

Nivo aplikacije

- ❑ Principi protokola nivoa aplikacije
- ❑ Web i HTTP

Primjeri mrežnih 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
- Cloud computing
- Fog computing
- ...

2: Nivo aplikacije 3

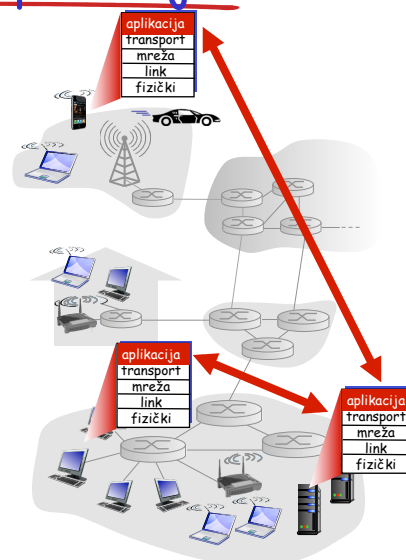
Kreiranje mrežne aplikacije

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

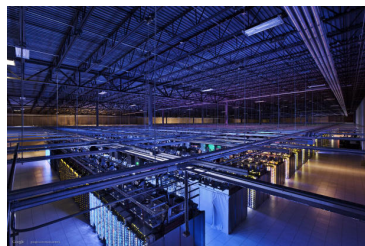
- mrežni uređaji na kičmi ne funkcionišu na nivou aplikacije
- ovakav dizajn dozvoljava brzi razvoj aplikacija



2: Nivo aplikacije 4

Google Data Centri

- ❑ Procijenjena cijena jednog data centra: stotine miliona \$
- ❑ Google svake godine potroši nekoliko milijardi \$ u nove data centre
- ❑ Svaki data centar troši stotine MW električne energije



2: Nivo aplikacije 5

Google Data Centri

Americas

Berkeley County, South Carolina
Council Bluffs, Iowa
Douglas County, Georgia
Jackson County, Alabama
Lenoir, North Carolina
Mayes County, Oklahoma
Montgomery County, Tennessee
Quilicura, Chile
The Dalles, Oregon

Asia

Changhua County, Taiwan
Singapore

Europe

Dublin, Ireland
Eemshaven, Netherlands
Hamina, Finland
St Ghislain, Belgium



<http://www.google.com/about/datacenters/inside/locations/index.html>

2: Nivo aplikacije 6

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

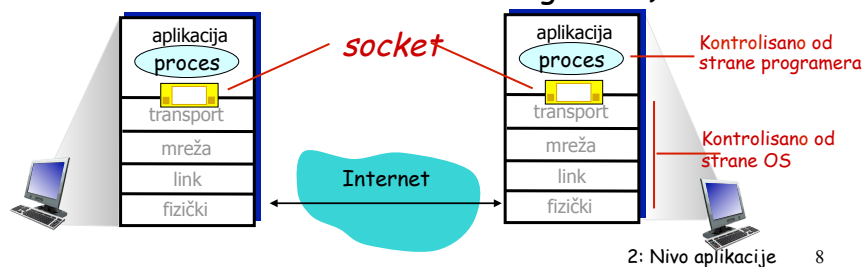
Server proces: proces koji čeka da bude kontaktiran

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

2: Nivo aplikacije 7

Soketi

- Proces šalje/prima poruke preko svog "socket"-a
- "socket" je analogan vratima
 - proces šalje poruke preko socketa
 - proces koji šalje se oslanja na transportnu infrastrukturu na drugoj strani vrata koja prenosi poruku do "socket" prijemnog procesa
- API: (1) izbor transportnog protokola; (2) mogućnost specificiranja nekoliko parametara (maksimalna veličina bafera i maksimalna veličina segmenta)



2: Nivo aplikacije 8

Adresiranje

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

2: Nivo aplikacije 9

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 šalju poruku i kako se odgovara na njih
- Javni (public) protokoli:**
 - ❑ Definisani u RFC-ovima
 - ❑ Dozvoljavaju interoperabilnost
 - ❑ npr, HTTP, SMTP
- Privatni (proprietary) protokoli:**
 - ❑ npr, Skype,...

2: Nivo aplikacije 10

Koji transportni servisi su potrebni aplikacijama?

Gubici podataka

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

Vrijeme

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

Brzina prenosa

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

Zaštita

- Enkripcija, integritet podataka, ...

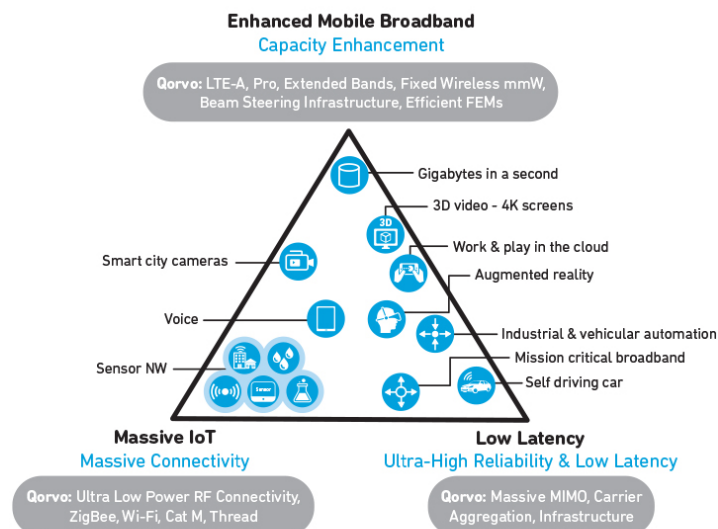
2: Nivo aplikacije 11

Transportni servisni zahtjevi zajednički za sve aplikacije

Aplikacija	Gubici	Brzina prenosa	Vrem. osjet.
file transfer	bez	elastičan	ne
e-mail	bez	elastičan	ne
Web dokumenti	bez	elastičan	ne
real-time audio/video	tolerantne	audio: 5kb/s-1Mb/s video: 10kb/s-5Mb/s	da, 100-tinak ms
stored audio/video	tolerantne	Isti kao gore	da, nekoliko s
Interaktivne igre	tolerantne	nekoliko kb/s i više	da, 100-tinak ms
instant messaging	bez	elastičan	da i ne

2: Nivo aplikacije 12

"5G trougao"



2: Nivo aplikacije 13

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

UDP servisi:

- Nepouzdan prenos podataka između procesa slanja i prijema
- Ne obezbeđuje: uspostavljanje veze, pozuđanost, kontrolu protoka, kontrolu zagušenju, tajming, ili garantovani opseg

P: Zašto oba? Zašto UDP?

2: Nivo aplikacije 14

Internet aplikacije: aplikacija, transportni protokoli

Aplikacija	Protokoli nivoa aplikacije	Transportni protokol
e-mail	SMTP [RFC 2821]	TCP
udaljeni terminal	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	Privatni ili javni	TCP ili UDP ili RTP
Internet telefonija	Privatni (Dialpad, Skype)	UDP (i TCP)

2: Nivo aplikacije 15

Zaštita i TCP

TCP & UDP

- Nema enkripcije
- Tekstualne lozinke se prenose preko Interneta

SSL

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

SSL je na nivou aplikacije

- Aplikacije koriste SSL biblioteke, koje "komuniciraju" sa TCP

SSL socket API

- Tekstualna lozinka se šalje enkriptovana preko Interneta

2: Nivo aplikacije 16

Web i HTTP

Termini

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

http://www.cftmn.ac.me/index.html

ime hosta

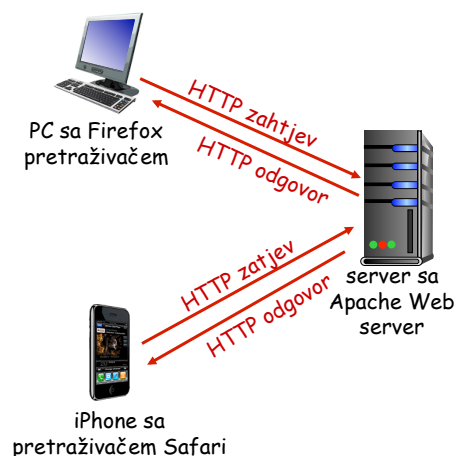
ime puta

2: Nivo aplikacije 17

Pregled HTTP-a

HTTP: hypertext transfer protokol

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



2: Nivo aplikacije 18

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 veza se zatvara

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 nekonzistentni, moraju biti ponovo razmotreni

2: Nivo aplikacije 19

HTTP konekcije

Neperzistentni (neistrajan) HTTP

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

Perzistentni HTTP

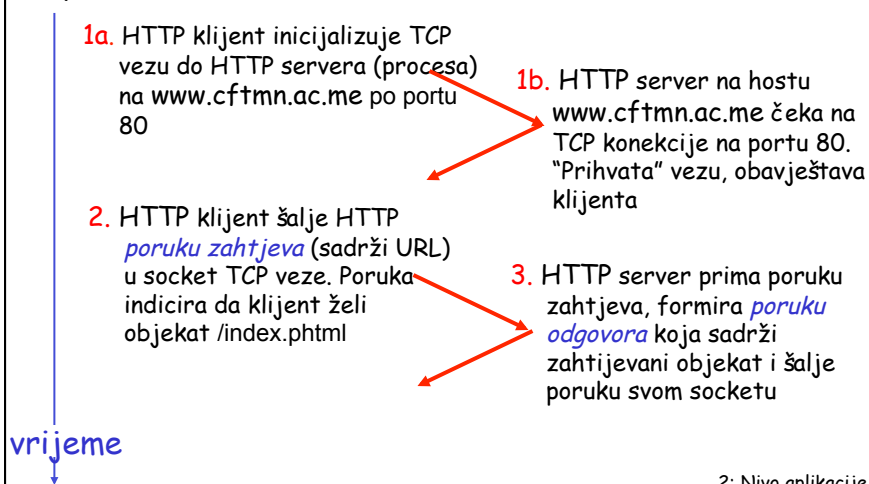
- Više objekata može biti poslato preko jedne TCP veze između klijenta i servera.

2: Nivo aplikacije 20

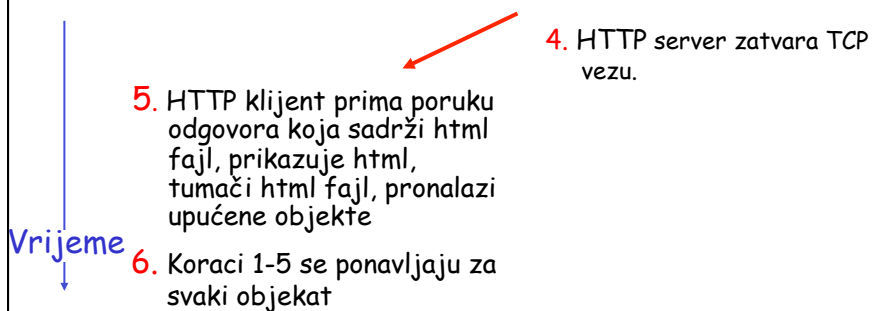
Neperzistentni HTTP

Pretpostavimo da korisnik unese sledeći URL

http://www.cftmn.ac.me/index.html



Neperzistentni HTTP(nastavak)



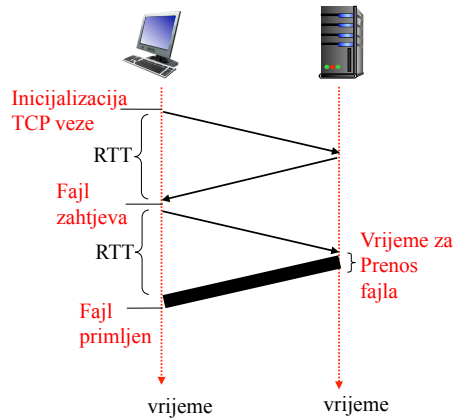
Modelovanje vremena odgovora

Definicija RTT (Round Trip Time): vrijeme prenosa malog paketa 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

ukupno = 2RTT+vrijeme prenosa fajla



2: Nivo aplikacije 23

Persistentni HTTP

Problemi neperzistentnog HTTP-a:

- Zahtjeva 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 klijenta/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

2: Nivo aplikacije 24

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.cftmn.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

2: Nivo aplikacije 25

Tipovi

HTTP/1.0

- GET
- POST
- HEAD
 - Traži od servera da pošalje traženi sadržaj (otklanjanje grešaka)

HTTP/1.1

- GET, POST, HEAD
- PUT
 - Uploaduje fajl na mjesto u Web serveru definisano u URL polju
- DELETE
 - Briše fajl definisan u URL polju

2: Nivo aplikacije 26

HTTP poruka odgovora

statusna linija
(protokol
statusni kod
statusna fraza)

Linije
zaglavlja

podaci, npr.,
zahtijevani
HTML fajl

```
HTTP/1.1 200 OK\r\n
Date: Mon, 27 Feb 2017 08:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Fri, 24 Feb 2017 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 ...
```

2: Nivo aplikacije 27

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

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

2: Nivo aplikacije 28

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

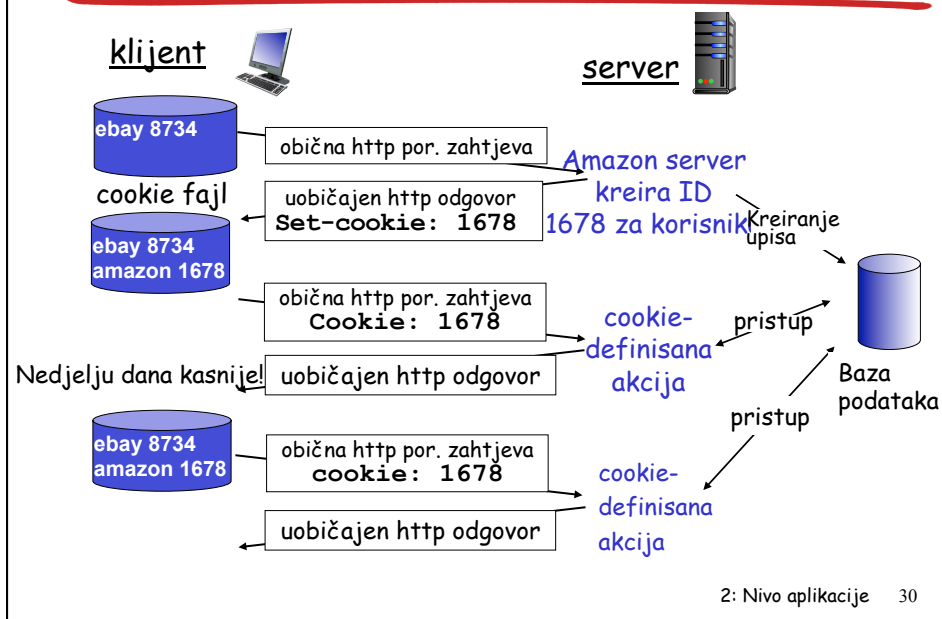
Primjer:

- Neko pristupa Internetu uvijek preko istog PC-a
- 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>

2: Nivo aplikacije 29

Cookies: vode računa o "stanju"(nastavak)



2: Nivo aplikacije 30

Cookies: vode računa o "stanju"(nastavak)

Šta cookies donose:

- ❑ autorizaciju
- ❑ "shopping cards"
- ❑ preporuke
- ❑ stanje korisnikove sesije (Web e-mail)

Pored toga

Cookies i privatnost:

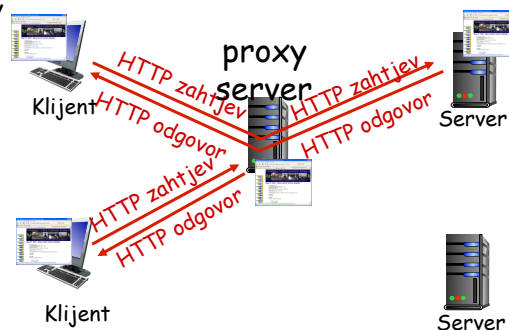
- ❑ Cookies dozvoljavaju sajtu da dosta nauči o korisniku
- ❑ Mogu se dostaviti imena i kontakt podaci
- ❑ Pretraživači koriste cookies da nauče više o korisnicima
- ❑ Kompanije dobijaju dodatne informacije preko weba

2: Nivo aplikacije 31

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>

2: Nivo aplikacije 32

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

2: Nivo aplikacije 33

Primjer:

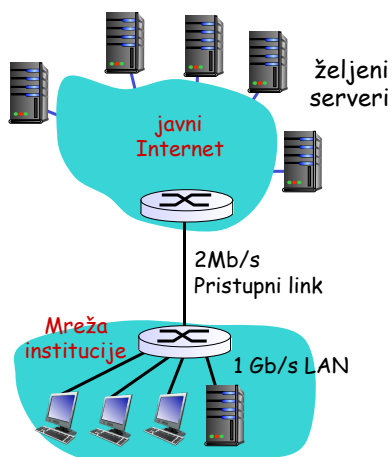
Pretpostavke:

- ❖ 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:

- ❖ Iskorišćenje LAN-a: 0.19%
- ❖ Iskorišćenje pristupnog linka **≈ 95%**
- ❖ Ukupno kašnjenje = kašnjenje na Internetu + kašnjenje u pristupu + LAN kašnjenje
= 2s + minuti + μ s

problem!



2: Nivo aplikacije 34

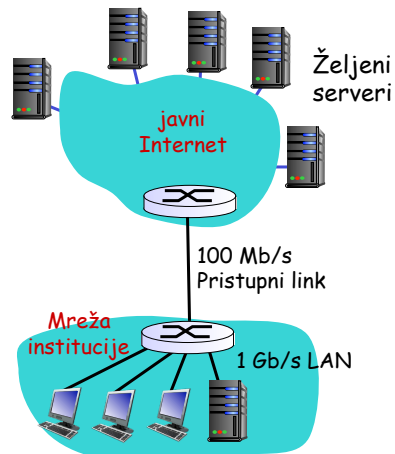
Primjer: brži pristupni link

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:

- ❖ Iskorišćenje LAN-a: 0.19%
- ❖ Iskorišćenje linka= 1.9%
- ❖ Ukupno kašnjenje= Internet kašnjenje + pristupno kašnjenje+ LAN kašnjenje
= 2s + ms + μ s



Troškovi: povećanje brzine pristupa je skupo!

2: Nivo aplikacije 35

Primjer: Lokalni proxy

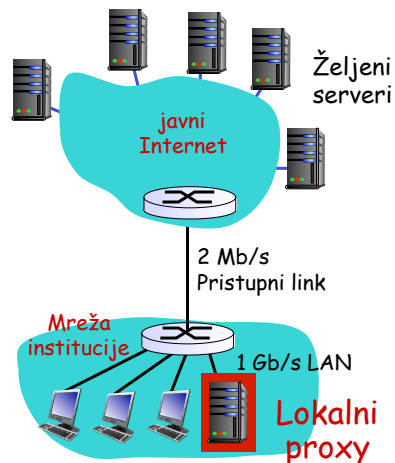
pretpostavke:

- ❖ 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 iskorišćenje: 0.19%
- ❖ Iskorišćenje pristupnog linka= ?
- ❖ Ukupno kašnjenje= ?

Kako izračunati iskorišćenje i kašnjenje?



Troškovi: proxy nije skup!

2: Nivo aplikacije 36

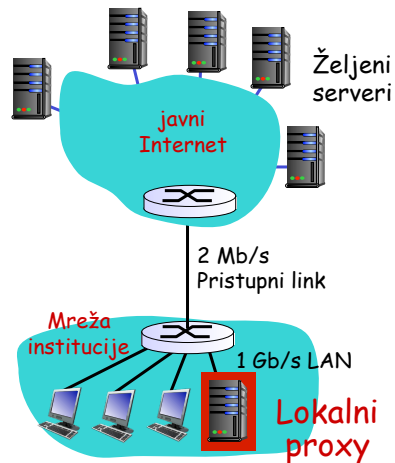
Primjer: Lokalni proxy

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:

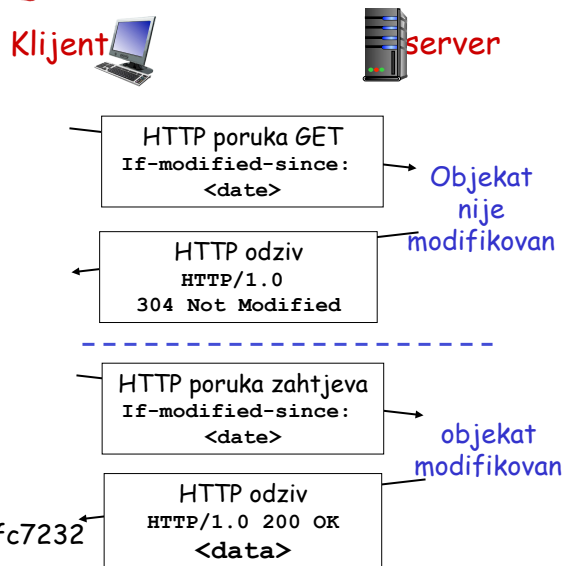
- 60% zahtjeva koristi pristupni link
- ❖ Brzina prenosa preko pristupnog linka = $0.6 * 1.14 \text{ Mb/s} = 1.14 \text{ Mb/s}$
 - iskorišćenje = $1.14 / 2 = 0.57 = 57\%$
- ❖ Ukupno kašnjenje
 - = $0.6 * (\text{kašnjenje od željenih servera}) + 0.4 * (\text{kašnjenje do proxy servera})$
 - = $0.6 (2s) + 0.4 (\sim \text{ms})$
 - = $\sim 1.2 \text{ s}$
 - Manje nego za 100Mb/s pristupni link



2: Nivo aplikacije 37

Conditional GET

- **Cilj:** ne slati objekat ako cache ima up-to-date sačuvanu verziju
- **cache:** 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>



2: Nivo aplikacije 38