

©material copyright 1996-2020
T.F. Kurose and K.W. Ross. All Rights Reserved

The screenshot shows a desktop environment with several windows open. On the left, there's a window for 'Second Life' with a virtual world interface. In the center, a 'Skype' window shows a contact list. On the right, there are two 'Facebook' windows, one showing a profile for 'Keith Flores' and another showing a search result for 'polytechnic of NYU'. Below these windows is a diagram illustrating the DHT Network for Public and Private Torrents. The diagram shows 'Public Trackers' (Pezbly, PTP, BitReactor, etc.) and 'Private Trackers' (Zamunda, HqReactor) connected to a 'DHT Network' of peers. A legend on the right lists various trackers: Azurius, Mainline, uTorrent, Kurobi, BitComet, ABC, and Tribler. Above the diagram are logos for 'Public Discovery Sites' (muhimbi, ISMUSIC, etc.) and 'Private Discovery Sites' (cinematik.net, HqReactor, etc.).

3: Nivo aplikacije 1

1

Nivo aplikacije

- Principi protokola nivoa aplikacije
- Web
 - HTTP

3: Nivo aplikacije 2

2

Primjeri Internet aplikacija

- E-mail
- Web
- "Instant messaging"
- "Remote login"
- "P2P file sharing"
- "Multi-user" mrežne igre
- "Streaming stored" video klipovi
- Internet telefon
- "Real-time" video konferencija
- "Grid computing"
- Društvene mreže
- ...

3: Nivo aplikacije 3

3

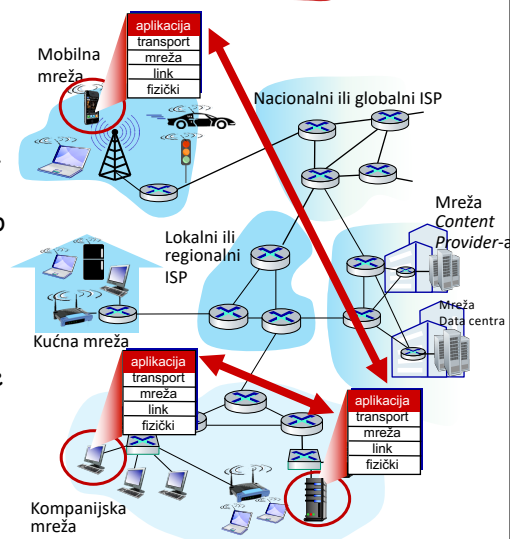
Kreiranje Internet aplikacije

Napisati programe koji

- se izvršavaju na različitim krajnjim sistemima i
- komuniciraju preko mreže.
- npr., Web: Web server software komunicira preko browser software

Ne piše se softver za uređaje na kičmi mreže

- mrežni uređaji na kičmi ne funkcionišu na nivou aplikacije
- ovakav dizajn dozvoljava brzi razvoj aplikacija



3: Nivo aplikacije 4

4

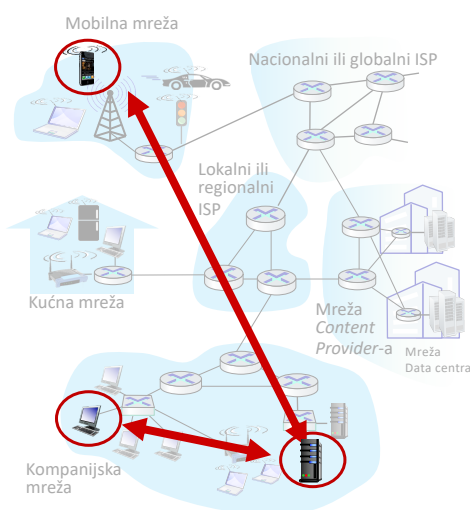
Arhitekture Internet aplikacija

- Klijent-server
- Peer-to-peer (P2P)
- Hibrid klijent-server i P2P
- ...

3: Nivo aplikacije 5

5

Klijent-server arhitektura



Server:

- Uvijek aktivan
- Po pravilu permanentna IP adresa
- Data centri

Klijenti:

- Komuniciraju sa serverom
- Mogu biti povremeno povezani
- Mogu imati dinamičku IP adresu
- Ne komuniciraju međusobno

Primjeri:

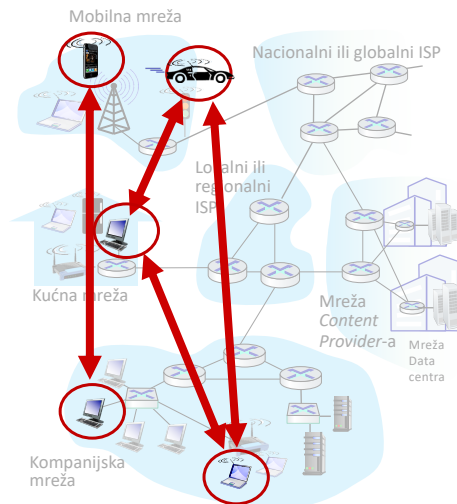
- Web, e-mail, file transfer

Nivo aplikacije 2-6

6

P2P arhitektura

- Proizvoljni krajnji sistemi mogu direktno komunicirati bez učešća servera
- Peer zahtijeva servis od drugog peer-a, nudeći servis drugim peer-ovima
 - *skalabilnost*- novi peer-ovi donose nove kapacitete, ali i nove zahtjeve
- Peer-ovi se povremeno povezuju i mogu da mijenjaju IP adrese
 - Složeno upravljanje
- Primjer: P2P file sharing



3: Nivo aplikacije 7

7

Hibrid Klijent-server i P2P arhitektura

Skype/Viber

- voice-over-IP P2P aplikacije
- centralizovani server: pronalaženje strane sa kojom se želi komunicirati
- klijent-klijent komunikacija je direktna bez posredovanja servera

Instant messaging

- Čatovanje dva korisnika je P2P
- Detektovanje prisutnosti i lokacije je centralizovano:
 - Korisnik registruje svoju IP adresu na centralni server kada hoće da čatuje
 - Korisnik kontaktira centralni server da pronade IP adrese korisnika sa kojima želi da čatuje

3: Nivo aplikacije 8

8

Komuniciranje procesa

Proces: program koji se izvršava na hostu.

- U samom hostu, dva procesa komuniciraju na bazi **inter-procesne komunikacije** (definisane u OS).
- Procesi na različitim hostovima komuniciraju razmjnom **poruka**

Klijent proces: proces koji inicijalizuje komunikaciju

Server proces: proces koji čeka da bude kontaktiran

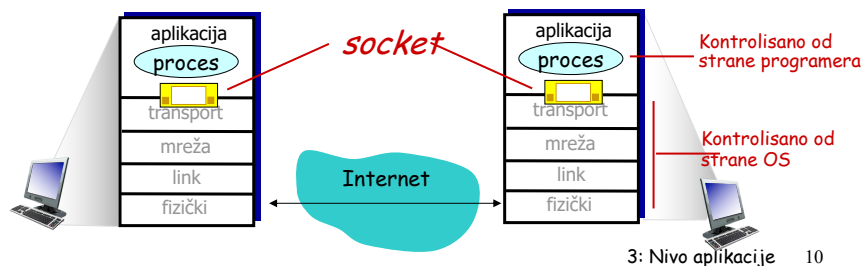
- Napomena: aplikacije sa P2P arhitekturom imaju i klijent i server procese

3: Nivo aplikacije 9

9

Socket-i

- Proces šalje/prima poruke preko svog *socket-a*
- *Socket* je sličan vratima
 - Proces šalje poruke preko *socket-a*
 - Proces koji šalje se oslanja na transportnu infrastrukturu na drugoj stani vrata koja prenosi poruku do *socket-a* prijemne strane
 - Po dva su *socket-a* angažovana (jedan na izvorištu drugi na odredištu)



10

Adresiranje

- Za proces koji prima poruke, mora postojati identifikator
- Svaki host ima jedinstvenu 32-bitnu IP adresu
- Prisjetiti se komande ipconfig...
- **P:** Da li je IP adresa hosta na kojem se proces izvršava dovoljna za identifikaciju procesa?
- **O:** Ne, mnogi procesi se mogu izvršavati na istom hostu
- Identifikator uključuje i IP adresu i broj porta vezan za proces na hostu.
- Primjer brojeva porta:
 - HTTP server: 80
 - Mail server: 25
- **VIŠE KASNIJE**

3: Nivo aplikacije 11

11

Protokol nivoa aplikacije definiše

- Tipove poruka koje se razmjenjuju, npr., zahtjevi & poruke odgovora
- Tipove sintaksi poruka: koja su polja & kako su odvojena
- Semantika polja, npr., značenje informacija u poljima
- Pravila vezana kada i kako se šalje poruka i kako se odgovara na njih
- **Javni (public) protokoli:**
 - Definisani u RFC-ovima
 - Dozvoljavaju interoperabilnost
 - npr, HTTP, SMTP
- **Privatni (proprietary) protokoli:**
 - npr, Skype, Viber, Zoom,...

3: Nivo aplikacije 12

12

Koji transportni servisi su potrebni aplikacijama?

Gubici podataka

- Neke aplikacije (npr., audio) mogu tolerisati određeni nivo gubitaka
- Druge aplikacije (npr., file transfer, telnet) zahtijevaju 100% pouzdani transfer podataka

Vrijeme

- Neke aplikacije (npr., Internet telefonija, interaktivne igre) zahtijevaju malo kašnjenje

Brzina prenosa

- Neke aplikacije (npr., multimedija) zahtijevaju preciziranje minimalne dostupne brzine prenosa
- Druge aplikacije ("elastične aplikacije") koriste onoliko opsega koliko mogu dobiti

Zaštita

- Enkripcija, integritet podataka, ...

3: Nivo aplikacije 13

13

Transportni servisni zahjevi zajednički za sve aplikacije

Aplikacija	Gubici	Brzina prenosa	Vrem. osjet.
File transfer	bez	elastičan	ne
E-mail	bez	elastičan	ne
Web	bez	elastičan	ne
Real-time audio/video	tolerantne	audio: 5kb/s-1Mb/s video: 10kb/s-5Mb/s	da, 10' ms
Stored audio/video	tolerantne	Isti kao gore	da, nekoliko s
Interaktivne igre	tolerantne	nekoliko kb/s i više	da, 10' ms
Instant messaging	bez	elastičan	da i ne

3: Nivo aplikacije 14

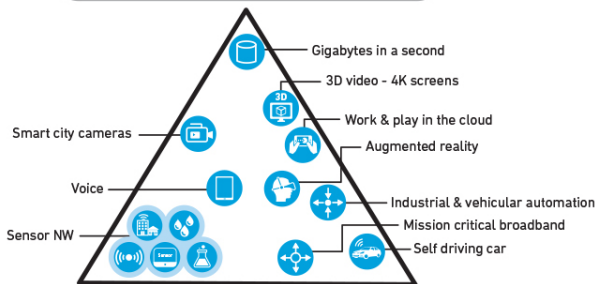
14

"5G trougao"

Velika brzina prenosa
(>100Mb/s, u piku više od 10Gb/s)

Enhanced Mobile Broadband Capacity Enhancement

Qorvo: LTE-A, Pro, Extended Bands, Fixed Wireless mmW,
Beam Steering Infrastructure, Efficient FEMs



10 do 100 puta više uređaja
nego što podržavaju 4G i
današnji WiFi

(Source: Qorvo, Inc., from ITU-R IMT 2020 requirements)

Malo kašnjenje (<1ms)
Mala vjerovatnoća gubitka (reda 10^{-9})

3: Nivo aplikacije 15

15

Servisi transportnih protokola Interneta

TCP servisi:

- ❑ **konektivnost:** uspostavljanje komunikacije se zahtijeva između klijentskih i serverskih procesa
- ❑ **pouzdan transport** između procesa slanja i prijema
- ❑ **kontrola protoka:** pošiljalac ne smije da "zaguši" prijemnik
- ❑ **kontrola zagušenja:** usporava pošiljaoca kada je mreža zagušena
- ❑ **ne obezbeđuje:** tajming, garantovanje minimalnog opsega, sigurnost

UDP servisi:

- ❑ nepouzdan prenos podataka između procesa slanja i prijema
- ❑ ne obezbeđuje: uspostavljanje veze, pouzdanost, kontrolu protoka, kontrolu zagušenja, tajming, garantovani opseg i sigurnost

P: Zašto oba? Zašto UDP?

3: Nivo aplikacije 16

16

Internet aplikacije: aplikacija, transportni protokoli

Aplikacija	Protokoli nivoa aplikacije	Transportni protokol
e-mail	SMTP [RFC 5321]	TCP
Interaktivne igre	WOW, FPS (privatni)	UDP ili TCP
Web	HTTP1.1 [RFC 7320]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	HTTP [RFC 7320], DASH	TCP
Internet telefonija	SIP [RFC 3261], RTP [RFC 3550], ili privatni (Skype,...)	TCP ili UDP

3: Nivo aplikacije 17

17

Zaštita i TCP

Vanilla TCP & UDP *socket*-i

- ❑ Nema kriptovanja
- ❑ Tekstualne lozinke se prenose preko Interneta

TLS (Transport Layer Security)

- ❑ Omogućava enkripciju TCP konekcije
- ❑ Integritet podataka
- ❑ Autorizacija od kraja do kraja

TLS

- ❑ Protokol nivoa aplikacije
- ❑ Aplikacije koriste TLS biblioteke, koje "komuniciraju" sa TCP
- ❑ Tekstualna lozinka se šalje kriptovana preko Interneta

3: Nivo aplikacije 18

18

Web i HTTP

Termini

- **Web stranica** se sastoji od **objekata**
- Objekat može biti HTML fajl, JPEG slika, Java "applet", audio fajl,...
- Web stranica se sastoji od **osnovnog HTML-fajla** koji može sadržati reference više objekata
- Svaki objekat se adresira sa **URL (Uniform Resource Locators)**
- Primjer URL:

http://www.ucg.ac.me/index.html
protokol ime hosta ime puta

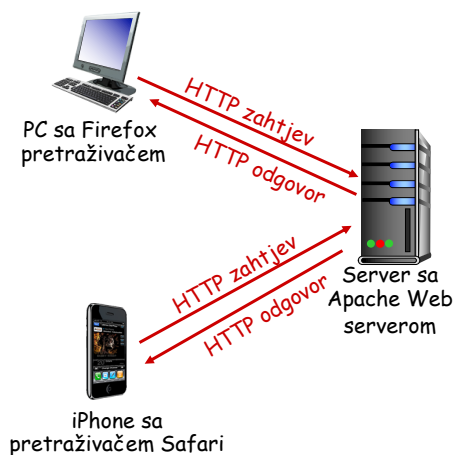
3: Nivo aplikacije 19

19

Pregled HTTP-a

HTTP: HyperText Transfer Protocol

- Web-ov protokol nivoa aplikacije
- klijent/server model
 - *klijent*: Web browser šalje zahtjeve, prima i prikazuje Web objekte
 - *server*: Web server šalje objekte kao odgovor na zahtjeve



3: Nivo aplikacije 20

20

Pregled HTTP-a (nastavak)

Koristi TCP:

- ❑ klijent inicijalizuje TCP vezu (kreira socket) prema serveru, port 80
- ❑ server prihvata TCP vezu od klijenta
- ❑ HTTP poruke zahtjeva i odgovora (poruke protokola nivoa aplikacije) se razmjenjuju između "browser"-a (HTTP klijent) i Web servera (HTTP server)
- ❑ TCP konekcija se zatvara poslije završetka razmjene HTTP poruka

HTTP je "stateless"

- ❑ server ne čuva informacije o prethodnim korisnikovim zahtjevima (ne raspoznaje korisnike)

Pored toga

Protokoli koji nadziru "stanje" su kompleksni!

- ❑ prethodno stanje mora biti nadzirano
- ❑ ako server/klijent "padne", njihovi uvidi u "stanje" mogu biti inkonzistentni, moraju biti ponovo razmotreni

3: Nivo aplikacije 21

21

HTTP konekcije

Neperzistentni (neistrajan) HTTP

- ❑ Najviše jedan objekat se šalje preko TCP konekcije.
- ❑ Povlačenje više objekata podrazumijeva otvaranje više konekcija

Perzistentni HTTP

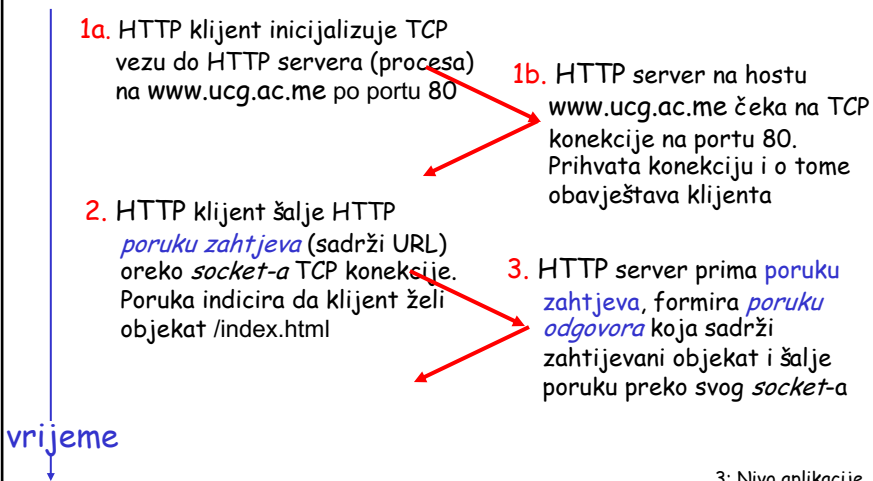
- ❑ Više objekata može biti poslato preko jedne TCP konekcije između klijenta i servera.

3: Nivo aplikacije 22

22

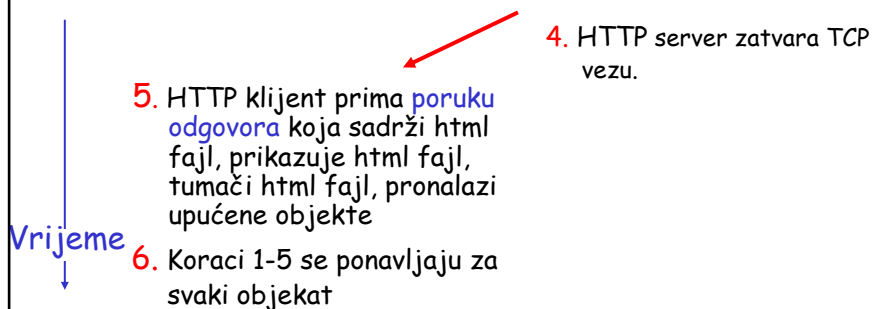
Neperzistentni HTTP

Pretpostavimo da korisnik unese sledeći URL
`http://www.ucg.ac.me/index.html`



23

Neperzistentni HTTP(nastavak)



24

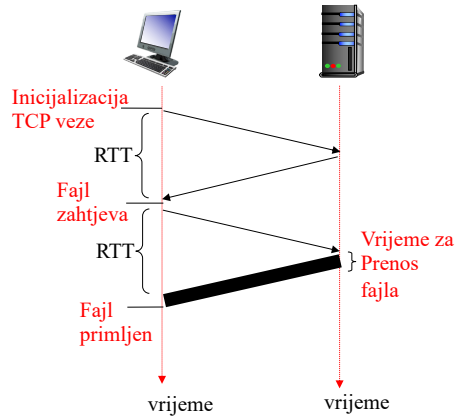
Neperzistentni HTTP: vrijeme odgovora

Definicija RTT (Round Trip Time): vrijeme prenosa paketa zanemarljive veličine od klijenta do servera i nazad.

Vrijeme odgovora:

- jedan RTT za inicijalizaciju TCP veze
- jedan RTT za HTTP zahtjev i vraćanje prvih nekoliko bajtova HTTP odgovora
- Vrijeme prenosa fajla

2RTT+vrijeme prenosa fajla



3: Nivo aplikacije 25

25

Perzistentni HTTP

Problemi neperzistentnog HTTP-a:

- zahtijeva 2 RTT po objektu
- OS mora raditi i dodijeliti resurse hosta za svaku TCP vezu
- Problem je što browser-i često otvaraju paralelne TCP veze za povlačenje zahtijevanih objekata

Perzistentni HTTP

- server zadržava vezu otvorenu poslije slanja odgovora
- sekvencijalne HTTP poruke između istog klijent/servera se šalju istom vezom
- Zatvara konekciju poslije određenog vremena neaktivnosti

Perzistentni bez "pipelining":

- Klijent šalje novi zahtjev samo kada je prethodni odgovor primljen
- jedan RTT za svaki upućeni objekat
- Kada nema zahtjeva TCP konekcija je slobodna

Perzistentni sa "pipelining":

- klijent šalje zahtjeve odmah po dobijanju referenci objekata
- Veličine svega po jedan RTT za svaki referencirani objekat

3: Nivo aplikacije 26

26

HTTP poruka zahtjeva

□ Dva tipa HTTP poruka: *zahtjev, odgovor*

□ **HTTP poruka zahtjeva:**

- ASCII (format čitljiv čovjeku)

Linija zahtjeva
(GET, POST,
HEAD komande)

Linije
zaglavlja

carriage return,
line feed na
početku linije
označavaju kraj zaglavlja

```
GET /index.html HTTP/1.1\r\n
Host: www.ucg.ac.me\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

carriage return karakter
line-feed karakter

3: Nivo aplikacije 27

27

Tipovi

POST:

- web stranice često sadrže forme za unos podataka
- Podaci koje je unio korisnik se šalju od klijenta server u tijelu HTTP POST poruke zahtjeva

GET:

- Služi za povlačenje objekata sa server
- Koristi se i za slanje korisnikovih podataka u sklopu URL polja HTTP GET poruke zahtjeva (poslije simbola '?'):

HEAD:

- Zahtjeva samo zaglavlje koja se šalju ako je specificirani URL zahtijevan GET porukom

PUT:

- *Upload*-uje novi fajl (objekat) na server
- U potpunosti mijenja fajl koji postoji na specificiranom URL-u sa sadržajem u tijelu HTTP POST poruke zahtjeva

www.google.com/animalsearch?monkeys&banana

3: Nivo aplikacije 28

28

HTTP poruka odgovora

statusna linija
protokol

statusni kod

statusna fraza

```
HTTP/1.1 200 OK\r\n
Date: Thu, 26 Sep 2013 18:09:20 CET\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2012 17:00:02
GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-
1\r\n
\r\n
data data data data data ...
```

Linije
zaglavlja

podaci, npr.,
zahtijevani
HTML fajl

3: Nivo aplikacije 29

29

HTTP kodovi statusnog odgovora

U prvoj liniji u server->klijent poruci odgovora.

Nekoliko primjera kodova statusa i odgovarajućih poruka:

200 OK

- Zahtjev uspješan, zahtijevani objekat se nalazi u poruci

301 Moved Permanently

- Zahtijevani objekat preseljen, nova lokacija specificirana u poruci (Lokacija:)

400 Bad Request

- Server ne razumije poruku zahtijeva

404 Not Found

- Zahtijevani dokument nije pronađen na ovom serveru

505 HTTP Version Not Supported

- Verzija HTTP protokola nije podržana

3: Nivo aplikacije 30

30

Cookies: vode računa o "stanju" (RFC 6265)

Mnogi Web sajtovi koriste cookies

Četiri komponente:

- 1) Linija zaglavlja Set-cookie u HTTP poruci odgovora
- 2) Linija zaglavlja Cookie u HTTP poruci zahtjeva
- 3) Cookie fajl se čuva na korisnikovom hostu i održava se od strane korisnikovog browser-a
- 4) Baza podataka na Web sajtu

Primjer:

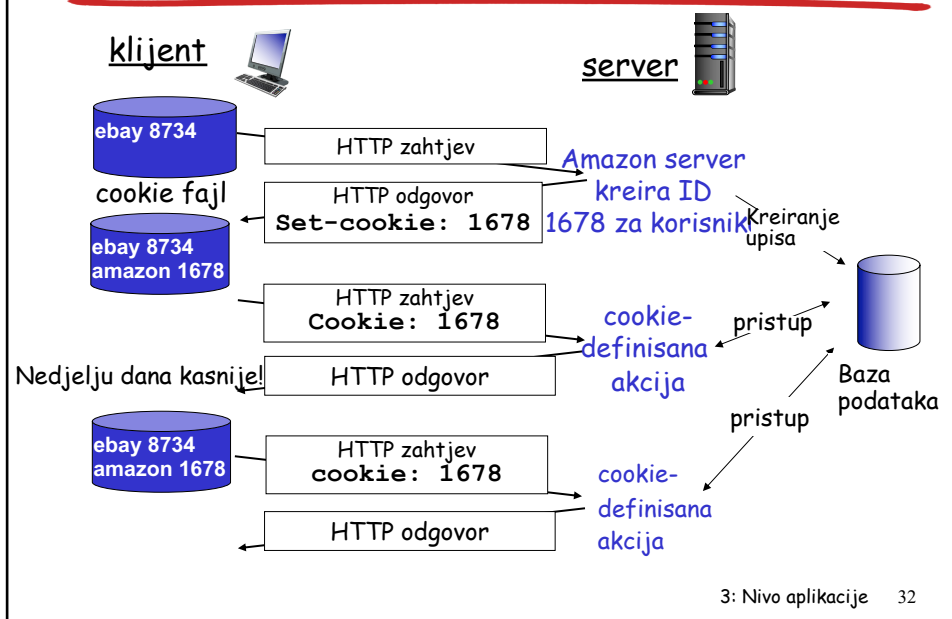
- Neko pristupa Internetu uvijek preko istog hosta
- Posjećuje specifične e-commerce sajtove po prvi put
- Kada inicijalni HTTP zahtjevi dođu na sajt, sajt kreira jedinstveni ID i kreira odgovarajuću informaciju u bazi podataka za ID

<https://tools.ietf.org/html/rfc6265>

3: Nivo aplikacije 31

31

Cookies: vode računa o "stanju" (nastavak)



32

Cookies: vode računa o “stanju”(nastavak)

Šta cookies donose:

- ☐ autorizaciju
- ☐ “shopping cards”
- ☐ preporuke
- ☐ stanje korisnikove sesije (Web e-mail)

Pored toga

Cookies i privatnost:

- ☐ Cookies dozvoljavaju sajtu da dosta nauči o korisniku
- ☐ Mogu se prikupiti imena i kontakt podaci
- ☐ Pretraživači koriste cookies da nauče više o korisnicima
- ☐ Kompanije dobijaju dodatne informacije preko weba

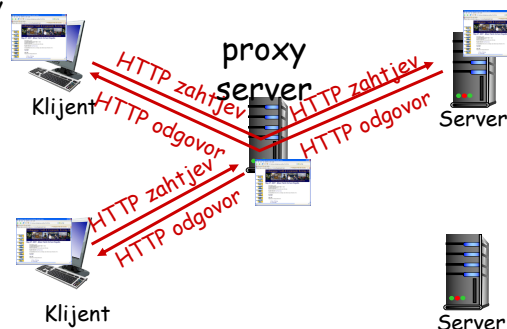
3: Nivo aplikacije 33

33

Web “caches” (proxy server)

Cilj: zadovoljenje klijentovog zahtjeva bez uključivanja originalnog servera

- ☐ Korisnik setuje browser: Web pristup preko proxy servera
- ☐ browser šalje sve HTTP zahtjeve proxy serveru
 - objekat u proxy-u: proxy šalje objekat
 - ili proxy zahtijeva objekat od željenog servera, tada vraća objekat klijentu



<https://tools.ietf.org/html/rfc7234>

3: Nivo aplikacije 34

34

Više o proxy serveru

- Proxy server radi i kao klijent i kao server
- Tipično proxy instalira ISP (univerzitet, kompanija, rezidencijalni ISP)

Zašto proxy server?

- Smanjuje vrijeme odziva na zahtjev.
- Smanjuje saobraćaj na linku institucije prema Internetu.
- Internet sa proxy serverom omogućava "slabim" provajderima sadržaja efikasniju predaju sadržaja

3: Nivo aplikacije 35

35

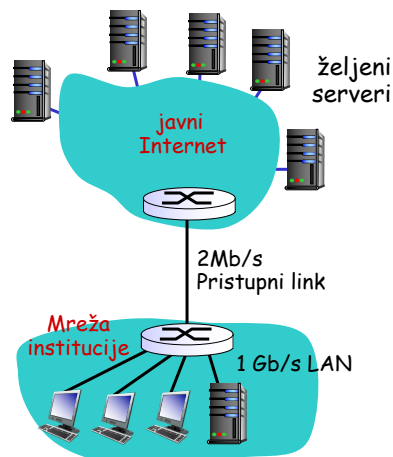
Primjer:

Pretpostavke:

- ❖ Srednja veličina objekta: 100000b
- ❖ Srednji broj zahtjeva prema željenim serverima: 19 zahtjeva/s
- ❖ Srednja brzina : 1.9Mb/s
- ❖ RTT od rutera institucije do željenog servera: 2s
- ❖ Brzina na pristupnom linku: 2Mb/s

Posledice:

- ❖ Iskorištenje LAN-a: 0.19% *problem!*
- ❖ Iskorištenje pristupnog linka = 95%
- ❖ Ukupno kašnjenje = kašnjenje na Internetu + kašnjenje u pristupu + LAN kašnjenje
= 2s + minuti + ms



3: Nivo aplikacije 36

36

Primjer: brži pristupni link

pretpostavke:

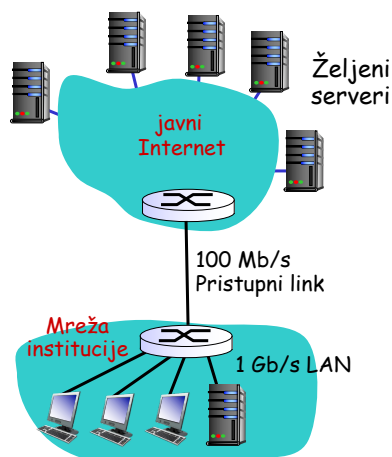
- ❖ Srednja veličina objekta: 100000b
- ❖ Srednji broj zahtjeva: 19 zahtjeva/s
- ❖ Srednja brzina: 1.9Mb/s
- ❖ RTT od rutera institucije do željenog servera: 2s
- ❖ Brzina pristupnog linka: 100Mb/s

posledice:

- ❖ Iskorištenje LAN-a: 0.19%
- ❖ Iskorištenje linka = 1.9%
- ❖ Ukupno kašnjenje = Internet kašnjenje + pristupno kašnjenje + LAN kašnjenje

$$= 2s + ms + ms$$

Troškovi: povećanje brzine pristupa je skupo!



3: Nivo aplikacije 37

37

Primjer: Lokalni proxy

pretpostavke:

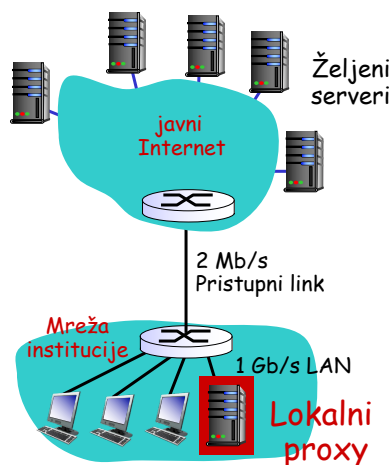
- ❖ Srednja veličina objekta: 100000b
- ❖ Srednja brzina zahtjeva: 19 zahtjeva/s
- ❖ Srednja brzina: 1.9Mb/s
- ❖ RTT od rutera institucije do željenog servera: 2s
- ❖ Brzina pristupa: 2Mb/s

posledice:

- ❖ LAN utilization: 0.19%
- ❖ Iskorištenje pristupnog linka = ?
- ❖ Ukupno kašnjenje = ?

Kako izračunati iskorištenje i kašnjenje?

Troškovi: proxy nije skup!



3: Nivo aplikacije 38

38

Primjer: Lokalni proxy

Izračunavanje iskorišćenja i kašnjenja:

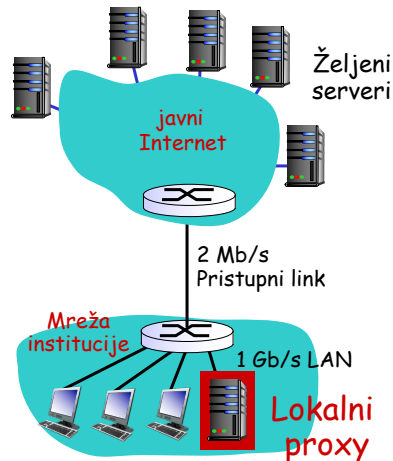
- Pretpostavimo da je vjerovatnoća pogađanja 0.4
 - 40% zahtjeva se posluži na proxy serveru, 60% zahtjeva na željenom serveru

Iskorišćenje pristupnog linka:

- 60% zahtjeva koristi pristupni link
- ❖ Brzina prenosa preko pristupnog linka = $0.6 * 1.9 \text{ Mb/s} = 1.14 \text{ Mb/s}$
- iskorišćenje = $1.14 / 2 = .57$

❖ Ukupno kašnjenje

- = $0.6 * (\text{kašnjenje od željenih servera}) + 0.4 * (\text{kašnjenje do proxy servera})$
- = $0.6 (2.0) + 0.4 (\sim \text{ms})$
- = $\sim 1.2 \text{ s}$
- Manje nego 100 Mb/s pristupni link

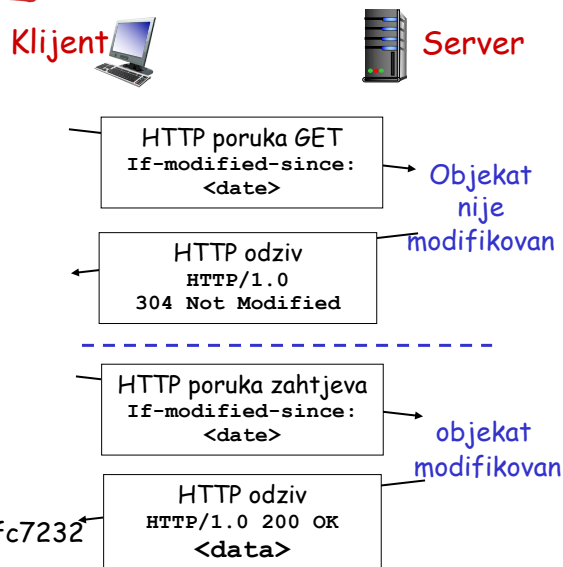


3: Nivo aplikacije 39

39

Conditional GET

- **Cilj:** ne slati objekat ako cache ima up-to-date sačuvanu verziju
- **Klijent:** specificira datum čuvanja kopije u HTTP zaglavlju
`If-modified-since: <date>`
- **Server:** odgovor ne sadrži objekat ako je sačuvana kopija up-to-date:
`HTTP/1.0 304 Not Modified`
<https://tools.ietf.org/html/rfc7232>



3: Nivo aplikacije 40

40

HTTP/2

Cilj: smanjiti kašnjenje u slučaju više objektnih zahtjeva

HTTP 1.1: uvodi više, pipeline GET poruka preko jedne TCP konekcije

- server odgovara *redosledno* (FCFS: first-come-first-served) na GET zahtjeve
- zbog FCFS može se desiti da mali objekat mora da čeka prenos iza velikog objekta (**head-of-line (HOL) blokiranje**)
- Oporavak od gubitaka (retransmisija izgubljeni TCP segmenta) usporava prenos objekata

3: Nivo aplikacije 41

41

HTTP/2

Cilj: smanjiti kašnjenje u slučaju više objektnih zahtjeva

HTTP/2: [RFC 7540, 2015] povećava fleksibilnost servera u slanju objekata klijentu:

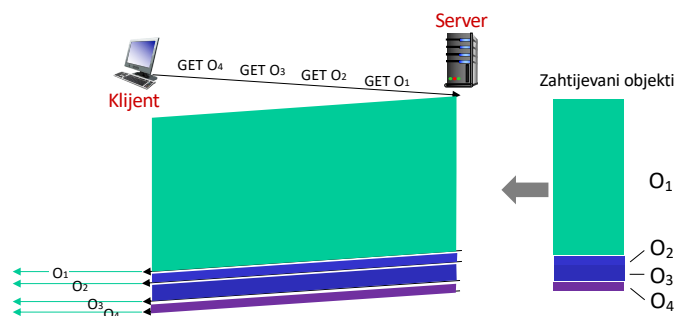
- Poruke zahtjeva, statusni kodovi, većina polja zaglavlja su isovjetna kao kod HTTP 1.1
- Redosled slanja zahtijevanih objekata je baziran na prioritetima koje je definisao klijent (ne mora biti FCFS)
- Omogućava *slanje* nezahtijevanih objekata klijentu
- Dijeli objekte na frejmove, raspoređuje frejmove radi smanjenja HOL blokiranja

3: Nivo aplikacije 42

42

HTTP/2: ublažavanje HOL blokiranja

HTTP 1.1: klijent zahtijeva veliki objekat (na primjer video fajl) i 3 manja objekta



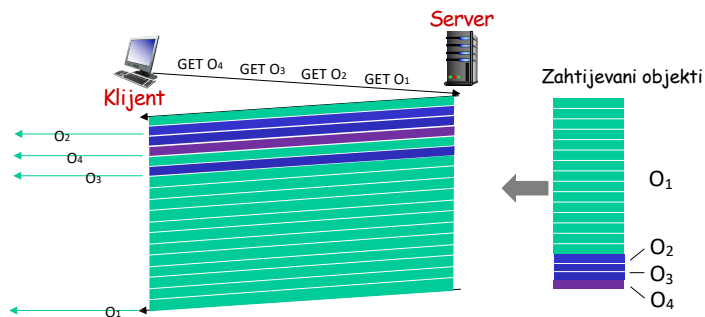
Slanje objekata po redosledu zahtjeva: O_2 , O_3 , O_4 čekaju iza O_1

3: Nivo aplikacije 43

43

HTTP/2: ublažavanje HOL blokiranja

HTTP/2: objekti se dijele ne frejmove, "izmiješano" slanje frejmova



O_2 , O_3 , O_4 se brzo prenose, malo kašnjenje se dodaje O_1

3: Nivo aplikacije 44

44

HTTP/3

HTTP/2 preko jedne TCP konekcije:

- Oporavak od gubitka paketa i dalje usporava prenos objekata
 - Kao i kod HTTP 1.1, *browser*-i otvaraju više paralelnih TCP konekcija radi smanjenja usporavanja i povećanja propusnosti
- Nema zaštite preko vanilla TCP konekcije

HTTP/3:

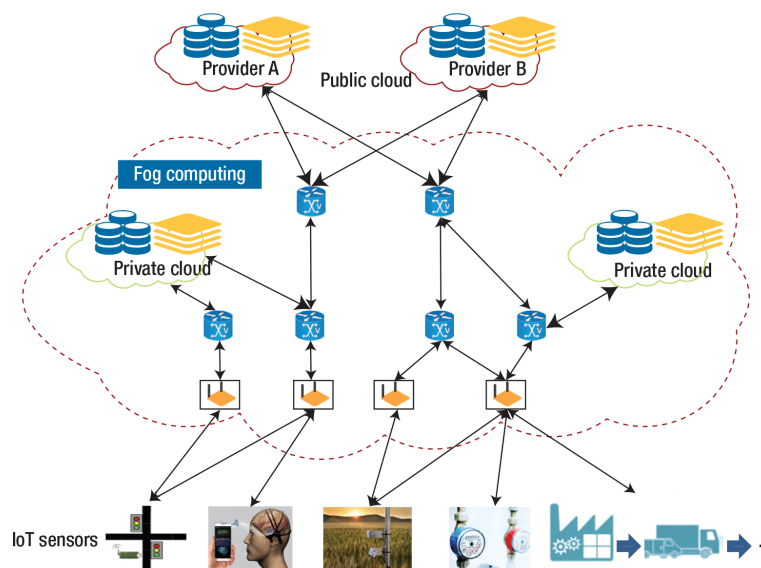
- Ima zaštitu
- Više pipeline prenosa preko UDP protokola

3: Nivo aplikacije 45

45

Fog computing

Fog Computing: Helping the Internet of Things Realize Its Potential
Issue No. 08 - Aug. (2016 vol. 49), ISSN: 0018-9162 pp: 112-116
DOI Bookmark: <http://doi.ieeecomputersociety.org/10.1109/MC.2016.245>
Amin Vahid Dastjerdi, University of Melbourne
Rajkumar Buyya, University of Melbourne



3: Nivo aplikacije 46

46

Nivo aplikacije

Ispitna pitanja

1. Arhitekture aplikacija
2. Navedi i objasni transportne servise koje zahtijevaju mrežne aplikacije
3. HTTP
4. Neperzistentni i perzistentni HTTP
5. HTTP poruke zahtjeva
6. HTTP poruke odgovora
7. Cookies
8. Web proxy servis
9. HTTP/2