



UNIVERZITET U
NOVOM SADU



FAKULTET
TEHNIČKIH NAUKA

Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija
Dekanat: 021 350-413 450-810; Centrala: 021 350-122
Računovodstov: 021 58-220; Studentska služba: 021 350-763
Telefon/fax: 021 58-133; ftndeans@uns.ac.rs



Sertifikovani
sistema
kvaliteta



Studijski program
Geodezija i geomatika

Seminarski rad
Integrirani sistemi premera
UDP prijem i slanje poruka

Profesor: dr Vladimir Bulatović
Student: Nikola Janković GG39/2011

Novi Sad, Jun 2015

Sadržaj

1	Uvod	3
2	Prenos podataka i osnove komunikacija.....	4
2.1	Računarska mreža	4
2.2	Tipovi mreža	4
3	Modeli slojevite računarske komunikacije	5
3.1	OSI referentni model	5
3.2	TCP/IP model.....	7
4	UDP protokol.....	8
4.1	Uvod.....	8
4.2	Zaglavlje UDP datagrama	8
4.3	UDP portovi	9
4.4	Multipleksiranje i demultipleksiranje	10
4.5	Primena UDP.....	10
4.6	Prednosti UDP.....	11
5	Klijent-server aplikacija u programskom jeziku C#.....	12
6	Prilog	13
	Literatura.....	16

1 Uvod

Potreba za informacijama naterala je čoveka da uspostavlja veze sa raznim izvorima informacija i da stvara mreže preko kojih će sebi olakšati prikupljanje, prenos, skladištenje i obradu podataka. Naglim razvojem računarske tehnologije poslednjih godina (povećanje performansi uz pad cena) i sa pravom eksplozijom Interneta, broj korisnika računara i računarskih mreža raste vrtoglavom brzinom. Sa sve moćnijom računarskom opremom svakodnevno se uvode novi servisi, a istovremeno se umrežavanju postavljaju viši standardi. Vremenom su se mrežni sistemi razvijali da bi danas dostigli nivo praktičnog efikasnog okruženja za razmenu podataka. [1]

Terminologija [2]

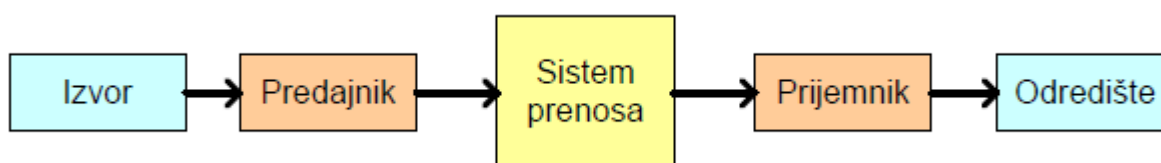
- ISP - Internet Service Provider - kompanija koja pruža internet pristup uređajima (SBB, Telekom, Orion)
- LAN - Local Area Network - mreža koja se održava na nekom manjem, lokalnom području (kućna mreža, poslovna mreža)
- WAN - Wide Area Network - mreža koja se održava na većem području (Internet)
- IP - Internet protocol address - numerička adresa koja odgovara nekom računaru u mreži. Kada se računari povezuju povezuju se pomoću adresa.
- IPv4 i IPv6 - Dva tipa adresa u upotrebi. IPv4 su starije i njihov problem je ograničenost u kvantitativnom smislu dok su IPv6 novije i ima više kombinacija slova i brojeva.
- Router - Uređaj koji preusmerava mrežni saobraćaj. Ima posao da šalje poruke ka internetu sa uređaja i da prima poruke sa interneta i prosledi ih uređaju.
- Gateway - Uređaj koji preusmerava saobraćaj između dve mreže najčešće WAN i LAN.
- DNS - Domain Name System - predstavlja konverziju iz naziva domena u IP adrese.
- localhost - naziv domena koji odgovara uređaju koji se trenutno koristi.
- Port - Numerička vrednost (1 - 65535) koja se koristi kada više aplikacija pristupaju istoj mreži.
- Protokoli - TCP, UDP, ICMP - različiti načini komuniciranja preko interneta.
- Paket - osnovna jedinica informacije u mreži.
- URL - Uniform Resource Locator - još poznato kao web adresa.
- HTTP - Hyper Text Transfer Protocol - standardni protokol koji koriste internet pretraživači za prikaz sadržaja.

2 Prenos podataka i osnove komunikacija

2.1 Računarska mreža

Računarska mreža se može posmatrati kao komunikacioni sistem, gde se informacija generisana na predajnoj strani (izvorište poruke) dostavlja željenom odreštu. Osnovni elementi komunikacionog sistema su [1]:

- Izvor (*source*) - generiše podatke za prenos,
- Predajnik (*transmitter*) - transformiše generisane podatke u oblik pogodan za prenos (npr. modem digitalne podatke iz PC računara transformiše u analogni signal koji se može preneti preko javne telefonske mreže - PSTN),
- Prenosni sistem (*transmission system*) - može biti jednostavna linija ili kompleksna mreža koja spaja izvor i odredište,
- Prijemnik (*reciever*) - prihvata signal iz prenosnog sistema i transformiše ga u oblik pogodan za odredište,
- Odredište (*destination*) - prihvata prenete podatke.



Slika 1. Komunikacioni sistem

2.2 Tipovi mreža

Podelu računarskih mreža moguće je vršiti po više kriterijuma. U skladu sa medijumom koji se koristi za prenos podataka računarske mreže mogu biti [1]:

- Kablovske mreže
- Bežične mreže

Po topologiji računarske mreže mogu biti:

- Bus network
- Star network
- Ring network
- Mesh network
- Star-bus network

Po vremenskoj postojanosti mreže mogu biti:

- fiksne
- privremene

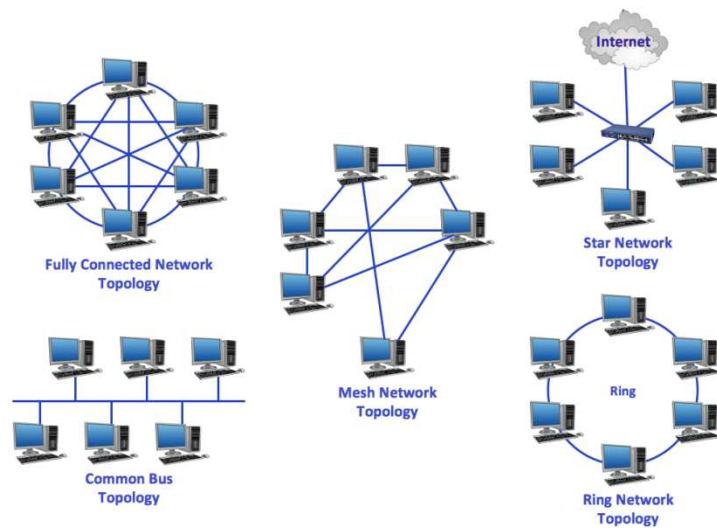
Po prostoru na kome se prostiru mogu biti:

UDP prijem i slanje poruka

- Personal Area Network (PAN)
- Local Area Network (LAN)
- Global Network (Internet)
- Wide Area Network (WAN)

Po arhitekturi (funkcionalnom odnosu članova) mogu biti:

- Host-based
- Client-Server
- Peer-to-peer



Slika 2. Topologije mreža

3 Modeli slojevite računarske komunikacije

3.1 OSI referentni model

OSI (Open Systems Interconnection) model je apstraktni opis dizajna protokola računarskih mreža, predstavljen u obliku sedam slojeva [3]. Razvijen je 1984. godine od strane Međunarodne organizacije za standarde (*International Organization for Standardization*). [4] Sve današnje mreže su bazirane na *OSI* modelu. *OSI* model definiše 7 slojeva (Tabela 1):

UDP prijem i slanje poruka

Tabela 1. OSI referentni model

Slojevi	Jedinica	Protokoli
<u>Aplikacija</u> Mrežni procesi vezani za aplikaciju	Podatak	HTTP , FTP , Telnet , DNS , DHCP , POP/SMT
<u>Prezentacija</u> Enkripcija i kodiranje podataka	Podatak	
<u>Sesija</u> Uspostavljanje sesije krajnjih korisnika	Podatak	NetBIOS, PAP , CHAP , SSH
<u>Transport</u> Veza, pouzdanost, transport	Segment Datagram	TCP , UDP
<u>Mreža</u> Logičko adresiranje i rutiranje	Paket	IP , ICMP , ARP , RARP
<u>Sloj veze</u> Fizičko adresiranje, pristup medijumu	Frejm (Okvir)	PPP , HDLC , Frame Relay
<u>Fizički sloj</u> Transmisija signala	Bit	Token Ring IEEE 802.11

- Fizički Sloj - Fizički sloj je zadužen za prenos bitova (nula i jedinica) putem komunikacionog kanala. Ovaj sloj definiše pravila po kojima se bitove prenose, koji električni napon je potreban, koliko bitova se šalje po sekundi i fizički format korišćenih kablova i konektora;
- Sloj veze - Sloj veze upravlja prenosom putem fizičkog sloja i omogućava prenos oslobođen grešaka na ovom i fizičkom sloju. Zadatak sloja veze jeste da zaštiti slojeve višeg nivoa od grešaka nastalih pri prenosu podataka. Takođe, s obzirom na to da je jedinica prenosa fizičkog sloja bit, sloj veze upravlja i formatom poruke (definiše početak i kraj poruke);
- Mrežni sloj - Zadatak mrežnog sloja jeste određivanje jedne ili više putanja kojima će poruka biti prosleđena od izvorišta do odredišta. Mrežni sloj je zadužen da u svakom čvoru mreže (stanici do odredišta) odredi koji je sledeći računar kome poruka treba biti prosleđena;
- Transportni sloj - Zadatak ovog sloja jeste obrada poruka na krajnjim tačkama – izvorištu i odredištu. Ovaj sloj uspostavlja, održava i prekida virtuelne veze za prenos podataka između izvorišta i odredišta. Transportni sloj je zadužen za nabavku mrežne adrese odredišta, podelu podataka u segmente pogodne za slanje, prilagođavanje brzine prenosa mogućnostima strane sa slabijim performansama, osiguravanje prenosa svih segmenata, eliminisanje dupliranih segmenata i sl. Takođe ovaj sloj može izvršiti i dodatnu kontrolu grešaka pri prenosu (dodatnu u smislu daje ona već izvršena na sloju veze);
- Sloj sesije - Sloje sesije je zadužen za uspostavljanje, održavanje i prekid logičkih sesija između krajnjih tačaka. Svrha sesija jeste definisanje stanja (ili faza) svakog dijaloga radi definisanja validnih akcija u svakom od stanja. Na osnovu toga se vrši

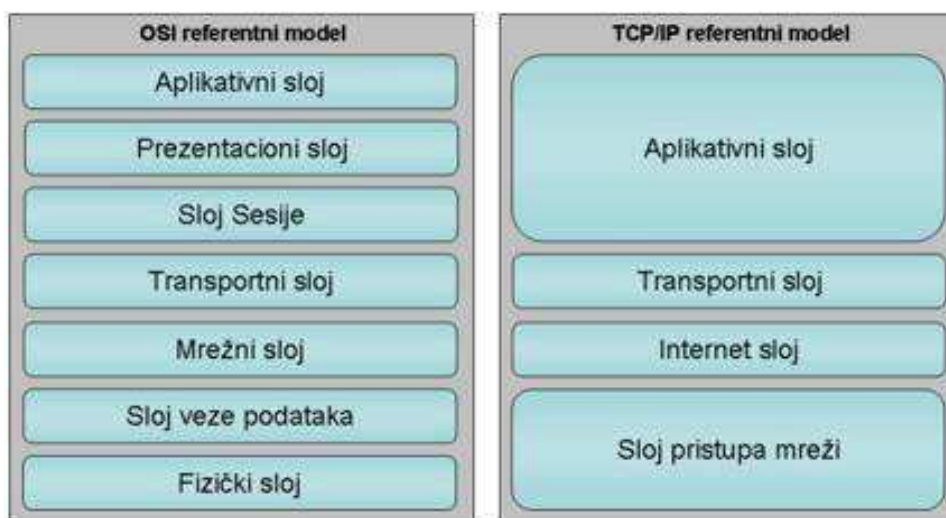
UDP prijem i slanje poruka

upravljanje transportnim slojem i provera podataka dobijenih od njega. Dodatna uloga sesija jeste obračunavanje sesija;

- Sloj prezentacije - Sloj prezentacije formatira podatke za prezentaciju korisniku. Zadatak ovog sloja jeste da uskladi format podataka između učesnika u komunikaciji i sloju aplikacije dostavi ove podatke u formatu koji on zahteva. Na primer, sloj prezentacije može originalne podatke dobijene od sloja aplikacije kompresovati radi efikasnijeg prenosa. Ovakve podatke sloj prezentacije na strani drugog učesnika ne može direktno proslediti sloju aplikacije već je pre toga neophodno izvršiti dekompresiju;
- Sloj aplikacije - Sloj aplikacije predstavlja interfejs mreže ka korisniku. Osnovna uloga ovog sloja je da omogući pristup mreži korisničkim programima.

3.2 TCP/IP model

Ovaj model je razvijen za potrebe Interneta i jednostavniji je od *OSI* modela. Jednostavnost ovog modela ogleda se u apstraktnom gledanju na najviše tri sloja *OSI* modela tako da *TCP/IP* model propisuje samo sloj aplikacije naspram slojeva aplikacije, prezentacije i sesije kod *OSI* modela (Slika 3). Takođe, fizički sloj i sloj veze podataka su objedinjeni u sloj pristupa mreži (Slika 3). Funkcije slojeva su identične kao kod *OSI* modela. Danas se u većini slučajeva koristi *TCP/IP* model kao referentni. [1]



Slika 3. OSI i TCP/IP modeli

4 UDP protokol

4.1 Uvod

User datagram protocol - UDP tj. protokol korisničkih datagrama, pored TCP predstavlja jedan od najčešće korišćenih transportnih protokola Interneta i lokalnih računarskih mreža. Nasuprot TCP, UDP ne omogućava pouzdan prenos podataka putem ostvarivanja virtuelne veze, kontrole grešaka, kontrole redosleda segmenata i ne prilagođava brzinu slanja podataka prijemnoj moći odredišta. Nepouzdan protokol, bez uspostavljanja direktne veze. Omogućava aplikacijama da šalju kapsulirane *IP* datagrame za koje ne moraju prethodno da uspostavljaju vezu. Nedostatak ovih funkcionalnosti čini UDP jednostavnijim protokolom, ali i ne garantuje pouzdan prenos podataka. [3]

TCP i UDP protokoli pripadaju Transportnom sloju u OSI i TCP/IP referentnim modelima.

Prilikom UDP konekcije, klijent dobija jedinstven port baziran na programu koji je započeo konekciju. UDP nije ograničen na 1-1 interakciju već je moguće 1-više interakcija. Sa određenim tehnikama moguće je čak i ostvariti više-više interakcija.

4.2 Zaglavlje UDP datagrama

Jedinica za prenos podataka UDP protokola je datagram (Slika 4). Struktura UDP datagrama je znatno jednostavnija od strukture TCP segmenata, jer je izostavljena većina kontrolnih informacija. [1]

UDP datagram	
16-bit source port number	16-bit destination port number
16-bit length	16-bit UDP checksum
Data	

Slika 4. Struktura datagrama

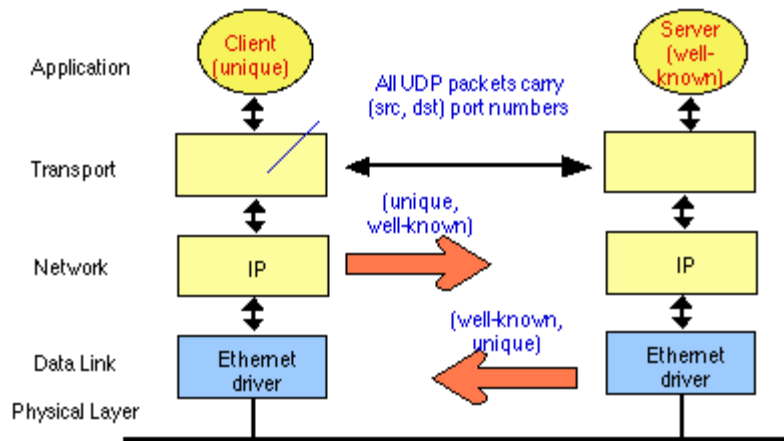
UDP zaglavlje se sastoji od 4 polja od kojih je svaki dužine 2 bajta [5]:

- **Source Port** - izvorni port tj. server
- **Destination Port** - odredišni port tj. klijent
- **UDP length** - broj bitova u UDP poruci
- **UDP Checksum** - proverava validnosti da verifikuje da podaci nisu oštećeni pre nego što su stigli do kraja. Koristi se standardni internet checksum algoritam.

4.3 UDP portovi

Najčešće, klijenti podešavaju port da bude jedinstven broj koji sami izaberu - baziran na programu koji je počeo konekciju. Pošto je ovaj broj vraćen od strane servera u odgovorima, ovo omogućava pošaljioaca kojem „razgovoru“ se prosleđuju poruke.

Serverski proces odsluškuje za UDP pakete primljene sa poznatih brojeva portova i prenosi lokalnom UDP lejeru da šalje pakete kod kojih se podudaraju krajnji port sa onim koji je odabrao server. Pritom odlučuje od kojeg klijenta stižu paketi pregledom IP adrese i jedinstvenog UDP porta.



Slika 5. Razmena poruka

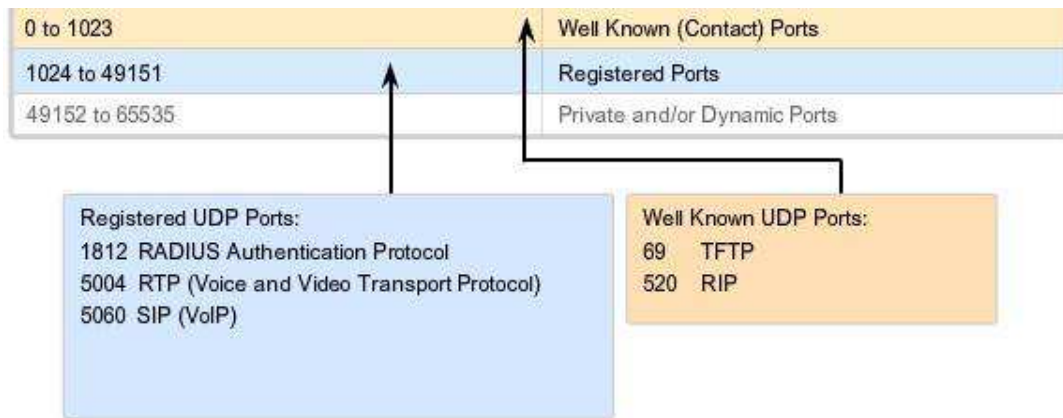
Da bi se razlikovali datagrami za svaku aplikaciju, *UDP* zaglavlja sadrže polja koja mogu jedinstveno identifikovati ove aplikacije. Ovaj jedinstveni identifikator se naziva broj porta. Port je zapravo 16-to bitni ceo broj koji se pridružuje svakoj aplikaciji koja komunicira preko mreže. U zaglavlju svakog datagrama nalazi se broj porta izvorišta i odredišta. Serverski procesi imaju statičke brojeve porta, klijentski port brojevi se dinamički određuju za svaku konverzaciju. Telo koje propisuje standarde vezane za adresiranje portova je *IANA*. Postoje različite vrste brojeva porta: *well known*, registrovani portovi i dinamički portovi.

Well Known (dobro poznati) port-ovi od 0 do 1023 – ovi brojevi su rezervisani za servise i aplikacije. Najčešće se koriste za aplikacije kao što su: *HTTP*, *POP3/SMTP* i *Telnet*.

Registrovani portovi (od 1024 do 49151) – ovi brojevi portova su dodeljeni korisničkim procesima ili aplikacijama. Ovi procesi su primarno individualne aplikacije koje je korisnik izabrao da instalira radije nego uobičajene aplikacije koje imaju *well known* portove.

Dinamički ili privatni portovi od 49152 do 65535 – takođe poznati i kao „*ephemeral ports*“. Ovi brojevi porta se dinamički dodeljuju korisničkim procesima ili aplikacijama koje zahtevaju konekciju.

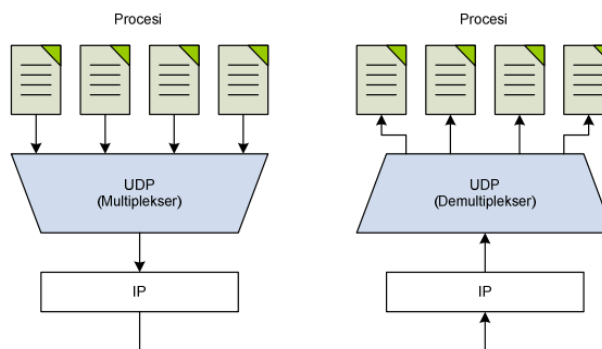
UDP prijem i slanje poruka



Slika 6. Well known i registrovani UDP portovi

4.4 Multipleksiranje i demultipleksiranje

Na hostu na kome se izvršava TCP/IP postoji samo jedan UDP. Međutim, može postojati nekoliko aplikacionih procesa koji u isto vreme žele da koriste usluge UDP-a. Da bi se razrešile ovakve situacije, na nivou UDP-a koristi se koncept multipleksiranja i demultipleksiranja.



Slika 7. Multipleksiranje i demultipleksiranje

Multipleksiranje se koristi na strani pošiljaoca kada više od jednog aplikacionog procesa želi da pošalje korisničke datagrame (odgovara relaciji više-prema-jedan: više aplikacionih procesa, jedan UDP). UDP prihvata poruke od različitih procesa (koji se identifikuju svojim brojevima porta), svakoj poruci pridodaje zaglavlje i kreirani korisnički datagram predaje IP-u.

Demultipleksiranje se koristi na strani prijemnika kada postoji više od jednog aplikacionog procesa koji očekuju korisničke datagrame (odgovara relaciji jedan-prema-više: jedan UDP, više aplikacionih procesa). UDP prima korisnički datagram od IP-a. Nakon provjere grešaka i odstranjivanja zaglavlja, UDP isporučuje poruku odgovarajućem aplikacionom procesu shodno broju porta

4.5 Primena UDP

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

UDP prijem i slanje poruka

prenosu, klijent jednostavno čeka neko vreme i pokušava ponovo. Primer aplikacije koja koristi UDP na ovaj način je već ranije pomenut 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 62 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. 500 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, 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 (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. [6]

4.6 Prednosti UDP

Nekada je bolje za prenos poruka koristiti *UDP* protokol umesto *TCP* protokola. Neki od tih slučajeva su:

- Prenos podataka aplikacija koje same osiguravaju pouzdani prenos, ili kada aplikacija dopušta manje gubitke;
- Slanje upita jednog računara drugom računaru, uz mogućnost ponavljanja upita ako odgovor ne stigne nakon isteka određenog vremenskog intervala;
- Kada je potrebno poslati manji blok podataka, veličine jednog paketa pa je jednostavnije i brže prenositi samo podatke, bez dodatnih kontrola, a u slučaju pogrešnog prijema podatke poslati ponovo

5 Klijent-server aplikacija u programskom jeziku C#

System.Net.Sockets - namespace za implementaciju Winsock interfejsa za developere koji uključuju pristup mrežama. TcpClient, TcpListener i UdpClient su klase koje sadrže detalje za konekciju internetu [7].

System.Net -namespace koji sadrži jednostavan interfejs za protokole koji se koriste u mrežama danas [7].

U primeru za server UDP kreira se klasa:

```
UdpClient listener = new UdpClient(listenPort);
```

Ovim kreiramo klasu odsluškivač za koju pozivamo metodu:

```
receive_byte_array = listener.Receive(ref groupEP);
```

Pozvana metoda vraća datagram prosleđen od korisnika.

Klijent aplikacija kreira socket:

```
Socket sending_socket = new Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);
```

I pokušava izvršiti konekciju ka serveru za slanje poruka - dodate metode za proveru validnosti samog postupka i try catch blok za hvatanje grešaka:

```
IPAddress send_to_address = IPAddress.Parse("192.168.40.201");
#region comments
// IPEndPoint je klasa koja oznacava prvi i poslednji objekat u komunikaciji
// U njoj su sadrzane IP adresa i port koji se koriste.
#endregion
IPEndPoint sending_end_point = new IPEndPoint(send_to_address, 11000);

Console.WriteLine("Unos teksta za prenos preko UDP.");
Console.WriteLine("Prazna linija za izlaz iz programa");
while (!done)
{
    Console.WriteLine("Tekst za slanje, prazna linija za izlaz");
    string text_to_send = Console.ReadLine();
    if (text_to_send.Length == 0)
    {
        done = true;
    }
}
```

6 Prilog

Klijent aplikacija

```
/*
 * Minimalni primer UDP za slanje podataka
 * Prenosi poruku paketa i prikazuje tekst u konzoli
 * radi sa programom UDP_Minimum_listener
 */
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
class Program
{
    static void Main(string[] args)
    {
        Boolean done = false;
        Boolean exception_thrown = false;
        #region comments
        // Kreiranje socketa. Fundamentalna jedinica za mrezne komunikacije.
        // Internet network: koji se koristi za internet komunikacije
        // Dgram - datagram za emitovanje svima radije nego specificnom odsluskivacu
        // UDP: poruke su formirane kao user datagram protocol.
        #endregion
        Socket sending_socket = new Socket(AddressFamily.InterNetwork,
        SocketType.Dgram,
        ProtocolType.Udp);
        #region comments
        // kreiranje adrese kojoj se zadaje adresa na koju se konektujemo pri slanju
        // podataka
        #endregion
        IPAddress send_to_address = IPAddress.Parse("192.168.40.201");
        #region comments
        // IPEndPoint je klasa koja oznacava prvi i poslednji objekat u komunikaciji
        // U njoj su sadrzane IP adresa i port koji se koriste.
        #endregion
        IPEndPoint sending_end_point = new IPEndPoint(send_to_address, 11000);

        Console.WriteLine("Unos teksta za prenos preko UDP.");
        Console.WriteLine("Prazna linija za izlaz iz programa");
        while (!done)
        {
            Console.WriteLine("Tekst za slanje, prazna linija za izlaz");
            string text_to_send = Console.ReadLine();
            if (text_to_send.Length == 0)
            {
                done = true;
            }
            else
            {
                // socket mora imati tekst za slanje
                // učitava se string koji se pretvara u niz bajtova.
                byte[] send_buffer = Encoding.ASCII.GetBytes(text_to_send);

                // Gde se salje.
                Console.WriteLine("slanje na adresu: {0} port: {1}",
                sending_end_point.Address,
                sending_end_point.Port);
                try
```

UDP prijem i slanje poruka

```
{
    sending_socket.SendTo(send_buffer, sending_end_point);
}
catch (Exception send_exception)
{
    exception_thrown = true;
    Console.WriteLine("Greska {0}", send_exception.Message);
}
if (exception_thrown == false)
{
    Console.WriteLine("Poruka nije poslata na adresu");
}
else
{
    exception_thrown = false;
    Console.WriteLine("Greska ukazuje na to da poruka nije poslata.");
}
}
}
}
```

Server aplikacija

```
/*
 * Svrha ovog programa je primer koriscenja UDP za primanje podataka.
 * Prima poruke sa klijenta i prikazuje ih u prozoru.
 * Radi sa programom UDP_minimum_talker
 * Pokrenu se oba programa i salju se poruke sa klijenta
 * Moguce je vise klijenta
 */
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
public class UDPListener
{
    private const int listenPort = 11000;
    public static int Main()
    {
        bool done = false;
        UdpClient listener = new UdpClient(listenPort);
        IPEndPoint groupEP = new IPEndPoint(IPAddress.Any, listenPort);
        string received_data;
        byte[] receive_byte_array;
        try
        {
            while (!done)
            {
                Console.WriteLine("Cekanje poruke");
                // Prijem poruke
                // Poziva funkciju receive od listenera iz klase UDPListener
                // Prosledjuje odsluskivacu groupEP - endpoint
                // It puts the data from the broadcast message into the byte array
                // Pretvara podatke poruke u niz bajtova recieved_byte_array
                receive_byte_array = listener.Receive(ref groupEP);
                Console.WriteLine("Primljena poruka od {0}", groupEP.ToString());
                received_data = Encoding.ASCII.GetString(receive_byte_array, 0,
                receive_byte_array.Length);
                Console.WriteLine("data follows \n{0}\n\n", received_data);
            }
        }
        catch (Exception e)
        {
            Console.WriteLine(e.ToString());
        }
        listener.Close();
        return 0;
    }
}
```

Literatura

- [1] V. Mladen and J. Aleksandar, *Uvod u Računarske mreže*. Beograd, Srbija: Univerzitet Singidunum, 2008.
- [2] HowToGeek. [Online]. <http://www.howtogeek.com/191139/22-common-network-jargon-terms-explained/>
- [3] V. Mladen and J. Aleksandar, *Računarske mreže*, Prvo izdanje ed. Beograd, Srbija: Univerzitet Singidunum, 2011.
- [4] Wikipedia. Wikipedia. [Online]. http://sh.wikipedia.org/wiki/OSI_model
- [5] G. Fairhurst. (2008, Nov.) [Online]. <http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/udp.html>
- [6] E. f. Niš. Deo materijala za pripremu ispita iz predmeta Računarske mreže i interfejsi. [Online]. <http://es.elfak.ni.ac.rs/rmif/Materijal/TCP-IP.pdf>
- [7] Wikibooks. (2014, Aug.) Wikibooks. [Online]. http://en.wikibooks.org/wiki/Communication_Networks/TCP_and_UDP_Protocols/UDP
- [8] Microsoft. MSDN. [Online]. <https://msdn.microsoft.com/en-us/library/system.net.sockets%28v=vs.110%29.aspx>