

Aplikacijska plast

Omrežne aplikacije

So glavni razlog za povezovanje računalnikov v omrežja (raison d'être)!

- ne potrebuje nujno GUI
- aplikacijski protokoli so običajno del aplikacije

Razlikovanje med:

- aplikacijo v računalniku, ki uporablja omrežje (kos programske opreme)
- omrežno aplikacijo (porazdeljena aplikacija iz več komponent)

Programer mora implementirati aplikacije, ki tečejo na različnih sistemih in komunicirajo med seboj. Podrobnosti omrežja zanj niso pomembne

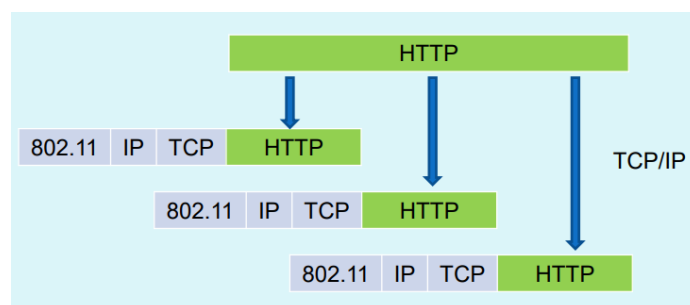
Protokoli aplikacijske plasti

Arhitekture aplikacij:

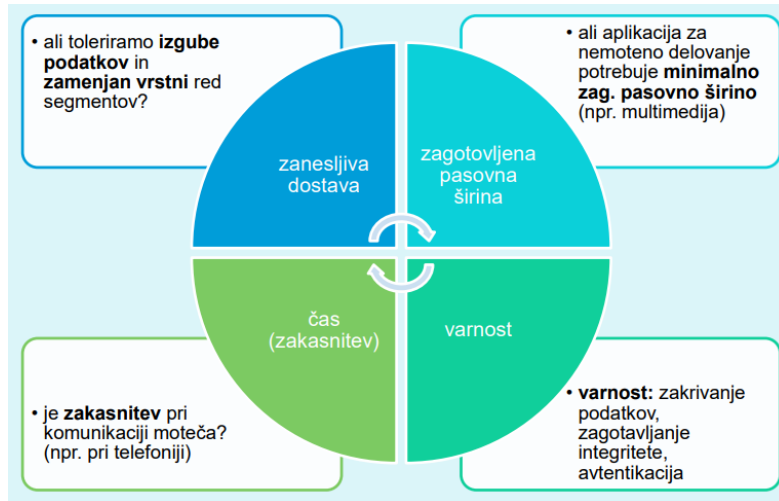
- strežnik in odjemalec: strežnik ves čas dostopen, odjemalci so priključeni le občasno (http, ftp, telnet, e-mail), najbolj pogosta
- peer-to-peer (P2P): končni sistemi direktno komunicirajo med seboj, člani so lahko nedostopni (BitTorrent, eMule)
- mešana arhitektura (Skype, takojšnje sporočanje)
- aplikacijski protokoli so običajno berljivi uporabniku
- lastništvo protokolov:
 - **javni** (odprti) protokoli:
 - HTTP (RFC 2616), SMTP, BitTorrent, specifikacije v dokumentih RFC
 - **lastniški** (zaprti, angl. proprietary) protokoli, npr. Skype
 - ni javnih specifikacij
 - delovanje ugotavljamo z obratnim inženiringom

Aplikacijska sporočila

Aplikacijsko sporočilo je protokolarna enota na aplikacijski plasti. Običajno je deljena in enkapsulirana v več paketov.



Kateri transportni protokol uporabiti?

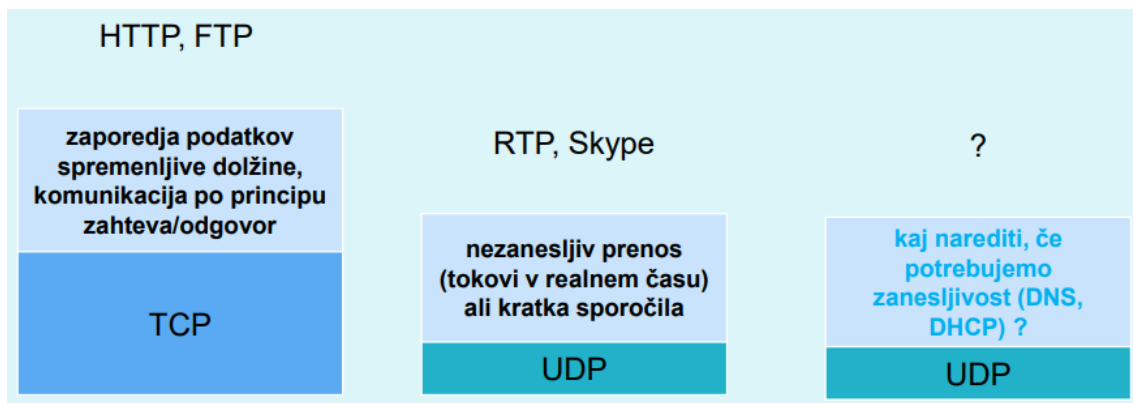


Zahteve aplikacij po storitvah transporta

Aplikacija	primeri aplikacijskih protokolov	dovoljena izguba	potrebna minimalna časovna širina	časovna občutljivost	običajen transportni protokol
prenos datotek	FTP	ne	ne (elastična)	ne	TCP
e-pošta	SMTP, POP3, IMAP	ne	ne (elastična)	ne	TCP
splet	HTTP	ne	ne (elastična)	ne	TCP
oddaljen dostop	telnet	da	~10 kb/s	da	TCP
multimedija (realni čas)	RTP, HTTP	da	10 kb/s – 5 Mb/s	da	TCP/UDP
multimedija (shranjena)	RTP, HTTP	da	10 kb/s – 5 Mb/s	da	TCP/UDP
interaktivne igre	HTTP, lastniški	da / ne (?)	~10 kb/s	da	TCP/UDP
IP telefonija	SIP, RTP, lastniški	da	100 kb/s – 3 Mb/s	da	TCP/UDP

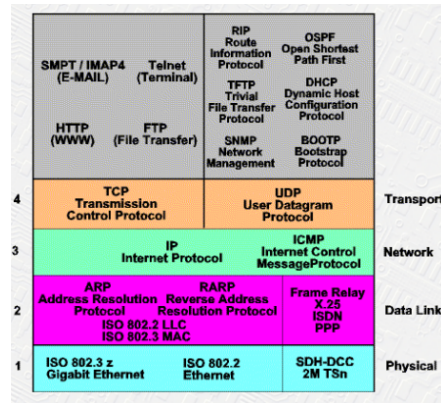
Elastična pasovna širina pomeni, da ni meje ampak se, če dovolj dolgo čakamo vse normalno naloži.

Izbira transportnega protokola in zanesljivost dostave

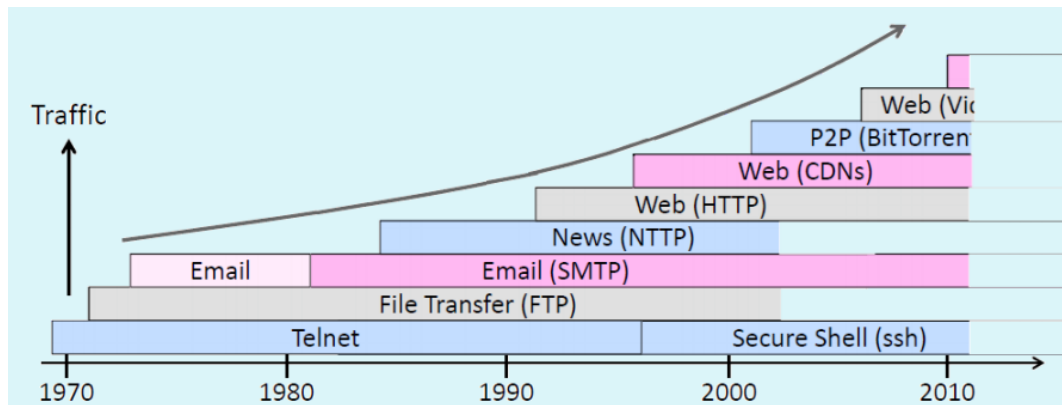


Če potrebujemo zanesljivost izberemo TCP in/ali jo sami implementiramo.

Aplikacijski protokoli



Razvoj protokolov skozi čas

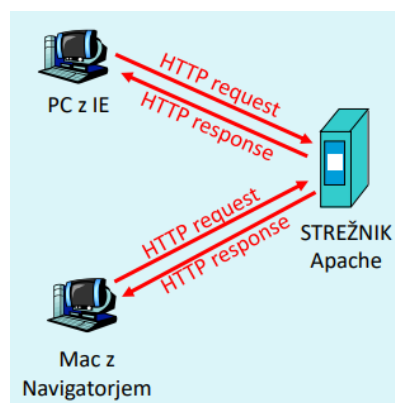


Protokol HTTP

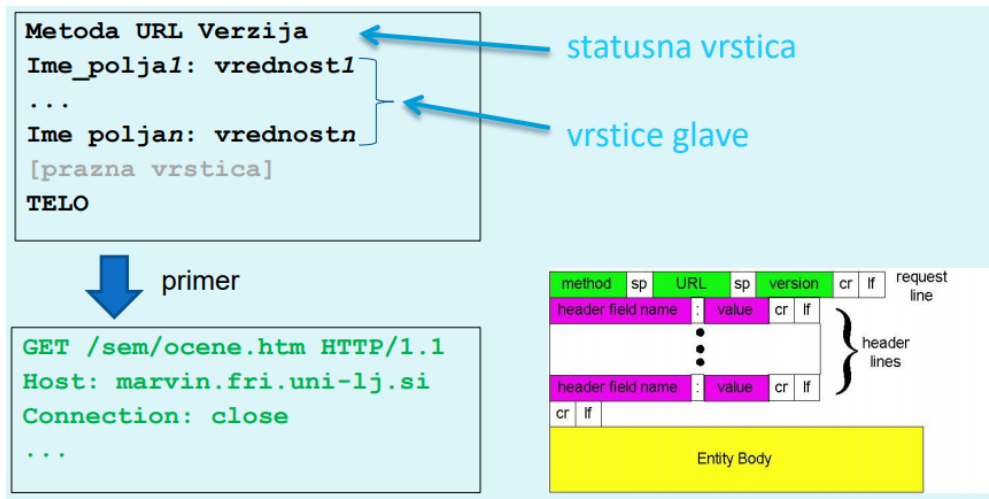
Splet in HTTP

Pojavi se v 1990-ih, revolucionarna aplikacija, omogoča:

- dostop do vsebin **na zahtevo**
- vsak lahko oglašuje
- iskanja, povezave, grafika, vmesniki, multimedija
- specifikaciji : RFC 1945 (HTTP 1.0) in RFC 2616 (HTTP 1.1)
- delovanje:
 - odjemalec naslovi TCP **zahtevo** (request) na vrata 80 strežnika
 - strežnik vrne http **odgovor** (response)
 - TCP poskrbi za potrditve, ponovitve, vrstni red
 - protokol brez stanj (stateless - ne hrani informacije o stanju)



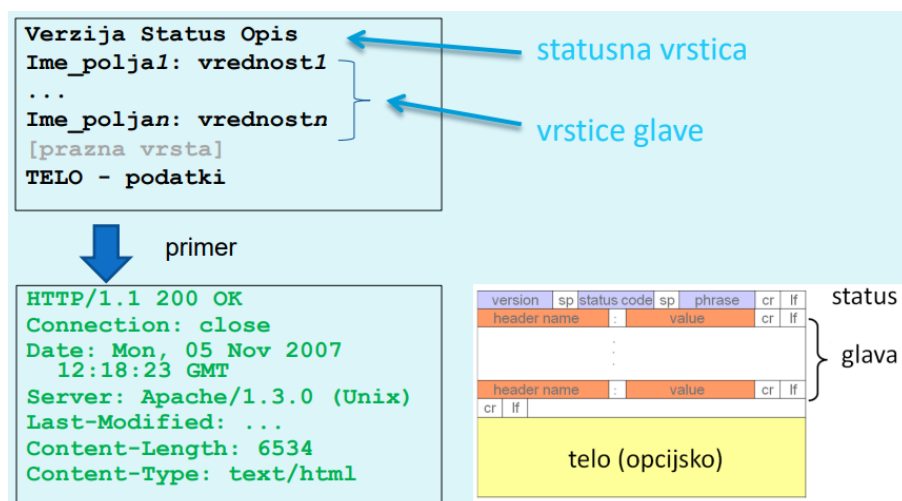
Oblika sporočila HTTP request



Metode pri HTTP request

GET	zahteva objekta
POST	zahteva objekta + poslane vrednosti (obrazci) <ul style="list-style-type: none"> obrazec lahko uporabi tudi metodo GET, vrednosti parametrov pa pošilja kot podaljšan naslov (http://...../search/rsr.aspx?lang=slv&id=31318)
HEAD	zahteva, na katero strežnik odgovori z odgovorom brez zahtevanega objekta (uporabno za razhroščevanje)
PUT	(HTTP 1.1) – nalaganje na strežnik (upload)
DELETE	(HTTP 1.1) – brisanje s strežnika
TRACE	razhroščevanje (echo-odmev zahtevka, podobno PING)
CONNECT	povezava preko medstrežnika
OPTIONS	povpraševanje o možnih opcijah pri zahtevku

Oblika sporočila HTTP odgovor

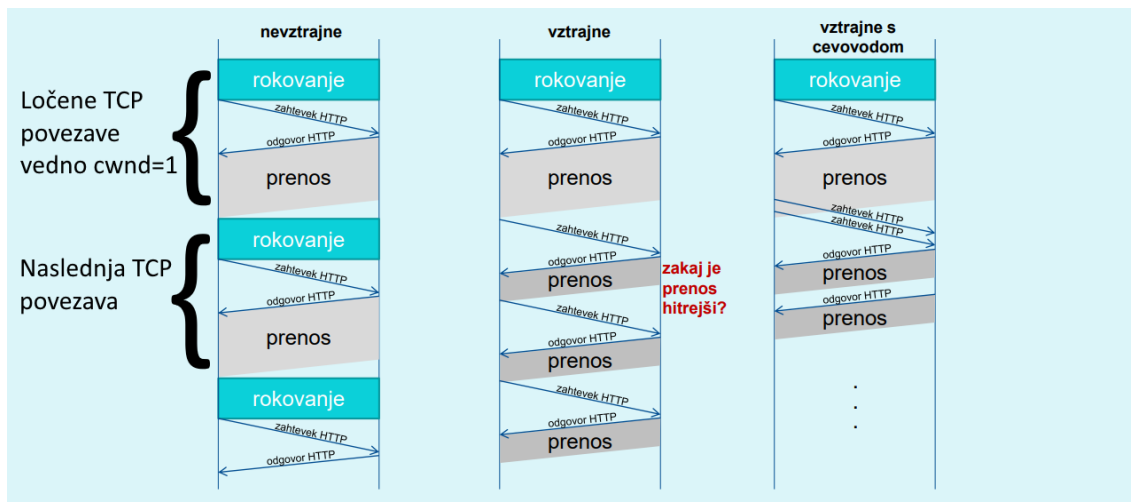


Statusi v HTTP odgovorih

- **1xx**: informativne kode
- **2xx**: uspešno
 - 200: OK
- **3xx**: preusmeritev
 - 301: Moved Permanently prestavljen dokument
- **4xx**: napake pri odjemalcu
 - 400: Bad Request – sintaksa
 - 404: Not Found – ni dokumenta
- **5xx**: napake na strežniku
 - 500: Internal Server Error
 - 505: HTTP Version Not Supported

Vrste HTTP povezav

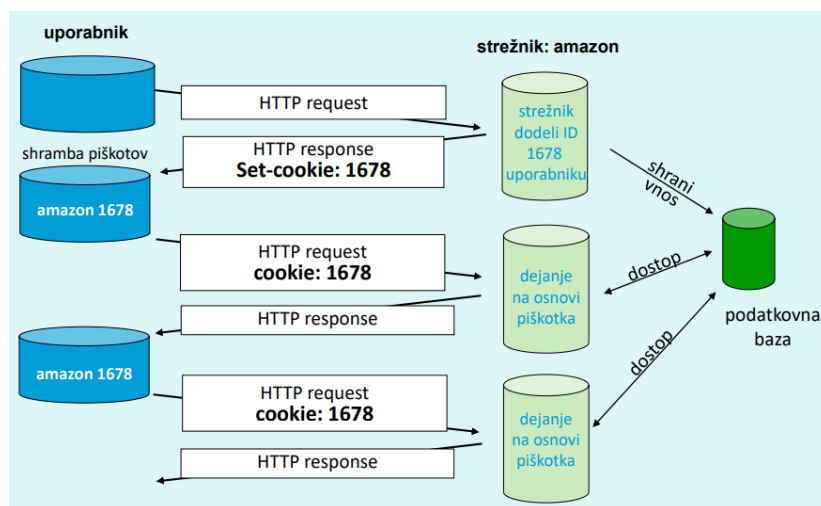
1. **nevztrajne (nonpersistent)**: za vsak prenašani objekt (stran, sliko) se vzpostavi nova TCP povezava
 - 2 RTT/objekt (rokovanje + prenos)
2. **vztrajne (persistent)**: strežnik uporabi isto povezavo za pošiljanje več objektov, strežnik pusti povezavo odprto
 - 1 RTT/objekt (prenos)
3. **vztrajne s cevovodom (persistent, pipelined)**: tekoče pošiljanje več zahtev naenkrat, brez čakanja na prejem prejšnjih



Vztrajno je hitrejšo, saj imamo le 1 TCP povezavo(=samo 1 rokovanje) hkrati pa se povečuje tudi cwnd. Pri nevztrajnem pa za vsako sporočilo vzpostavljamo novo povezavo (+1 RTT za rokovanje) in cwnd je vedno 1.

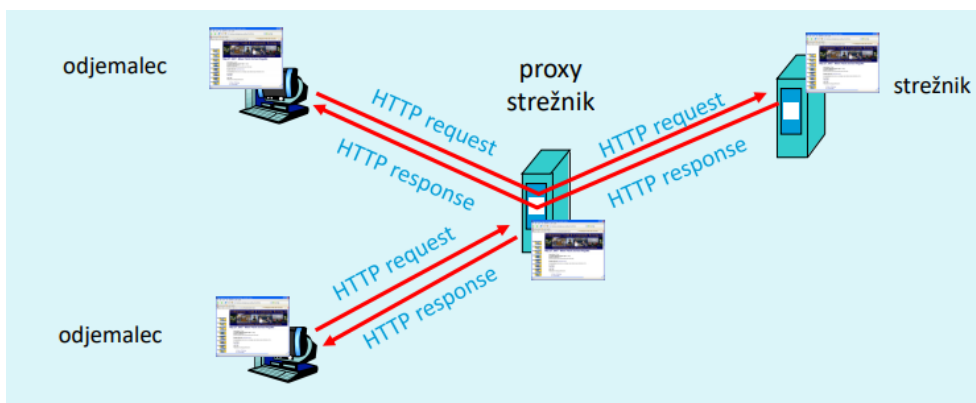
Uporaba piškotkov

- HTTP ne pomni stanja povezav (stateless), zato med seboj ne razlikuje odjemalcev in ne pomni zgodovine
- nad plastjo HTTP (brez stanja) se ustvari sejna plast (s stanji)
- bogatejša uporabniška izkušnja:
 - avtorizacija
 - nakupovalni vozički
 - priporočila
 - shranjeni podatki o uporabnikih
- problem: varovanje zasebnosti
- rešitev: uporaba piškotkov (identifikator, ki ga strežnik dodeli uporabniku in ga uporabnik lokalno shrani).
- Elementi:
 - datoteka z ID piškota shranjena na uporabnikovem računalniku
 - strežnik hrani evidenco o uporabnikih izdanih piškotov
 - ID piškota dodamo v glavo zahteve
 - ID piškota dodamo v glavo odgovora

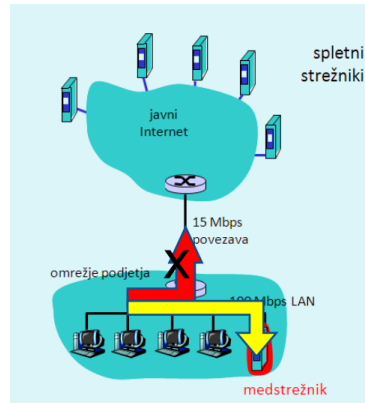


Medstrežnik (proxy strežnik)

Web cache, proxy strežnik (navadno pri ISP-ju) odgovarja na zahteve namesto strežnikov, ima svoje kopije spletnih strani (če je nima, jo zahteva od pravega strežnika). Odjemalec mora biti ustrezno konfiguriran.



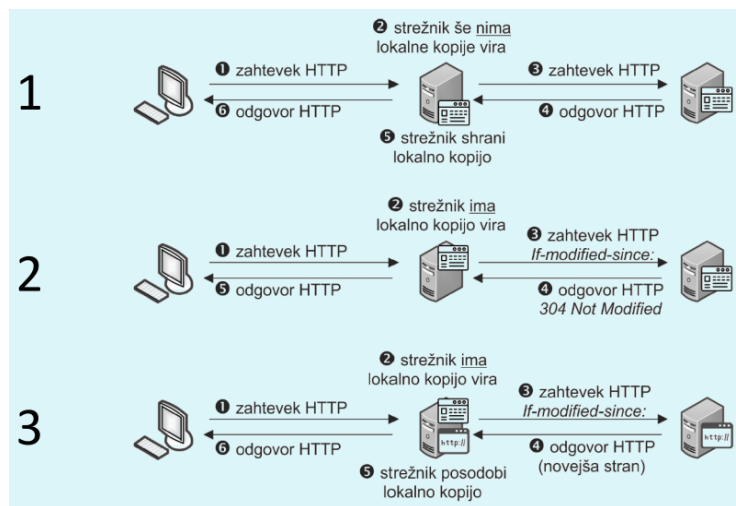
Zakaj posredniki?



- medstrežnik igra vlogo odjemalca in strežnika
- omogoči hitrejši odgovor odjemalcu
- manj prometa na dostopni povezavi do javnega omrežja (če to plačujemo, lahko tudi nekaj privarčujemo)
- pogojna zahteva (če je prišlo do spremembe strani)

- vrstica glave:
If-modified-since: Wed, 31 Oct 2007 09:32:22
- strežnik pošlje novo stran ali
HTTP/1.1 304 Not Modified (prazno telo)

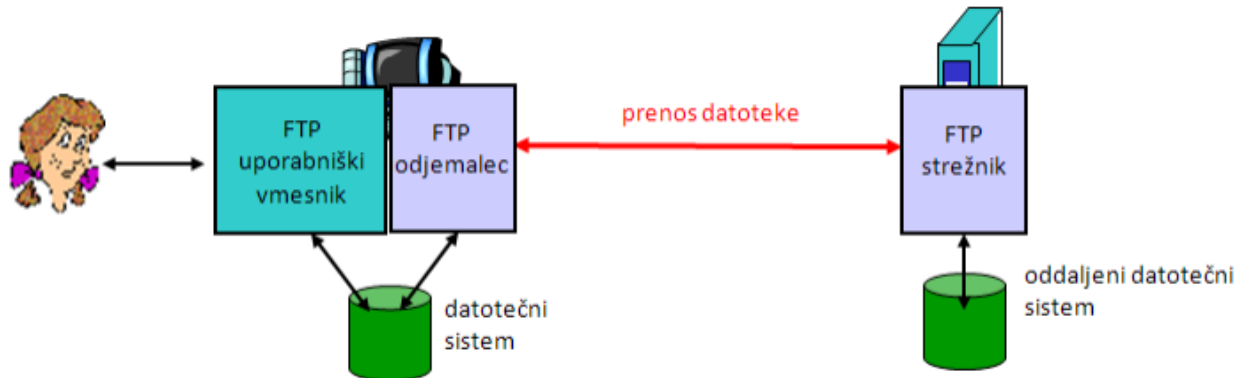
Možni scenariji



Protokol FTP

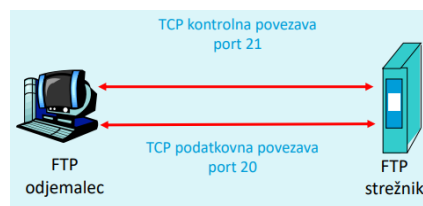
Prenos datotek (File Transfer Protocol)

- Prenos datotek z oddaljenega računalnika k uporabniku in obratno • Protokol, ki hrani stanje povezave: strežnik ve, kdo je odjemalec (avtentikacija), spremlja, kateri imenik pregleduje itd.



2 ločeni TCP povezavi na FTP strežnik, za odpiranje druge povezave skrbi pasivni FTP:

- vrata TCP 21 (kontrolna povezava): ukazi za prenos datotek, uporabniško ime/geslo, menjava map...
- vrata TCP 20 (prenos podatkov) na zahtevo odjemalca strežnik odpre povezavo, po kateri prenaša podatke



Sporočila FTP protokola

Nadzorna povezava uporablja 7-bitne ASCII ukaze.

- Ukazi in odgovori
 - USER ime
 - PASS geslo
 - LIST
 - RETR ime_datoteke
 - STOR ime_datoteke
- Odgovori strežnika
 - 331 Username OK, password required
 - 125 Data connection open, transfer starting
 - 452 Error writing file
 - 425 Can't open data connection

Elektronska pošta (SMTP, POP3, IMAP)

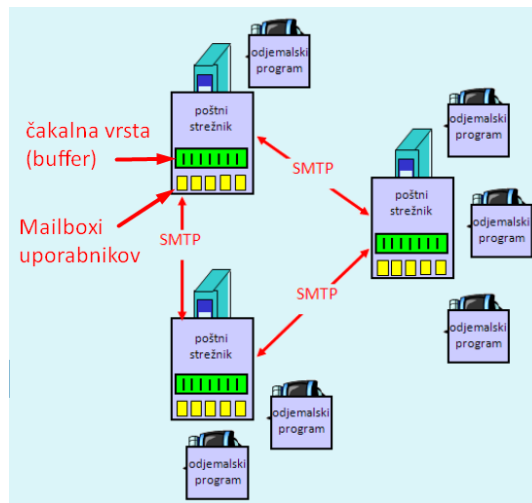
Elektronska pošta kot aplikacija

Prednosti:

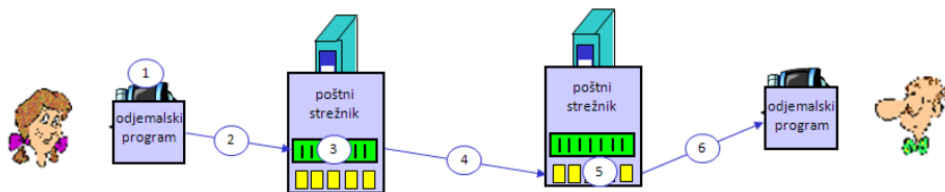
- hitra, poceni, enostavna dostava
- priponke, povezave, HTML

Arhitektura e-poštne aplikacije:

- **poštni strežniki**, ki hranijo:
 - poštno predale
 - imajo izhodno vrsto sporočil
- **odjemalski programi** (user agent): tekstovni, grafični
- **protokol** za prenos sporočil (SMTP)



Pot elektronskega sporočila



1. uporabnik uporabi odjemalca, da sestavi sporočilo
2. odjemalec pošlje sporočilo strežniku
3. strežnik shrani sporočilo v izhodno vrsto
4. strežnik vzpostavi TCP povezavo s poštnim strežnikom prejemnika in prenese sporočilo
5. prejemnikov poštni strežnik shrani sporočilo v ustrezen poštni predal
6. prejemnik uporabi svoj odjemalski program za prenos in branje sporočila

Simple Mail Transfer Protocol (SMTP), star več kot 30 let.

- ```
Delivered-To: <smith@aspalliance.com>
Received: by 10.220.195.140 with SMTP id ecl2csl503ovcb;
 Thu, 10 Nov 2011 05:52:18 -0800 (PST)
Received: by 10.182.152.37 with SMTP id uvwsm18527r9obb.53.132093
 Thu, 10 Nov 2011 05:52:17 -0800 (PST)
Return-Path: <mllout@maillist.codeproject.com>
Received: from mail.maillist.codeproject.com (mail.mailhost.usps.net) [66.158.102.1] by mx.google.com with ESMTP id n2sil4870lzyhm.88.2011.11.11.
 Thu, 10 Nov 2011 05:52:17 -0800 (PST)
Received-SFPP: pass (google.com: domain of mllout@maillist.codeproje
Message-ID: <4ebdbd71.02bebec0a.3783.6bdSMTPIP ADDED=>mx.google.com>
Received: from jobjel.jobjelcommunication.com ([192.168.s.180])
 by mail.maillist.codeproject.com (Postfix) with ESMTP ID B6E0A
 for <smith@aspalliance.com>; Thu, 10 Nov 2011 08:51:45 -0800
MIME-Version: 1.0
From: "The Code Project"
 <mllout@maillist.codeproject.com>
To: "Steve Smith"
 <smith@aspalliance.com>

Date: 10 Nov 2011 08:52:16 -0500
Subject: CodeProject | Daily News ~ Clean up after Visual Studio
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: base64
```
- ```
PCFEtONUWVFIEhtUtUgVFCVEtIDICItLy9XMOMvLORURCBIVEIMIDQumDEGvhJn  
bmldPGdlvbmfSY9fYttIgmHODHA6ly93dcidzHub3JuNLRSL2oBwwOLZvwXjn  
lmR0CZI+IAOKGPbnmVwDqosAgVhdZ48gdUGubAG+rGFpbhkTwmV3cyAGLSdBDbGVh  
bi1kChznRLci1BEwxHNjVwgUSRIJyGUgUSrl2YGvPC9oaXRzRT4NCjcvtZXRHghOdhAdZFxi  
AYKxviNbvnRlbmgVhlWSig72UGdvGDvdOdIGv4DC9odGsIoSyBaEGvTzOVWFml  
bzGOHDOWULStEIC8+DO8gL2lhVygDDoSITOSMtTygzMezxMSDsOTPKFGvgYZHK+dQ8  
Zy24THNOUEpSEpkXAiwBgVGfSm5vbmhU72se3zb3HuNLPvlvDGvYPl1lvhwBgnytlmu  
ZYsIkcgRhlgddGEIERhaNxISe134dgYCgoQ2xlYw4ggXkgXgYL2onCGvmldzvTFNG  
dWRqbzwzgRDGSOGvPRKHMOHEwlTHRSG5oyINrkLEqy3ZNzi14NCjwcLnSQomCVZHke
```

```
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
```

vrstice glave

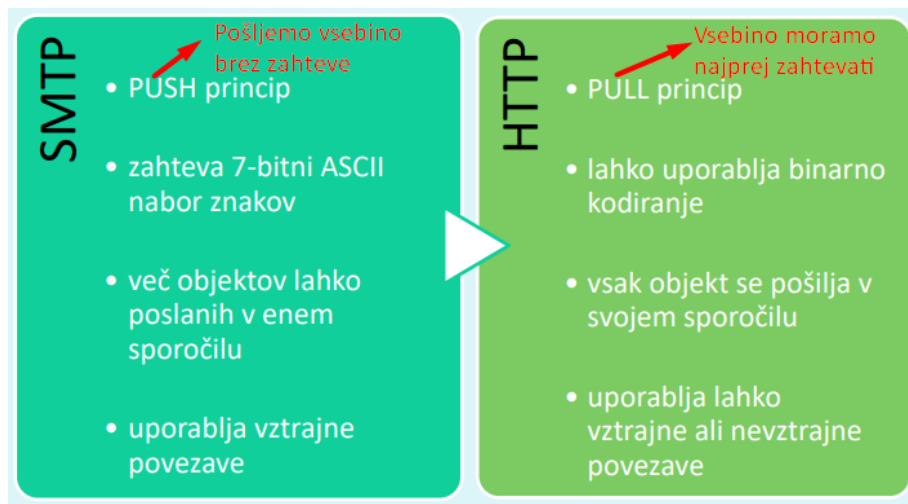
Celotna glava poštnega sporočila

```
Received: from ns.fri.uni-lj.si ([212.235.188.18]) by fri-postar1.fri1.uni-lj.si with Microsoft SMTPSVC(6.0.3790.4675);
Sun, 13 Feb 2011 18:27:58 +0100
Received: from AG (unknown [212.235.188.19])
by ns.fri.uni-lj.si (Postfix) with ESMTP id 17586833A
for <zoran.bosnic@fri.uni-lj.si>; Sun, 13 Feb 2011 18:27:58 +0100 (CET)
Received: from localhost ([212.235.188.18]) by AG with Microsoft SMTPSVC(6.0.3790.4675);
Sun, 13 Feb 2011 18:26:55 +0100
X-Virus-Scanned: amavisd-new at fri.uni-lj.si
Received: from ns.fri.uni-lj.si ([127.0.0.1])
by localhost (ns.fri.uni-lj.si [127.0.0.1]) (amavisd-new, port 10024)
with ESMTP id CSPiVHX2mvZ8 for <zoran.bosnic@fri.uni-lj.si>;
Sun, 13 Feb 2011 18:26:52 +0100 (CET)
Received: from web4 ([198.115.93.97]) by web4.editorialmanager.com with Microsoft SMTPSVC(6.0.3790.4675);
Sun, 13 Feb 2011 12:26:39 -0500
MIME-Version: 1.0
From: "Knowledge and Information Systems" <srilakshmi.patru@springer.com>
Sender: em.kais.0.210cb1.358f57e1@editorialmanager.com
To: "Zoran Bosni?" <zoran.bosnic@fri.uni-lj.si>
Date: 13 Feb 2011 12:26:39 -0500
Subject: Decision on your Manuscript #KAIS-2487
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 7bit
Message-ID: <WEB40glxRhrlJANJQ0w0000f@web4.editorialmanager.com>
X-OriginalArrivalTime: 13 Feb 2011 17:26:39.0849 (UTC) FILETIME=[2C987D90:01C8CBA3]
Return-Path: em.kais.0.210cb1.358f57e1@editorialmanager.com
X-AntiVirus: checked (incoming) by AntiVir MailGuard (Version: 10.0.1.38; AVE: 8.2.4.166; VDF: 7.11.3.52)
```

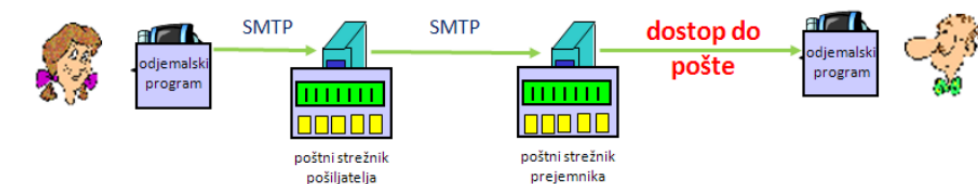
oznake
prejemnih
strežnikov

podatki o
sporočilu

Primerjava SMTP in HTTP



Dostop do poštnega predala



- SMTP: dostava pošte do prejemnikovega strežnika
- POP3, IMAP, HTTP: protokoli za prejem pošte s poštnega strežnika
 - POP: Post Office Protocol [RFC 1939]: avtentikacija in prenos
 - IMAP: Internet Mail Access Protocol [RFC 1730]: avtentikacija, prenos, organizacija po mapah, naprednejše storitve
 - HTTP: gmail, Hotmail, Yahoo! Mail, etc.

Primerjava protokolov

- POP
 - preprost, omejena funkcionalnost, TCP vrata 110
 - 3 faze: avtorizacija, prenos sporočil in oznak, posodabljanje (QUIT)
 - slabosti: lokalno urejanje pošte, ne hrani stanja med sejami
- IMAP
 - kompleksen, zahtevnejši, več funkcionalnosti
 - uporabnik lahko določi mape na strežniku
 - možen prenos le delov sporočil
 - hrani sejo
 - Večja obremenitev strežnika
- HTTP
 - brskalnik, dostop od koderkoli, brezplačni ponudniki
 - mape kot pri IMAP
 - skripte na HTTP strežniku komunicirajo s poštnim strežnikom

Primer POP3 ukazov

```
S: +OK POP3 server ready
O: user zoranb                uporabnik se predstavi
S: +OK
O: pass tralala                nezaščiten!
S: +OK user successfully logged on

O: list
S: 1 678                      1. sporočilo je veliko 678B
S: .                          ni več sporočil
O: retr 1                      prenesi sporočilo 1
S: <vsebina sporočila>
S: .
O: dele 1                      briši sporočilo 1
O: quit
S: +OK POP3 server signing off
```

Imenska storitev - DNS

DNS (Domain Name System)

Naprave identificiramo z imenom (www.google.com) in IP številko

Storitve DNS:

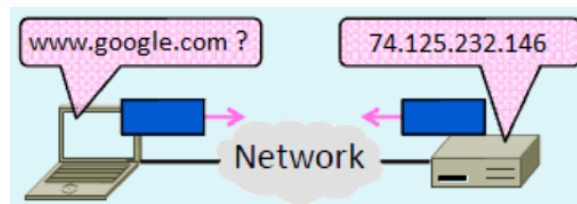
1. **preslikovanje** imen v IP naslove

- lahko tudi več imen v isti IP (transparentnost)
- lahko tudi isto ime v različne naslove IP (porazdeljevanje bremena)

2. porazdeljena (hierarhična) **podatkovna zbirka**

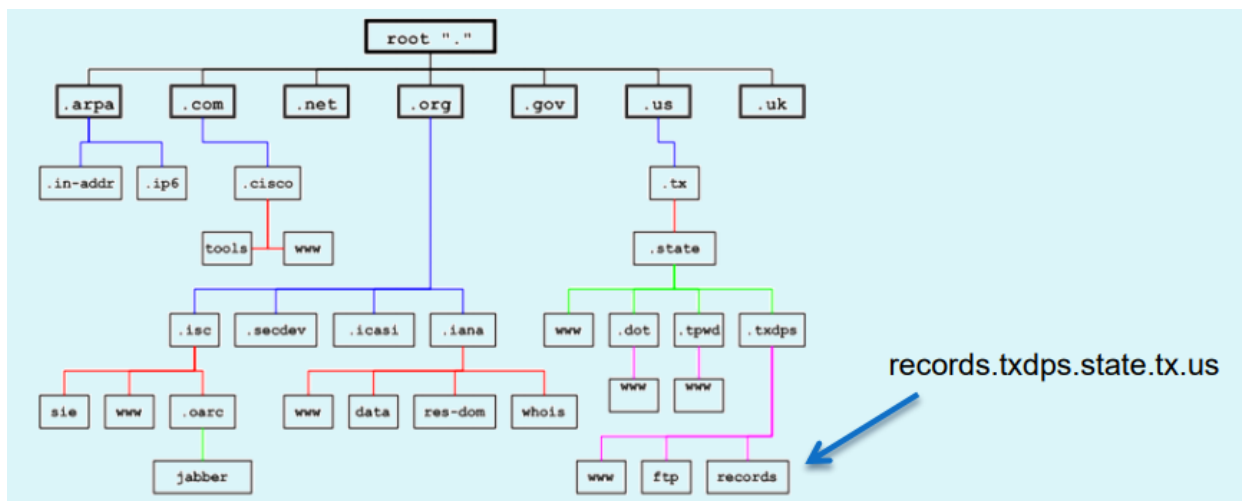
3. **protokol za poizvedovanje po zbirki**

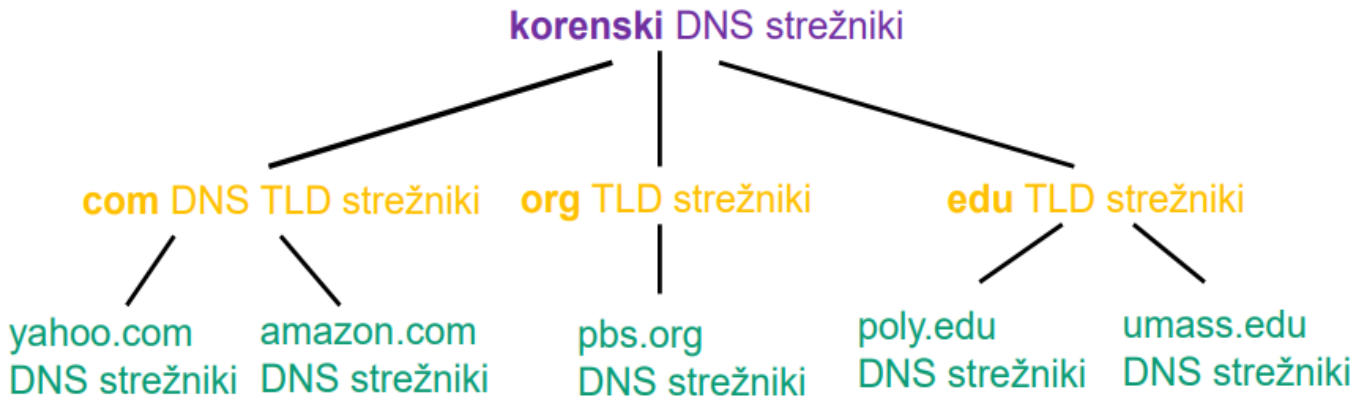
Zakaj bi več imen vodilo v isti IP? Da ni single point of failure in za load balancing.



Hierarhična organizacija

- Če bi imeli en centralni strežnik:
 - enotna točka odpovedi, oddaljena podatkovna baza, velik obseg prometa, težko vzdrževanje
 - rešitev ni skalabilna
- rešitev - hierarhična organizacija **imen strežnikov** in **povpraševanj**



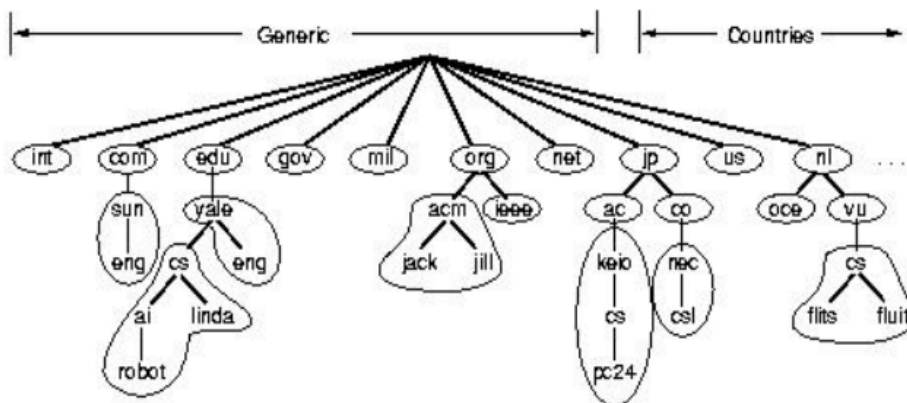


Uporabnik, ki potrebuje IP za www.amazon.com:

- povpraša korenski strežnik po naslovu TLD strežnika za domeno .com
- povpraša .com TLD DNS strežnik po naslovu avtoritativnega strežnika amazon.com
- povpraša strežnik podjetja amazon.com o IP naslovu za www.amazon.com

Hierarhična organizacija strežnikov

- 13 korenskih strežnikov (A-M), vsak je replicirana gruča, ti so privzeto vneseni v OS
- TLD (Top-Level Domain) strežniki:
 - generične domene: 7 prvotnih (com, edu, gov, mil, org, net, biz) in ~1500 dodatnih (info, aero, museum, actor, blog, news, restaurant...)
 - ~255 domen za države: si, it, de, tv, am, gl
 - komercializacija domen ([instagram.am](http://instagram.com), youtu.be, bi.ng, ti.me, pep.si, redd.it)
- avtoritativni strežniki: organizacija z javnimi računalniki (UL: uni-lj)



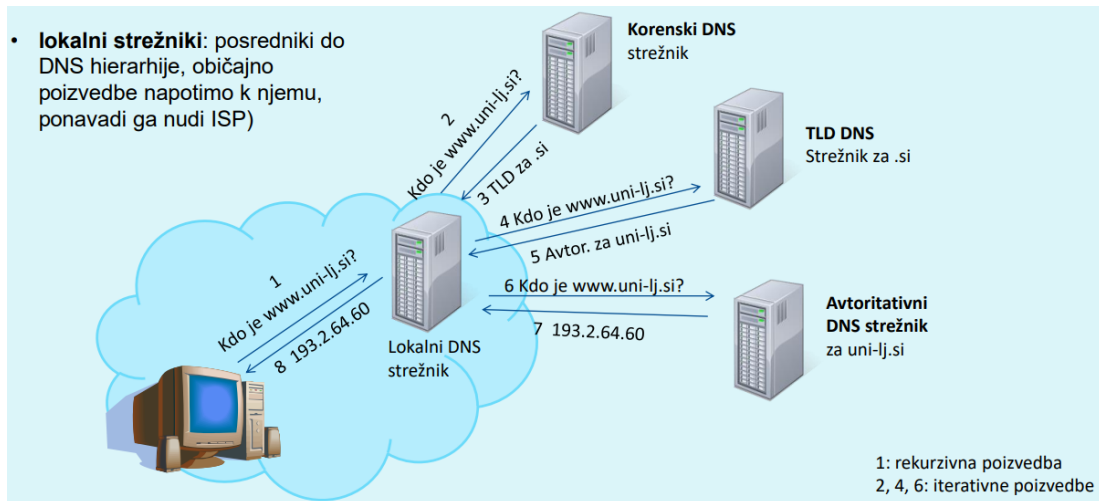
- korenski
- TLD
- avtoritativni za poddomene

13 Korenskih strežnikov

List of Root Servers

Hostname	IP Addresses	Manager
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	VeriSign, Inc.
b.root-servers.net	192.228.79.201, 2001:500:84::b	University of Southern California (ISI)
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4	US Department of Defence (NIC)
h.root-servers.net	128.63.2.53, 2001:500:1::803f235	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	VeriSign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:3::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

Lokalni DNS strežnik, rekurzivna poizvedba



Lokalni DNS strežnik ni del hierarhije. Je le pomočnik, ki razbremeni nas in uporablja predpomnjenje.

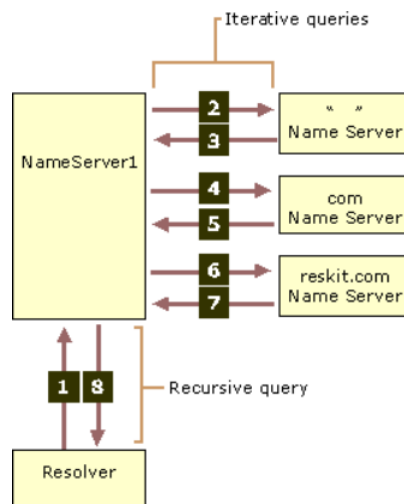
Rekurzivna in iterativna poizvedba

Iterativna poizvedba:

- strežnik vrne bodisi končni odgovor ali pa naziv strežnika za naslednje povpraševanje (primer: lokalni strežnik iterativno povpraša ostale)

Rekurzivna poizvedba:

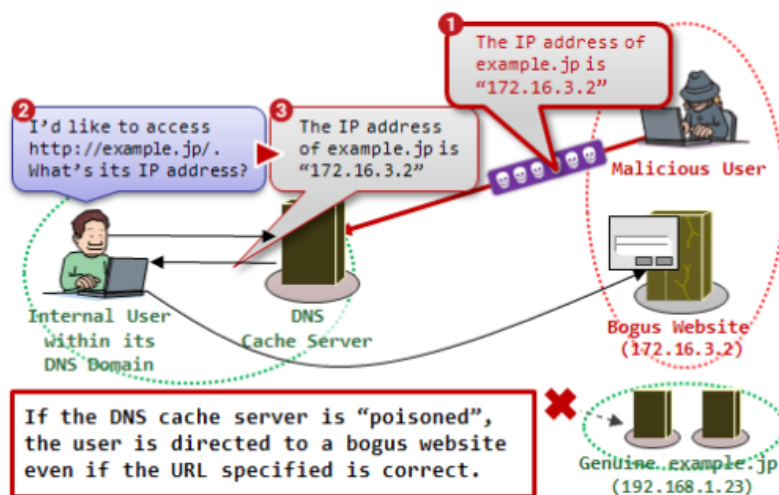
- strežnik poišče preslikavo imena in vrne odgovor (primer: naša poizvedba lokalnemu strežniku)
- razbremenimo končne kliente komunikacije in povpraševanja
- možnost centralnega predpomnenja v lokalnem strežniku!



DNS predpomnjenje (DNS caching)

- DNS strežnik si lahko zapomni prejete odgovore
- s predpomnjenjem dosežemo hitrejši odziv in manj prometa v omrežju (pomembno, ker DNS povzroča del čakanja pri HTTP zahtevkih),
- zapomni si lahko tudi naslove TLD strežnikov (razbremeni korenskega)
- možna tarča napadov (DNS poisoning)

DNS poisoning napade preprečujemo z DNSSEC. Napad deluje podobno kot zastrupljanje stikalnih tabel, v DNS strežnik se doda lažna pot, ki običajno vodi na zlonamerno stran.



DNS zapisi

OBLIKA ZAPISA (RR - resource record):

(Name, Value, Type, TTL)

Authoritative data fortos.vu.nl				
cs.vu.nl.	86400	IN	SOA	star boss (9527,7200,7200,241920,86400)
cs.vu.nl.	86400	IN	MX	1 zephyr
cs.vu.nl.	86400	IN	MX	2 top
cs.vu.nl.	86400	IN	NS	star
star	86400	IN	A	130.37.56.205
zephyr	86400	IN	A	130.37.20.10
top	86400	IN	A	130.37.20.11
www	86400	IN	CNAME	star.cs.vu.nl
ftp	86400	IN	CNAME	zephyr.cs.vu.nl
flits	86400	IN	A	130.37.16.112
flits	86400	IN	A	192.31.231.165
flits	86400	IN	MX	1 flits
flits	86400	IN	MX	2 zephyr
flits	86400	IN	MX	3 top
rowboat		IN	A	130.37.56.201
		IN	MX	1 rowboat
		IN	MX	2 zephyr
little-sister		IN	A	130.37.62.23
laserjet		IN	A	192.31.231.216

- **TTL:** čas veljavnosti zapisa
- **Type = A** (address): Name: ime računalnika, Value: IP številka
 - ti zapisi so shranjeni v avtoritativnih strežnikih za svoje gostitelje
 - AAAA predstavlja naslov IPv6
- **Type = NS** (name server): Name: ime domene, Value: ime avtoritativnega DNS strežnika
 - ti zapisi običajno v TLD strežnikih za iskanje avtoritativnega strežnika neke domene
- **Type = CNAME** (canonical name): Name: alias ime, Value: pravo (kanonično) ime
 - primer: www.ibm.com je dejansko servereast.backup2.ibm.com
- **Type = MX** (mail exchange): Name: alias poštnega strežnika, Value: pravo (kanonično) ime poštnega strežnika

Zapisa CNAME in MX omogočata, da lahko spletni in poštni strežnik naslavljamo z istim imenom (npr. <http://yahoo.com> in xyz@yahoo.com).

Protokol DNS

- deluje po principu izziv - odgovor
- UDP, vrata 53
- strežnik ne hrani stanja povezav, skrbi za ponovno pošiljanje
- 16-bitno polje (ID), ki povezuje zahteve in odgovore
- DNSSEC (DNS Security Extensions): razširitev DNS, ki zagotavlja večjo varnost

identifikator	zastavice	12 bajtov
število poizvedb	število zapisov v odgovoru	
število avtoritativnih zapisov v odgovoru	število dodatnih zapisov v odgovoru	
poizvedbe (spremenljivo število)		parametri poizvedbe: URL, tip, razred
odgovori (spremenljivo število zapisov)		
avtoritativni odgovori (spremenljivo število zapisov)		deli odgovorov: URL, tip, razred, TTL, dolžina, odgovor
dodatni zapisi (spremenljivo število zapisov)		

Novi DNS vnosi

0. Namesti vse potrebne lokalne strežnike in jih vpiši v avtoritativni DNS strežnik podjetja
1. podjetje registrira domeno www.mojafirma.com pri registrarju, posreduje mu imena in IP številke svojih avtoritativnih strežnikov
2. registrar dopolni bazo TLD strežnikov z zapisoma:
 - (mojafirma.com, dns1.mojafirma.com, NS)
 - (dns1.mojafirma.com, 212.212.212.1, A)
3. v avtoritativni strežnik vnesemo zapis tipa A za spletni strežnik, MX zapis za poštni strežnik domene

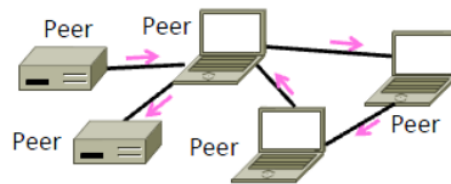
P2P storitve (BitTorrent, Skype)

P2P je družina protokolov.

Arhitektura P2P

- ni strežnika, ki je nenehno prižgan
- izmenjava podatkov med poljubnima končnima sistemoma
- odjemalci sodelujejo po potrebi, pri priklopu menjavajo IP naslov
- izzivi: varnost, iniciativnost pri sodelovanju, NAT

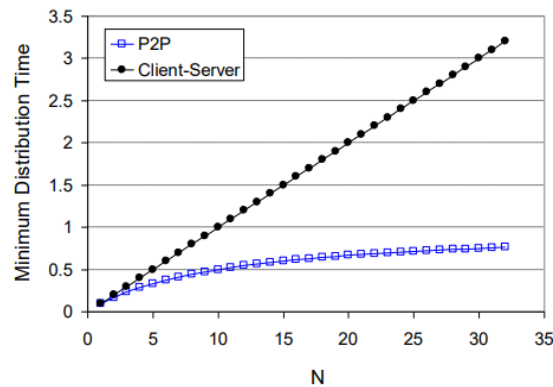
Z NATom je problem, ker pri P2P pride dohodni promet, brez da bi ga ciljnj sistem zahteval, kar pa povzroči težave pri naslavljanju z NAT saj ta še nima odprte povezave.



Skalabilnost sistema P2P

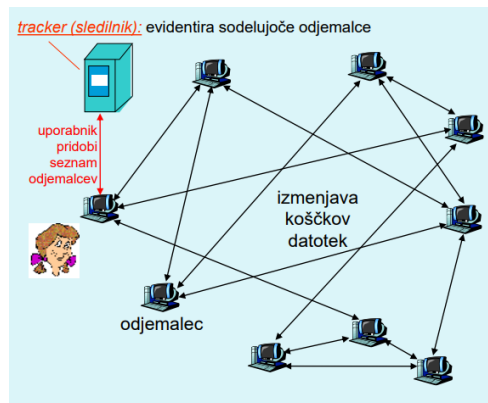
Skalabilnost je performančna lastnost sistema, da še vedno dobro dela če povečujemo število uporabnikov.

- sistem klient-stežnik: čas prenosa linearno narašča s številom odjemalcev (N krat prenos istega podatka)
- sistem P2P: čas prenosa podatka je krajši, ker izmenjava poteka tudi med klienti (proste prenosne kapacitete) -> prednost: večja skalabilnost sistema



BitTorrent

- sistem P2P za izmenjavo datotek
- torrent: skupina odjemalcev, ki si delijo kose (chunks) datotek
- odjemalci se lahko pridružijo/odidejo kadarkoli
- pridružitve novega odjemalca:
 - če še nima koščkov datotek, je cilj pridobiti s časom
 - prijavi se sledilnemu strežniku (**tracker**) in od njega pridobi seznam drugih odjemalcev
 - od vseh izjemalcev odjemalec izbere **podmnožico sosedov (neighbors)** in izvaja izmenjavo **samo z njimi ne z vsemi**.

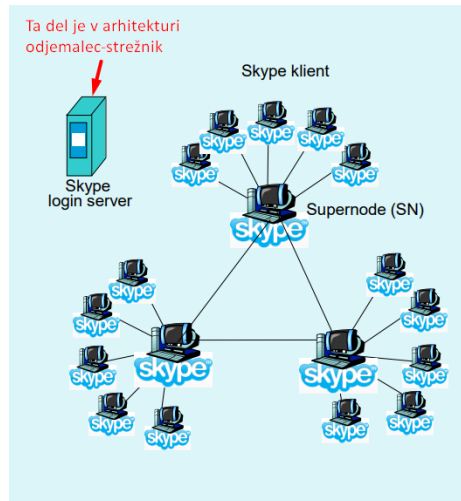


- strategija komunikacije:
 - sosede vpraša po razpoložljivih koščkih datotek in zahteva najprej tiste manjkajoče, ki so **najbolj redki med sosedi (rarest first!)**
 - med prejemanjem koščkov datotek, pošilja koščke tudi drugim odjemalcem
 - **pravičnost**: opazujemo hitrost prejetja od sosedov in jim pošiljamo koščke s **sorazmerno visoko hitrostjo**
 - **vzpodbuda za sodelovanje**: odjemalec lahko po prejemu datoteke ostane (radodarno) ali odide iz torrenta (sebično)
- **aplikacijska sporočila** imajo kontrolna bita za status odjemalca: zamašen (angl. choked) in zainteresiran(angl. interested)
- na začetku je stanje vsake povezave pri odjemalcu: nezainteresiran/zamašen
- pretok podatkov se izvaja, kadar je eden zainteresiran, drugi pa ni zamašen
- namen zastavice za zamašitev: **nadzor zasičenja preko različnih TCP povezav**

Grob opis trackerja: podatkovna baza, ki navzven pokaže podatke o torrentu.

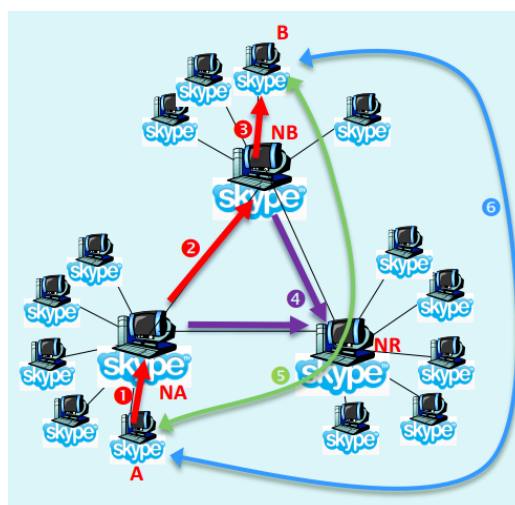
Skype

- tudi P2P: komunikacija med poljubnimi uporabniki
- **lastniški aplikacijski protokol**, podrobnosti neznane (nekateri lastnosti pridobljene z **obratnim inženiringom**)
- strežnik za prijavo preverja podatke o uporabnikih
- nadzorna vozlišča (supernodes):
 - hranijo preslikave uporabniško ime -> IP naslov
 - skrbijo za povezovanje med uporabniki



Premostitvena vozlišča (relay) - rešitev težav z NATom

- NAT povzroča težave v P2P arhitekturah, ker zunanji odjemalci ne morejo direktno kontaktirati odjemalca za prehodom NAT, ker ta ni vzpostavil odhodne povezave
- rešitev:
- odjemalca A in B vzpostavita zvezo preko svojih nadzornih vozlišč NA in NB (1-3),
- pri tem NA in NB izbereta tretje premostitveno (relay) nadzorno vozlišče (NR), s katerim A in B vzpostavita sejo(4-5) (ker sta odjemalca prva vzpostavila zvezo, lahko prejmeta dohodni promet od premostitvenega vozlišča)
- premostitveno vozlišče poskuša zagotoviti neposredno povezavo med odjemalcema (6)



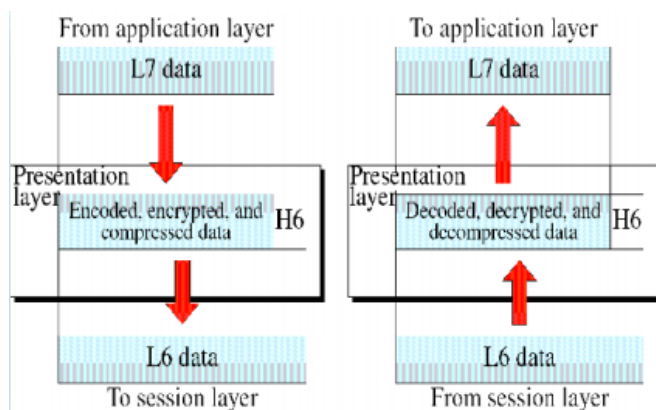
Podporne storitve (predstavitevna plast, sejna plast)

Aplikacijske storitve

- delovati morajo v skladu s tehnološko infrastrukturo. Da lahko to zagotovimo, aplikacija potrebuje podporne storitve, ki v OSI modelu ležijo med aplikacijsko plastjo in omrežjem
- podporne storitve:
 - predstavitevna plast: usklajevanje predstavitve podatkov
 - sejna plast: logično povezovanje med aplikacijskimi procesi

Storitve predstavitvene plasti

1. **predstavitev podatkov (binarni tipi):** različni sistemi predstavljajo binarne tipe na različen način; uporaba sintakse ASN.1 (Abstract Syntax Notation 1) zmanjša število vseh možnih preslikav
2. **predstavitev alfanumeričnih znakov (združljivost kodnih strani):** znaki so predstavljeni s števkami po kodnem sistemu, potrebno zagotoviti, da se prenašajo pravi znaki
3. **stiskanje podatkov:** omogoča zmanjšanje velikosti podatkov in s tem pohitritev prenosa (jpeg, mpeg)
4. **zaščita podatkov (kriptiranje):** varnostni mehanizem, ki omogoča vpeljavo zaupnosti v komunikacijo



Storitve sejne plasti

- **naloge:**
 - vzpostavljane, rušenje, vzdrževanje sej med aplikacijama (potek dialoga med aplikacijama)
 - odgovornost za obnovitvene točke (checkpoints) seje in obnovo (recovery), če seja ne uspe
 - sinhronizacija podatkov iz različnih tokov in virov
- **protokoli:** H.245 (Call Control Protocol for Multimedia Communication), RTCP (Real-time Transport Control Protocol), SCP (Session Control Protocol), ...
- **možni odnosi** med transportno in sejno plastjo:

