



Komunikacije v avtomatiki

Sloj aplikacije in sloj prenosa

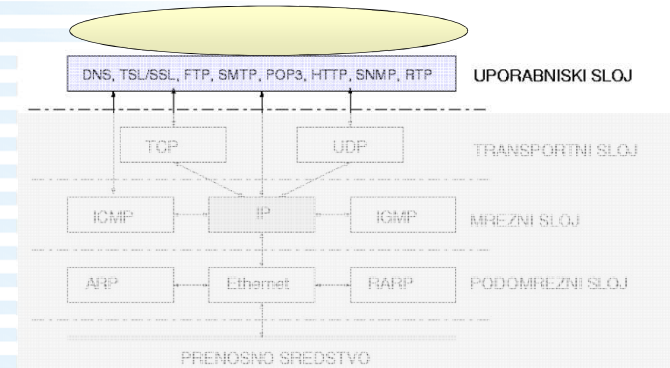
Stanislav Kovačič



<http://vision.fe.uni-lj.si>

2013/14

Sloj uporabe



Sloj aplikacije se poslužuje storitev nižjih slojev, ki so dane na „zgorjem“ vmesniku transportnega sloja

Sloj uporabe

Bistvena naloga:

Sloj uporabe zagotavlja storitve končnim uporabnikom omrežja sicer pa so v okviru tega sloja ponujene tudi nekatere storitve, ki jih potrebujejo nižji sloji (IP)

E-pošta

POP3 (Post Office Protocol)

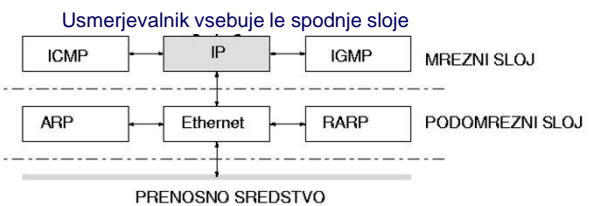
FTP (File Transfer (Protocol))

DNS (Domain Name System)

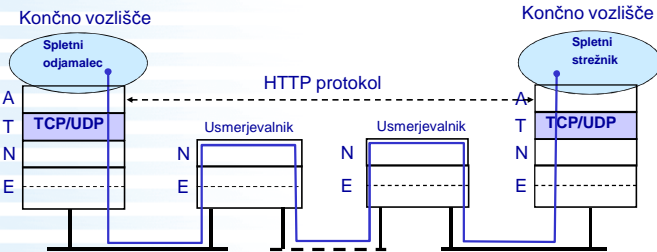
DHCP (Dynamic Host Configuration Protocol)

WWW (World Wide Web)

Sloj uporabe



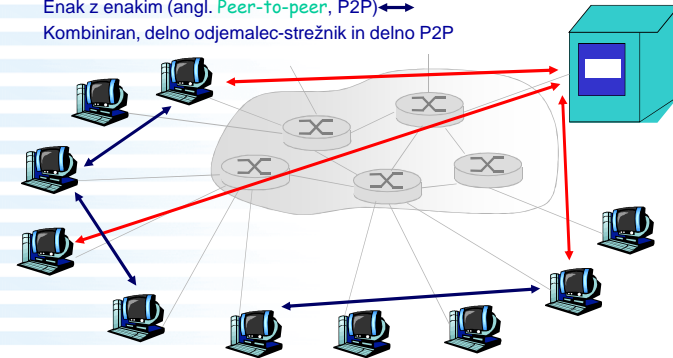
Sloj uporabe - primer



Na primer:
 Aplikacijski sloj zagotavlja spletnemu brskalniku dostop do spletnih strani oddaljenega spletnega strežnika (HTTP).
 Za prenos skrbi prenosni sloj (TCP/UDP)
 Za pot prenosa skrbi omrežni sloj (IP)
 Za dostop do omrežja poskrbita podatkovno linjski in fizični sloj - Ethernet

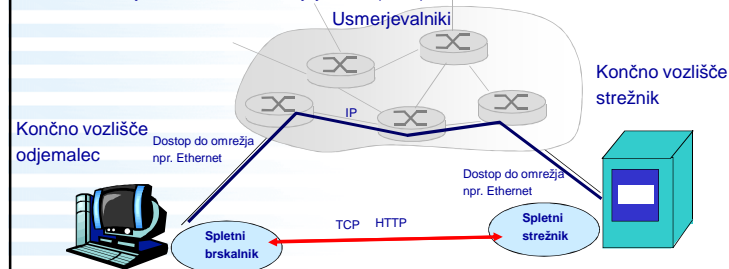
Dva osnovna koncepta aplikacij

Odjemalec-strežnik (angl. *Client-server*) ↔
 Enak z enakim (angl. *Peer-to-peer, P2P*) ↔
 Kombiniran, delno odjemalec-strežnik in delno P2P



Sloj uporabe - še drugačen pogled

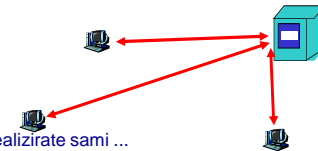
V končnih vozliščih se izvajajo: HTTP, TCP, IP, Ethernet
 V usmerjevalnikih se izvajajo: IP, Ethernet
 V umerjevalnikih se NE izvajajo aplikacije
 V usmerjevalnikih se NE izvajaja TCP (UDP)



Razvojni inženir aplikacij se ne ukvarja z razvojem programske opreme usmerjevalnikov.
 Razvija programsko opremo, ki se bo izvajala na končnih vozliščih.

Odjemalec strežnik

- Strežnik stalno deluje (za odjemalce to ni nujno)
- Strežnik ima stalen (fiksni ter znan naslov), za odjemalce to ni nujno.
- Pobudo za komunikacijo (zahtevo) da odjemalec, strežnik se odzove.
- Strežnik streže več odjemalcev, ki jih je lahko tudi zelo veliko.
- Strežniška končna vozlišča lahko tvorijo gručo oz. "farmo" strežnikov
- Odjemalci ne komunicirajo direktno med seboj, lahko pa posredno prek strežnika
- Nekaj najbolj znanih primerov:
 - elektronska pošta
 - svetovni splet
 - prenos datotek
 - enostaven koncept, lahko ga realizirate sami ...
 - ... in ga tudi boste v okviru vaj.



P2P

- Vsako končno vozlišče je hkrati odjemalec in strežnik.
- Odjemalec je tisti, ki začne komunikacijo.
- Bolj splošen koncept kot odjemalec-strežnik
- Skalabilen, ker je sam po sebi porazdeljen.

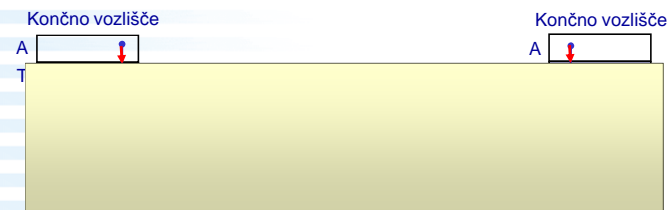
- Nekaj najbolj znanih primerov:

- BitTorrent
- eMule
- Skype
- senzorska in "ad-hoc" omrežja.



Komunikacijske vtičnice

- Vmesnik med aplikacijami in komunikacijskimi prenosnimi storitvami je zgornji vmesnik transportnega sloja (TCP/UDP).
- **Komunikacijske vtičnice:** "socket"



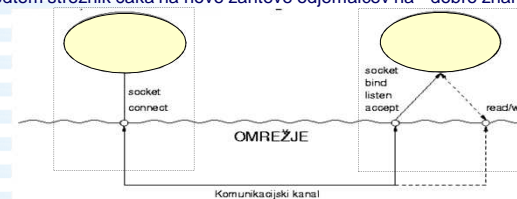
Bistvena naloga prenosnega sloja:
prenos podatkov med končnimi vozlišči za vse potrebe uporabnikov
To nalogo pa zagotavlja s pomočjo storitev nižjega (omrežnega) sloja, ki skrbi za povezljivost ter seveda obeh nižjih slojev.

Omrežne aplikacije - API

- Omrežne aplikacije vključujejo izvrševanje opravil na vsaj dveh napravah (končnih vozliščih) in komunikacijo med opravili, kot npr:
 - osebni računalnik <-> osebni računalnik
 - mikrokrmilnik <-> mikrokrmilnik
 - osebni računalnik <-> mikrokrmilnik
 - videonadzorna ("IP") kamera <-> iPhone
 - industrijski krmilnik (PLK) <-> senzor
 - ipd.
- Programu, ki se izvršuje na/v napravi, pravimo proces ali opravilo (proces nastane iz programa, ko se le-ta izvršuje)
- Razlika med "navadnim" programom/procesom in omrežno aplikacijo je le ta, da vključuje komunikacijo med procesi na različnih vozliščih.
- Podlago za izvedbo komunikacije med procesi daje sistemska programska oprema oziroma operacijski sistem (OS).
- Komunikacijske storitve OS so ponujene v obliki (programskega) vmesnika – API.

Komunikacijske vtičnice

- Vtičnice omogočajo aplikaciji (procesu) pridobiti komunikacijski kanal
- Komunikacija je asimetrična po načelu odjemalec-strežnik
- Strežnik pridobi en konec obojesmerne komunikacijskega kanala
- Potem čaka na "dobro znanem naslovu" na zahteve odjemalcev
- Odjemalec pridobi drugi konec obojesmerne komunikacijskega kanala
- Odjemalec poda zahtevo za vzpostavitev zveze s strežnikom
- Strežnik "odpre" nov komunikacijski kanal, na katerem streže zahtevam odjemalca.
- Komunikacija potem poteka v obe smeri, dokler se ne podre oz zapre.
- Medtem strežnik čaka na nove zahteve odjemalcev na "dobro znanem naslovu".



Komunikacijske vtičnice - številke vrat

- Kaj je "dobro znan naslov" strežnika?
 - To je naslov, ki je znan vsem zainteresiranim odjemalcem.
- Kaj mora poznati odjemalec?
 - Naslov končnega vozlišča (IP naslov), na katerem "domuje" aplikacijski strežnik.
- Ali zadostje zgolj naslov vozlišča?
 - Pravzaprav ne, na enem vozlišču lahko domuje več strežnikov.
 - Prenosni sloj (TCP/UDP) opravlja prenos za vse potencialne strežnike.
 - Potrebno je torej dodatno določilo, številka ali "naslov" prenosnega sloja.
- Naslov komunikacijske vtičnice, kamor je pripet strežnik (naslov aplikacijskega strežnika) sestavljata:
omrežni naslov: številka vrat
IP naslov: (TCP ali UDP) vrata ("port")
- Na primer:
193.2.72.170:80 (80 je naslov spletnega strežnika)
- ITD: 21: ftp, 25: smtp, 80: http, 110: pop3

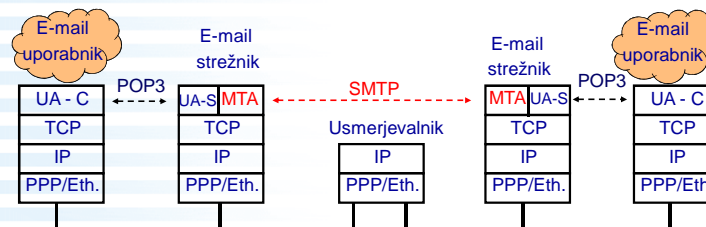
E-mail - elektronska pošta

- Sprva se je e-pošta prenašala "direktno" med uporabnikoma storitve, oz. direktno med uporabnikovima vozliščema.
- Dandanes se e-pošta prenaša med poštnimi strežniki, do katerih imajo dostop končni uporabniki storitev.
- Končni uporabniki dostopajo do strežnikov po protokolu **POP3** (ali **IMAP** ali **HTTP**)
- POP3 Protokol omogoča preproste operacije, prenos pošte na lokalni disk, ...
- Strežniki komunicirajo – prenašajo pošto – po protokolu **SMTP**
- Med sabo strežniki komunicirajo „direktno“,
tj strežnik končnega uporabnika (pošiljatelja) s strežnikom končnega uporabnika (prejemnika), brez posredovanja vmesnih strežnikov.

E-mail - elektronska pošta

- **E-pošta:**
verjetno največ uporabljena storitev, RFC 822, RFC 2822,, RFC 5321
- ena prvih (poleg prenosa datotek – ftp, in daljinskega terminala –telnet) *Internetnih* storitev. V zgodnjih letih TCP/IP omrežij pravzaprav realizirana na osnovi prenosa datotek, katerih vsebina je bila pač e-pošta).
- E-sporočilo sestavljata (poleg "ovojnice"):
 - glava
 - vsebina
- Sprva je zadoščalo ASCII kodiranje vsebine, dandanes je to premalo,
- MIME (The Multipurpose Internet Mail Exchange), strukturira vsebino in določa kako kodirati ne-ASCII vsebino. (RFC 1241, RFC2045, ...)

E-mail - elektronska pošta



- UA – C: Uporabnikov agent – odjemalec komunicira z UA – S: strežnikom uporabnikovega agenta po protokolu POP3 (ali IMAP)
- MTA: poštni strežnik - agent za prenos pošte (Mail Transfer Agent) komunicira z oddaljenimi poštnimi strežniki po protokolu SMTP (Simple Mail Transfer protocol)

E-mail - smtp - elektronska pošta

- **SMTP**: je enostaven znakovni - ASCII protokol
 - Poštni strežnik, ki komunicira po protokolu SMTP, posluša na vratih 25
 - Odjemalcu, ki se prijavi, sporoči, da je pripravljen
 - Odjemalec se javi s **HELO** (ukazi so 4-črkovni)
 - Strežnik odgovori s številsko kodo odgovora (in besedilom)
 - Odjemalec najavi odhodno pošto, **MAIL FROM**
 - Strežnik jo potrdi
 - Odjemalec najavi, komu pošilja, **RCPT TO**
 - Strežnik to potrdi
 - Sledi prenos, **DATA**
- SMTP je protokol, ki prenaša drugam – pošilja.

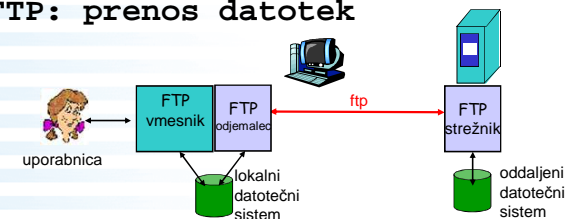
E-mail - pop3 - elektronska pošta

- **POP3** (Post Office Protocol, version 3), RFC 1939
- POP3 strežnik, ki komunicira po protokolu POP3, posluša na vratih 110
- Protokol je enostaven ASCII protokol
- Odjemalec se prijavi z imenom (**USER** ime) in geslom (**PASS** geslo)
- Odjemalec na primer preveri prispelo pošto, ukaz **LIST**
- Odjemalec nato zahteva prenos pošte, ene za drugo, **RETR 1**
- Odjemalec ev. zahteva brisanje pošte na strežniku, **DELE 1**
- I.T.D.
- Odjemalec zaključi transakcijo, **QUIT**

E-mail - smtp - elektronska pošta

```
S: 220 vasaposta.si
C: HELO nasaposta.si
S: 250 Hello nasaposta.si, pleased to meet you
C: MAIL FROM: <stane@nasaposta.si>
S: 250 stane@nasaposta.si ... Sender ok
C: RCPT TO: <mojca@vasaposta.si>
S: 250 mojca@vasaposta.si ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Je kaksno vprasanje?
C: Je kaksen komentar?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 vasaposta.si closing connection
```

FTP: prenos datotek



- prenos datoteke v eno ali drugo stran
- model odjemalec strežnik
 - **odjemalec**: pobuda za prenos
 - **Strežnik**: oddaljen računalnik
- ftp strežnik posluša na vratih 21

2: Application Layer (Kurose, Ross)

FTP: ločen prenos ukazov in podatkov

- FTP odjemalec se poveže s strežnikom na vratih 21
- Odjemalec se predstavi (ime, geslo)
- odjemalec izbere direktorij na strežniku
- ko strežnik dobi ukaz za prenos, strežnik odpre nov komunikacijski kanal na vratih 20
- ko je datoteka prenešana, strežnik zapre podatkovni kanal



- Strežnik odpre nov TCP kanal za prenos datoteke
- Nadzor prenosa: izven podatkovnega kanala
- FTP vzdržuje stanje: tekoči direktorij, avtentikacija, ...

Application Layer (Kurose, Ross)

DNS - Domain Name System

- Omrežje za naslavljanje vozlišč uporablja številske vrednosti naslovov – naslove: 193.2.72.151
- Uporabnikom je bližje “poimenovanje” vozlišč z imenom vozlišč, na primer: stane.fe.uni-lj.si
- Potrebna je povezava (preslikava) med imeni in naslovi.
- Poleg tega je potrebna ločitev imena od naslova (če se kdo preseli na nov naslov, mu zato še ni treba spremeniti imena ☺)

FTP ukazi in odgovori

Primeri ukazov:

- ASCII protokol
- **USER imeuporabnika**
- **PASS geslo**
- **LIST** vsebina direktorija
- **RETR imedatoteke**
- **STOR imedatoteke**

Primeri odgovorov:

- kode/pojasnila odgovorov
- 331 Username OK, password required
- 125 data connection already open; transfer starting
- 425 Can't open data connection
- 452 Error writing file

DNS - Domain Name System

- V zgodnjih letih omrežij TCP/IP je bila preslikava med imeni in naslovi urejena “lokalno” na posameznih vozliščih.
- V posebni datoteki (“hosts”) je bil za vsako ime naveden pripadajoč omrežni naslov.
- Vzdrževanje svežega stanja lokalnih datotek je sčasoma postalo nemogoče.
- Razvil se je DNS (RFC 1034, RFC 1035):
 - hierahično poimenovanje vozlišč
 - porazdeljena podatkovna baza, ki jo vzdržujejo imenski strežniki

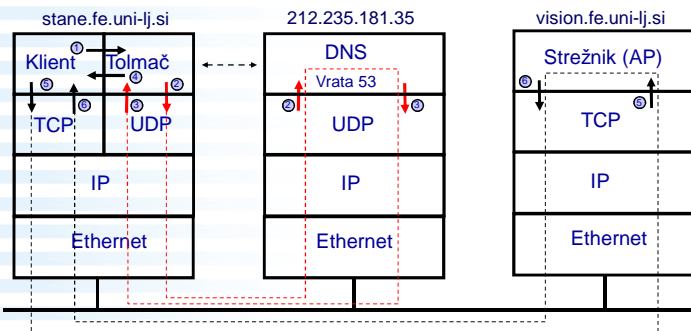
DNS - sistem domenskih imen - osnove

- Uporabniški proces (AP) želi komunicirati z oddaljenim procesom na drugem vozlišču, na primer z imenom: vision.fe.uni-lj.si
- IP naslova ne pozna, zato povpraša svojega imenskega "tolmača", na primer z:
gethostbyname(Ime vozlišča), to je
gethostbyname('vision.fe.uni-lj.si');
- Tolmač pošlje UDP paket z DNS poizvedbo lokalnemu imenskemu strežniku, na primer z naslovom IP = 212.235.181.35
- Imenski strežnik v UDP paketu odgovori z iskanim IP naslovom vozlišča vision.fe.uni-lj.si
- Komunikacija med oddaljenima aplikacijama sedaj lahko steče.

DNS

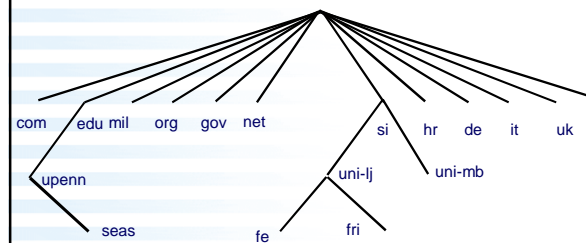
- Večino povpraševanj razreši lokalni imenski strežnik sam.
- V primeru, da lokalni imenski strežnik ne pozna imena, povpraša imenski strežnik, ki je zadložen za dotično "višjo" domeno.
- Poizvedovanje se nadaljuje (rekurzivno ali iterativno), dokler ime oz. naslov ni razrešeno, če seveda obstaja.
 - rekurzivno: s poizvedbo nadaljuje naslednji strežnik (na primer, lokalni strežnik povpraša "višji" strežnik, višji strežnik povpraša naprej, itd.)
 - iterativno: povprašani strežnik vrne naslov strežnika, ki naj se ga povpraša kot naslednjega. (na primer, lokalni strežnik povpraša "višji" strežnik, višji strežnik javi naslov strežnika, ki naj ga vpraša lokalni strežnik, itd.)

DNS imenski strežnik



- Komunikacija odjemalec (AP) / strežnik (AP)
- DNS imenski strežnik posluša na vratih 53 na poizvedbo tolmača in se na poizvedbo odzove - vrne iskani naslov strežnika AP.

DNS - hierarhična struktura imen



- preko 200 domen gornjega nivoja (generične, državne) pod-domene, pod-pod-domene, ...
- Hierarhičen imenski prostor omogoča bolj učinkovito upravljanje z imeni (dodeljevanje, iskanje, ...)

DNS - hierarhična struktura imen

- Imenski strežniki vzdržujejo imensko podatkovno bazo
- "Osnovni imenski zapis" poleg "imena domene" vsebuje še druga določila:
 - Čas veljavnosti
 - Tip
 - Vrednost
- Obstaja več tipov zapisov:
 - Tip A: vsebuje IP naslov (končnega) vozlišča – najpomembnejši tip
 - TIP NS: ime imenskega strežnika za dano domeno
 - TIP HINFO: podatki o dotičnem vozlišču (OS; CPE; ...)
- DNS poizvedba/odgovor (protokol) sta enostavna, tekstovna in se prenašata v UDP paketih.

HTTP in WWW

Za prenos HTTP paketov skrbi TCP:

- strežnik pasivno čaka na zahtevo za vzpostavitev zveze s strani odjemalca (na vtičnici 80)
- odjemalec da pobudo za vzpostavitev zveze (odpre vtičnico)
- strežnik sprejme zahtevo odjemalca (na vtičnici, št. vrat 80)
- odjemalec in strežnik si izmenjata sporočila po protokolu HTTP, seveda za to potrebujeta TCP prenos v obe smeri
- TCP na koncu zapre zvezo

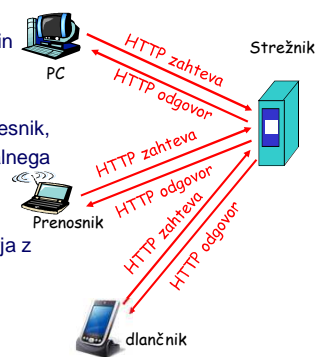
HTTP ne vzdržuje stanja protokola skozi čas

- Neperzistenten način – prenos enega objekta in sprostitve TCP zveze
- Perzistenten način – prenos večih objektov po istem TCP kanalu in nazadnje sprostitve zveze

HTTP in WWW

HTTP: hypertext transfer protocol

- protokol za prenos spletnih vsebin
- koncept odjemalec/strežnik
- odjemalec:
 - spletni brskalnik, uporabniški vmesnik, interpretacija opisnega/označevalnega jezika (html) in komunikacija po protokolu (http)
- strežnik: spletni strežnik odgovarja z vsebinami (objekti) na zahteve odjemalcev



HTTP zahteva

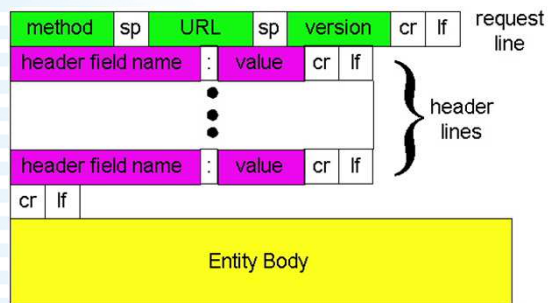
HTTP zahteva je v ASCII "berljivem" zapisu

Vrstica z zahtevo
(GET, PUT, POST, HEAD, DELETE)

glava
GET /page.html HTTP/1.1
Host: www.nasstreznik.si
User-agent: Mozilla/4.0
Connection: close

konec → (dodatni cr lf)

HTTP splošna oblika zahteve



Kurose, Ross: Application Layer

HTTP kode odzivov

Strežnik -> odjemalec

200 OK

- Zahteva uspeša, zahtevani objekt vsebovan (sledi) v odgovoru

301 Moved Permanently

- Zahtevani objekt se je preselil, njegova nova lokacija vsebovana (sledi) v odgovoru

400 Bad Request

- Neznana zahteva

404 Not Found

- Zahtevani dokument ne obstaja

505 HTTP Version Not Supported

HTTP odziv

Statusna vrstica
(protokol, koda odgovora, fraza odgovora)

glava

```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 .....
Content-Length: 6821
Content-Type: text/html
```

podatki, npr zahtevani HTML dokument

```
data data data data data ...
```

Poskus s telnet na 80

1. Telnet na strežnik

```
telnet vision.fe.uni-lj.si 80
```

Odpre TCP povezavo na vrata 80 na vision.fe.uni-lj.si

2. Tipkaj GET HTTP zahtevo:

```
GET / HTTP/1.1
Host: vision.fe.uni-lj.si
Dvakrat CR LF
```

3. Poglejte odziv HTTP strežnika ;)

Viri

J. Kurose, K. Ross, Computer Networking, 5th ed. Pearson, 2010.

Viri na Internetu:

Wikipedia

<http://en.wikipedia.org/>

<http://www.w3.org/>