

Uvod u veb i internet tehnologije

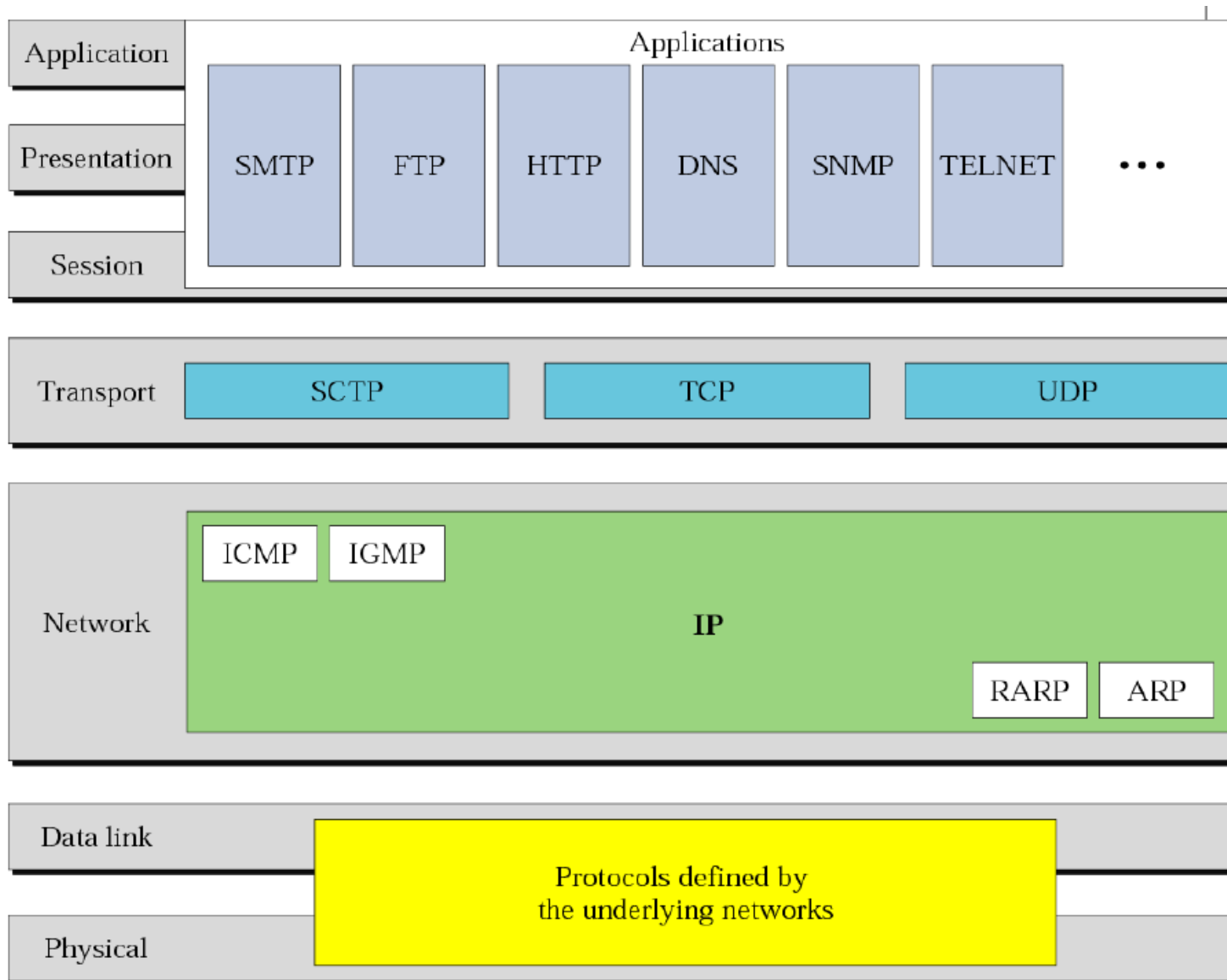




Slojevi kod računarskih mreža transportni sloj



Protokoli i slojevi





Transportni sloj

Transportni sloj (transport layer) - ima zadatak da prihvata podatke sa viših slojeva, deli ih na manje jedinice (pakete), šalje te pakete na odredište korišćenjem nižih slojeva

- Obično se na ovom sloju razlikuju dve vrste protokola: protokoli sa uspostavljanjem konekcije i protokoli bez uspostavljanja konekcije
 - Protokoli koji zahtevaju uspostavljanje konekcije garantuju da će poslati podaci zaista i stići na odredište u istom redosledu u kojem su i poslati
 - Protokoli bez uspostavljanja konekcije ne daju ovakve garancije, ali je prenos podataka obično brži
- Za razliku od protokola mrežnog sloja koji moraju da budu implementirani u svakom čvoru lanca komunikacije, protokoli transportnog sloja moraju biti implementirani jedino na krajnjim čvorovima komunikacije (host čvorovima)



Transportni sloj (2)

- Ruteri (uredaji koji posredno učestvuju u komunikaciji prenošenjem paketa) obično nisu svesni detalja transportnih protokola
- Transportni protokoli se, dakle, mogu smatrati protokolima kojim komuniciraju dva host računara
- S obzirom da na istom host računaru obično postoji više različitih programa koji imaju potrebu za komunikacijom (svaki korišćenjem zasebnog aplikacionog protokola, ali zajedničkim korišćenjem transportnog protokola), zadatak transportnih protokola je i da vrše tzv. **multipleksovanje**
 - Multipleksovanje se obično ostvaruje kroz koncept **portova** koji predstavljaju brojeve na osnovu kojih se određuje kom programu pokrenutom na host računaru pripada paket primljen na transportnom sloju
- Najkorišćeniji protokoli ovog sloja (koji se koriste u okviru Interneta) su **Transfer Control Protocol (TCP)** i **User Datagram Protocol (UDP)**



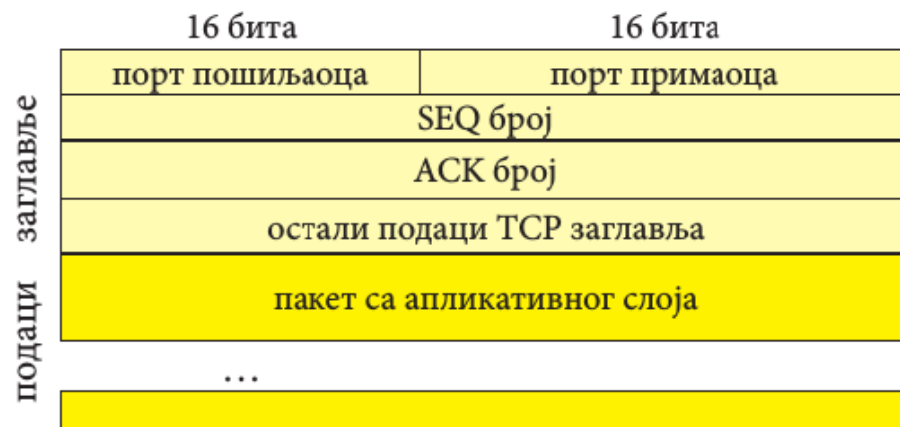
Protokoli transportnog sloja

- Poruka se deli na **pakete** koji se nezavisno šalju (komutiranje paketa)
- Više delova iste poruke može paralelno da putuje kroz mrežu
- Svaki paket se dopunjuje informacijama potrebnim za njegovu dostavu
- Na transportnom sloju paketi se nazivaju **segmenti**
- Komunikacija se organizuje ne samo kao komunikacija između dva uređaja, već između dva programa koji se na njima izvršavaju
- Paket mora da sadrži informaciju o uređaju i softveru koji paket prima i koji paket šalje
- Na transportnom nivou se paketima dodaju identifikatori softvera - portovi, a adrese uređaja tek na mrežnom sloju



TCP protokol transportnog sloja

- TCP (Transmission Control Protocol) je protokol transportnog sloja u okviru Interneta koji pre komunikacije vrši uspostavljanje pouzdane konekcije između dva hosta
 - Kanal komunikacije je dvosmeran (eng. full duplex)
 - Konekcija se uspostavlja tako što klijent i server razmene tri poruke (three way handshake):
 - Klijent traži uspostavljanje konekcije, server potvrđuje da prihvata konekciju i konačno klijent potvrđuje da je konekcija uspostavljena
 - Prava komunikacija može da započne tek nakon što je konekcija uspostavljena, što može da traje neko vreme





TCP protokol transportnog sloja (2)

- TCP garantuje pouzdanost prenosa podataka (reliable transfer) čime se garantuje da će paketi koji su poslani biti primljeni (i to u istom redosledu u kojem su poslani). S obzirom da niži mrežni slojevi ne garantuju dostavu paketa:
 - TCP protokol mora da se stara o tome da paketi koji zalutaju automatski budu ponovno poslani, kao i da na prihvatnoj strani automatski permutuje primljene pakete tako da odgovaraju redosledu slanja
 - Da bi ovo moglo da bude realizovano, uvodi se potvrda prijema paketa (acknowledgment), tj. nakon prijema jednog ili više paketa, vrši se slanje poruke pošaljiocu koja govori da su ti paketi zaista primljeni
 - Pošaljioc, na osnovu ovoga, može da odluči da ponovno pošalje paket koji je ranije već bio poslat, u slučaju da u određenom vremenskom periodu ne dobije potvrdu prijema



TCP protokol transportnog sloja (3)

- TCP uvodi kontrolu i korekciju grešaka (error correction)
 - Ovo je dodatna slaba provera (vrši se samo kontrola parnosti), jer se pretpostavlja da se jača provera (obično CRC) vrši na nižim slojevima
 - Ipak, u praksi se pokazuje da ova provera ima smisla i uspeva da uoči i ispravi veliki broj grešaka koje promaknu ostalim kontrolama
- TCP uvodi i kontrolu brzine protoka (flow control)
 - Njom se kontroliše brzina slanja kako se ne bi desilo da brzi uređaji šalju pakete brzinom većom od one kojom spori uređaji mogu da ih prime (npr. brz računar koji šalje podatke na spor mobilni telefon)



TCP protokol transportnog sloja (4)

- Važna odlika TCP protokola je da vrši kontrolu zagušenja (congestion control)
 - Pojava zagušenja se javlja kada više čvorova pokušava da pošalje podatke kroz mrežu koja je već na granicama svoje propusne moći
 - U takvim situacijama, dešava se da brzina komunikacije u celoj mreži opada za nekoliko redova veličina
 - Naime, broj izgubljenih paketa se višestruko povećava jer unutrašnji čvorovi mreže (ruteri) ne mogu da prihvate nove pakete zato što su im prihvatni baferi prepuni
 - TCP pokušava da detektuje ovakve situacije i da u tim slučajevima uspori sa slanjem paketa dok se mreža ne rastereti
 - Jedna od tehnika koje se koriste u cilju smanjenja zagušenja je da se pri početku komunikacije paketi šalju sporije (slow-start), a da se brzina slanja postepeno povećava kada se utvrdi da paketi zaista i stižu na odredište



TCP protokol transportnog sloja (5)

- Činjenica da TCP protokol da vrši kontrolu zagušenja je jedan od razloga zbog čega TCP spada u grupu sporijih protokola
- Stoga se TCP ne koristi se za aplikacije kod kojih je brzina prenosa presudna



UDP protokol transportnog sloja

- **UDP** (User Datagram protocol) je protokol transportnog sloja u okviru Interneta koji ne vrši uspostavljanje konekcije pre započinjanja komunikacije
 - Prilikom korišćenja UDP protokola ne vrši se potvrda prijema poslatih paketa, tako da se komunikacija može smatrati nepouzdanom
 - Osnovni razlozi korišćenja UDP protokola su, pre svega, njegova brzina - zbog toga se uglavnom koristi od strane aplikacija koje imaju potrebu za komunikacijom u realnom vremenu (real time), kao što su npr. audio-video prenosi, internet telefonija, igrice i sl.
 - Takođe, UDP se koristi za aplikacione protokole koji daju elementarne mrežne usluge i vrše kontrolu mreže (npr. DHCP, DNS, SNMP)



Sistem imena domena

- IP adrese su pogodne za korišćenje od strane računara, ali nisu pogodne za ljudsku upotrebu
- Stoga je uveden je sistem imena domena (domain name system - DNS) – adrese hostova (servera) zadaju se u tekstualnom obliku
- DNS se smatra „telefonskim imenikom” Interneta, koje imenima domena dodeljuje razne informacije (najčešće IP adrese)
 - Na primer, već pomenuti studentski server Matematičkog fakulteta u Beogradu ima domen `alas.matf.bg.ac.rs`
- Domeni su hijerarhijski organizovani i čitaju se s desna na levo
 - Na primer, domen `rs` označava Republiku Srbiju, `ac.rs` označava akademsku mrežu u Srbiji, `bg.ac.rs` njen čvor u Beogradu, `matf.bg.ac.rs` označava Matematički fakultet, dok `alas.matf.bg.ac.rs` označava konkretan studentski server koji tako imenovan u čast velikog matematičara Mihaila Petrovića Alasa
- Domeni najvišeg nivoa mogu biti bilo nacionalni (kao u navedenom primeru), bilo generički (npr. `.com`, `.org`, `.net`), a novom regulativom je liberalizovano korišćenje domena najvišeg nivoa



Sistem imena domena (2)

- Domeni se koriste u okviru jedinstvenih lokatora resursa na Vebu (URL), u okviru adresa elektronske pošte, itd.
- Prilikom preslikavanja domena u adrese, koriste se usluge distribuirane DNS baze podataka
 - Specijalizovani DNS serveri čuvaju delove ove baze
 - Ovi serveri su hijerarhijski organizovani i njihova hijerarhija uglavnom prati hijerarhiju domena

```

Command Prompt

DNS Suffix Search List. . . . . : matf.bg.ac.rs

Ethernet adapter Ethernet 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . : matf.bg.ac.rs
    Description . . . . . : Intel(R) PRO/1000 PM Network Connection
    Physical Address. . . . . : 00-24-21-5F-EE-06
    DHCP Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix . : 
    Description . . . . . : Intel(R) 82566DM-2 Gigabit Network Connection
    Physical Address. . . . . : 00-24-21-5F-EE-05
    DHCP Enabled. . . . . : No
    Autoconfiguration Enabled . . . . : Yes
    Link-local IPv6 Address . . . . . : fe80::a177:9abb:a200:158e%12(Preferred)
    IPv4 Address. . . . . : 147.91.67.138(Preferred)
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 147.91.67.1
    DHCPv6 IAID . . . . . : 201335841
    DHCPv6 Client DUID. . . . . : 00-01-00-01-1C-29-0A-22-00-24-21-5F-EE-05

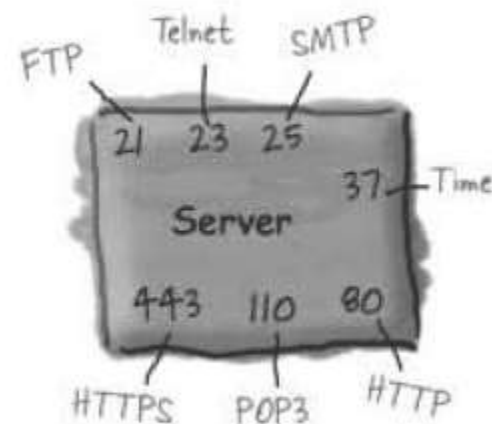
    DNS Servers . . . . . : 147.91.64.4
                           147.91.66.2
    NetBIOS over Tcpip. . . . . : Enabled
  
```



TCP порт

- Различите апликације користе различите портове
 - HTTP сервер подразумевано користи порт 80
 - Telnet сервер подразумевано користи порт 23
 - FTP сервер подразумевано користи порт 21
 - POP3 порт 110, а SMTP порт 25
 - Сервер за време порт 37
- Порт представља једнозначни идентификатор – број између 0 и 65535
- Порт представља логичку везу између конкретног софтвера и хардвера на ком се тај софтвер извршава
- Порт не представља место на које се прикључује уређај, већ број који означава приступну тачку за комуникацију са сервером

Well-known TCP port numbers for common server applications



Using one server app per port, a server can have up to 65536 different server apps running.



TCP порт (2)

- Ако не би било броја порта, онда сервер не би знао са којом клијентском апликацијом комуницира, нити по ком се протоку та комуникација реализује
 - Могло би се догодити, на пример, да прегледач, уместо да комуницира са веб сервером, покушава успостави комуникацију са сервером електронске поште
- TCP портови између 0 и 1023 су резервисани за познате сервисе и не препоручује се да се ти портови користе за „нове“ серверске програме
- Ако се извршавају серверски програми на рачунарској мрежи компаније, тада са администратором система треба проверити који су портови слободни, а који заузети



TCP/IP programski interfejs

- Većina savremenih operativnih sistema i programskih jezika daje direktnu podršku za korišćenje Internet (TCP/IP familije) protokola
- Podrška za korišćenje ovih protokola u okviru programa se realizuje kroz koncept soketa (socket)
- Soket je apstrakcija kojom se programeru predstavlja kanal komunikacije (zasnovan bilo na TCP bilo na UDP protokolu)
- Programer piše podatke u soket ili čita podatke iz soketa, obično na sličan način kao da je u pitanju obična datoteka, a operativni sistem se brine o svim aspektima stvarne mrežne komunikacije



TCP/IP programski interfejs (2)

```
import java.io.*;
import java.net.*;
class TCPClient {
    public static void main(String argv[]) throws Exception {
        String userSentence, modifiedSentence;
        BufferedReader inFromUser =
            new BufferedReader(new InputStreamReader(System.in));
        Socket clientSocket = new Socket("hostname", 6789);
        DataOutputStream outToServer =
            new DataOutputStream(clientSocket.getOutputStream());
        BufferedReader inFromServer =
            new BufferedReader(new InputStreamReader(
                clientSocket.getInputStream()));
        userSentence = inFromUser.readLine();
        outToServer.writeBytes(userSentence + '\n');
        modifiedSentence = inFromServer.readLine();
        System.out.println("FROM SERVER: " + modifiedSentence);
        clientSocket.close();
    }
}
```



TCP/IP programski interfejs (3)

```
import java.io.*;
import java.net.*;
class TCPServer {
    public static void main(String argv[]) throws Exception {
        String clientSentence, modifiedSentence;
        ServerSocket welcomeSocket = new ServerSocket(6789);
        while(true) {
            Socket connectionSocket = welcomeSocket.accept();
            BufferedReader inFromClient =
                new BufferedReader(new InputStreamReader(
                    connectionSocket.getInputStream()));
            DataOutputStream outToClient =
                new DataOutputStream(connectionSocket.getOutputStream());
            clientSentence = inFromClient.readLine();
            modifiedSentence = clientSentence.toUpperCase() + '\n';
            outToClient.writeBytes(modifiedSentence);
        }
    }
}
```



Zahvalnica

Delovi materijala ove prezentacije su preuzeti iz:

- Skripte iz predmeta Uvod u veb i internet tehnologije, na Matematičkom fakultetu Univeziteta u Beogradu, autor prof. dr Filip Marić
- Prezentacija iz predmeta Uvod u veb i internet tehnologije, na Matematičkom fakultetu Univeziteta u Beogradu, autor dr Vesna Marinković
- Skripte iz predmeta Informatika na Univerzitetu Milano Bicocca, autor dr Dario Pescini