

Računarske mreže
(2OER5O03)

HTTP i SMTP

Auditivne vežbe

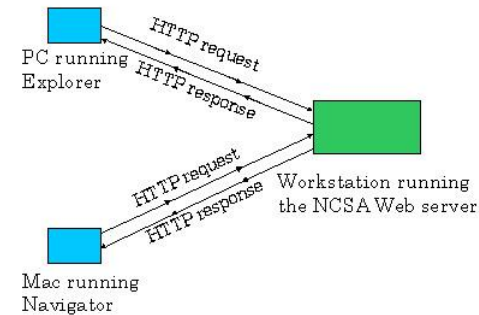


Računarske mreže

HTTP, E-mail

HTTP

- Hypertext Transfer Protocol
- Protokol aplikativnog nivoa
- Osnova Web-a
- Klijenti i serveri komuniciraju razmenjivanjem poruka
- Osnovni resurs čine Web dokument
- Pristup preko URL-a
- Protokol transportnog nivoa: TCP
- Port: 80



URL

■ <http://www.someSchool.edu:80/someDepartment/server.cgi?str1=1&str2=3>

- protokol
- adresa hosta i port
- putanja do resursa na datom hostu
- naziv dokumenta ili servisa
- parametri upita (samo kod servisa)

Neperzistentna veza

1. Klijent inicira konekciju (zadata adresa i port 80) – kreira se TCP soket konekcija
2. Klijent šalje HTTP zahtev kroz kreiranu konekciju (sadrži deo URL-a objekta koji se zahteva)
3. Server prima zahtev kroz konekciju uspostavljenu u tački 1, pronalazi traženi objekat i inkapsulira ga u HTTP odgovor koji šalje kroz uspostavljenu konekciju
4. Server inicira zatvaranje TCP konekcije (ali se ona fizički ne zatvara dok ne stigne potvrda klijenta da je korektno primio odgovor)
5. Klijent prima odgovor, konekcija se zatvara, ekstrahuje objekat iz odgovora i vrši njegovu obradu (parsiranje)
6. Ukoliko je objekat sadržao reference na druge objekte (npr. HTML stranica sadrži slike), tada se se koraci 1-5 ponavljaju za svaku od slika.

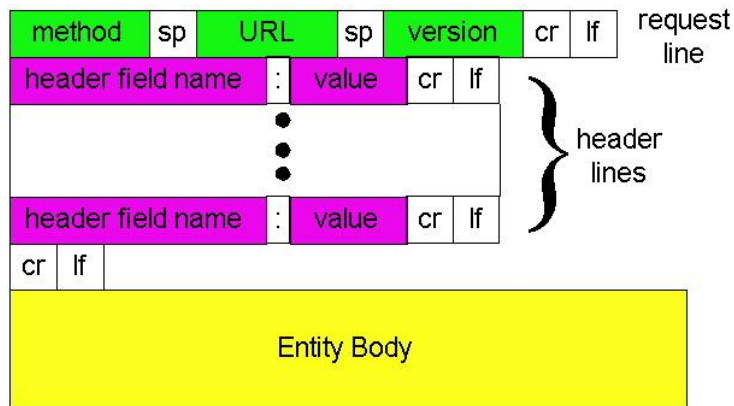
Neperzistentna veza

- Veza je neperzistentna jer se TCP veza raskida nakon svakog prenetog objekta
- Ako dokument ima 10 slika, potrebno je 11 konekcija da bi se preneo celi dokument
- Svaka konekcija zahteva trostruku *handshake* proceduru (potrebno je 2RTT - *round trip time* za uspostavljanje veze)
- Problem sporog starta kod svake konekcije
- RFC 1945 - Hypertext Transfer Protocol -- HTTP/1.0
- <http://www.faqs.org/rfcs/rfc1945.html>

Perzistentna veza

- Kod perzistentne veze ona ostaje otvorena i nakon povratka traženog objekta
- Zahtevi koji slede (od istog klijenta) idu preko iste veze
- Veza se zatvara kada se ne koristi izvesno vreme
- Postoje dva tipa veza
 - ▷ bez *pipeline*-a – novi zahtev se može poslati tek po kompletiranju prethodnog. Svaki novi zahtev troši 1 RTT (umesto 2 RTT kod neperzistentne veze). Nakon odgovora server čeka da pristigne novi zahtev.
 - ▷ sa *pipeline*-om
- HTTP 1.1 podrazumeva perzistentnu vezu sa *pipeline*-om
- RFC 2068 - Hypertext Transfer Protocol -- HTTP/1.1
- <http://www.faqs.org/rfcs/rfc2068.html>

HTTP zahtev



GET /somedir/page.html HTTP/1.1

Connection: close

User-agent: Mozilla/4.0

Accept: text/html, image/gif, image/jpeg

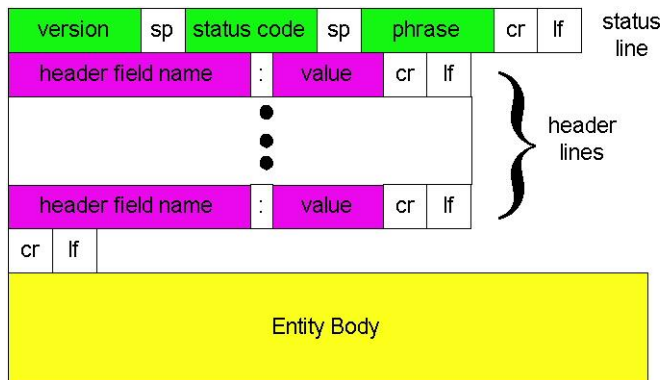
Accept-language: fr

< dodatni prazni red >

Najčešći tipovi zahteva:

- GET
- POST
- HEAD

HTTP odgovor



HTTP/1.1 200 OK

Connection: close

Date: Thu, 26 Apr 2005 07:00:15 GMT

Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT

Content-Length: 6821

Content-Type: text/html

data data data data data ...

Neki kodovi HTTP odgovora:

200 OK

301 Moved Permanently: **nova**
lokacija specificirana je u Location:

400 Bad Request

404 Not Found

505 HTTP Version Not Supported

Authentication

Neki sajtovi zahtevaju od korisnika da unesu svoje korisničko ime i šifru. HTTP omogućuje automatizaciju ovog procesa, mada je pristup vrlo slab i ne može se koristiti u nekim “ozbiljnijim” aplikacijama

1. Klijent šalje običan zahtev bez specijalnih zaglavlja
2. Server odgovara sa **401 Authorization Required** i navodi zaglavlje **WWW-Authenticate:** u kome se specificiraju detalji prijavljivanja
WWW-Authenticate: Basic realm="User Visible Realm"
3. Po prijemu ovakvog odgovora, klijent otvara dijalog u kome korisnik unosi korisničko ime i šifru, a zatim šalje zahtev sa uključenim zaglavljem **Authorization:** u kome navodi korisničko ime i šifru
user:password → Base64 →
Authorization: Basic QWxhZGRpbjpPcGVuU2VzYW1l

Nakon dobijanja prvog odgovora, klijent nastavlja da šalje korisničko ime i šifru sa svakim narednim zahtevom

Cookies

- Mehanizam za praćenje aktivnosti korisnika (obzirom da je HTTP *stateless* protokol)
- Kada klijent prvi put kontaktira server koji koristi *cookies*, odgovor servera sadrži i zaglavlje Set-cookie:
 - ▶ Set-cookie: 1678453
- Klijent dobijene podatke u Set-cookie liniji dodaje u zasebnu tekstualnu datoteku koja se smešta na klijentskoj strani. Kod sledećeg pristupa istom serveru, klijent dodaje Cookie: zaglavlje u svoj zahtev i upisuje ogovarajući sadržaj
 - ▶ Cookie: 1678453
- Detalji se mogu naći u RFC 2109

Uslovno GET

Klijent zahteva nekeširani objekat

```
GET /fruit/kiwi.gif HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
```

Web server šalje odgovor sa zahtevanim objektom

```
HTTP/1.0 200 OK
Date: Wed, 12 Aug 1998 15:39:29
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 09:23:24
Content-Type: image/gif
data data data data data ...
```

Klijent prikazuje objekat i snima ga u lokalni keš, zajedno sa *last-modified* podatkom. Kada se nekada kasnije zahteva isti objekat, a on postoji u lokalnom kešu, šalje se uslovni zahtev zajedno sa zapamćenim *last-modified* podatkom.

```
GET /fruit/kiwi.gif HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
If-modified-since: Mon, 22 Jun 1998 09:23:24
```

Ako objekat nije modifikovan server odgovara kodom 304, čime kaže klijentu da može da koristi lokalnu kopiju

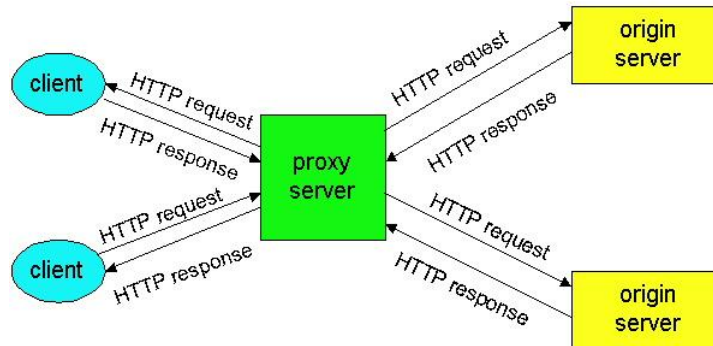
```
HTTP/1.0 304 Not Modified
Date: Wed, 19 Aug 1998 15:39:29
Server: Apache/1.3.0 (Unix)
(empty entity body)
```

Web Cache

- **Web Cache** ili **Proxy Server** je mrežni entitet koji odgovara na HTTP zahteve klijenata kako bi poboljšali performanse mreže
- Ima svoje skledište (diskove) u kome čuva kopije zahtevanih objekata
- Ako se klijent (browser) konfigurira na odgovarajući način, svaki zahtev klijenta se najpre upućuje Web Cache-u

Web Cache

- Klijent uspostavlja TCP vezu sa proxy serverom i šalje zahtev za dobijanje nekog objekta
- Proxy server traži kopiju u svom lokalnom skladištu i ukoliko postoji šalje je klijent
- Ako ne postoji, šalje se zahtev izvornom serveru, pribavlja traženi objekat, smešta ga u svoj keš, a kopiju vraća klijentu



Elektronska pošta

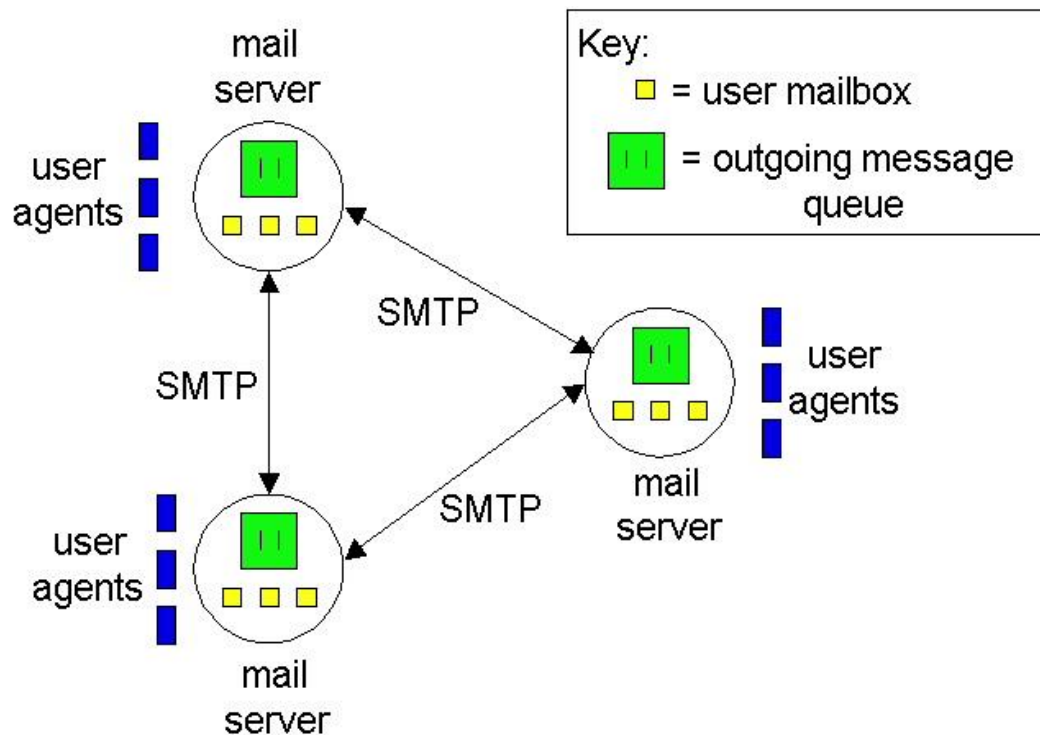
■ SMTP

■ POP3



E-mail

Simple Mail Transfer Protocol (SMTP)



Simple Mail Transfer Protocol (SMTP)

- Protokol aplikativnog nivoa
- Koristi **TCP** kao protokol transportnog nivoa (koristi port **25**)
- SMTP server radi kao klijent kada šalje poštu, a kao server kada je prima
- Poruke koje se šalju smeštaju se u izlazni *queue* i pokušava **direktno** da kontaktira **odredišni** server. Ako u tome ne uspe pri prvom pokušaju, može pokušati još par puta (zavisi od konfiguracije). Ako ne uspe, šalje poruku pošiljaocu o neuspehu slanja
- Detaljnije u **RFC 821**

SMTP komunikacija

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

Razlike u odnosu na HTTP

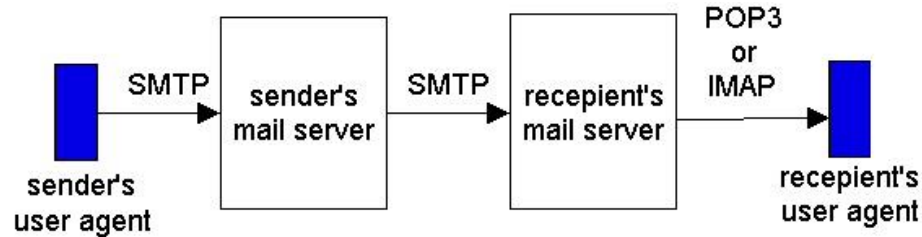
- SMTP je *push* protokol (HTTP je *pull*)
- SMTP zahteva da čitava poruka bude ASCII kodirana (nema binarnih podataka)
- SMTP zahteva da na kraju svake poruke bude linija koja sadrži samo tačku (CRLF.CRLF)
- SMTP smešta sve delove poruke u okviru jednog objekta (tekst, slike i sl. sve je u okviru istog mejla)

Pristupni protokoli

SMTP omogućuje komunikaciju između mail servera, dok se za komunikaciju sa klijentima koriste tzv. pristupni protokoli. Najčešće su to:

- **POP3** (Post Office Protocol - Version 3) – RFC 1939
- **IMAP** (Internet Mail Access Protocol) – RFC 1730
- **HTTP** (npr. Yahoo, Hotmail, ...)

POP3



Vrlo je jednostavan, koristi **TCP** vezu na portu **110** i komunikacija se obavlja u tri faze:

- autorizaciju
- transakciju i
- ažuriranje

POP3 faza autorizacije

■ Koristi dve osnovne komande

▷ user <user name>

▷ pass <password>

■ Nakon uspostavljanja TCP veze na portu 110, autorizacija ima sledeći tok:

+OK POP3 server ready

user alice

+OK

pass hungry

+OK user successfully logged on

Ukoliko se javi greška, server vraća kod **-ERR**

POP3 faza transakcije

Koristi četiri osnovne komande: **list**, **retr**, **dele** i **quit**

C: list

S: +OK 2 messages (1410 octets)

S: 1 498

S: 2 912

S: .

C: retr 1

S: +OK 498 octets

S: blah

S:blah

S: .

C: dele 1

S: +OK message deleted

C: retr 2

S: +OK 912 octets

S: blah

S:blah

S: .

C: dele 2

S: +OK message deleted

C: quit

S: +OK POP3 server closing connection

User agent može biti konfigurisan kao "*download and delete*" ili "*download and keep*". Ako se koristi "*download and delete*" iza svakog **retr** izdaje se i **dele** komanda.

Tek nakon izdavanja **quit** komande, POP3 server prelazi u fazu ažuriranja i tek tada zaista briše mejlove.

URLs

RFC 1945 - Hypertext Transfer Protocol -- HTTP/1.0

<http://www.faqs.org/rfcs/rfc1945.html>

RFC 2068 - Hypertext Transfer Protocol -- HTTP/1.1

<http://www.faqs.org/rfcs/rfc2068.html>

RFC 821 - Simple Mail Transfer Protocol

<http://www.faqs.org/rfcs/rfc821.html>

RFC 822 - Standard for the format of ARPA Internet text messages

RFC 2045 - Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies

RFC 2046 - Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types

RFC 2047 - MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text

RFC 2048 - Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures

RFC 2049 - Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples

RFC 1939 - Post Office Protocol - Version 3

PITANJA

