

The screenshot displays a desktop environment with several applications open:

- Second Life:** A virtual world interface with options like 'What is Second Life?', 'World Map', 'Shopping', and 'Buy Land'.
- Skype:** A window showing a contact list for 'Keith W. Rows' and options for adding contacts and creating conferences.
- Facebook:** A profile page for 'Keith Rows' with a photo and some text.
- YouTube:** A search results page for 'polytechnic of NYU'.

Below the applications is a diagram illustrating the DHT Network for Public and Private Torrents:

- Public Trackers:** Includes sites like 'Public Discovery Sites' and 'Private Discovery Sites'.
- Private Trackers:** Includes sites like 'Zetund Tracker' and 'Hbtracker Tracker'.
- Peers:** Represented by icons of people connected to the network.
- DHT Network:** A decentralized network where peers can find each other without a central server.

3: Nivo aplikacije 1

Nivo aplikacije

- Principi protokola nivoa aplikacije
- Web
- HTTP

Neke mrežne aplikacije

- 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

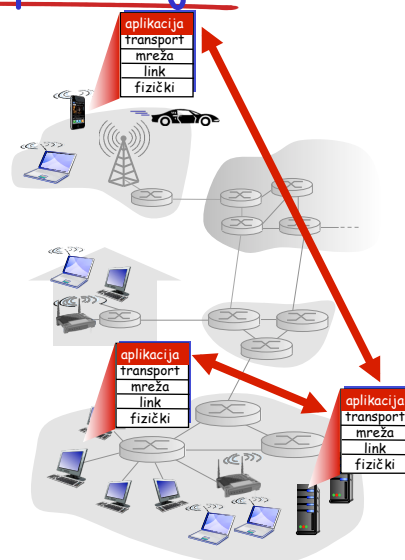
Kreiranje mrežne aplikacije

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



3: Nivo aplikacije 4

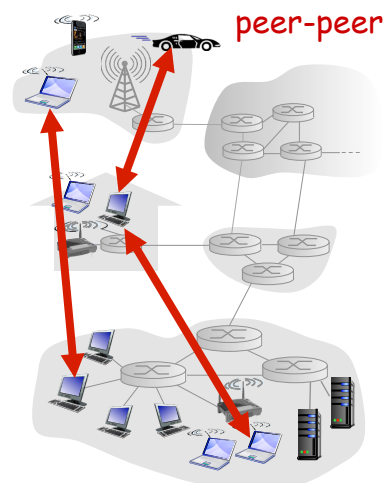
Arhitekture mrežnih aplikacija

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

3: Nivo aplikacije 5

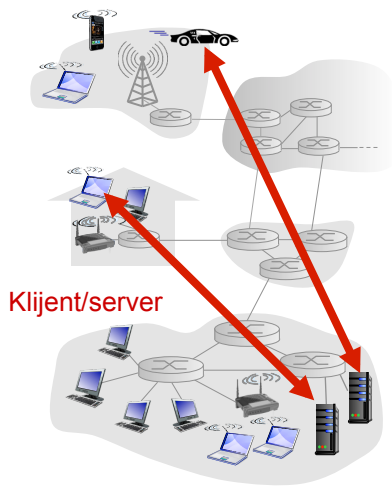
P2P arhitektura

- 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



3: Nivo aplikacije 6

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

Nivo aplikacije 2-7

Hibrid Klijent-server i P2P arhitektura

Skype

- voice-over-IP P2P aplikacija
- centralizovani server: pronalaženje adrese udaljene strane:
- klijent-klijent konekcija 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

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

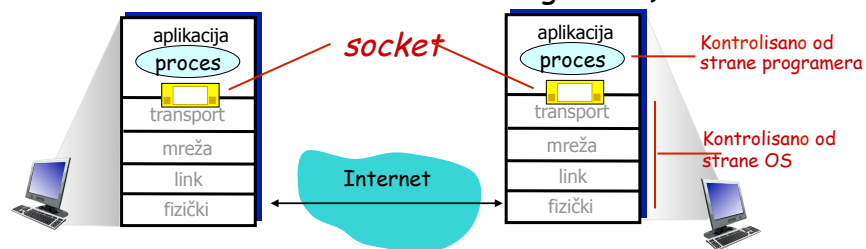
Server proces: proces koji čeka da bude kontaktiran

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

3: Nivo aplikacije 9

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 specifikiranja nekoliko parametara (maksimalna veličina bafera i maksimalna veličina segmenta)



3: Nivo aplikacije 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?
 - ❑ 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

3: Nivo aplikacije 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 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, Viber,...

3: Nivo aplikacije 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

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

3: Nivo aplikacije 14

Servisi transportnih protokola Interneta

TCP servisi:

- ❑ *konektivnost*: uspostavljanje komunikacije se zahtijeva između klijentskih i serverskih procesa
- ❑ *pouzdana 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:

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

P: Zašto oba? Zašto UDP?

3: Nivo aplikacije 15

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 (npr., Dialpad, Skype)	UDP (i TCP)

3: Nivo aplikacije 16

Zaštita i TCP

TCP & UDP

- Nema kriptovanja
- Tekstualne lozinke se prenose preko Interneta

SSL (Secure Sockets Layer)

- 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 kriptovana preko Interneta

3: Nivo aplikacije 17

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 sadrži više referenci objekata
- Svaki objekat se adresira sa **URL (Uniform Resource Locators)**
- Primjer URL:

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

ime hosta

ime puta

3: Nivo aplikacije 18

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



3: Nivo aplikacije 19

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 inkonzistentni, moraju biti ponovo razmotreni

3: Nivo aplikacije 20

HTTP konekcije

Neperzistentni (neistrajan) HTTP

- Najviše jedan objekat je poslat preko TCP konekcije.
- Povlačenje više objekata podrazumijeva otvaranje više konekcija

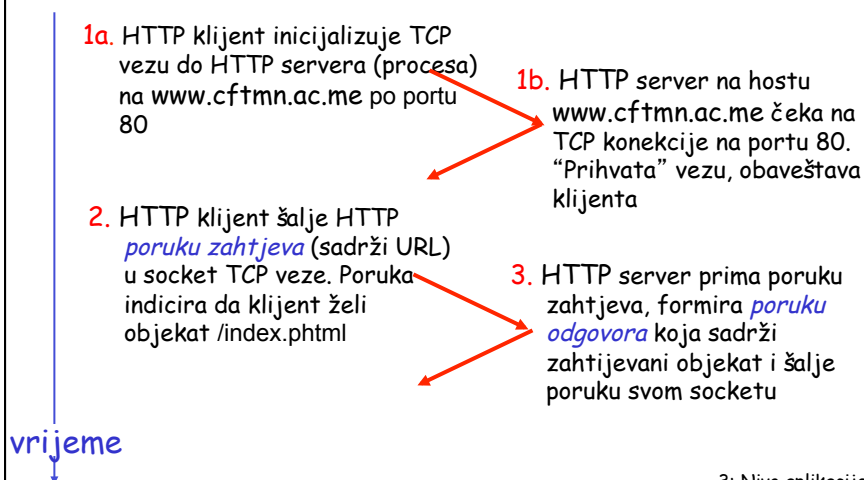
Perzistentni HTTP

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

3: Nivo aplikacije 21

Neperzistentni HTTP

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



3: Nivo aplikacije 22

Neperzistentni HTTP(nastavak)

- Vrijeme ↓
4. HTTP server zatvara TCP vezu.
 5. HTTP klijent prima poruku odgovora koja sadrži html fajl, prikazuje html, tumači html fajl, pronalazi upućene objekte
 6. Koraci 1-5 se ponavljaju za svaki objekat

3: Nivo aplikacije 23

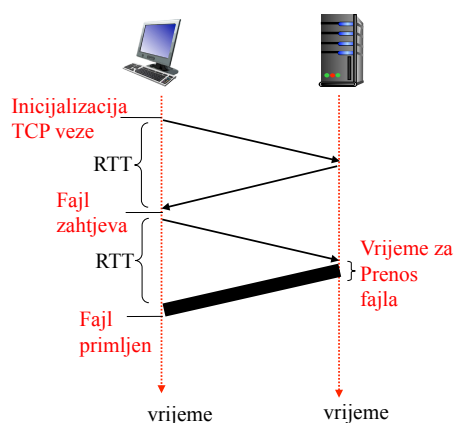
Modelovanje vremena odgovora

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 = $2RTT + \text{vrijeme prenosa fajla}$



3: Nivo aplikacije 24

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 25

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

3: Nivo aplikacije 26

Tipovi

HTTP/1.0

- GET
- POST
- HEAD
 - Pita servera da pusti 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

3: Nivo aplikacije 27

HTTP poruka odgovora

statusna linija
(protokol statusni kod statusna fraza)

Linije
zaglavlja

```
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 ...
```

podaci, npr.,
zahtijevani
HTML fajl

3: Nivo aplikacije 28

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

3: Nivo aplikacije 29

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

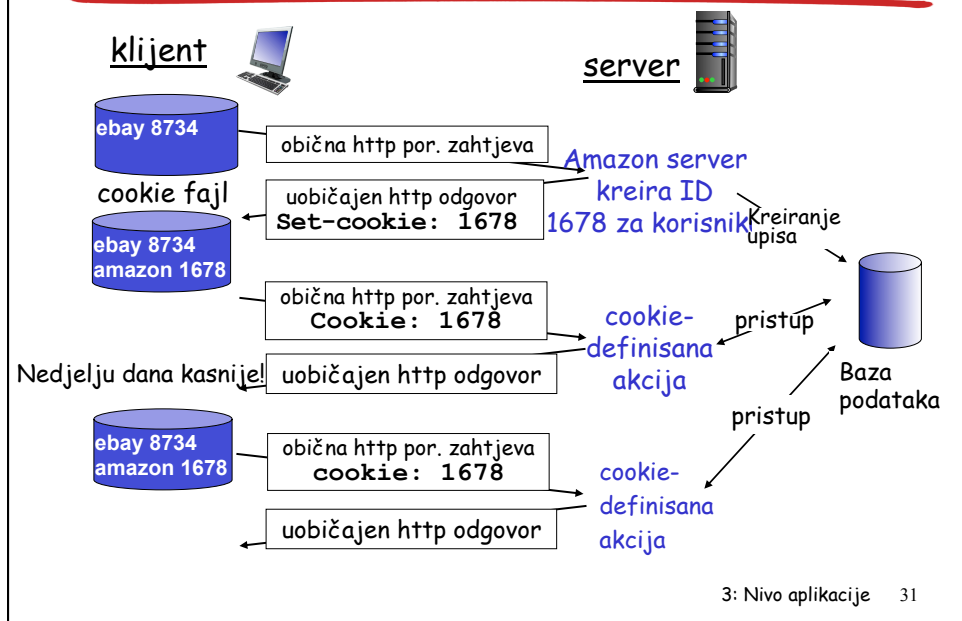
Primjer:

- 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>

3: Nivo aplikacije 30

Cookies: vode računa o "stanju" (nastavak)



Cookies: vode računa o "stanju" (nastavak)

Šta cookies donose:

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

Pored toga

Cookies i privatnost:

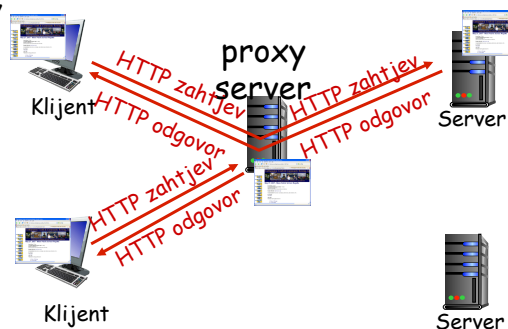
- Cookies dozvoljavaju sajtu da dosta nauči o korisniku
- Mogu se dostaviti imena i kontakt podaci
- Pretraživači koriste cookies da nauče više o korisnicima
- Kompanije dobijaju dodatne informacije preko weba

3: Nivo aplikacije 32

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 33

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 34

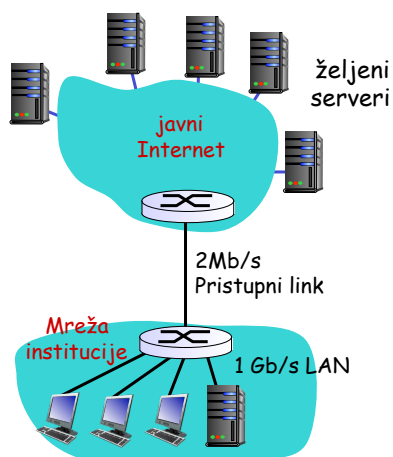
Primjer:

Pretpostavke:

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

Posljedice:

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



3: Nivo aplikacije 35

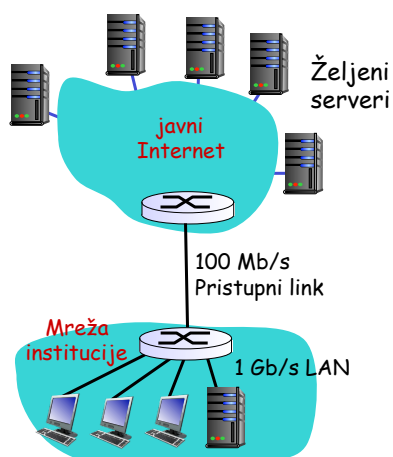
Primjer: brži pristupni link

pretpostavke:

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

posljedice:

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



Troškovi: povećanje brzine pristupa je skupo!

3: Nivo aplikacije 36

Primjer: Lokalni proxy

pretpostavke:

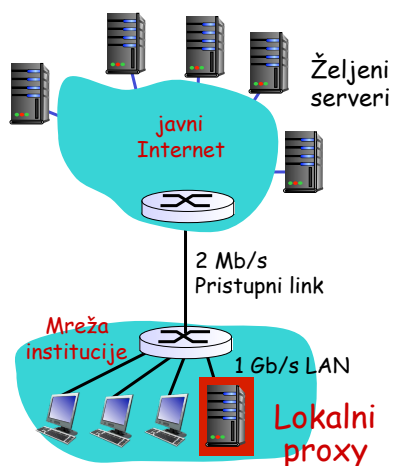
- ❖ Srednja veličina objekta: 100000 bita
- ❖ Srednja brzina zahtjeva: 19/sec
- ❖ Srednja brzina: 1.9 Mb/s
- ❖ RTT od rutera institucije do željenog servera: 2 s
- ❖ Brzina pristupa: 2 Mb/s

posledice:

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

Kako izračunati iskorišćenje i kašnjenje?

Troškovi: proxy nije skup!



3: Nivo aplikacije 37

Primjer: Lokalni proxy

Izračunavanje iskorišćenja i kašnjenja:

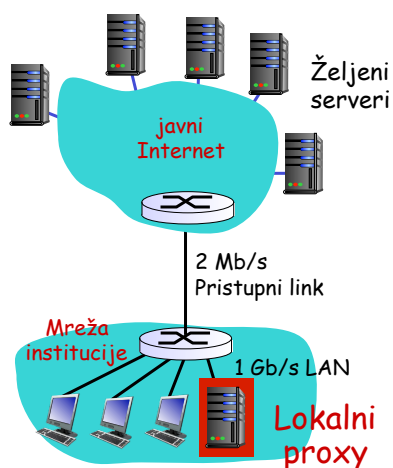
- Pretpostavimo da je vjerovatnoća pogađanja 0.4
 - 40% zahtjeva se posluži na proxy serveru, 60% zahtjeva na željenom serveru

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 100Mb/s pristupni link



3: Nivo aplikacije 38

Conditional GET

