

## Vežba 6 – Protokol korisničkih datagrama (UDP)

### Teorijske osnove

UDP (engl. *User Datagram Protocol*) protokol definiše uslugu nepouzdanu isporuku korisničkih podataka (datagrama) preko IP-a. Konceptualno, jedina bitna razlika između UDP datagrama i IP datagrama je u tome što UDP sadrži brojeve portova, što omogućava predajnoj aplikaciji da se obrati tačno određenoj aplikaciji na određenoj mašini.

Osobine UDP-a:

- Nepouzdana isporuka podataka
- Nije obavezna uspostava konekcije pre slanja podataka.
- Ne koristi *sequence number*, *acknowledgment number*, niti obavlja kontrolu toka podataka. Ne postoji zaštita od zagušenja prijemnika velikim brojem poruka.
- Kada se pošalju podaci, nema načina da se sazna da li su stigli ili da li su stigli redosledom kojim su poslani. Međutim, podaci koji stižu, stižu brzo što je bitno za neke vrste servisa tj. aplikacija kojima je brzina važnija od pouzdanosti isporuke podataka (npr. real-time audio i video).

Većina operativnih sistema omogućava simultano izvršenje više aplikativnih programa. U teoriji operativnih sistema program pripravan za izvršenje ili u fazi izvršenja se naziva procesom. Komunikaciona programska podrška treba da omogući komunikaciju distribuiranih procesa programa koji se izvršavaju na različitim računarima. Međutim zadavanje određenog procesa na određenom računaru kao odredišta datagrama je donekle pogrešno iz sledećih razloga.

1. Procesi nastaju i nestaju dinamički pri čemu pošiljalac datagrama nema dovoljno informacije da bi identifikovao proces na drugom računaru.
2. Poželjno je da postoji mogućnost zamene procesa prijemnika datagrama bez potrebe da se o tome informišu pošiljaoci (npr. zbog lokalnog restarta računara).
3. Odredišta treba da se identifikuju prema funkcijama koje izvode bez potrebe da se poznaje proces koji funkciju izvodi (npr. treba omogućiti pošiljaocu da kontaktira server datoteka (engl. *File server*) bez potrebe da on zna koji konkretan proces na određenoj mašini izvodi funkciju server datoteka).

4. U slučaju da jedan proces podržava više funkcija, mora postojati način da taj proces odredi koju tačno funkciju pošiljaoc datagrama želi.

Iz tih razloga, umesto razmišljanja o procesu kao odredištu datagrama, treba zamišljati da svaki računar ima skup apstraktnih odredišnih tačaka, koje se nazivaju prolazi protokola (engl. *Protocol ports*). Ovi prolazi se identifikuju prirodnim (pozitivnim celim) brojevima. Lokalni OS je zadužen da obezbedi mehanizam procesima za definisanje i pristup prolazima protokola. Najčešće se koristi mehanizam poštanskih sandučića (engl. *Mailbox*).

Na većini sistema mehanizam pristupa prolazu protokola je sinhron, odnosno proces koji želi da primi poruku iz praznog prolaza se blokira do pojave prve poruke za taj prolaz, čime se proces aktivira (vraća u red pripravnih procesa) radi obrade te poruke. S druge strane, mehanizam pristupa prolazu je i baferovan, naime u slučaju prijema poruke na nekom prolazu u trenutku kad nema procesa koji na nju čeka, ta poruka se smešta u konačan red poruka, odakle je proces kasnije preuzima bez prethodnog blokiranja. Da bi se omogućila komunikacija u mreži, pošiljaoc mora da specificira IP adresu odredišnog računara i broj prolaza protokola za funkciju koju želi, ali i broj prolaza protokola na koji treba slati odgovor (tzv. izvorišni broj prolaza protokola) i naravno IP adresu izvorišne mašine. Na taj način proces koji primi poruku može odgovoriti procesu pošiljaocu. UDP razlikuje više procesa na jednoj mašini i omogućava predajniku (engl. *Sender*) i prijemniku (engl. *Receiver*) da svakoj UDP poruci dodaju 2 16-bitna broja, koji identifikuju prolaze izvora i odredišta (Slika1).

<b>UDP SOURCE PORT (UDP PROLAZ IZVORIŠTA)</b>	<b>UDP DESTINATION PORT (UDP PROLAZ ODREDIŠTA)</b>
<b>UDP MESSAGE LENGTH (DUŽINA UDP PORUKE)</b>	<b>UDP CHECKSUM (KONTROLNA SUMA)</b>
<b>DATA (PODACI)</b>	
...	

Slika 1. Oblik UDP poruke

- Polje “*UDP SOURCE PORT*” je neobavezno. Ako se koristi, specificira prolaz na koji treba slati odgovore. Ako se ne koristi, treba da bude 0. Pošto se portovi zadaju kao 2-bajtni neoznačeni brojevi, 65536 različitih mogućih UDP portova je dostupno po hostu.
- Polje “*UDP MESSAGE LENGTH*” sadrži broj okteta u UDP datagramu, uključujući zaglavlje i korisničke podatke (min. dužina je 8 bajtova, tj. dužina zaglavlja). Dužina je takođe 2-bajtni neoznačeni ceo broj, pa je broj bajtova u datagram-u ograničen na 65536-8 za zaglavlje. Međutim, to je teoretska granica. U praksi je veličina znatno manja.
- Polje “*UDP CHECKSUM*” sadrži kontrolnu sumu koja se izračunava nad pseudozaglavljem UDP datagrama (Slika 2). Ako se kontrolna suma ne poklopi, native mrežni software tiho odbaci datagram, ne obaveštavajući o tome ni pošiljaoca ni primaoca.
- Količina podataka u UDP datagramu je na mnogim platformama ograničena na 8192 bajta (8K). Implementacije ne moraju prihvatiti datagrame sa više od 576 bajtova ukupno, uključujući podatke i zaglavlja. Zato, treba biti jako obazriv sa programima

koji šalju ili primaju UDP pakete sa više od 8K podataka. Veći deo vremena, veći paketi se jednostavno skrate na 8K. Za maksimalnu bezbednost, data deo UDP paketa treba da bude 512 bajtova ili manji, iako se to ograničenje negativno odražava na performanse u poređenju sa paketima veće veličine.

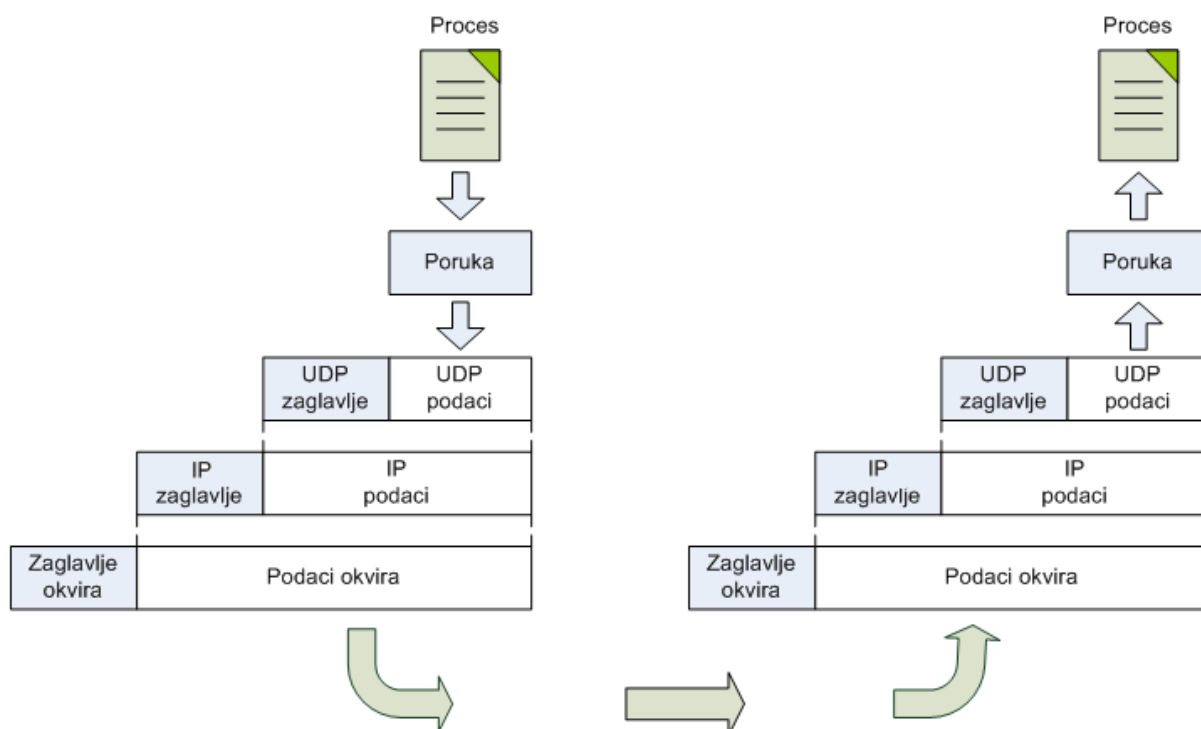
<b>SOURCE IP ADDRESS (IP ADRESA IZVORIŠTA)</b>		
<b>DESTINATION IP ADDRESS (IP ADRESA ODREDIŠTA)</b>		
<b>ZERO (NULA)</b>	<b>PROTO (17 ZA UDP)</b>	<b>UDP LENGTH (DUŽINA UDP DATAGRAMA)</b>

Slika 2. Oblik pseudozaglavlja UDP datagrama

<b>APLIKACIJA</b>
<b>UDP PROTOKOL</b>
<b>INTERNET (IP)</b>
<b>MREŽNA SPREGA</b>

Slika 3. Nivoi protokola  
(UDP je transportni protocol)

UDP poruka se smešta u IP datagram, a on u fizički okvir (Slika 4).



Slika 4. UDP enkapsulacija

UDP port se realizuje kao red čekanja (engl. *queue*). OS stvara ovaj red na zahtev aplikacije. Aplikacija može da zada ili promeni veličinu reda čekanja. Nakon prijema UDP datagrama, UDP proverava broj prolaza sa brojevima koji su trenutno u upotrebi. Ako se zadati prolaz ne koristi, šalje se ICMP poruka “port unreachable” i UDP datagram se odbacuje. U suprotnom, UDP ulančava UDP datagram u red čekanja, osim ako je on pun, kad dolazi do greške i odbacivanja primljenog UDP datagrama.

Postoji nekoliko slučajeva kada je bolje koristiti UDP umesto TCP protokola:

- Kada je blok podataka koji treba poslati mali, veličine jednog paketa – jednostavnije je, brže i efikasnije prenositi samo podatke (uz zaglavlje UDP-a), pa u slučaju pogrešno primljene poruke ponoviti slanje, nego uspostavljati vezu i proveravati pouzdanost prenosa.
- UDP koriste poruke tipa upita koje jedan računar šalje drugom, pri čemu se ako odgovor ne stigne u nekom određenom vremenu, zahtev ponovi ili se od njega odustane.
- Neke aplikacije imaju sopstvene tehnike za pouzdani prenos podataka i ne zahtevaju korišćenje TCP protokola, tako da je tada bolje koristiti UDP.

Neki rezervisani portovi koje koristi UDP:

- Echo – 7
- Daytime – 13
- Nameserver – 53
- Bootpc – 68
- RPC – 111
- NTP – 123

- Razlika između TCP i UDP često se objašnjava analogijom između telefonskog i poštanskog sistema.
- TCP je poput telefonskog sistema. Kada pozovete broj, telefon se javi i konekcija između dve strane je uspostavljena. Dok pričate, znate da druga strana čuje vaše reči onim redom kojim ih pričate. Ako je telefon zauzet ili se niko ne javlja, odmah to znate.
- UDP, suprotno, je kao poštanski sistem. Šaljete poštu na neku adresu. Većina pisama stigne, ali neka mogu biti izgubljena. Pisma verovatno stižu redom kojim su poslata, ali ne postoji garancija. Što smo dalje od primaoca, verovatnije je da će pošta biti izgubljena ili stići nekim drugim redosledom. Ako je to problem, možemo pisati redne brojeve a zatim tražiti od primalaca da ih uredi i pošalju pismo koje kaže koja pisma su stigla tako da možemo ponovo da pošaljemo ona koja nisu. Međutim, mi i primalac to moramo da dogovorimo unapred. Pošta to neće uraditi za nas.

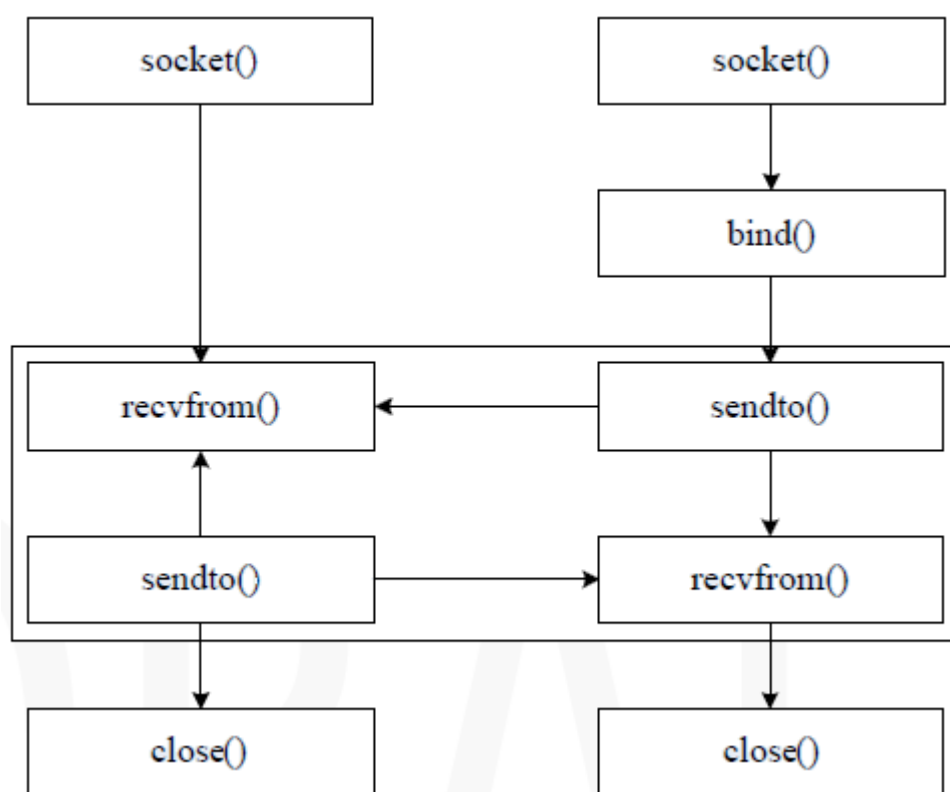
## Primene UDP-a

Jedna oblast gde je UDP naročito koristan su izvesne klijent-server konfiguracije. Često, klijent šalje kratak upit serveru i očekuje kratak odgovor. Ukoliko se upit ili odgovor izgube u prenosu, klijent jednostavno čeka neko vreme i pokušava ponovo. Primer aplikacije koja koristi UDP na ovaj način je *Daytime* servis. Program kome je potrebna informacija o tekućem vremenu šalje UDP datagram sa zahtevom *Daytime* serveru. Server odgovara UDP datagramom sa upisanim tekućim datumom i vremenom. Da bi se obavila ova prosta konverzacija, nije potrebna nikakva prethodna priprema ili uspostavljanje konekcije, dovoljno je razmeniti dve kratke poruke.

Druga oblast primene UDP protokola su *real-time* multimedijalne aplikacije, kao što su: Internet radio, Internet telefonija, muzika-na-zahtev, video konferencije, video-na-zahtev i druge. Zajednička karakteristika svih ovih aplikacija je prenos kontinualnog toka digitalizovanog zvuka i/ili videa. Na predajnoj strani, zvuk (ili video) se konvertuje u niz digitalnih odmeraka. Odmerak je binarni broj koji ukazuje na trenutnu amplitudu signala. Odmerci se generišu frekvencijom koja je dovoljno visoka da omogući vernu reprodukciju (npr. 44kHz za muziku). Odgovarajući proces deli generisani tok odmeraka na segmente (od po npr. 100 odmeraka) i pakuje ih u UDP datagrame koje šalje prijemnoj strani. Na taj način, brzi tok odmeraka, konvertovan je u tok UDP datagrama. Odgovarajući proces na prijemnoj strani dobija UDP datagrame, izdvaja odmerke i reprodukuje ih tempom koji odgovara frekvenciji odmeravanja. U prenosu UDP datagrama može se ispoljiti džiter (engl. *jitter*), a pojedini datagrami mogu biti izgubljeni u prenosu, što narušava kvalitet reprodukcije. Međutim, s obzirom da se radi o *real-time* toku (neprekidnom) retransmisija izgubljenih datagrama nije moguća (jer nema vremena za čekanje), kao ni neka stroga kontrola protoka. Iz tog razloga za pomenute aplikacije se koristi UDP (a ne TCP), a aplikaciji se prepušta da prevaziđe tj. ublaži probleme koji nastaju gubitkom ili kašnjenjem paketa. Na primer, umesto da se traži ponovno slanje izgubljenog paketa, aplikacija na prijemu može sama da pokuša da rekonstruiše deo zvuka koji nedostaje. Takođe, umesto da odmerke odmah reprodukuje može privremeno da ih smešta u bafer, čime će uneti izvesno kašnjenje u reprodukciji, ali zato će moći da toleriše veće kašnjenje pojedinih paketa, koje će kada stignu da umetne na pravo mesto u baferu.

## ZADACI VEŽBE

- Na osnovu primera iz vežbe 4 (TCP klijent/server) napisati UDP klijent i UDP server programe koji obavljaju sledeće funkcije:
  - Server kreira UDP utičnicu, povezuje je sa lokalnom adresom i protokolom tj. komunikacionom linijom i čeka na dolazeće UDP datagrame.
  - Posle svakog primljenog datagrama server šalje nepromenjeni datagram nazad odgovarajućem klijentu. Server ispisuje IP adresu i prolaz protokola svakog od klijenata.
  - Klijent čita podatke sa standardnog ulaza sve dok ne primi CR/LF kombinaciju znaka. Pročitane znake klijent šalje do servera čiji su naziv i prolaz protokola specificirani (ime servera i prolaz protokola na kome server sluša). Klijent na osnovu *gethostbyname* funkcije određuje IP adresu servera. Kada jednom klijent pošalje datagram čeka dok ne primi datagram od servera i ispisuje ga na standardni izlaz.
  - Implementirati funkciju koja vodi statistiku o UDP paketima na strani servera: minimalna, maksimalna, prosečna veličina paketa(korisnički podaci) na osnovu IP adrese klijenta.



Slika 5. UDP utičnice - dijagram sekvenci (Berkeley API). Klijent(levo) - Server(desno) arhitektura