УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД
Департман за рачунарство и аутоматику
Одсек за рачунарску технику и рачунарске комуникације

ДИПЛОМСКИ – МАСТЕР РАД

Кандидат: **Предраг Ковач**
Број индекса: **E9477**

Тема рада: **Програмски пакет за анализу мрежног саобраћаја**

Ментор рада: **проф. др Мирослав Поповић**

Нови Сад, април, 2011.

УНИВЕРЗИТЕТ У НОВОМ САДУ ● **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:			
Идентификациони број, **ИБР**:			
Тип документације, **ТД**:		Монографска документација	
Тип записа, **ТЗ**:		Текстуални штампани материјал	
Врста рада, **ВР**:		Дипломски – мастер рад	
Аутор, **АУ**:		Предраг Ковач	
Ментор, **МН**:		проф. др Мирослав Поповић	
Наслов рада, **НР**:		Програмски пакет за анализу мрежног саобраћаја	
Језик публикације, **ЈП**:		Српски / латиница	
Језик извода, **ЈИ**:		Српски	
Земља публиковања, **ЗП**:		Република Србија	
Уже географско подручје, **УГП**:		Војводина	
Година, **ГО**:		2011	
Издавач, **ИЗ**:		Ауторски репринт	
Место и адреса, **МА**:		Нови Сад; трг Доситеја Обрадовића 6	
Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога)		поглавља:6/страна:35/слика:39	
Научна област, **НО**:		Електротехника и рачунарство	
Научна дисциплина, **НД**:		Рачунарска техника	
Предметна одредница/Кључне речи, **ПО**:		анализа мрежног саобраћаја, ТЦП/ИП протоколи, формат записа ПЦАП датотека, Интернет	
УДК			
Чува се, **ЧУ**:		У библиотеци Факултета техничких наука, Нови Сад	
Важна напомена, **ВН**:			
Извод, **ИЗ**:		Овај рад описује начин на који је реализован програмски пакет за анализу мрежног саобраћаја. У теоријским основама представљени су основни појмови везани за структуру *.pcap* датотека и апликационих протокола. Опис рјешења представља начин на који је конципирано и реализовано рјешење. У раду је приказано и тестирање развијеног програмског пакета. У закључку је анализирано шта је постигнуто и изнијет је поглед на правце даљег развоја.	
Датум прихватања теме, **ДП**:			
Датум одбране, **ДО**:			
Чланови комисије, **КО**:	Председник:	др Илија Башичевић	
	Члан:	др Небојша Пјевалица	Потпис ментора
	Члан, ментор:	др Мирослав Поповић	

UNIVERSITY OF NOVI SAD ● **FACULTY OF TECHNICAL SCIENCES**
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Accession number, **ANO**:			
Identification number, **INO**:			
Document type, **DT**:		Monographic publication	
Type of record, **TR**:		Textual printed material	
Contents code, **CC**:		Master Thesis	
Author, **AU**:		Predrag Kovac	
Mentor, **MN**:		PhD Miroslav Popovic	
Title, **TI**:		Software for analysis of network traffic	
Language of text, **LT**:		Serbian	
Language of abstract, **LA**:		Serbian	
Country of publication, **CP**:		Republic of Serbia	
Locality of publication, **LP**:		Vojvodina	
Publication year, **PY**:		2011	
Publisher, **PB**:		Author's reprint	
Publication place, **PP**:		Novi Sad, Dositeja Obradovica sq. 6	
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)		chapters:6/pages:35/pictures:39	
Scientific field, **SF**:		Electrical Engineering	
Scientific discipline, **SD**:		Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, **S/KW**:		network traffic analysis, TCP/IP protocols, PCAP file format, Internet	
UC			
Holding data, **HD**:		The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, **N**:			
Abstract, **AB**:		This paper describes a software package for analysis of network traffic. The first chapter contains theoretical foundations of .pcap files and application protocols. The second chapter describes the software implementation. The testing of the developed software is explained as well. In the conclusion chapter, there is a summary of what has been achieved and possible directions for future developments of software.	
Accepted by the Scientific Board on, **ASB**:			
Defended on, **DE**:			
Defended Board, **DB**:	President:	PhD Ilija Basicevic	
	Member:	PhD Nebojsa Pjevalica	Menthor's sign
	Member, Mentor:	PhD Miroslav Popovic	

SADRŽAJ

SPISAK SLIKA

SKRAĆENICE

FDDI - *Fiber Distributed Data Interconnect*

RFC - *Request for Comments*

IP - *Internet Protocol*

ICMP - *Internet Control Message Protocol*

TCP - *Transmission Control Protocol*

UDP - *User Datagram Protocol*

RTP - *Real-time Transport Protocol*

RTCP - *Real-time Transport Control Protocol*

SIP - *Session Initiation Protocol*

FTP - *File Transfer Protocol*

SMTP - *Simple Mail Transfer Protocol*

SNMP - *Simple Network Management Protocol*

POP3 - *Post Office Protocol*

HTTP - *Hypertext Transfer Protocol*

HTTPS - *Hypertext Transfer Protocol Secure*

DNS - *Domain Name System*

1. Uvod

Potrebno je napisati aplikaciju koja analizira *log* zapis nastao korišćenjem ***WinPCap*** biblioteke. Neophodno je izvršiti analizu i organizovati prikupljene podatke na način koji će omogućiti efikasno stvaranje sledećih statističkih izvještaja:

- Procenat neobrađenih paketa *.pcap* datoteke nastao usled nerealizovanih metoda za obradu protokola na nekom od *ISO-OSI* nivoa ili usled nekog nepredviđenog scenarija u njihovoj obradi.
- Spisak aplikacionih protokola sa informacijama vezanim za broj TCP/UDP tokova i količinu prenesenih podataka u bajtima.
- Informacije o tome koliki procenat mrežnog (IP) saobraćaja čine svaki od aplikacionih i transportnih protokola.
- Informacija o tome koliki procenat transportnog (UDP) saobraćaja čini RTP protokol.
- Statistika o UDP tokovima, minimalno, maksimalno, prosječno trajanje.
- Informacije o paru IP adresa između kojih je bilo najviše, odnosno najmanje UDP tokova.
- Informacije o minimalnom, maksimalnom i prosječnom trajanju RTP tokova.
- Informacija o satu maksimalnog opterećenja za svaki od protokola.
- Informacije o opterećenju po satima za svaki od protokola, organizovanih u izlaznoj datoteci u formi pogodnoj za grafički prikaz u nekom od raspoloživih programskih paketa.
- Informacije o broju parova IP adresa čiji je broj UDP tokova u intervalima, koji se zadaju kao parametar komandne linije.

- Statistika o TCP/UDP paketima: minimalna, maksimalna, prosječna veličina, varijansa broja TCP/UDP paketa, po toku, paru IP adresa, u *.pcap* datoteci.
- Informacije o SIP sesijama.
- Informacije o RTP tokovima.
- Izvještaj o obavještenjima o nepredviđenim situacijama.
- Sve izvještaje prikazati u izlaznim tekstualnim datotekama.

Prilikom realizacije programskog rješenja korišćen je objektni pristup. Aplikacija je napisana u programskom jeziku *C++*, u okruženju *Microsoft Visual Studio 2005*. Mrežni saobraćaj koji se analizira snimljen je uz pomoć *libpcap* biblioteke, odnosno njene Windows verzije *WinPcap* u *.pcap* binarnoj datoteci. Snimljeni saobraćaj se zapisuje u binarnu datoteku sa ekstenzijom *.pcap*.

U programskom paketu za analizu saobraćaja, koristićemo naziv ***PcapAnalyzer***, su realizovane metode za učitavanje *.pcap* binarne datoteke i analizu učitanog sadržaja kako bi se izdvojili podaci neophodni za statističku analizu definisanu na početku tekućeg poglavlja.

Diplomski-Master rad sadrži 6 poglavlja:

- *Uvod*, postavka zadatka i kratak opis realizacije programskog paketa (razvojno okruženje, datoteke čiji zapis predstavlja ulazne podatke...)
- *Teorijske osnove rada*, kratak pregled teorijskih osnova neophodnih za realizaciju zadatka.
- *Opis programskog paketa*, kraći opis modula, njihova namjena i funkcionalnosti.
- *Eksperimenti*, rezultati testiranja i analiza izlaznih izvještaja.
- *Zaključak*, pregled realizovanih funkcionalnosti, prednosti i mane, pravci daljeg razvoja.
- *Literatura*.

2. Teorijske osnove rada

Za razliku od industrijske ere gdje su glavni bili fizički i finansijski resursi, današnje vrijeme, poznato kao informatička era, kao ključni resursi, karakterišu znanje i informacije. Ono što je neophodno učiniti, i što se nameće kao stalna potreba, jeste blagovremena dostupnost znanja i informacija.

Da bi se to postiglo, uvode se računarske mreže koje se povezuju u jednu globalnu mrežu i na taj način i znanje i informacije su dostupni svima. S povećanjem vrsta usluga dostupnih putem računarskih mreža, javlja se potreba i za poboljšanjem karakteristika mreža.

U sklopu poboljšanja karakteristika mreža, kao jedna od veoma važnih aktivnosti se javlja i potreba za snimanjem i analizom mrežnog saobraćaja. Na osnovu zaključaka zasnovanih na analizi mrežnog saobraćaja vrše se poboljšanja karakteristika mreža (zaštita, povećanje brzine prenosa, itd.).

Jedan od načina snimanja mrežnog saobraćaja jeste snimanje saobraćaja uz pomoc *libpcap* biblioteke, odnosno *WinPcap* njene verzije za Windows. Saobraćaj se snima u binarne datoteke sa ekstenzijom *.pcap*.

Za izradu programskog paketa za analizu zapisa saobracaja neophodne su teorijske osnove vezane za poznavanje mehanizama funkcionisanja komunikacionih protokola, formata njihovih paketa, itd.. Takođe je neophodno poznavati format zapisa binarne *.pcap* datoteke.

Sve informacije vezane za protokole koje su neophodne za izradu projektnog zadatka se mogu naći u RFC *(Request for Comments)* dokumentima pod različitim brojevima za različite protokole (npr. *RFC 3261* za SIP protokol, *RFC 3550* za RTP protokol itd.).

2.1 Format zapisa *.pcap* datoteke

Struktura *.pcap* datoteke je prikazana na slici 2.1. Datoteka je u binarnom zapisu i počinje zaglavljem *(Global Header)* sa opštim karakteristikama. Poslije zaglavlja idu zapisi zapisanih paketa.

Slika 2.1 Struktura *.pcap* datoteke, opšti oblik

Struktura zaglavlja *(Global Header)* datoteke je prikazana na slici 2.2.

```
typedef struct pcap_hdr_s {
        guint32 magic_number;   /* magic number */
        guint16 version_major;  /* major version number */
        guint16 version_minor;  /* minor version number */
        gint32  thiszone;       /* GMT to local correction */
        guint32 sigfigs;        /* accuracy of timestamps */
        guint32 snaplen;        /* max length of captured packets, in octets */
        guint32 network;        /* data link type */
} pcap_hdr_t;
```

Slika 2.2 Struktura zaglavlja *.pcap* datoteke

Svaki zapisani paket počinje zaglavljem. Struktura zaglavlja paketa *(Packet Header)* je prikazana na slici 2.3.

```
typedef struct pcaprec_hdr_s {
        guint32 ts_sec;         /* timestamp seconds */
        guint32 ts_usec;        /* timestamp microseconds */
        guint32 incl_len;       /* number of octets of packet saved in file */
        guint32 orig_len;       /* actual length of packet */
} pcaprec_hdr_t;
```

Slika 2.3 Struktura zaglavlja zapisanog paketa

Značenja polja zaglavlja sa prethodnih slika iz odjeljka 2.1 se mogu naći na http://wiki.wireshark.org/Development/LibpcapFileFormat odakle su slike i preuzete.

2.2 Session Initiation Protocol (SIP) protokol

SIP protokol je signalizacioni protokol koji je našao široku primjenu u kontroli multimedijalnih komunikacionih sesija (kao što su video i glasovni pozivi u VoIP telefoniji). Koristi se za uspostavu, prekid i izmjene sesija koje mogu imati jedan ili više medijskih tokova.

To je protokol na aplikacionom nivou napravljen da bude nezavisan od protokola nižeg, transportnog nivoa, tako da može kao transportni protkol koristiti i *Transmission Control Protocol* (TCP), *User Datagram Protocol* (UDP), ili *Stream Control Transmission Protocol* (SCTP).

Prva verzija SIP protokola je nastala 1996. Njegova najnovija verzija je detaljno opisana u dokumentu RFC 3261.

SIP je tekstualni protokol sa sintaksom sličnom HTTP protokolu. Postoje dva različita tipa SIP poruka: zahtjevi i odgovori.

SIP requests

Request name	Description	Defined in
INVITE	Indicates a client is being invited to participate in a call session.	RFC 3261
ACK	Confirms that the client has received a final response to an INVITE request.	RFC 3261
BYE	Terminates a call and can be sent by either the caller or the callee.	RFC 3261
CANCEL	Cancels any pending request.	RFC 3261
OPTIONS	Queries the capabilities of servers.	RFC 3261
REGISTER	Registers the address listed in the To header field with a SIP server.	RFC 3261
PRACK	Provisional acknowledgement.	RFC 3262
SUBSCRIBE	Subscribes for an Event of Notification from the Notifier.	RFC 3265
NOTIFY	Notify the subscriber of a new Event.	RFC 3265
PUBLISH	publishes an event to the Server.	RFC 3903
INFO	Sends mid-session information that does not modify the session state.	RFC 6086
REFER	Asks recipient to issue SIP request (call transfer.)	RFC 3515
MESSAGE	Transports instant messages using SIP.	RFC 3428
UPDATE	Modifies the state of a session without changing the state of the dialog.	RFC 3311

Slika 2.4 SIP zahtjevi i njihovi opisi

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142

(Alice's SDP not shown)
```

Slika 2.5 Izgled SIP poruke, zahtjev

Na slici 2.4 je prikazan spisak SIP zahtjeva i njihov opis, a na slici 2.5 je dat format INVITE zahtjeva za uspostavu poziva. Iza metode INVITE slijedi URI (*Universal Resource Identifier)* strane kojoj se šalje zahtjev za uspostavu veze. Isti URI se nalazi i u ***To*** polju . URI strane koja poziva se nalazi u ***From*** polju. Polje ***CSeq*** sadrži broj i naziv zahtjeva, svi odgovori na ovaj zahtjev će u ovom polju imati istu vrijednost. Takođe polja ***Via, To, From, Call-ID*** će imati iste vrijednosti kao i u INVITE zahtjevu.

SIP odgovori se mogu na osnovu trocifrenih kodova podijeliti u sledeće grupe:

- 1XX Informacioni odgovori (*Informational Responses*), koji govore da je u toku uspostavljanje sesije (183) , uključeno zvono terminalskog aparata (180) itd.
- 2XX Signali potvrde (*Successful Responses)* signali potvrde o prijemu zahtjeva.
- 3XX odgovori vezani za preusmjeravanje (*Redirection Responses).*
- 4XX odgovori vezani za neuspjeh klijentovog zahtjeva (*Client Failure Responses).*
- 5XX odgovori vezani za neuspjeh serverske strane (*Server Failure Responses).*
- 6XX odgovori vezane za greške globalnog otkaza (*Global Failure Responses).*

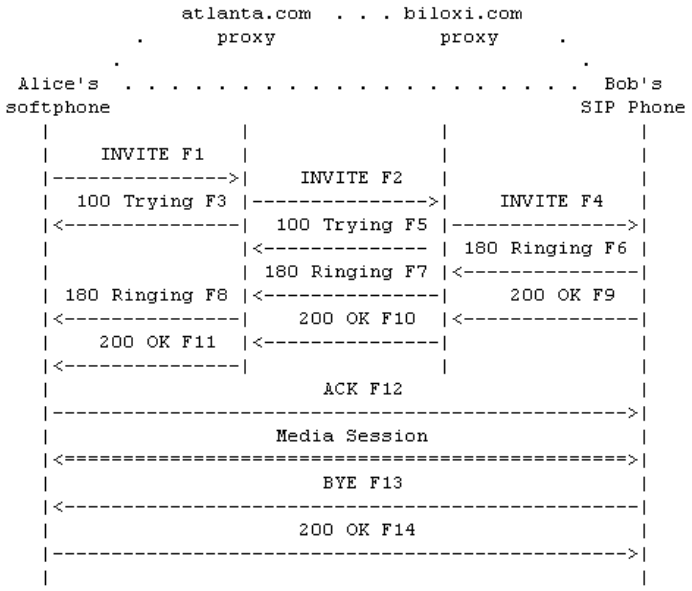
```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
 ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131

(Bob's SDP not shown)
```

Slika 2.6 Izgled SIP poruke, odgovor

Na slici 2.6 je prikazan SIP odgovor. Iza aktuelne verzije SIP protokola slijedi trocifreni kod odgovora iza koga slijedi tekstualni opis odgovora. Ukoliko uporedimo polja poruke ***To, Via, From, Call-ID, Cseq*** na slikama 2.5 i 2.6 vidjećemo da su identični sadržaji tih polja. Na taj način se uz **tag** vrijednosti iz polja ***From, To*** vrši jednistvena identifikacija jedne sesije/poziva.

U sklopu zahtjeva INVITE i odgovora 200 OK, a ponekad i odgovora 183 *Session Progress*, se nalazi i SDP (*Session Description Protocol*) poruka sa informacijama o medijskim tokovima (tip, portovi, adrese za povezivanje, kodeci...).

Na slici 2.7 je prikazan scenario uspostave sesije nakon koje slijedi njen kratak opis.



Slika 2.7 Scenario uspostave Sesije između Alice i Bob-a

Kao što se može na slici 2.7 vidjeti, *Alice* šalje zahtjev INVITE *Bob-u* za uspostavljanje sesije. U sklopu zahtjeva se šalje i SDP poruka sa informacijama o medijskom toku (tip, kodeci, protokol za prenos, port ..). Preko njihovih SIP servera zastupnika zahtjev stiže do *Boba* odakle se po prispeću zahtjeva INVITE šalje odgovor 180 koji signalizira zvonjenje. Onog trenutka kad *Bob* podigne slušalicu odgovor sa kodom 200 se upućuje ka *Alice* sa SDP porukom koja sadrži povratne ifnormacije o medijskom toku između *Bob i Alice*. Kada *Alice* primi odgovor 200 OK salje ACK, i time je završen INVITE/200/ACK *"three-way-handshake"* metod uspostave sesije. Sesija se završava kada jedna od strana pošalje zahtjev BYE, u ovom slučaju to je *Bob*, a druga strana pošalje odgovor 200 OK. To je kraći opis scenarija uspostave sesije. Detaljniji opis se može naći u RFC 3261.

Pored informacija vezanih za uspostavu sesije kao i definisanje portova i IP adresa za RTP tokove, neophodno je i imati informaciju o količini prenesenih podataka SIP protokolom. Taj podatak se nalazi u zaglavlju SIP poruka *Content-Length*.

2.3 Real-time Transport Protocol (RTP) protokol

RTP protokol definiše standardizovan format paketa za prenos glasovnih i video podataka preko Interneta. Intenzivno se koristi u komunikacionim sistemima i sistemima za zabavu gdje su prisutni medijski tokovi.

Koristi se u kombinaciji sa *RTP Control Protocol* (RTCP) protokolom. Dok RTP služi za prenos medijskih sadržaja (npr. glasovnih i video podataka) RTCP nadgleda statistiku prenosa, kvalitet usluge i pomaže sinhronizaciju više medijskih tokova. RTP koristi parne portove, a RTCP neparne. RTCP port je veći za 1 od RTP porta, RTP toka koga nadgleda.

RTP protokol je detaljno opisan u dokumentu RFC 3550. Na slici 2.8 je prikazano zaglavlje RTP paketa. Budući da je od interesa na RTP nivou ustanoviti količinu prenešenih podataka (veličina RTP paketa, bez zaglavlja), na osnovu polja u zaglavlju realizovan je algoritam koji određuje količinu podataka.

RTP packet header

bit offset	0-1	2	3	4-7	8	9-15	16-31
0	Version	P	X	CC	M	PT	Sequence Number
32	Timestamp						
64	SSRC identifier						
96	CSRC identifiers ...						
96+32×CC	Profile-specific extension header ID						Extension header length
128+32×CC	Extension header ...						

Slika 2.8 Format zaglavlja RTP paketa

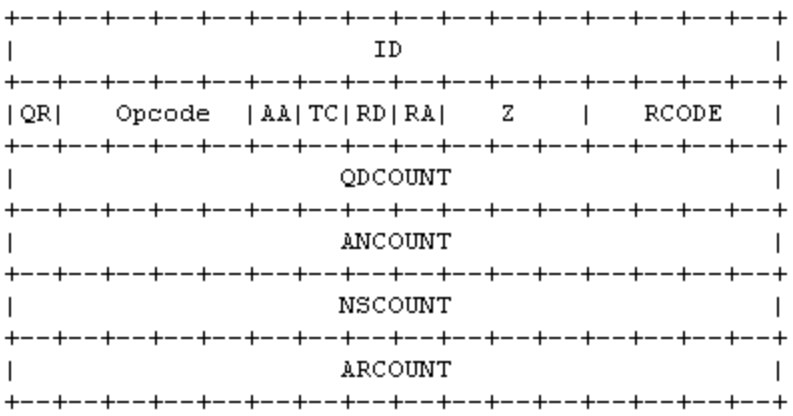
Polja zaglavlja paketa su detaljno opisana u dokumentu RFC 3550.

2.4 Domain Name System (DNS) protokol

Domain Name System (DNS) je hiijerarhijski sistem naziva za računare, servise i druge resurse povezane na Internet ili privatnu mrežu. Omogućava preslikavanje (prevođenje) simboličkog imena u 32-bitnu IP adresu i obratno, koristeći skup geografski distribuiranih programskih servera (primjenom prakse klijent-server).

Ime mašine je ljudima razumljiv identifikator, koji se sastoji od niza ASCII znakova a daje se po imenima geografskih lokacija ili po nazivima dijelova radne organizacije. 32-bitna adresa je namijenjena za upotrebu od strane računara.

Na slici 2.9 je dat format zaglavlja DNS poruke, veličine 12 bajta.



Slika 2.9 Format zaglavlja DNS poruke

Sve informacije vezane za DNS protokol se nalaze u dokumentu RFC 1035. Budući da je potrebno ustanoviti broj prenesenih podataka korišćenjem DNS protokola, na osnovu informacija o zaglavlju moguće je implementirati algoritam koji će obaviti pomenutu operaciju.

2.5 File Transfer Protocol (FTP) protokol

File Transfer Protocol (FTP) protokol se koristi za prenos datoteka između dva čvora u mreži. Zasnovan je na *klijent-server* arhitekturi i koristi dvije odvojene TCP veze za prenos komandi (kontrolna veza, eng. *control connection*) i prenos podataka (veza za podatke, eng. *data connection*).

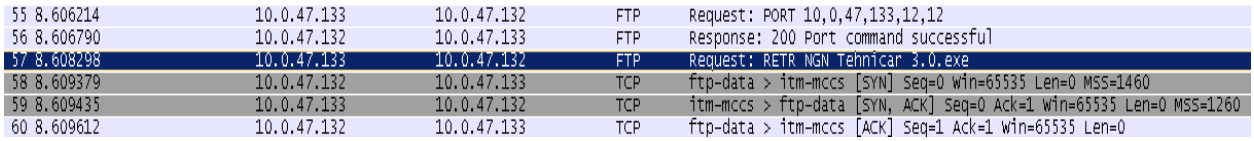
Kontrolnim vezama se upravlja sesijom između klijenta i servera (tj. razmjena komandi, identifikacije, lozinke). FTP zahtjevi i odgovori su u formi *Telnet* komandi (niz karaktera, praznih mjesta, koje završavaju sekvencom CRLF).

Uspostavu kontrolne veze inicira klijent slanjem zahtjeva serveru na port 21. FTP može biti pokrenut u dva režima: *aktivnom* i *pasivnom* .

2.5.1 Aktivni FTP režim (*active mode*)

U aktivnom režimu FTP server se ponaša aktivno. Klijent kontrolnom vezom šalje komandu PORT sa IP adresom i brojem porta koji će se koristiti prilikom prenosa podataka (tj. za uspostavu veze za podatke). Na PORT komandu server odgovara komandom: 200 *Port command succesfull* ukoliko podržava aktivni režim. Nakon slanja komande RETR sa odgovarajućim paramterima, server će inicirati uspostavu veze za podatke i pokrenuti prenos podataka ka klijentu.

Na slici je 2.10 prikazan izgled pomenutih komandi.



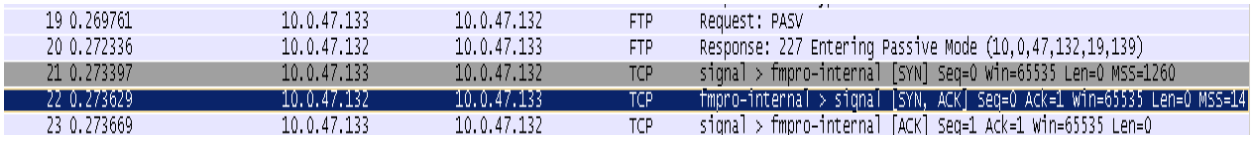
Slika 2.10 Prikaz FTP komandi u aktivnom modu

Kao što je na slici prikazano PORT komanda se šalje sa 6 parametara odvojenih zarezima, gdje su prva 4 parametra IPv4 adrese u tačkastoj decimalnoj notaciji, a dva poslednja parametra se koriste za računanje vrijednosti porta po formuli. Ako parametre označinmo sa p1, ..., p6, formula za računanje vrijednosti porta je: port = p5*256 + p6.

Nakon komande PORT slijedi odgovor da je podržan aktivni FTP režim. Komandom RETR se zahtjeva prenos datoteke nakon čega slijedi uspostava veze za prenos u naredna tri reda. Ono što je bitno naglasiti kada je riječ o aktivnom modu da kod veze za prenos podataka vrijednost porta na serveru je standardna i iznosi 20. Detaljnije informacije o tekućoj verziji FTP protokola se mogu naći u RFC 959.

2.5.2 Pasivni FTP režim(*passive mode*)

U pasivnom režimu FTP server se ponaša pasivno. Klijent šalje komandu PASV kojom se od servera traži da pošalje podatke za vezu kojom će se prenositi podaci. U odgovoru 227 *Entering Passive Mode* server šalje IP adresu i parametre za računanje porta po algoritmu iz poglavlja 2.5.1. U pasivnom modu klijent inicira uspostavu veze za prenos podataka. Na slici 2.11 je prikazan izgled pomenutih komandi.



Slika 2.11 Prikaz FTP komandi u pasivnom modu

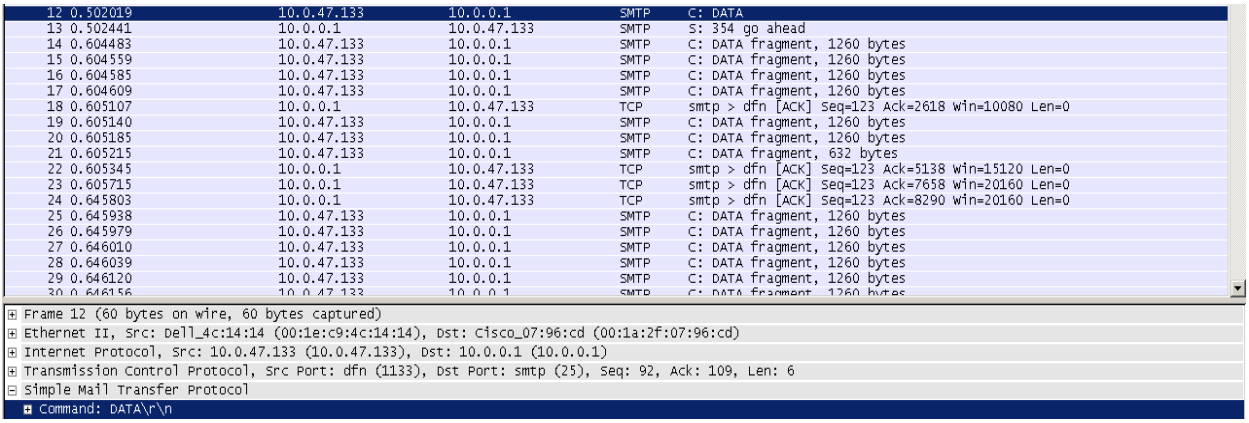
Kao što se može na slici 2.11 vidjeti format zapisa parametara u odgovoru 227 je sličan kao i kod komande PORT u prethodnom poglavlju. Nakon odgovora 227 slijedi uspostava veze za prenos podataka i sa slike se može vidjeti da uspostavu pomenute veze inicira klijent.

Iz priloženog se može zaključiti da podaci koji su bitni za statistiku vezanu za FTP protokol su podaci koji se prenesu vezom za podatke (*eng. data connection*).

2.6 Simple Mail Transfer Protocol (SMTP)protokol

Simple Mail Transfer Protocol (SMTP) protokol je široko rasprostranjen protokol namijenjen za prenos elektronske pošte u računarskim mrežama. Koristi se za transport odlazne pošte (*eng. Outgoing mail transport*) a kao transportni protokol koristi TCP protokol i port 25.

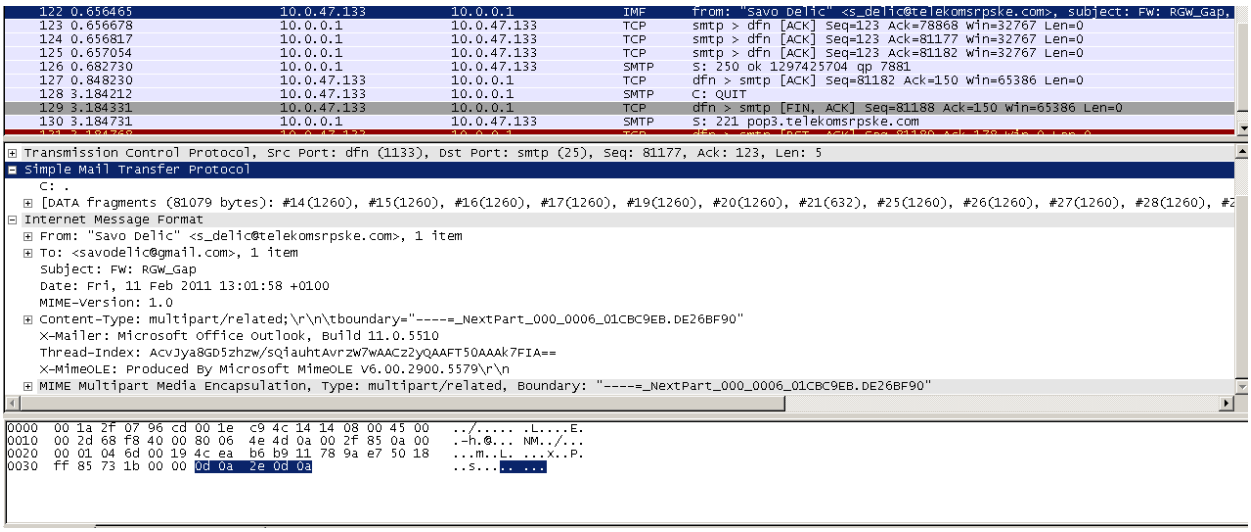
Kao i kod FTP protokola i SMTP se oslanja na Telnet format komandi. Detaljnije informacije o SMTP protokolu i njegovim komandama su date u RFC 5321. Za statistiku o podacima vezanim za SMTP protokol u ovom radu, od interesa je komanda DATA kojom se prenosi sadržaj elektronske pošte. Komandom DATA se prenosi elektronska pošta u TCP segmentima koji se sklapaju na prijemnoj strani i formira se poruka u IMF (*Internet Message Format)* formatu. Za označavanje kraja podataka koji trebaju biti preneseni (tj. elektronske poste) se koristi niz znakova <CR><LF><.><CR><LF>. Na narednim slikama će biti prikazane komande DATA i njen poslednji segment kada dolazi do sklapanja svih preostalih segmenata i formiranja poruke u IMF formatu.



Slika 2.12 *Wireshark* prikaz komande DATA i njenih segmenata

Na slici 2.12 prikazana je DATA komanda kojom se prenosi elektronska pošta u dijelovima (TCP segmentima). Slanje podataka kreće nakon dobijanja potvrdnog 354 *go ahead* odgovora od strane SMTP servera.

Na slici 2.13 je prikazan u *Wireshark-u* poslednji TCP segment koji se odnosi na DATA komandu, sklapanjem sa ostalim TCP segmentima i formiranje jedinstvene elektronske poruke u IMF formatu. Kao što se može primjetiti u poslednjem redu na slici, označeno je 5 okteta u heksadecimalnom zapisu: 0d0a2e0d0a. To su ASCII vrijednosti sekvence <CR><LF><.><CR><LF> koja označava kraj podataka koje treba prenijeti (kraj elektronske poruke).



Slika 2.13 *Wireshark* prikaz poslednjih segmenata i formirane IMF poruke

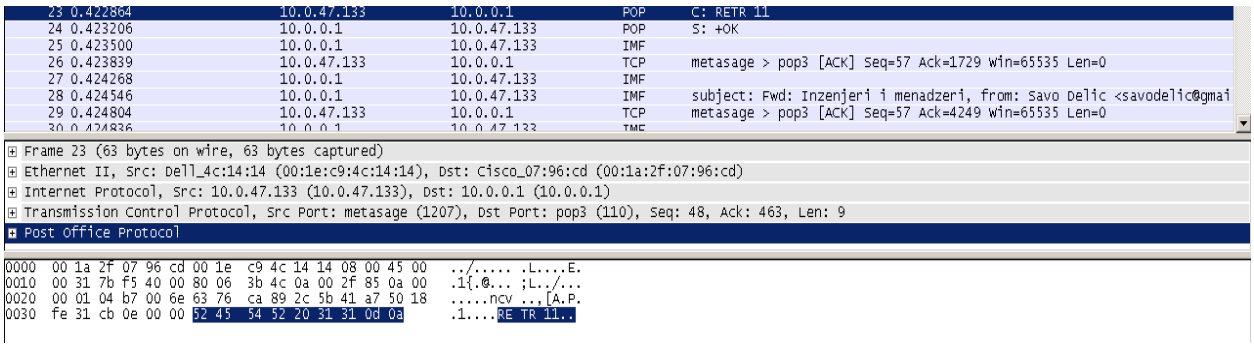
2.7 Post Office Protocol version 3 (POP3) protokol

Post Office Protocol je protokol aplikacionog nivoa koji se korisiti za prihvatanje elektronske pošte sa servera korišćenjem TCP protokola, kao transportnog protokola. POP protokol je razvijan kroz nekoliko verzija, verzija 3 predstavlja standardnu verziju POP protokola pa se stoga koristi oznaka POP3. POP3 server koristi *well-known* port 110.

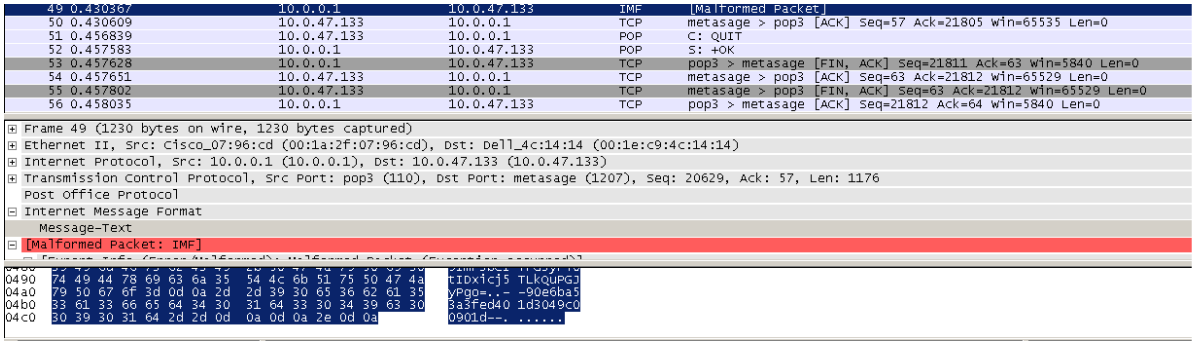
Kao SMTP i FTP protokol, i POP3 protokol je tekstualni protokol koji sadrži komande koje koriste Telnet format komandi.

Detaljnije informacije o POP3 protokolu i njegovim komandama se mogu naći u RFC 1939. Komanda od interesa u ovom radu jeste komanda RETR za prihvatanje elektornske pošte sa POP3 servera. Slično kao kod SMTP protokola, samo ovaj put u suprotnom smjeru, odvija se prenošenje elektronske pošte preko TCP segmenata. Sekvenca <CR><LF><.><CR><LF> označava kraj podataka koje treba prenijeti (tj. kraj poruke, odnosno elektronske pošte). Na sledećim slikama će biti prikazane RETR komanda, njeni segmenti i formirana poruka u IMF formatu po prispeću poslednjeg segmenta sa sekvencom <CR><LF><.><CR><LF>.

Na slici 2.14 je prikazana RETR komanda i početak prenosa elektronske poruke u TCP segmentima.



Slika 2.14 *Wireshark* prikaz komande RETR i njenih početnih segmenata



Slika 2.15 *Wireshark* prikaz poslednjeg segmenta RETR komande

Na slici 2.15 je prikazan poslednji segment RETR komande i formirana elektronska poruka u IMF formatu. Takođe se može vidjeti i poslednjih 5 okteta koji predstavljaju kraj poruke tj. sekvencu <CR><LF><.><CR><LF>.

2.8 HyperText Transport Protocol (HTTP) protokol

HTTP je protokol aplikativnog niova, za prenos HTML stranica od Web servera do Web pretraživača. Budući da omogućuje pouzdanu isporuku podataka, kao protokol nižeg nivoa koristi TCP protokol gdje koristi standardni port 80, odnosno 8080 ukoliko je riječ o serveru zastupniku (eng. *proxy server).*

Zasnovan je na *klijent-server* arhitekturi, odnosno razmjeni zahtjeva (eng. *request*) i odgovora (eng. *response*).

Postoje dvije verzije HTTP vrotokola HTTP 1.0 i HTTP 1.1. Kod verzije 1.0 za svaki zahtjev/odgovor se otvara po jedna TCP veza, dok kod verzije 1.1 se jedna TCP veza može koristiti za razmjenu više zahtjev/odgovor parova.

Zahtjevi, odgovori i druge karakteristike HTTP protokola su detaljno opisane u dokumentu RFC 2616, gdje je opisan HTTP verzija 1.1 koja je najzastupljenija.

```
GET /index.html HTTP/1.1
Host: www.example.com
```

Slika 2.16 Prikaz HTTP zahtjeva (GET metoda)

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8
```

Slika 2.17 Prikaz HTTP odgovora

Na slici 2.16 je prikazan HTTP zahtjev za stranicom *index.html*. Takođe je navedena i verzija HTTP protokola. Ostala polja na slici nisu navedena.

Slika 2.17 prikazuje HTTP odgovor sa poslatom HTML stranicom. Polje koje je od posebnog interesa za statistiku jeste polje *Content-Length* i ono označava veličinu sadržaja u oktetima (bajtima).

2.9 Protokoli aplikativnog nivoa kod kojih je rađena “gruba” statistika

Hypertext Transfer Protocol Secure (HTTPS), *Simple Network Management Protocol* (SNMP), *Real-Time Transport Control Protocol* (RTCP) i *H.323* familija protokola nisu teorijski razmatrani zato što zbog prirode informacije koje nose u sebi ili nekih ograničavajućih okolnosti (npr. šifrovane informacije vezane za HTTPS) nad njima je obavljena tzv. „gruba“ statistika, gdje su na aplikativnom nivou evidentirani svi paketi transportnog nivoa koji su u sebi sadržali izvorni/odredišni port karakterističan za pomenute protokole:

- Za HTTPS je standardni port 443 za TCP protokol.

- Za RTCP, neparni port za jedan veći od odgovarajućeg RTP protokola.
- Za SNMP, portovi 161 i 162 za UDP protokol.
- Za H.323, portovi 1718 i 1719 za UDP, odnosno 1720, 1721, 1722, 1723 za TCP protokol.

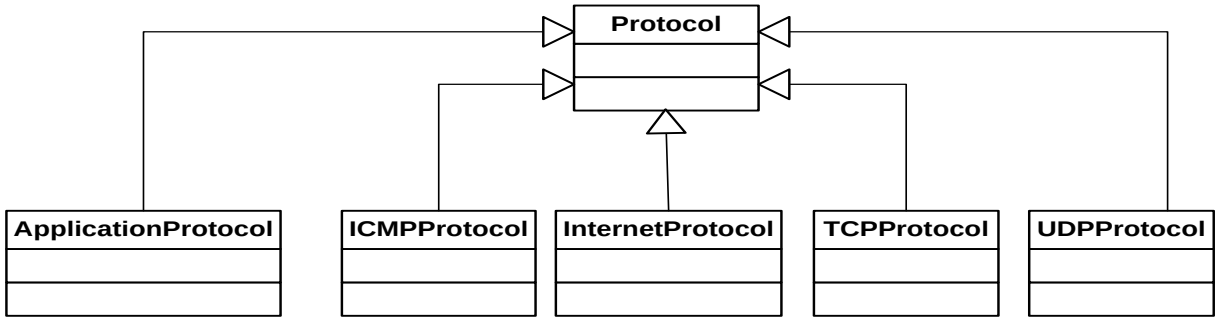
3. Opis programskog paketa

Razvoj programa je realizovan u tri faze:

- Uočavanje programskih cjelina i formiranje UML opisa sistema.
- Izrada programskog modela (pisanje aplikacije).
- Testiranje i analiza dobijenih rezultata.

Naime, neophodno je uočiti programske cjeline i veze između njih, kako bi se mogao formirati UML opis sistema. Na osnovu UML opisa sistema, vrši se programska implementacija, nakon čega se dobija konačna aplikacija. Nakon završetka, aplikacija se testira za različite ulazne datoteke (obično manjeg kapaciteta 100KB) i provjerava se njena robusnost testiranjem velikih *.pcap* datoteka (stotine MB).

U nastavku su dati *UML* dijagrami klasa i model projekta.



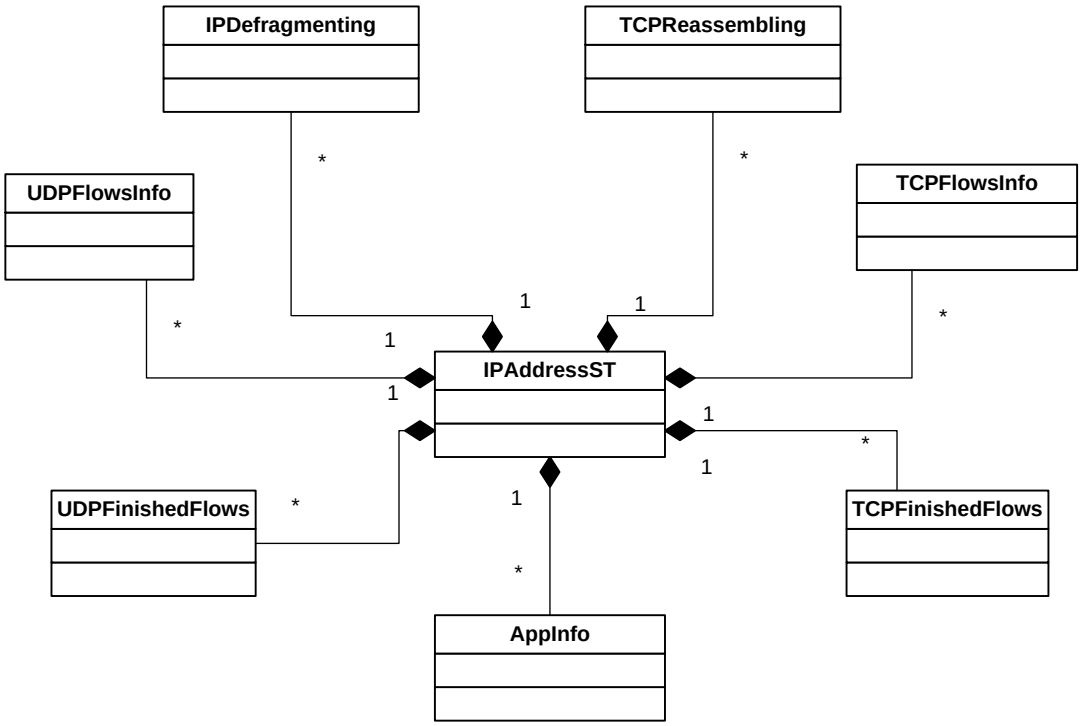
3.1Graf nasleđivanja klasa koje modeluju podakte vezane za protokole

Slijedi opis klasa sa slike 3.1 :

- Klasa ***Protocol*** je klasa-predak svih ostalih klasa koje predstavljaju protokole različitih nivoa. Kao zajednička karakteristika svih protokola koje su od interesa u ovom radu, može se navesti naziv protokola i količina prenešenih podataka po satima. Metode realizovane u okviru ove klase se upravo odnose na manipulaciju

pomenutim podacima (maksimalno opterećenje, opterećenje po satim, broj prenesenih podataka u bajtima...).

- Klasa ***ApplicationProtocol*** predstavlja jedan protokol aplikacionog nivoa sa svojim parametrima i karakteristikama a koje su sadržane u zahtjevima (broj porta, broj TCP/UDP protokola..).
- Klasa ***ICMPProtocol*** predstavlja ICMP protokol (*Internet Control Message Protocol).*
- Klasa ***InternetProtocol*** predstavlja IPv4 protokol *(Internet Protocol version 4).*
- Klasa ***TCPProtocol*** predstavlja TCP protokol *(Transmission Control Protocol).*
- Klasa ***UDPProtocol*** predstavlja UDP protokol *(User Datagram Protocol).*



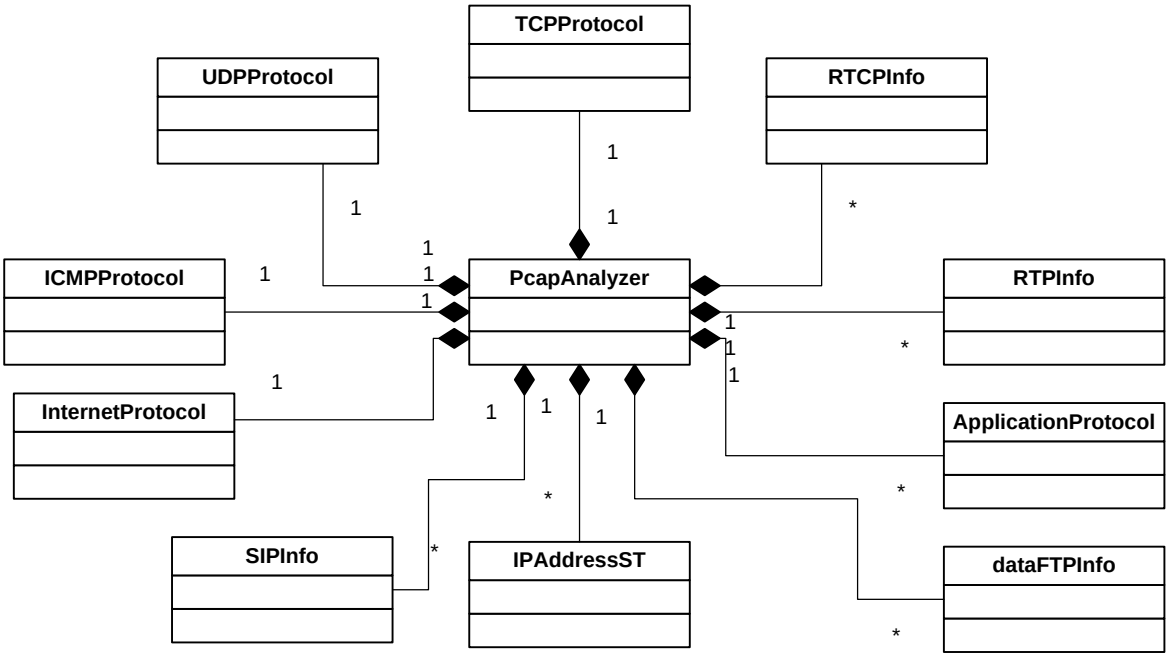
3.2 UML dijagram klasa koje sadrže informacije o paru IP adresa TCP/UDP tokovima koji pripadaju tom paru IP adresa, i algoritmima za defragmentaciju na IP nivou i desegmentaciju na TCP nivou

Slijedi opis klasa sa slike 3.2:

- Klasa ***IPAddressST*** sadrži sve neophodne informacije koje su nam od interesa a koje su karakteristične za par IP adresa(broj TCP/UDP tokova, trajanje tokova i sl.)
- Klasa ***UDPFlowsInfo*** sadrži neophodne parametre za praćenje jednog UDP toka. Kada se tok završi informacije o njemu se prosleđuju u objekat klase ***UDPFinishedFlows.***

- Klasa ***TCPFlowsInfo*** sadrži sve neophodne parametre za praćenje jednog TCP toka. Kada se tok završi informacije o njemu se prosleđuju u objekat klase ***TCPFinishedFlows.***
- Klasa ***AppInfo*** predstavlja klasu o informacijama vezanim za broj TCP/UDP tokova kod detektovanih aplikacionih protokola, sadrži broj porta i broj TCP/UDP tokova. Ovi podaci su aktuelni za tekući par IP adresa.
- Klasa ***TCPReassembling*** sadrži u sebi implementiran algoritam za sklapanje TCP segmenata na prijemnoj strani za jedan TCP tok. Algoritam omogućava sklapanje segmenata u cjelinu i mogućnost pregrupisavanja pristiglih segmenata ukoliko ne stižu po redosledu generisanja na predajnoj strani, tj. vodi se računa da segmenti budu složeni u redosledu koji odgovara rastućoj vrijednosti sekvencijalnih brojeva TCP segmenata. TCP segment je opisan klasom ***TCPSegment***.
- Klasa ***IPDefragmenting*** sadrži u sebi implementiran algoritam za defragmentaciju pristiglih IP fragmenata i slaganje po redosledu nastanka na predajnoj strani. IP fragment je opisan klasom ***IPFragment***.

Na slici 3.3 je dat UML dijagram klasa analizatora *.pcap* datoteka.



3.3 Dijagram klasa pcap Analizatora

Slijedi opis klasa sa slike 3.3 :

- Klasa ***PcapAnalyzer*** je glavna klasa u projektu i ona u sebi sadrži objekte ostalih klasa i metode koje će omogućiti ispunjavanje zahtjeva navedenih u Poglavlju 1. Metode koje su definisane u ovoj klasi omogućavaju otvaranje *.pcap* datoteke,

pravilnu interpretaciju bitskog sadržaja datoteke i razvrstavanje podataka od interesa u oblike pogodne za dobijanje željenih statističkih proračuna. Takođe, u klasi su definisane metode koje generišu izlazne datoteke i u njih smještaju rezultate proračuna.

- Klasa ***RTPInfo*** sadrži informacije o jednom RTP *(Real-time Transport Protocol*) toku. Potreba za ovom klasom nastaje usled činjenice da ne postoji standardni broj porta koji se odnosi na RTP protokol.
- Klasa ***RTCPInfo*** sadrži informacije o jednom RTCP *(Real-Time Transport Control Protocol)* toku. Javlja se kao prateći dio RTP toka na portu uvećanom za jedan u odnosu na tekući RTP tok.
- Klasa ***SIPInfo*** sadrži informacije o SIP sesijama u obrađivanoj *.pcap* datoteci.
- Klasa ***dataFTPInfo*** sadrži informacije o FTP tokovima za podatke.

Ostale klase sa slike 3.3 su objašnjene u tekućem poglavlju.

3.1 Metode za otvaranje i obradu *.pcap* datoteke

Metoda *parseFile* klase ***PcapAnalyzer*** vrši otvaranje binarne *.pcap* datoteke, učitavanje njegnog zaglavlja, analizu informacija iz zaglavlja, a zatim u iterativnom postupku, dok ne dođe do kraja datoteke, prihvata određenu količinu bajtova koji reprezentuju jedan zabilježeni paket. Analizom zaglavlja zabilježenog paketa provjerava da li je u pitanju IP *(Internet Protocol)* paket i ako jeste poziva metodu *parseIP* koja obavlja analizu IP paketa. Ukoliko nije riječ o IP paketu, zabilježeni paket se evidentira kao ne-procesiran kako bi se izračunalo koji procenat *.pcap* datoteke je uspješno obrađen. Takođe, u ovoj metodi se vrši otvaranje izlazne tekstualne datoteke ***report.txt*** u koju se zapisuju rezultati analize paketa po nivoima (mrežni, transportni, aplikacioni). O tome detaljnije u poglavlju o testiranju.

Metoda *parseIP* prihvata niz bajtova koji reprezentuje IP paket i vrši analizu zaglavlja paketa kako bi procjenio da li pripada paket TCP ili UDP protokolu. Takođe, u okviru *parseIP* metode se vrši i provjera da li je došlo do fragmentacije IP paketa.

Metode *parseTCP, parse UDP,* obrađuju TCP, UDP pakete i kom protokolu aplikativnog nivoa pripadaju paketi. Metode za obradu paketa na aplikativnom nivou nose opšti naziv *parseXXXX,* gdje je *XXXX*, skraćenice za određeni aplikacioni protokol (npr. SIP, DNS, RTP...).

3.2 Metode za realizaciju algoritma defragmentacije na IP (mrežnom) nivou (*IP defragmentation*)

Prilikom prihvatanja IP paketa u metodi *parseIP* da bi detektovali da li je riječ o fragmentu ili ne neophodno je u zaglavlju IP paketa provjeriti flag *MF* (eng. *more fragments)*, odnosno vrijednost polja *Fragment Offset.*

Ukoliko je riječ o fragmentu tada će se vrijednosti flaga *MF* i polja *Fragment Offset* pojavljivati u sledećim kombinacijama njihovih vrijednosti:

- MF=1 i Fragment Offset = 0, riječ je o prvom fragmentu,
- MF=1 i Fragment Offset != 0, fragment koji nije ni prvi ni poslednji,
- MF=0 i Fragment Offset != 0, poslednji fragment.

Svakom fragmentisanom paketu se dodjeljuje jedan objekat klase ***IPDefragmenting*** koji je jednoznačno određen izvornom/odredišnom IP adresom, i vrijednošću polja *Identification* i *Protocol.* Metodom *AddFragment* se vrši dodavanje u listu fragmenata sa određenim karakteristikama, a pozicija fragmenta u listi fragmenata se određuje na osnovu vrijednosti polja *Fragment Offset*. Na taj način se postiže pravilan redosled fragmenata u listi čak i u slučaju da fragmenti ne dolaze po hronološkom redosledu nastanka.

Nakon što dođu svi fragmenti i ispravno se slože u listu, tada se u bafer prebacuju metodom *loadFromIPDtoBuff* da bi se metodom *moveFromIPtoTransportLevel* ponovo sklopljen IP paket (tj. dio vezan za podatke IP paketa) proslijedio protokolu višeg nivoa (transportnim protokolima ili ICMP protokolu).

3.3 Metode za evidenciju UDP tokova

U okviru metode *parseUDP* se poziva *refreshIPAdressDataUDP* koja na osnovu podataka dobijenih iz zaglavlja UDP paketa kao i podataka iz IP zaglavlja proslijeđenih metodi *parseUDP*, provjerava da li postoji vec u evidenciji registrovan tekući par IP adresa (objekat klase ***IPAddressST***), odnosno u okviru njega da li postoji registrovan UDP tok sa tekućim parametrima (izvorni, odredišni broj porta, dužina trajanja…) tj. objekat klase ***UDPFlowsInfo***.

Metoda *checkUDPFlowStatus* klase ***IPAddressST*** provjerava postojanje UDP toka sa tekućim parametrima, ukoliko ne postoji dodaje ga u listu aktivnih tokova, ukoliko postoji pozivanjem metode *checkFlowState* klase ***UDPFlowsInfo*** provjerava za tok sa tekućim parametrima, da li je proteklo 90 sekundi između prethodnog i sadašnje pristiglog paketa, odnosno da li je završen tok. Ukoliko je završen tok, vrši se prebacivanje u listu završenih

tokova i inicijalizacija novog toka sa tekućim parametrima. Ukoliko nije tok završen vrši se ažuriranje podataka. Podaci o završenom UDP toku se smještaju u objekat klase ***UDPFinishedFlows***.

3.4 Metode za realizaciju algoritma sklapanja TCP segmenata na prijemnoj strani (*TCP Reassembling*)

Algoritam sklapanja TCP segmenata na prijemnoj strani je sadržan u okviru klase ***TCPReassembling***. Svakom TCP toku prilikom uspostavljanja TCP veze se pridružuje jedan objekat pomenute klase kako bi segmenti koji se budu prenosili tim tokom na prijemnoj strani bili sklopljeni u jedinstvenu cjelinu. Ukoliko segmenti ne pristižu na prijemnu stranu po redosledu njihovog slanja (eng. *Out of order)* algoritam će se pobrinuti da po prispeću poslednjeg segmenta svi budu složeni po željenom redosledu, odnosno da bude niz okteta identičan onom koji je poslat sa aplikativnog nivoa predajne strane.

Formiranje objekata klase ***TCPReassembling*** je direkno povezano za uspostavljanje TCP veze između dva čvora u mreži. Prilikom analize paketa na transportnom nivou, tj. TCP paketa, posmatraju se vrijednosti flagovi SYN, ACK, FIN, tj. postupak uspostavljanja TCP veze kroz trostruku razmjenu paketa (eng. *Three-way handshake)*.

Pozivom metode *addTCPR* određenom toku se dodjeljuje objekat klase ***TCPReassembling*** koji će omogućiti spajanje pristiglih TCP segmenata. Objekat klase ***TCPReassembling*** je određen parom izvorni/odredišni adresa/port i vrijednošću sekvencijalnog broja prvog segmenta sa podacima koji se očekuje.

Svaki objekat klase ***TCPReassembling*** sadrži u sebi listu objekata klase ***TCPSegment*** koji predstavljaju listu TCP segmenata koje treba sklopiti na prijemnoj strani jednog toka. Oni su jednoznačno određeni svojim sekvencijalnim brojevima na osnovu kojih se utvrđuje tačan poredak segmenata.

Kad pristignu svi segmenti i kada se organizuju u pravilan redosled tada se oni prebace u jedinstven bafer i proslijede odgovarajućem protokolu aplikativnog nivoa. Metodom *addSeg*, odnosno *addMailSeg*, u zavisnosti od toga o kom aplikativnom protokolu je riječ, se vrši dodavanje segmenata u listu. Prebacivanje u bafer niza okteta se postiže pozivom metode *loadFromTCPRToBuff*, nakon čega su okteti poredani u nizu kao i na aplikativnom nivou predajne strane.

Nakon prebacivanja okteta iz liste segmenata u bafer za prosleđivanje aplikativnom nivou, lista sa segmentima se briše i podešavaju se parametri za sledeći niz segmenata koji će doći tekućim TCP tokom.

Prilikom raskidanja TCP veze se brišu objekti za realizaciju algoritma sklapanja TCP segmenata na prijemnoj strani.

3.5 Metode za evidenciju TCP tokova

Slično kao i kod UDP tokova, postoje i za TCP tokove klase čiji objekti sadrže informacije o TCP tokovima. To su klase ***TCPFlowsInfo*** i ***TCPFinishedFlows.*** Prilikom analize zaglavlja TCP paketa, posmatraju se flagovi SYN, ACK, FIN i na osnovu njih se vrši registracija, ažuriranje i prebacivanje tokova u listu završenih tokova.

Prilikom uspostavljanja TCP veze, nakon što se realizuje uvodna sekvenca od tri paketa (eng. *three-way hand shake*) pozivom metode *addTCPFlow* dodaje se objekat klase ***TCPFlowsInfo*** za evidenciju o tekućem TCP toku.

Metodom *editTCPFlow* se vrši ažuriranje količine prenesenih podataka tekućim TCP tokom, na taj način sto se količina podataka aktuelnog TCP paketa (eng. *payload*) dodaje na vrijednost količine već prenesenih podataka.

Prilikom raskidanja TCP veze (flag FIN = 1) podaci o aktuelnom toku se prebacuju u listu završenih tokova, kreiranjem objekta klase ***TCPFinishedFlows.***

3.6 Metode za evidenciju RTP/RTCP sesija

RTP *(Real-Time Transport Control Protocol)* protokol nema standardni broj porta preko koga prima/šalje podatke, tako da je neophodno imati evidenciju podataka o RTP/RTCP sesijama. Informacije o RTP/RTCP sesijama se čuvaju u objektima klasa ***RTPInfo*** odnosno ***RTCPInfo.*** Onoga trenutka kada se uspostavi SIP sesija, tada postoje svi neophodni podaci o RTP/RTCP tokovima pa se metodom *addRTPInfo* dodaje u evidenciju informacija o RTP/RTCP tokovima na osnovu izvornih/odredišnih adresa/portova prikupljenih iz SIP poruka.

Napomena: Budući da nije unaprijed bio poznat broj parova IP adresa, broj TCP/UDP tokova, RTP/RTCP sesija,…, za evidenciju podataka o njima korišćene su dinamičke strukture STL *(Standard Template Library)* biblioteke tj. *vektori* i *liste.*

4. Eksperimenti

Prilikom izrade aplikacije, kao potvrdu ispravnosti algoritma interpretacije zapisnih paketa binarne *.pcap* datoteke, korišćen je uvid u prikaz iste u programskom paketu ***Wireshark***. Da bi vizuelno poređenje bilo moguće, u aplikaciji je stvorena izlazna tekstualna datoteka ***report.txt*** gdje su prikazani zapisani paketi u obliku koji je skoro identičan prikazu paketa u ***Wiresharku***, tako da je maksimalno olakšana provjera ispravnosti algoritma za učitavanje i obradu zapisanih paketa .

Na slici 4.1 dat je prikaz zapisanog paketa u datoteci ***report.txt.***

```
 7  ------------------------------------------------
 8  Frame : 1
 9      Arrival time :Mon Jul 04 11:32:20 2005
10      ts_usec= 839312
11      Capture Length: 92
12      Frame Length: 92
13
14      Destination :(255:255:255:255:255:255)
15      Source      :(0:224:237:1:110:189)
16
17      INTERNET PROTOCOL INFO
18          Version: 4
19          Header Length: 20 Bytes
20          Differentiated services field : 0
21          Total Length: 78
22          Identification  = 27020
23          Flags: 0
24          Fragment offset = 0
25          Time To Live: 128
26          Protocol = 17
27          Source: 192.168.1.2
28          Destination : 192.168.1.255
29
30      UDP PROTOCOL INFO
31          Source port: 137
32          Destination port: 137
33          Length = 58
34
35  ------------------------------------------------
```

Slika 4.1 Prikaz zapisanog paketa u datoteci ***report.txt***

Za informacije o TCP, UDP tokovima su napravljene dvije izlazne tekstualne datoteke: ***reportIPAddr_TCP*** za TCP tokove, odnosno ***reportIPAddr_UDP*** za UDP tokove. Na slici 4.2 i 4.3 su prikazi informacija o TCP/UDP tokovima po parovima IP adresa.

```
*************************************************************************
Source Address       : 192.168.1.2
Destination Address  : 192.168.1.1

    Transferred bytes number: 13252
    Min packet size in current IP address pair: 33
    Max packet size in current IP address pair: 300
    Average packet size in current addres pair: 43.9709
    Variance:8017

UDP FLOW INFO:
    Num UDP Flows:103
    ------------------------------------------------------
    Source port        : 2712
    Destination port  : 53
    Duration          : 3
    Number of transffered bytes: 102
    Transferred packets number : 3

    Min packet size    : 34
    Max packet size    : 34
    Average packet size: 34
    Variance           : 0
    ------------------------------------------------------
    Source port        : 2713
    Destination port  : 53
    Duration          : 9
    Number of transffered bytes: 220
    Transferred packets number : 5

    Min packet size    : 44
    Max packet size    : 44
    Average packet size: 44
    Variance           : 0
    ------------------------------------------------------
```

Slika 4.2 Prikaz infomacija o UDP tokovima u datoteci ***reportIPAddr_UDP***

Kao što se može vidjeti na slici 4.2, podaci o jednom toku sadrže izvornu/odredišnu adresu/port, kao i broj prenesenih bajtova po toku, odnosno paru IP adresa i statistiku vezanu za veličinu paketa (najveći, najmanji, prosječan, varijansa).

```
************************************************************
Source Address       : 192.168.192.8
Destination Address  : 192.168.7.57

    Transferred bytes number: 343703
    Min packet size in current IP address pair: 10
    Max packet size in current IP address pair: 1380
    Average packet size in current addres pair: 1009.82
    Variance:54645.3

TCP FLOW INFO:
    Num TCP Flows:16
    ------------------------------------------------------
    Source port        : 8080
    Destination port   : 1099
    Duration           : 0
    Number of transffered bytes: 35922
    Transferred packets number : 31

    Min packet size    : 38
    Max packet size    : 1380
    Average packet size: 1158.77
    Variance           : 203787
    ------------------------------------------------------
    Source port        : 8080
    Destination port   : 1095
    Duration           : 12
    Number of transffered bytes: 3222
    Transferred packets number : 5

    Min packet size    : 438
    Max packet size    : 702
    Average packet size: 644.4
    Variance           : 10662.6
    ------------------------------------------------------
```

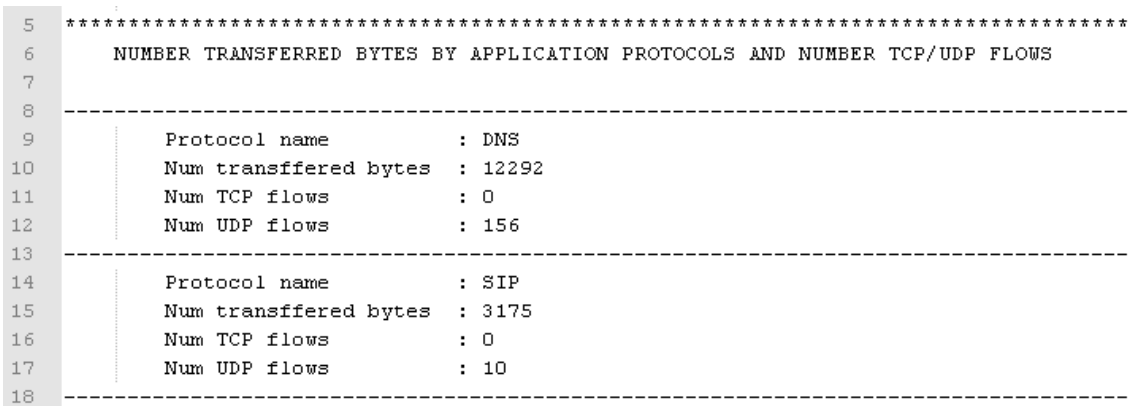
Slika 4.3 Prikaz informacija o TCP tokovima u datoteci ***reportIPAddr_TCP***

U izlaznoj tekstualnoj datoteci ***Show_Results*** se nalaze rezultati sadržani u specifikaciji zahtjeva u Poglavlju 1. Objašnjenja prikaza izlaznih podataka će biti navedena ispod sledećih slika.

```
1  ********************************** SHOW RESULTS ***********************************
2          Captured packets number          : 691
3          Not processed packets number     : 166
4          Not processed packets percentage : 24.0232%
5  ***********************************************************************************
```

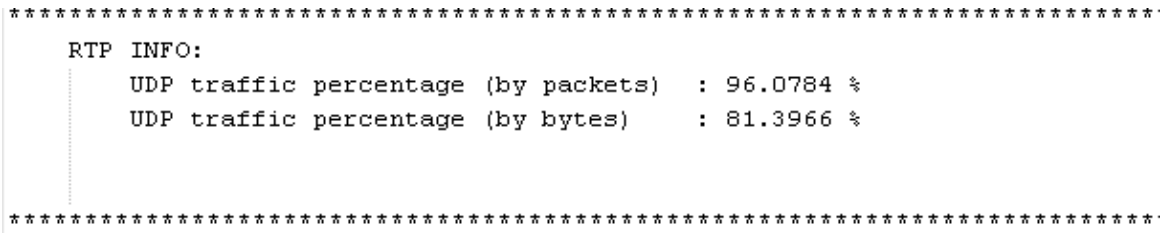
Slika 4.4 Prikaz procenta obrađenosti *.pcap* datoteke

Na slici 4.4 je prikazan izlazni izvještaj o broju paketa koji su zapisani u *.pcap* datoteci, broju neobrađenih paketa, i procentu neobrađenih paketa. Paket se smatra neobrađenim u slučaju njegove nepotpune obrade (imajući u vidu mrežni, transportni, i aplikacioni nivo) tj. smatra se neobrađenim čak i ako je obrađen na nižim nivoima. Veličina procenta neobrađenosti je uvedena zbog toga da se ne bi stvorila lažna predstava o tome da su proračuni dobijeni nad svim paketima iz *.pcap* datoteke.



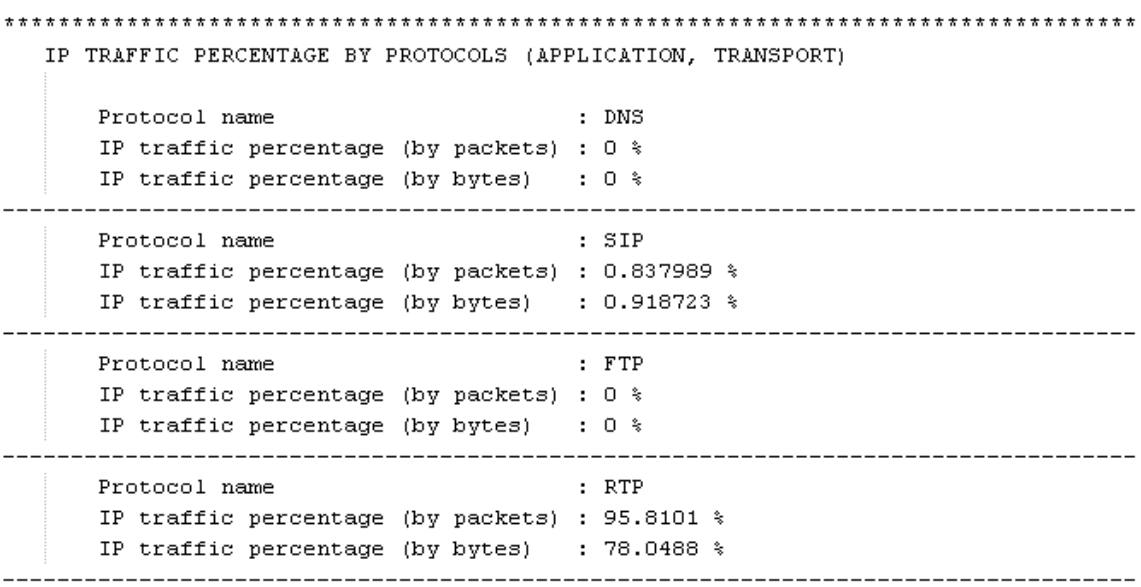
Slika 4.5 Prikaz broja prenesenih bajtova i broja TCP/UDP tokova po ap. Protokolima

Slika 4.5 prikazuje dio datoteke ***Show_Results.txt*** koja sadrži podatke o količini prenesenih bajtova i broju TCP/UDP tokova po aplikacionim protokolima. Zbog veličine slike ovdje nisu navedeni svi protokoli iz spiska realizovanih protokola nego samo SIP i DNS protokol.



Slika 4.6 Informacija o procentu učešća RTP protokola u UDP saobraćaju

Kao jedan od zahtjeva u specifikaciji u Poglavlju 1 jeste i procenat učešća RTP protokola u UDP saobraćaju, što je i prikazano na slici 4.6.



Slika 4.7 Procenat učešća protokola transportnog i aplikacionog nivoa u IP saobraćaju

Slika 4.7 prikazuje prikaz procenta učešća svih realizovanih protokola transportnog (TCP, UDP) i aplikativnog nivoa u mrežnom IP saobraćaju (po broju paketa i broju prenesenih okteta).

```
73  ********************************************************************************
74      UDP FLOWS INFO (MIN, MAX AND AVERAGE DURATION)
75
76  --------------------------------------------------------------------------------
77      MIN:
78
79          Duration (seconds): 0
80          Transferred bytes : 34
81  --------------------------------------------------------------------------------
82      MAX:
83
84          Duration (seconds): 1528
85          Transferred bytes : 8856
86  --------------------------------------------------------------------------------
87      AVERAGE:
88
89          Duration (seconds): 92.0646
90          Transferred bytes : 1920.06
91
92  ********************************************************************************
```

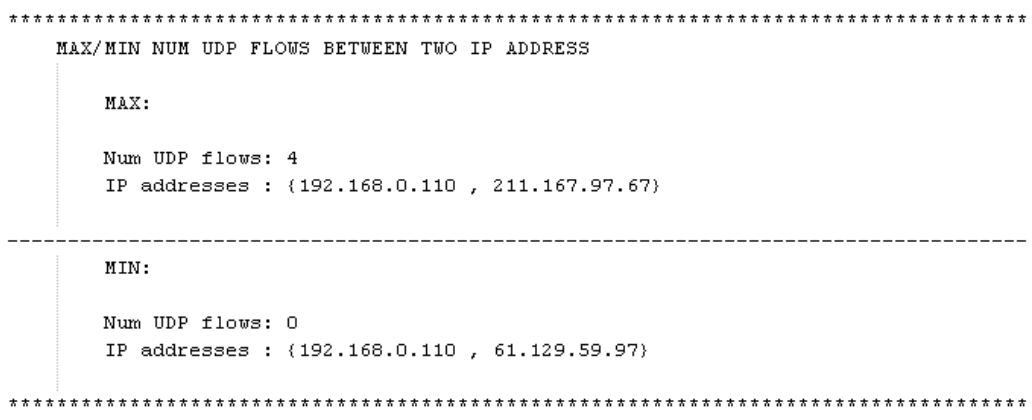
Slika 4.8 Statistika o dužini trajanja UDP tokova

Slika 4.8 predstavlja prikaz statistike o dužini trajanja UDP tokova. Najduži najkraći i prosječan UDP tok. Na slici 4.8 piše da je najkraći tok trajao 0 sekundi, to znači da je trajao kraće od jedne sekunde.

```
127
128 ********************************************************************************
129     THE HOUR OF PROTOCOL MAX LOAD:
130
131 --------------------------------------------------------------------------------
132         Name : DNS
133         Hour : 11
134         Load : 12292
135 --------------------------------------------------------------------------------
136         Name : SIP
137         Hour : 11
138         Load : 3175
139 --------------------------------------------------------------------------------
140         Name : FTP
141         Hour : 0
142         Load : 0
143 --------------------------------------------------------------------------------
144         Name : RTP
145         Hour : 11
146         Load : 1440
147 --------------------------------------------------------------------------------
```

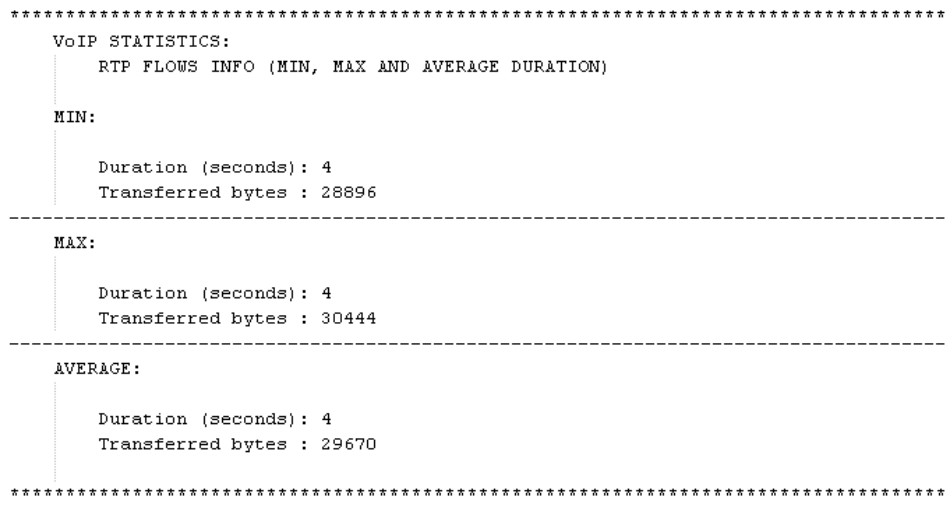
Slika 4.9 Sat maksimalnog opterećenja za sve protokole

Na slici 4.9 je dat prikaz izvještaja o satu maksimalnog opterećenja za sve protokole. Zbog veličine slike, na slici su prikazani rezultati samo za nekoliko protokola. Evidencija o količini prenesenih podataka po satima i statistika vezana za pomenute podatke je prisutna kod svih protokola u aplikaciji.



4.10 Parovi IP adresa između kojih je bilo najviše, odnosno najmanje UDP tokova

Slika 4.10 prikazuje statistiku vezanu za par IP adresa između kojih je bilo najviše, odnosno najmanje UDP tokova.



4.11 Informacije o RTP tokovima (količina prenesenih podataka i trajanje)

Slika 4.11 daje prikaz o trajanju i količini prenesenih podataka za RTP tokove. Radi se o minimalnoj, maksimalnoj i srednjoj vrijednosti.

```
********************************************************************************************
    THE HOUR OF PROTOCOL MAX LOAD:

--------------------------------------------------------------------------------------------
        Name : DNS
        Hour : 0
        Load : 0
--------------------------------------------------------------------------------------------
        Name : SIP
        Hour : 10
        Load : 646
--------------------------------------------------------------------------------------------
        Name : FTP
        Hour : 0
        Load : 0
--------------------------------------------------------------------------------------------
        Name : RTP
        Hour : 10
        Load : 54880
--------------------------------------------------------------------------------------------
        Name : RTCP
        Hour : 0
        Load : 0
--------------------------------------------------------------------------------------------
```

4.12 Sat maksimalnog opterećenja za sve protokole

Na slici je 4.12 je dat prikaz izvještaja o satu maksimalnog opterećenja za sve protokole. Zbog veličine slike, na slici su prikazani rezultati samo za nekoliko protokola. Evidencija o količini prenesenih podataka po satima i statistika vezana za pomenute podatke je prisutna kod svih protokola u aplikaciji.

```
****************************************** UDP FLOW STAT  ******************************************
    Intervals number =2

Report shows num udp flows by ip address in range:
[0, 1)
[1, 2]

-------------------------------------------------------------------------------------------------------
1
10
-------------------------------------------------------------------------------------------------------
```

4.13 Prikaz broja parova IP adresa koje imaju broj UDP tokova u datom intervalu

Izlazna datoteka ***UDPFlowsStat.txt*** sadrži broj parova IP adresa koje u ovom testnom slučaju sadrže broj UDP tokova u intervalima: [0,1), [1,2]. Broj intervala se prosleđuje kao parametar komandne linije. Gornja granica intervala je maksimalni broj UDP tokova u analiziranoj datoteci. Prikaz njenog sadržaja datoteke je ***UDPFlowsStat.txt*** na slici 4.13.

```
1  ****************************************** LOADING REPORT FOR PROTOCOLS(in Bytes) *****
2  ---------------------------------------------------------------------------------------
3  Protocol name:  IP
4  0
5  0
6  0
7  0
8  0
9  0
10 0
11 0
12 0
13 0
14 0
15 75710
16 0
17 0
18 0
19 0
20 0
21 0
22 0
23 0
24 0
25 0
26 0
27 0
28 ---------------------------------------------------------------------------------------
29 Protocol name:  ICMP
```

4.14 Opterećenje po satima za sve protokole

Izlazna datoteka ***Loading_report_for_protocols.txt*** sadrži opterećenja po satima za sve protokole prisutne u *.pcap* datoteci. Podaci su organizovani u datoteci na način koji omogućuje jednostavan unos u neki od programa koji imaju mogućnost prikaza u obliku grafika. Na slici 4.14 je prikazan dio pomenute datoteke.

```
****************************************** SIP SESSIONS INFO ************************
---------------------------------------------------------------------------------------
Source Address                  : 10.127.251.3
Destination Address             : 10.252.64.110

Source Connection Address       : 10.127.251.3
Source Connection Port          : 2234

Destination Connection Address : 10.252.64.153
Destinaiton Connection port     : 34198


Via: SIP/2.0/UDP 10.127.251.3:5060;branch=z9hG4bK949ab6ae8355F7D1
From: "490000" <sip:+38751490000@mtel.ba>;tag=3DE02C58-EE3F63DB
To: <sip:065768044@mtel.ba;user=phone>
CSeq: 1 INVITE
Call-ID: 2a334a44-59762107-76563f1a@10.127.251.3

Media Attribute:

    a= rtpmap:8 PCMA/8000

    a= rtpmap:127 telephone-event/8000

    a= fmtp:127 0-15
---------------------------------------------------------------------------------------
```

4.15 Prikaz informacija o SIP sesijama

Datoteka ***SipSessions.txt*** sadrži izvještaje o SIP sesijama prisutnim u analiziranoj *.pcap* datoteci(slika 4.15).

```
************************************** RTP FLOWS INFO ****************************************************
-----------------------------------------------------------------------------------------------------------
Source Address      : 10.127.251.3
Source Connection Port         : 2234

Destination Address : 10.252.64.153
Destinaiton Connection port    : 34198


Codec:
    rtpmap:8 PCMA/8000
-----------------------------------------------------------------------------------------------------------
Source Address      : 10.252.64.153
Source Connection Port         : 34198

Destination Address : 10.127.251.3
Destinaiton Connection port    : 2234


Codec:
    rtpmap:8 PCMA/8000
-----------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------
```

4.16 Informacije o RTP tokovima

Datoteka ***RTPFlowsInfo.txt*** sadrži informacije o RTP tokovima (izvorne/odredišne adresa/port i kodek). Na slici 4.16 je prikazan njen sadržaj.

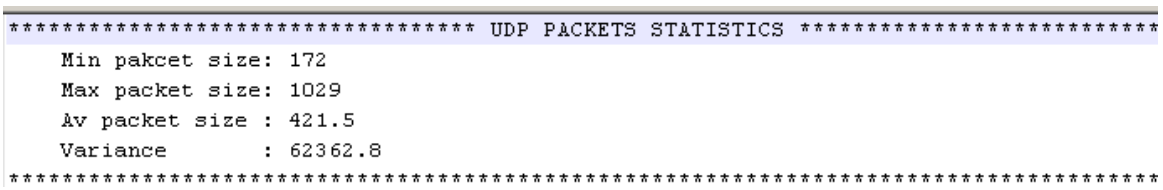
U cilju svoebuhvatnije analize *.pcap* datoteke, generiše se i izlazni izvještaj koji sadrži obavještenja (upozorenja) o nekim nepredviđenim situacijama (ukoliko paket nije procesiran na nekom od ISO–OSI nivoa, nije utvrđeno prisustvo tijela poruke u nekim situacijama i sl.).

```
*******************************   WARNING LIST   ************************************
    Date     : Mon Mar 28 09:23:21 2011
    Filename : SIP-RTP.pcap

-------------------------------------------------------------------------------
NumFrame: 3
    Method Info: PcapAnalyzer::parseSIP(..)
    Warning: There is no message body RESPONSE 183 Session Progress !!!
-------------------------------------------------------------------------------
NumFrame: 5
    Method Info: PcapAnalyzer::parseSIP(..)
    Warning: There is no message body RESPONSE 200 OK !!!
-------------------------------------------------------------------------------
NumFrame: 6
    Method Info: PcapAnalyzer::parseUDP(..)
    Warning: Packet is not proccessed on UDP (transport) level!!! : SIP-RTP.pcap
-------------------------------------------------------------------------------
NumFrame: 7
    Method Info: PcapAnalyzer::parseUDP(..)
    Warning: Packet is not proccessed on UDP (transport) level!!! : SIP-RTP.pcap
-------------------------------------------------------------------------------
```

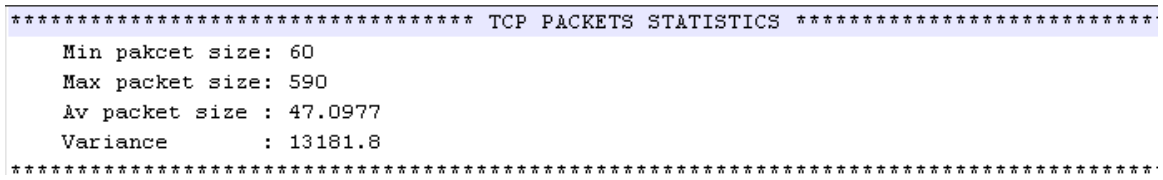
4.17 Spisak upozorenja o nepredviđnim situacijama

Kao što se može na slici 4.17 vidjeti, sva obavještenja su data u istom formatu. Broj zapisanog paketa, informacije o klasi i metodi gdje je došlo do generisanja obavještenja i tekst obavještenja koji opisuje situaciju.

Na slikama 4.18 i 4.19 se nalaze informacije o minimalnoj, maksimalnoj i prosječnoj velicini TCP/UDP paketa, u *.pcap* datoteci. Podaci o UDP paketima se nalaze u datoteci ***UDP_Pkt_Stat.txt***, a o TCP paketima u datoteci ***UDP_Pkt_Stat.txt***.



4.18 Izgled datoteke ***UDP_Pkt_Stat.txt***



4.19 Izgled datoteke ***TCP_Pkt_Stat.txt***

Prilikom razvoja aplikacije za provjeru tačnosti algoritama obrade *.pcap* datoteka korišćene su *.pcap* datoteke veličine do 2 MB tj. malog kapaciteta. Razlog za njihovo korišćenje jeste lakša manipulacija datotekom i brža provjera proračuna u *Wireshark-u.* To je otprilike do 2000 zapisanih paketa.

Sledeći korak u testiranju je obrada *.pcap* datoteka mnogo većeg kapaciteta i praćenje ponašanja aplikacije, tj. da li će doći do nasilnog prekida rada i neke nepredviđene situacije.

U tu svrhu, izvršeno je testiranje gdje su korišćene sledeće *.pcap* datoteke:

- ***unibs20090930.anon.pcap*** veličine ***317MB***
- ***unibs20091001.anon.pcap*** veličine ***236MB***

Oba testna slučaja su uspješno obavljena, u toku analize navedenih *.pcap* datoteka nije došlo do nasilnog prekida rada aplikacije.

5. Zaključak

Realizovana je aplikacija koja analizira *log* zapis nastao korišćenjem ***WinPCap*** biblioteke (*.pcap* datoteke). Kao rezultat njenog izvršavanja dobijaju se sledeći izvještaji:

- Procenat neobrađenih paketa *.pcap* datoteke nastao usled nerealizovanih metoda za obradu protokola na nekom od *ISO-OSI* nivoa ili usled nekog nepredviđenog scenarija u njihovoj obradi.
- Spisak aplikacionih protokola sa informacijama vezanim za broj TCP/UDP tokova i količinu prenesenih podataka u bajtima.
- Informacije o tome koliki procenat mrežnog (IP) saobraćaja čine svaki od aplikacionih i transportnih protokola.
- Informacija o tome koliki procenat transportnog (UDP) saobraćaja čini RTP protokol.
- Statistika o UDP tokovima, minimalno, maksimalno, prosječno trajanje.
- Informacije o paru IP adresa između kojih je bilo najviše, odnosno najmanje UDP tokova.
- Informacije o minimalnom, maksimalnom i prosječnom trajanju RTP tokova.
- Informacija o satu maksimalnog opterećenja za svaki od protokola.
- Informacije o opterećenju po satima za svaki od protokola, organizovanih u izlaznoj datoteci u formi pogodnoj za grafički prikaz u nekom od raspoloživih programskih paketa.
- Informacije o broju parova IP adresa čiji je broj UDP tokova u intervalima, koji se zadaju kao parametar komandne linije.
- Statistika o TCP/UDP paketima: minimalna, maksimalna, prosječna veličina, varijansa broja TCP/UDP paketa, po toku, paru IP adresa, u *.pcap* datoteci.

- Informacije o SIP sesijama.
- Informacije o RTP tokovima.
- Izvještaj o obavještenjima o nepredviđenim situacijama.

Prednosti ovog programskog paketa se ogledaju u mogućnosti obrade velikih *.pcap* datoteka (stotine MB), pošto ih je u programskom paketu *Wireshark* nemoguće obraditi (korišćena verzija 1.2.9). Nedostaci se ogledaju u relativnom malom broju realizovanih protokola i proračuna u poređenju sa standardnim paketima za analizu mrežnog saobraćaja.

Pravci daljeg razvijanja i usavršavanja ove aplikacije mogu ići u smjeru realizacije preostalih nerealizovanih protokola na svim nivoima kako bi se omogućila analiza većeg broja protokola prisutnih u mrežnom saobraćaju. Takođe, neophodno je proširiti algoritme realizacije postojećih protokola ukoliko uvođenje novih proračuna to bude zahtjevalo, kao i otklanjanje eventualnih grešaka ukoliko se pojave u toku eksploatacije.

Neki od konkretnih koraka bi mogli biti:

- Realizacija algoritma provjere kontrolnih suma kod binarnih protokola (IP, TCP, UDP…)
- Mogućnost analize saobraćaja sa mreža koje nisu *Ethernet* (FDDI, Token Ring,..).
- Realizacija algoritma sklapanja tijela poruke u HTTP protokolu, kada se šalje u dijelovima uz prisustvo polja *Transfer-Encoding*.

6. Literatura

[1] Wikipedia, the free encyclopedia, www.wikipedia.org

[2] http://wiki.wireshark.org/Development/LibpcapFileFormat

[3] RFC 3261, http://www.ietf.org/rfc/rfc3261.txt

[4] RFC 3550, http://www.ietf.org/rfc/rfc3550.txt

[5] RFC 1035, http://www.ietf.org/rfc/rfc1035.txt

[6] RFC 959, http://www.ietf.org/rfc/rfc959.txt

[7] RFC 5321, http://www.ietf.org/rfc/rfc5321.txt

[8] RFC 1939, http://www.ietf.org/rfc/rfc1939.txt

[9] RFC 2616, http://www.ietf.org/rfc/rfc2616.txt

[10] RFC 791, http://www.ietf.org/rfc/rfc791.txt

[11] RFC 793, http://www.ietf.org/rfc/rfc793.txt

[12] RFC 768, http://www.ietf.org/rfc/rfc768.txt