



Univerza v Ljubljani

Fakulteta
za računalništvo
in informatiko

Aplikacijska plast

© Mojca Ciglarič

Omrežne aplikacije so razlog za obstoj omrežij!

Omrežne aplikacije

- 80. leta: tekstovne (e-pošta, oddaljen dostop, prenos datotek, novice, klepet)
- Sredi 90. let: aplikacija SPLET
- Večpredstavne aplikacije: pretočni video, spletni radio, spletni telefon, video konference...
- Večuporabniške omrežne igre
- Okrog 2000: IM (takojšnje sporočanje) in P2P izmenjava datotek
- Danes: družabna omrežja, oblachno računalništvo
- Kaj je “*killer*” aplikacija?

Temeljna načela omrežnih aplikacij

- Teče na več **končnih** napravah (ni isto kot aplikacija, ki le uporablja omrežje)
- Več (različnih?) programov / procesov
- Primeri:
 - spletni strežnik in odjemalec: strežnik ves čas dostopen, lahko gruča; znan naslov. Odjemalci so lahko nedostopni, med sabo direktno ne komunicirajo.
 - P2P: člani so lahko nedostopni. Robustnost, skalabilnost.. Upravljanje je težko.
 - Hibridni pristop (npr. Skype, IM – potreben osrednji strežnik za del funkcionalnosti, ostalo gre P2P)

Komunikacija med procesi

- Komunicirajo procesi, ne programi!
- Proces: program, ki teče na končnem sistemu (“živ” primerek programa; skupek vseh virov, potrebnih za izvedbo programa).
- Izmenjava sporočil.
- Omrežna aplikacija: pari procesov, ki si izmenjujejo sporočila.
Par = odjemalec + strežnik .
- **Odjemalec**: proces, ki sproži komunikacijo.
- **Strežnik**: proces, ki čaka, da ga bo kdo kontaktiral.

Protokoli aplikacijske plasti

- Protokol določa **pravila za izmenjavo sporočil**.
 - Vrste sporočil (npr. zahteva, odgovor, potrditev...)
 - Zgradbo sporočila (polja, meje med polji...)
 - Pomen sporočila (kaj je v nekem polju)
 - Kdaj in kako proces oddaja sporočila in kako reagira na prejeta sporočila
- Javni (odprti) protokoli, npr. HTTP (RFC 2616)
 - Specifikacije (RFC): www.ietf.org
- Lastniški (zaprti) protokoli, npr. Skype
- Protokol je le DEL aplikacije!
- Aplikacijski protokoli so navadno lepo berljivi uporabniku

Dopolnite...

Aplikacija	Izguba	Pasovna širina	Časovna občutljivost
Prenos datotek	NE	Elastična	NE
E-pošta	NE	Elastična	NE
Zvok/ slika v realnem času	DA	Zvok: nekaj kb/s- 1Mb/s Slika: 10kb/s – 5 Mb/s	Nekaj 100 ms
Shranjen zvok/ slika	DA	-II-	Nekaj s
Interaktivne igre	? (DA)	1-10 kb/s	Nekaj 100 ms
IM	NE	Elastična	? (DA)

Uporaba TCP in UDP za aplikacijske protokole

Aplikacija	Protokol	Transportni p.
E pošta	SMTP	TCP
Oddaljen dostop	telnet	TCP
Prenos datotek	ftp	TCP
Splet	http	TCP
Multimedija	http, rtp	TCP ali UDP
IP telefonija	SIP, RTP, lastniški	Pogosto UDP



PREGLED APLIKACIJSKIH PROTOKOLOV

Splet in HTTP

- Je aplikacija (roj. 1990), ki omogoča
 - Vsebine „kar hočeš, kadar hočeš“ – na zahtevo.
 - Vsak lahko objavlja karkoli
 - Iskanje, povezave, grafika, vmesniki, multimedija
- Osnovni specifikaciji RFC 1945 (HTTP 1.0), 2616 (HTTP 1.1)
- Delovanje:
 - Spletno stran sestavljajo objekti (html, jpg, applet, audio, flash...)
 - Vsak objekt ima svoj URL naslov (gostitelj+pot)
 - Odjemalec naslovi http zahtevo (request) na vrata 80 strežnika
 - Strežnik vrne http odgovor (response) z zahtevanim objektom.
- TCP poskrbi za potrditve, ponovitve, vrstni red.
- Protokol **brez stanj** (stateless)

HTTP povezave (zgodovinsko)

- Nonpersistent (minljive, ne trajne):
 - Za vsak objekt se vzpostavi nova TCP povezava (zamudno zaradi rokovanja, obremenjuje strežnik)
- Persistent (trajne):
 - Strežnik pusti po pošiljanju povezavo še odprto, po njej lahko pošlje še več datotek
 - Brez cevovodov: odjemalec da novo zahtevo, ko prejme prejšnji objekt
 - S cevovodi (pipelined): odjemalec da novo zahtevo, ko naleti na referenco na nov objekt (**privzeti način**).

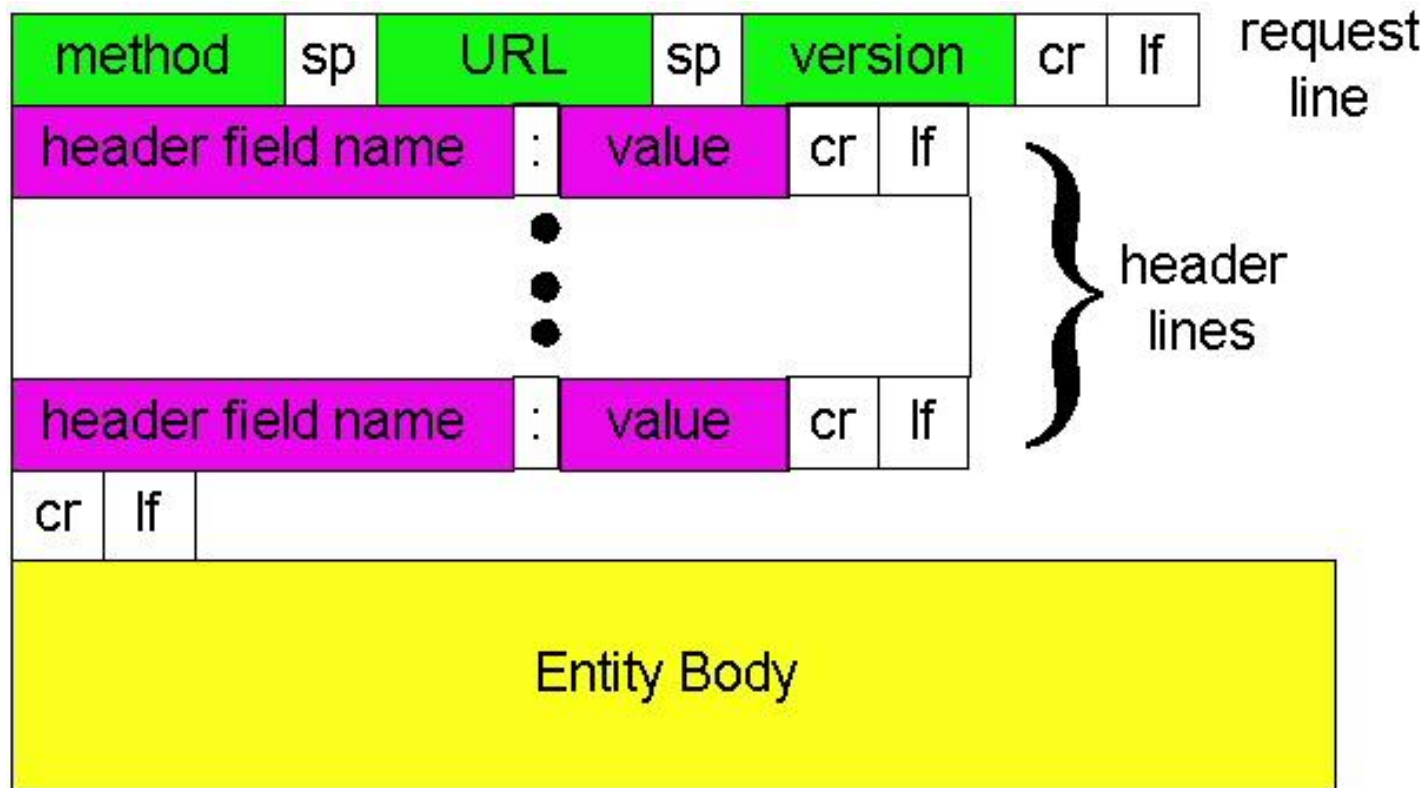
Format sporočila: http zahteva

Metoda URL Verzija ← Statusna vrstica- zahteva
Ime polja1: vrednost ←
... ←
Ime poljan: vrednost ←
[prazna vrstica]
TELO

Vrstice glave
(header lines)

```
GET /sem/ocene.htm HTTP/1.1  
Host: marvin.fri.uni-lj.si  
Connection: close  
...
```

Format HTTP zahteve



HTTP zahteva: metode

- GET: zahteva objekta
- POST: zahteva objekta + deli objekta imajo poslane vrednosti (html forms)
- Obrazec lahko uporabi tudi metodo GET, vrednosti parametrov pa pošilja kot podaljšan naslov
(www.google.si/search?q=kolokvij)
- HEAD: zahteva za HTTP odgovor, vendar brez zahtevanega objekta (uporaba: razhroščevanje)
- PUT (HTTP 1.1) – upload na strežnik
- DELETE (HTTP 1.1) – brisanje s strežnika
- TRACE (za razhroščevanje, podobno pingu)
- CONNECT (povezava preko proxyja)
- OPTIONS (povpraševanje po opcijah v zahtevi)

Format sporočila: http odgovor

Verzija Status Opis ← Statusna vrstica
Ime polja1: vrednost
... ← Vrstice glave
Ime poljan: vrednost (header lines)
[prazna vrsta]
TELO

HTTP/1.1 200 OK

Connection: close

Date: Mon, 05 May 2015 12:18:23 GMT

Server: Apache/2.2.3 (Debian)

Last-Modified: ...

Content-Length: 13534

Content-Type: text/html

HTTP status

(poglej RFC za cel seznam!)

- 1xx: informativne kode (100: Continue)
- 2xx: uspešno (200: OK)
- 3xx: preusmeritev (301: Moved Permanently- prestavljen dokument + vrne novi naslov Location : ...)
- 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).

Piškotki

- Specifikacija RFC 2109
- Strežnik brez piškotkov ne loči zahtev različnih odjemalcev, nima zgodovine, nima spomina.
- Sestavni deli
 - Piškotkova vrstica v glavi zahteve (ID piškotka)
 - Piškotkova vrstica v glavi odgovora (ID piškotka)
 - Odjemalčeva datoteka piškotkov
 - Strežnikova zaledna podatkovna zbirka izdanih piškotov in povezanih odjemalcev

Scenarij uporabe

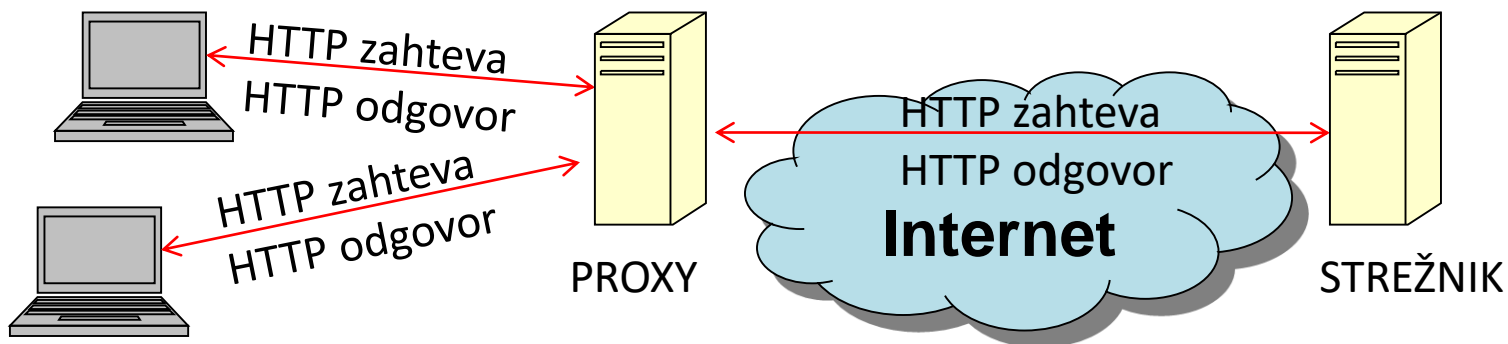
- Odjemalec pošlje HTTP zahtevo brez piškotkove vrstice
- Dobi HTTP odgovor z vrstico **Set-cookie:1234567** (ID)
- Odjemalec: dopolni datoteko piškotkov in vse naslednje zahteve s piškotkovo vrstico.
- Strežnik shranjuje podatke o uporabniku...
- Nad plastjo HTTP (brez stanj) se tako ustvari sejna plast (s stanji).
- Bogatejša uporabniška izkušnja (avtorizacija, košarica, stanje – spletna pošta, personalizacija)…, sporno glede zasebnosti.

Vrste piškotkov

- Session cookie – samo za čas trajanja seje (nima roka trajanja)
- Persistent cookie (tracking) – daljši rok trajanja , npr.1 leto
- http- only (ni dostopen skriptom) – manj nevarnosti za krajo
- 3rd party – od strani, katere naslov ni v naslovni vrstici (npr. oglaševalci)
- Zombie cookie – se spet pojavi, ko ga pobrišemo (obstaja rezervna kopija in nek skript poskrbi, da se po brisanju restavrira)
- Napadi: kraja piškotka in ugrabitev seje, zastrupljen piškotek in DoS

Posredniški strežnik

- Imenuje se tudi Web cache, proxy server (navadno pri ISP-ju)
- Pošilja sporočila namesto strežnikov in odjemalcev
- Ima svoje kopije spletnih strani (samo sveže).
- Ustrezno konfiguriran odjemalec!
- Če posrednik zahtevane strani nima pri sebi, jo zahteva od pravega strežnika.



Zakaj posredniki?

- Manj prometa
- Hitrejši odgovor odjemalcu
- Ozka grla
- MANJ IZPOSTAVLJENI ODJEMALCI (napad, anonimnost)
- Pogojna zahteva (je pomnjena stran zastarela?)
 - metoda **Conditional GET**
 - vrstica glave:
`If-modified-since: Wed, 31 Oct 2011 09:32:22`
 - Strežnik pošlje novo stran ali
`HTTP/1.1 304 Not Modified` (prazno telo)

Prenos datotek – protokol FTP

- Prijava na oddaljeni računalnik + prenos datotek z oddaljenega računalnika k uporabniku in obratno.
- 2 ločeni TCP povezavi na FTP strežnik:
 - Nadzor (vrata 21) na zahtevo odjemalca (trajna): uporabniško ime, geslo, CD ukazi, ukazi za prenos datotek
 - Prenos podatkov – datotek (vrata 20) na zahtevo strežnika (minljiva – za vsako datoteko nova!) – to je aktivni način
- **Protokol s stanji:** strežnik ve, kdo je odjemalec, kateri imenik pregleduje...
- Potreben je odjemalski program (UA)!
- Pasivni način: odjemalec ne more sprejeti povezave od strežnika, zato tudi podatkovno vzpostavi sam

FTP: sporočila

- RFC 959. Nadzorna povezava: 7-bitni ASCII
- Ukazi
 - USER ime; PASS geslo; LIST**
 - RETR ime_dat** (retrieve = get)
 - STOR ime_dat** (store = put)
- (Nekateri) odgovori strežnika
 - 331 Username OK, password required
 - 125 Data connection open, transfer starting
 - 452 Error writing file
 - 425 Can't open data connection

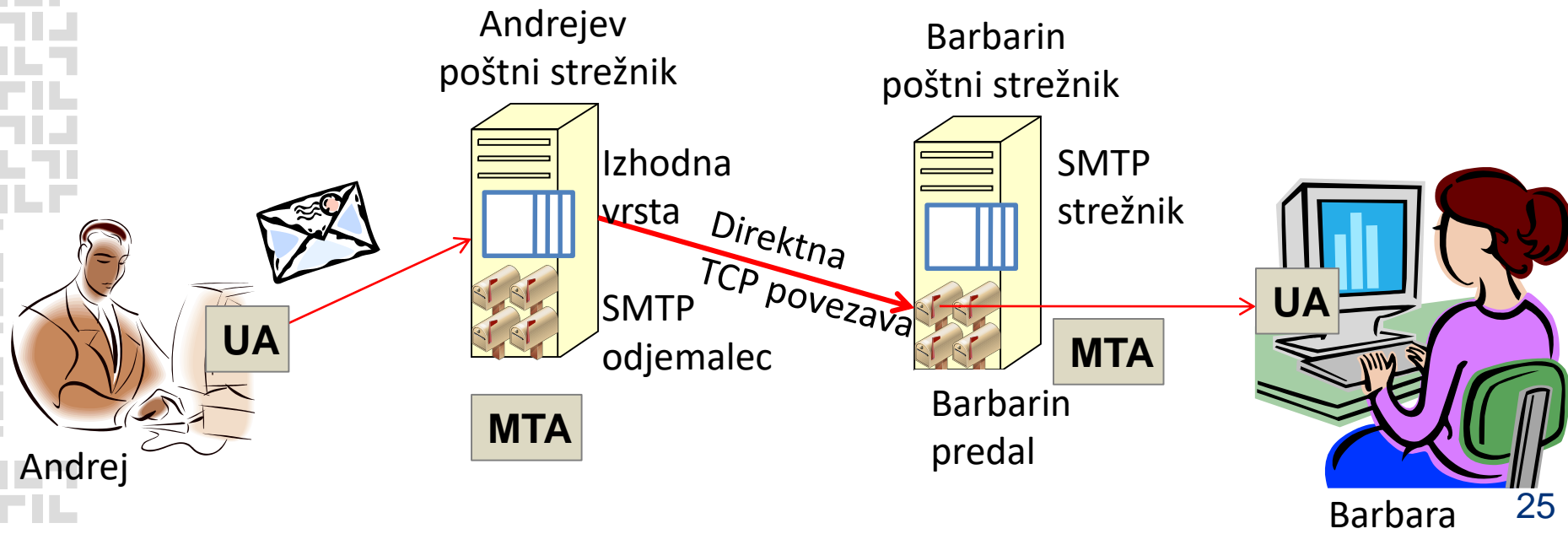
Protokoli elektronske pošte: SMTP, POP3, IMAP

Osnovni elementi sistema:

- Poštni **strežniki**
 - Poštni predali (vhodna pošta)
 - Izhodna vrsta sporočil
- **Odjemalski** programi (UA): tekstovni, grafični
- **Protokol** za prenos sporočil (SMTP)
- Pošiljatelj – pošiljateljev UA – pošiljateljev strežnik – prejemnikov strežnik – prejemnikov UA – prejemnik.

SMTP

- RFC 2821. Protokol je star več kot 30 let!
- Strežnik posluša na TCP vratih 25
- 7-bitni ASCII (tudi za telo sporočila)
- Binarne priponke je potrebno prekodirati v ASCII. In na prejemni strani nazaj v binarno.



SMTP

- Odjemalec: SMTP strežnik, ki pošilja sporočilo
 - Strežnik: SMTP strežnik, ki sprejema sporočilo
 - Povezava na vrata 25
-
1. Aplikacijsko rokovanje
 - Medsebojna predstavitev
 - Odjemalec: e-mail naslov pošiljatelja in prejemnika
 2. Prenos sporočila (lahko več po isti povezavi)
 3. Rušenje TCP povezave


Primer - SMTP seja med dvema strežnikoma

S: 220 fri.uni-lj.si strežnik se predstavi
O: **HELO** mojmail.si odjemalec se predstavi
S: 250 Hello mojmail.si, pleased to meet you
O: **MAIL FROM:** <miha@mojmail.si>
S: 250 miha@mojmail.si ... Sender ok
O: **RCPT TO:**<mojcac@fri.uni-lj.si>
S: 250 <mojcac@fri.uni-lj.si> ... Recipient ok
O: **DATA**
S: 354 Enter mail, end with "." on a line by itself
O: Zdravo, Mojca!
O: Nujno me poklici, ko prides domov, LP Miha.
O: .
S: 250 Message accepted for delivery.
O: **QUIT** ali pa zopet MAIL FROM: <...
S: 221 fri.uni-lj.si closing connection

Fri Format sporočila (RFC 822, 2822)

Ime polja: vrednost **From: miha@mojmail.si**
... **To: mojcac@fri.uni-lj.si**
Ime polja : vrednost **Subject: Poklici me**
[prazna vrsta]
Telo sporočila **Zdravo, Mojca!**
 Nujno me poklici, ko
 prides domov, LP Miha.

Vrstice glave
(header lines)



Pomembno: razlika med SMTP ukazi in polji v glavi!

Prejemni strežnik

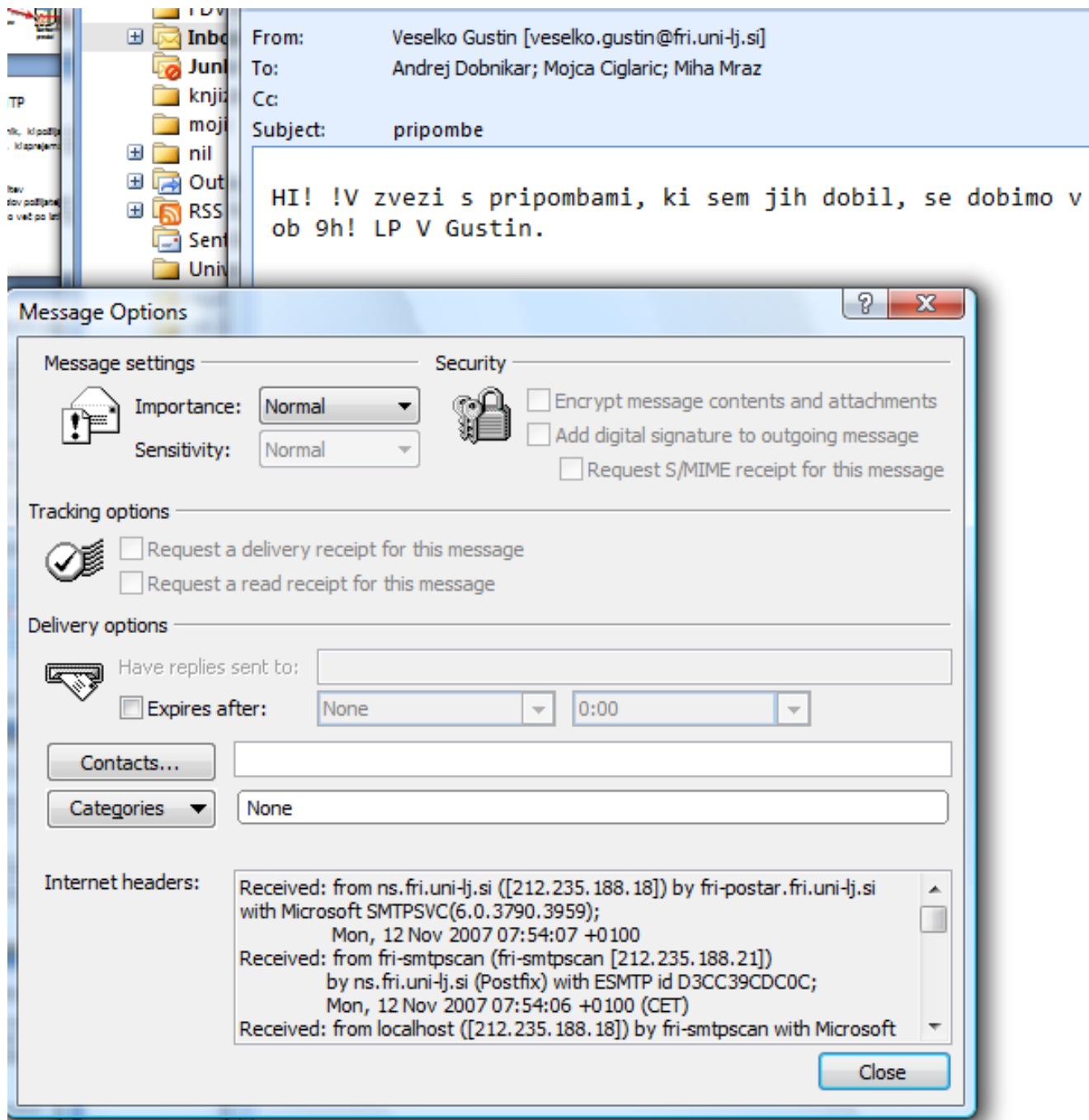
- V glavo doda vrstico Received
- Teh vrstic je lahko več (npr. če se pošta posreduje - forward)

`Received: from fri.uni-lj.si by
gmail.com;`

`4 Nov 2010 15:29:42 GMT`

`Received: from mojmail.si by fri.uni-
lj.si;`

`4 Nov 2010 15:27:33 GMT`



Celotna glava prejšnjega sporočila...

Received: from ns.fri.uni-lj.si ([212.235.188.18]) by fri-postar.fri.uni-lj.si with Microsoft SMTPSVC(6.0.3790.3959);

Mon, 12 Nov 2007 07:54:07 +0100

Received: from fri-smtpscan (fri-smtpscan [212.235.188.21])

by ns.fri.uni-lj.si (Postfix) with ESMTP id D3CC39CDC0C; Mon, 12 Nov 2007 07:54:06 +0100 (CET)

Received: from localhost ([212.235.188.18]) by fri-smtpscan with Microsoft SMTPSVC(6.0.3790.0);

Mon, 12 Nov 2007 07:54:05 +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 LBI165vdxDib; Mon, 12 Nov 2007 07:54:04 +0100 (CET)

Received: from fri.uni-lj.si (lra-3.fri.uni-lj.si [193.2.76.71])

by ns.fri.uni-lj.si (Postfix) with ESMTP id D61259CDC0C; Mon, 12 Nov 2007 07:54:03 +0100 (CET)

Message-ID: <4737F884.9080703@fri.uni-lj.si>

Date: Mon, 12 Nov 2007 07:53:56 +0100

From: Veselko Gustin <veselko.gustin@fri.uni-lj.si>

Organization: University of Ljubljana, Faculty of Computer and Information Science

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4) Gecko/20030624 Netscape/7.1 (ax)

X-Accept-Language: en-us, en

MIME-Version: 1.0

To: <andrej.dobnikar@fri.uni-lj.si>, Mojca Ciglaric <mojca.ciglaric@fri.uni-lj.si>, <miha.mraz@fri.uni-lj.si>

Subject: pripombe

Content-Type: text/plain; charset=us-ascii; format=flowed

Content-Transfer-Encoding: 7bit

X-OriginalArrivalTime: 12 Nov 2007 06:54:05.0541 (UTC) FILETIME=[D0E7F150:01C824F8]

Return-Path: veselko.gustin@fri.uni-lj.si

MIME razširitve sporočila

- Multipurpose Internet Mail Extensions
- RFC 2045 in 2046: razširitvi starega RFC 822.
- ČŠŽÁÃÇÊËΩξ☺ , večpredstavna sporočila
- Nova polja glave

MIME-Version:

Content-Type:

Content-Transfer-Encoding:

Kodiranje (Encoding)

- Quoted-printable
 - Za (8-bitni) tekst z malo ne-angleškimi znaki. Berljivo.
`Subject: =?iso-8859-2?Q?RE:_Obisk_na_va=B9i_=B9oli?=`
- Base 64
 - Abeceda iz 64 znakov (A-Z, a-z, 0-9, "+" in "/")
 - 3x8 bitov → 4x6 bitov → 4 ASCII znaki
 - Velika režija (137% + 814 bitov glava)
- Binary (novejši RFC 3030)
- Primer: jpg priponka (decode, jpeg dekompresija)

Primerjava SMTP in HTTP

- Podobnosti
 - Prenos datotek
 - Trajne (persistent) povezave (HTTP: možne)
- Razlike
 - HTTP: pull (potegnem vsebino s strežnika)
SMTP: push (oddajni strežnik pošto porine prejemnemu)
 - SMTP: 7-bitno ASCII kodiranje, HTTP ne
 - HTTP: vsak objekt enkapsulira v svoj HTTP odgovor, SMTP: vse objekte maila zavije v eno sporočilo

Dostop do poštne predala

Včasih: oddaljen dostop do strežnika (telnet), nato neposredno branje iz poštne predala...

Danes

- Dohodna pošta: POP3, IMAP ali HTTP dostop
 - PULL (prenos pošte k sebi)
- Pošiljanje odhodne pošte na strežnik: SMTP
 - PUSH

POP3

- Preprost, omejena funkcionalnost
- UA odpre TCP povezavo na vrata 110
- 3 faze
 - Avtorizacija
 - Transakcija (prenos sporočil, oznake za brisanje, statistika)
 - Posodabljanje (odjemalec : QUIT, strežnik izvede brisanje)
- Slabosti: lokalno urejanje pošte, dostop z več računalnikov.

POP3 ukazi - primer

S: +OK POP3 server ready

O: user mojcac uporabnik se predstavi

S: +OK

O: pass tralala nezaščiten!

S: +OK user successfully logged on (ali pa -ERR)

O: list

S: 1 678 1. sporočilo je veliko 678 bytov

S: . Ni več sporočil...

O: retr 1 prenesi sporočilo 1

S: (tralala hopsasa ...)

S: .

O: dele 1 briši sporočilo 1

O: quit

S: +OK POP3 server signing off

IMAP in HTTP

- IMAP
 - Kompleksen, zahtevnejši, več funkcionalnosti
 - Uporabnik lahko določi mape na strežniku
 - Vsako sporočilo je v mapi
 - UA lahko prenese tudi le dele sporočil
 - Večja obremenitev strežnika
- HTTP dostop do pošte
 - Brskalnik, dostop od koderkoli, brezplačni ponudniki
 - Mape kot pri IMAP
 - Dostop do map in sporočil omogočajo skripte na HTTP strežniku, te npr. prek IMAP komunicirajo s poštnim strežnikom.

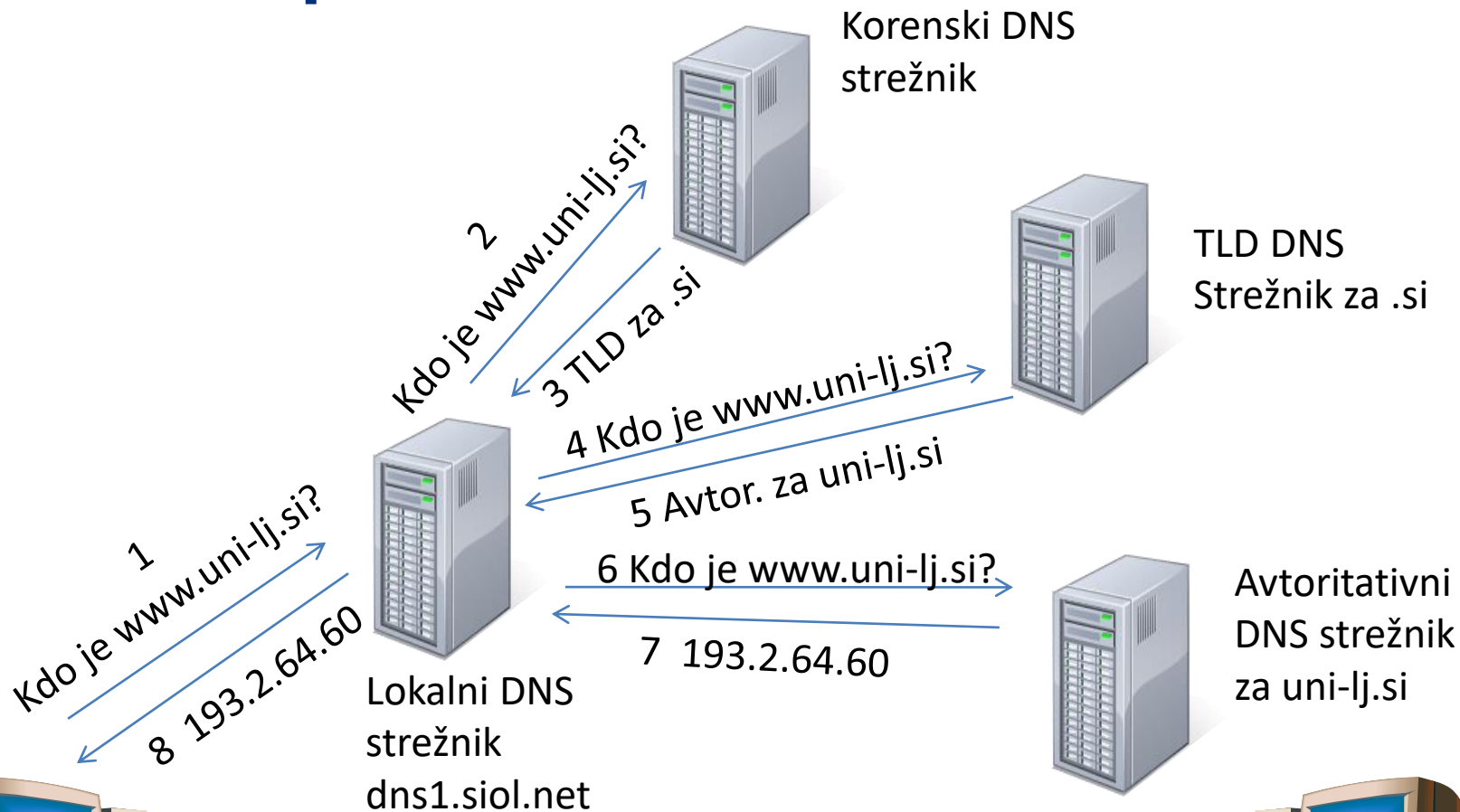
DNS

- IP številka ali znano ime (www.google.com)?
- Bistvena omrežna funkcionanost, ne direktno za uporabnika. RFC 1034, 1035, ...
- DNS vključuje
 - Porazdeljeno podatkovno zbirko
 - Protokol za poizvedovanje po njej
- Storitve
 - Preslikava med imeni in IP številkami
 - Alias (več imen za isto IP številko) hostov in poštnih strežnikov
 - Porazdeljevanje bremena (več IP številke za isto ime)

Organizacija

- Zakaj ne le en strežnik?
- 13 korenskih strežnikov (A-M), vsak je replicirana gruča
- Posamezni TLD (Top-Level Domain) strežniki
 - com, org, net, edu, biz, info, si, fr, it, de, ...
- Avtoritativni strežniki
 - Organizacija z javno dostopnimi računalniki (UL: uni-lj)
- Lokalni strežniki
 - posredniki do DNS hierarhije

Primer poizvedbe



1: rekurzivna poizvedba
2,4,6: iterativne poizvedbe

DNS caching

- DNS strežnik si zapomni prejete odgovore (za določen čas, npr. 2 dni)
- Njegov odgovor ne bo avtoritativen
- Manj poizvedb, hitrejši odziv
- Zapomni si lahko tudi naslove TLD strežnikov (razbremenijo korenskega)

DNS zapisi

- RR = Resource Record (Name, Value, Type, TTL)
- TTL: kdaj zapis izbrisati
- Type = A: Name - ime rač., Value – IP številka (AAAA za IPv6)
- Type = NS: Name – ime domene, Value – ime avtoritativnega DNS strežnika.
- Type = CNAME: Name – alias ime, Value – pravo (kanonično) ime
- Type = MX: Name – alias poštne strežnika, Value – pravo (kanonično) ime poštne strežnika

DNS strežniki in zapisi

- Avtoritativni DNS strežnik ima zapise tipa A za vse “svoje” gostitelje.
- Ne-avtoritativni DNS strežnik (**dns1.siol.net**)
 - ima *lahko* zapis tipa A (cache!) za nekega gostitelja
(**www.uni-lj.si, 193.2.64.60, A**)
 - Ima NS zapis za domeno tega gostitelja
(**uni-lj.si, dns1.uni-lj.si, NS**)
 - Ima A zapis za DNS strežnik te domene
(**dns1.uni-lj.si, 193.2.64.45, A**)

DNS sporočila

- Poizvedba in odgovor. Format je enak.
- Glava 12 bytov, več polj
 - ID sporočila 16 bitov
 - Zastavice (zahteva ali odgovor, želim rekurzijo, možna rekurzija, avtoritativni odgovor...)
 - Število vprašanj, število odgovorov (RR-jev), št. avtoritativnih in št. dodatnih RR
- Poizvedba (ali več) (ime, tip, npr. A/MX)
- Odgovor(i) (RR zapisi za ime)
- Avtoritete (RR zapisi drugih avt. strežnikov)
- Dodatni podatki (RR)
- Nslookup – za vpogled v bazo sistema

FRi Kako raste DNS zbirka podatkov?

- Registracija domene lrk.si in dodelitev ranga IP števil
- Določitev primarnega in sekundarnega (backup) avtoritativnega DNS strežnika
- Registrar: vnos NS in A zapisov zanj v TLD DNS strežnik:
 - `lrk.si`, `dns1.lrk.si`, NS
 - `dns1.lrk.si`, `123.123.122.5`, A
- Vnos A zapisa za spletni strežnik, MX zapisa za poštni strežnik domene v avt. DNS strežnik
 - Statično (ročni vnos)
 - Dinamično (z DNS sporočili – RFC 2136)

Storitev aplikacijske plasti je še več...

- **Standardne**
 - Oddaljen dostop (telnet, RFC15 → RFC 854 in drugi),
 - Novice (NNTP, RFC 977, 3977 in drugi)
 - Imenik (LDAP)...
- **Nestandardne**
 - Iskanje,
 - P2P izmenjava datotek
 - ...
- **Podporne** (sejna + predstavitvena plast po OSI)
 - Predstavitev podatkov (preslikave med kodnimi tabelami, ASN.1)
 - Stiskanje (jpeg, mpeg...)
 - Zaščita vsebine (kriptiranje...)
 - Logično povezovanje aplikacijskih procesov – vodenje seje
 - ...

Nestandardne storitve: P2P

- Dinamično omrežje, nestalni člani
- Ponovni priklop z drugim IP
- Izmenjava podatkov med poljubnima končnima sistemoma

- Osrednji strežnik (Napster)
- Popolna enakost (osnovna Gnutella, Kazaa)
 - Poplavljanje poizvedb ali omejeno poplavljanje
- Popolna enakost + super vozlišča (eMule, eDonkey)
 - Prioritete uporabnikov, paralelno pretakanje, vrste zahtev
- Podobno: BitTorrent: iskanje je tu ločeno od prenašanja

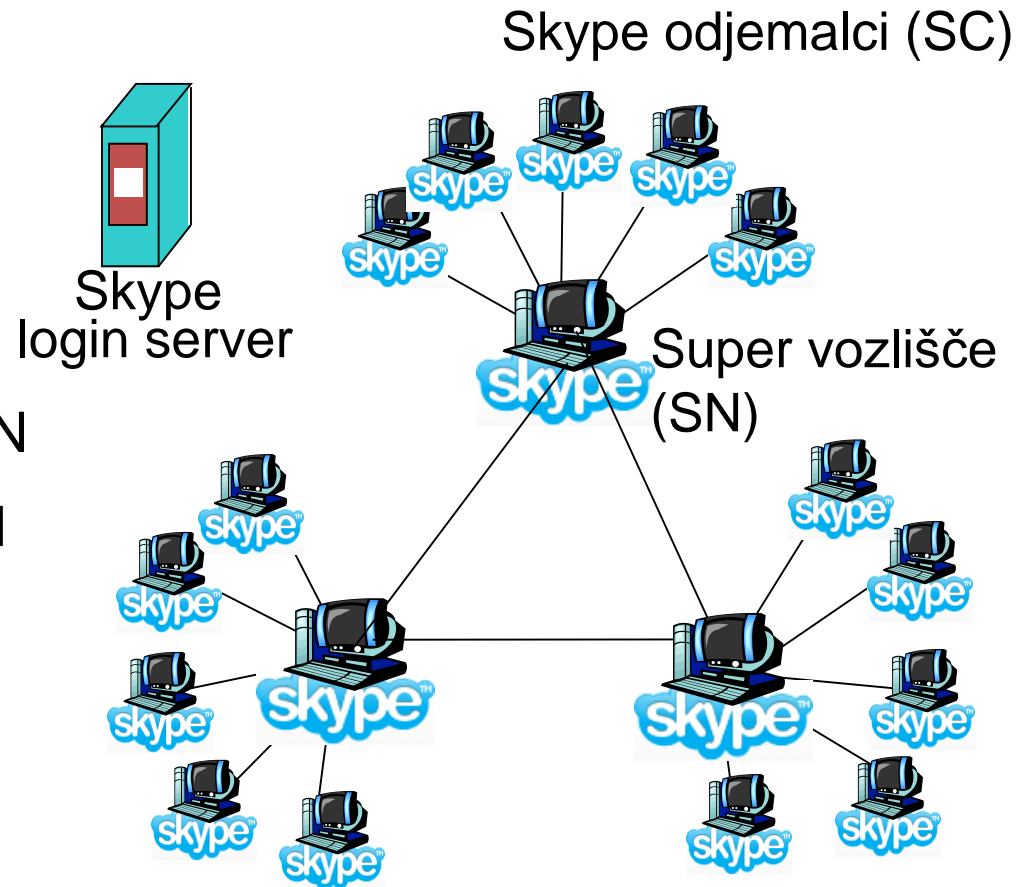


BitTorrent

- Torrent: skupina odjemalcev, ki si delijo kose datotek (256 KB)
- Sledilnik – tracker- evidentira odjemalce in seznam posreduje drugim odjemalcem.
- Peer, ki se vključuje: nima datoteke. Od sledilnika dobi seznam peerov in se poveže na nekatere (sosedje)
- Periodično sprašuje sosede, katere kose imajo (različne!) –
- Med prenosom k sebi tudi prenaša k drugim peerom (4-im, ki k njemu prenašajo najhitreje + vsakih 30 sekund naključno še enemu - optimistically unchoke)
- Peeri s hitrejšim prenosom drugim najdejo boljše partnerje in hitreje dobijo celo datoteko,

Skype

- Hierarhično prekrivno omrežje (2 nivoja)
- Indeks vsebuje pare uporabniško ime – IP naslov – porazdeljen po SN
- Težava: komunikacija med peeroma, ki sta oba za NATom: potreben je posrednik, ki ni za NATom (relay): vsak se poveže na relay, ta posreduje ves promet.



Podporne storitve sejne plasti

- Vsebina: **logično povezovanje apl. procesov**
- TCP model: logično povezovanje opredeli programer
- OSI model: predlog standardnih funkcij
- Sejne storitve
 - So na voljo aplikacijski plasti: SSPT nudi dostop do funkcij logičnega povezovanja, nadzora,...
 - Uporabljajo storitve transportne plasti (idealni kanal)

Sejna in transportna povezava

- Možni odnosi:



Sejna povezava

Transportna povezava



Več sejnih povezav

Transportna povezava



Sejna povezava

Več transportnih povezav

- Multipleksiranje se izvaja na nižjih plasteh.

OSI: struktura seje

- Sejna povezava
 - Seja: ena ali več aktivnosti (ena naenkrat)
 - Aktivnost: en ali več dialogov (en naenkrat)
 - Aktivnost lahko zajema več kot eno sejo
 - Prekinitev, zamrznitev, bujenje, ponovitev
- Žetoni pomagajo strukturirati sejno povezavo
 - Podatkovni (pošiljanje)
 - Rušilni (sproščanje povezave)
 - Sinhronizacijski
 - Glavne sinhronizacijske točke (potrditev, čakanje)
 - Pomožne (ni potrditve – nepovezana storitev)

OSI: funkcije sejne plasti

Različni nivoji kakovosti sejne storitve! Funkcionalni sklopi:

1. Jedro: osnovna povezana storitev, dvosmerni kanal
2. Usklajeno sproščanje logičnega kanala
3. Izmenično dvosmerni kanal
4. Sinhronizacija med sejo
5. Nadzor in upravljanje aktivnosti
6. Sporočanje o neregularnostih

OSI nivoji kakovosti sejnih protokolov:

1, 1+3, 1+4, 1+5+6, 1-6 (full)