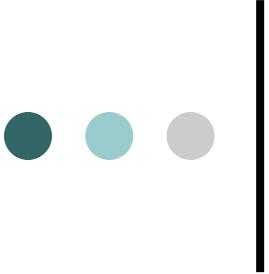
Cookie

- **Zaključak:**
- **Uprkos stateless prirodi** komunikacije unutar Web sistema – **ima smisla** (i veoma je korisno) **razmotriti pojma sesije** u Web komunikaciji
- U okviru koje bi **stanje korisnika bilo sačuvano kroz veći broj interakcija** (zahteva i odgovora)
- To bi pomoglo serveru da **identifikuje određeni pretraživač** – omogućavajući mu da **poveže skup** inače **nezavisnih zahteva** sa sesijom
- Takvi zahtevi, zajedno, **čine jednu sesiju**



Cookie

- **Rešenje (ideja)** – dovoljno je da **server vodi evidenciju o IP adresama** sa kojih se upućuju zahtevi za stranicama
- Međutim, **ova ideja nije dobra**
- **Prvo**, mnogi korisnici koriste računare koje **dele sa drugim korisnicima**
- **IP adrese može da identifikuje računar, ali ne i korisnika** koji trenutno radi na računaru



Cookie

- **Drugo, mnogi korisnici *Internetu pristupaju putem provajdera Internet usluga* od kojih, nakon konektovanja, *dobijaju IP adresu na privremeno korišćenje*, koja ne mora biti ista kao IP adresa koju su koristili prilikom prethodnog konektovanja**
- **Da bi se opisani problem rešio, razvijena je tehnika pod nazivom cookies (kolačići)!**

Kreiranje i čuvanje Cookie-a

- Kreiranje i čuvanje cookie-a zavisi od konkretne realizacije - osnovni princip je uvek isti
- **Mali fajl** (ne veći od **4KB**) ili **string** koji se pridodaje svakom **zahtevu ili odgovoru** koji se razmenjuju između klijenta i servera
- **Pravi ga server** i, zajedno sa traženom Web stranicom, **šalje ga klijentu**



Kreiranje i čuvanje Cookie-a

- **Sadrži**

- **ime domena servera**,
- **podatke** koje je server prikupio **od klijenta**
 - korisničko ime,
 - registracioni broj
- **dodatne podatke** koje imaju značaj u konkretnoj situaciji

Kreiranje i čuvanje Cookie-a

Na strani klijenta (kada primi odgovor)

- **Pretraživač** automatski (a) **izdvaja cookie** iz primljenog odgovora i (b) **smešta ga na hard disk** u direktorijum predviđen za čuvanje cookie

Kada klijent šalje zahtev serveru

- **Pretraživač** (a) **proverava** da li u direktorijum za cookie, **postoji cookie** kojeg je **poslao dati server**
- Ako takav cookie postoji, (b) pretraživač ga **pridodaje zahevu i šalje serveru**



Kreiranje i čuvanje Cookie-a

- **Server, kada primi zahtev, zna** da se radi o **starom, prepoznatljivom** korisniku, a ne o nekom novom
- **Dakle**, cookie **pravi** i **koristi** (konzumira) **server**
- **Pretraživač čuva** cookie do sledećeg obraćanja serveru, ali nikada **ne čita sadržaj** cookie-a, niti njegov sadržaj otkriva korisniku



Format Cookie-a

- Cookie može sadržati **do pet polja**:
(Domain, Path, Content, Expires, Secure)
- Domain** (domen). Ovo polje sadrži **ime domena** iz koga cookie potiče (**domen servera**)
- Path** (putanja). Ovo polje sadrži **putanju u strukturi direktorijuma servera** koja ukazuje na **deo njegovog stabla direktorijuma**
 - Unutar tog dela **sve stranice mogu koristiti cookie**
 - Kosa crta "/" u ovom polju znači celokupno stablo



Format Cookie-a

- **Content** (sadržaj). Sadržaj ovog polja je oblika ***ime = vrednost***
- Ime i vrednost mogu biti ***bilo koja informacija koju server želi da uvrsti u cookie***



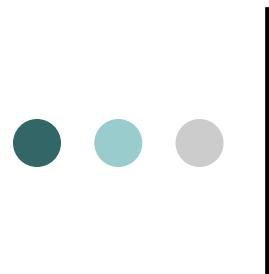
Format Cookie-a

- **Expires** (*rok važenja*). Ovo polje sadrži **datum i vreme kada prestaje važnost** cookie-ja
- Drugi način – korišćenje **Max-Age** direktive
- Ako ovo polje ne postoji, pretraživač uništava cookie prilikom svog zatvaranja (sesije)
- Ovakav cookie se zove **neprezistentni cookie**
- Ukoliko su datum i vreme sadržani u cookie-ju, za cookie se kaže da je **perzistentan**
- Ostaće na klijentu sve do isteka navedenog roka



Format Cookie-a

- **Secure** (Sigurnost). Ako je postavljeno na Yes, nalaže pretraživaču da cookie može da se **vrati serveru samo u šifrovanom obliku**
- Ova mogućnost se koristi kod **aplikacija za elektronsko poslovanje u bankovnim transakcijama** i drugim aplikacijama koje zahtevaju zaštitu informacija



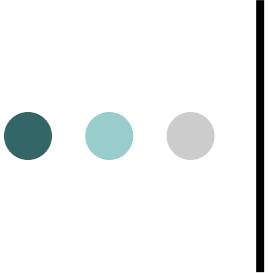
HTTP i Cookie

- **Cookie**-ima su posvećena **dva zaglavla**
- Preko zaglavla **Cookie** klijent vraća serveru **sadržaj cookie-a** kojeg je ranije poslat klijentu od strane neke mašine **iz domena servera**
- **Server šalje cookie klijentu** u obliku sadržaja zaglavla **Set-Cookie**
- Kao što znamo, klijent je u obavezi da **zapamti cookie na svoj hard disk**, i vrati ga serveru pri svakom narednom obraćanju

HTTP i Cookie

- **Server** aplikacije mogu koristiti **Set-Cookie** zaglavlje na sledeći način:

```
Set-Cookie: <name>=<value>
[; Comment=<value>] [; Max-Age=<value>]
[; Expires=<date>] [; Path=<path>]
[; Domain=<domain name>] [; Secure]
[; Version=<version>]
```



HTTP i Cookie

- **Primer1:**

```
HTTP/1.1 200 OK
```

```
Set-Cookie: Name="Leon"; Path="/test/"; Domain=".rutgers.edu";  
Version=1
```

Nalaže pretraživaču da uključi **zaglavlje Cookie** sa vrednošću Leon kad god pravi zahtev za server Rutgers za resurse čiji URL počinje sa /test/.
Izostavljanje informacija o **roku važenja** – označava da je ovaj Cookie važeći samo dok traje tekuća sesija pretraživača

HTTP i Cookie

- **Primer2:** Primer iznajmljivanja filma

Počinjemo sa podnošenjem registracije posetom URL-a koji nam omogućuje pouzdanu prijavu na sajt za iznajmljivanje filmova:

```
GET /movies/register HTTP/1.1
Host: www.sample-movie-rental.com
Authorization: ...
```

Autorizacioni
kredencijali

Server prepoznaje i potvrđuje korisnika – šalje odgovor koji uključuje Set-Cookie zaglavlje sa ID-em kupca

```
HTTP/1.1 200 OK
Set-Cookie: CUSTOMER="Rich"; Path="/movies"; Secure; Version='1'
...
```



HTTP i Cookie

- **Primer2 (nastavak)**: Primer iznajmljivanja filma

Od tog trenutka, kad god pretraživač podnese zahtev namenjen :

http://www.sample-movie.rental.com/movies/*

```
GET /movies/rent-recommended HTTP/1.1
Host: www.sample-movie-rental.com
Cookie: $Version="1"; CUSTOMER="Rich"; $Path="/movies"
```

Uključiće u taj zahtev i **Cookie zaglavlje** koje **sadrži ID kupca**