VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

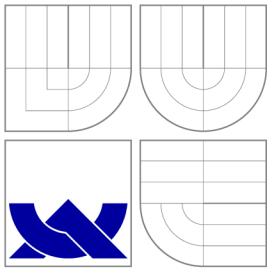
MONITOROVÁNÍ IP TELEFONNÍ SÍTĚ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

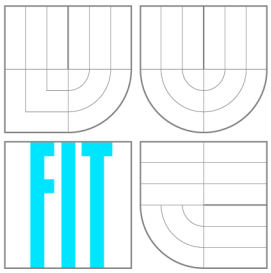
AUTOR PRÁCE
AUTHOR

TOMÁŠ DROZDA

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MONITOROVÁNÍ IP TELEFONNÍ SÍTĚ

MONITORING VOIP NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ DROZDA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR MATOUŠEK, Ph.D.

BRNO 2014

Abstrakt

Tato bakalářská práce se zabývá problematikou monitorování VoIP síti. Hlavním cílem práce bylo navrhnout a následně naprogramovat monitorovací systém pro IP telefonní síte, které používají signalizační protokol SIP, především za účelem účtování. Dalším důležitým cílem bylo prozkoumat možnosti získání geografické polohy komunikujících stanic. Teoretická část práce se věnuje popisu protokolu SIP. Zbytek práce je věnován popisu návrhu a implementace systému. Systém byl implementován jako sada konzolových aplikací, které jsou doplněny o webovou aplikaci, která slouží pro zobrazování získaných dat. Geolokace byla řešena za pomoci služby IPInfoDb, která poskytuje API pro získávání geolokační informací.

Abstract

This thesis deals with the monitoring VoIP networks. The main objective of this work was to design and then to program monitoring systems for IP telephony network using SIP signaling protocol, primarily for billing purposes. Another important objective was to explore the possibility of obtaining geographic location of the communicating stations. The theoretical part of the thesis deals with the description of the SIP protocol. The rest of the thesis is focused to the design and implementation of the system. The system was implemented as a set of console applications that are complemented by a web application that is used to display the obtained data. Geolocation was solved with the help of IPInfoDb services that provides API for obtaining geolocation information.

Klíčová slova

geolokace, VoIP, SIP, monitorovaní VoIP

Keywords

geolocation, VoIP, SIP, VoIP monitoring

Citace

Tomáš Drozda: Monitorování IP telefonní sítě, bakalářská práce, Brno, FIT VUT v Brně, 2014

Monitorování IP telefonní sítě

Prohlášení

Prehlasujem, že túto bakalárku správcu som vypracoval samostatne pod vedením pána Ing. Petra Matouška, P.h.D..

.....
Tomáš Drozda
20. května 2014

Poděkování

Chcel by som sa poďakovať môjmu vedúcemu Ing. Petrovi Matouškovi, P.h.D. za jeho ústretový prístup, ochotu a vecné pripomienky k riešeniu danej problematike. V neposlednom rade chcem poďakovať rodine, priateľke a kamarátom za prejavenú podporu.

Obsah

Kapitola 1

Úvod

Keďže technológia VoIP je čoraz častejšie nasadzovaná v praxy a tak narastá dôležitosť vykonávania monitorovania siete VoIP. Za pomoci monitorovacích nástrojov má sieťový administrátor prehľad o stave jednotlivých zariadení, ako napríklad či je telefón pripojený do siete. Administrátorovi je umožnené získavať informácie o prebiehajúcich telefónnych hovorov (výpis, vyhľadávanie...) či vykonavaných registračných požiadavkov. Nie menej dôležitú rolu v prostredí monitorovania zohrávajú štatistiky. V prípade VoIP sa jedná o štatistiky zobrazujúce vyťaženie telefónnych liniek v jednotlivých hodinách, či najviac komunikujúce zariadenia. Za pomoci týchto informácií je možné vykonávať účtovanie alebo opatrenia pre zlepšenie kvality siete VoIP.

Táto práca sa zaoberá monitorovaním VoIP siete využívajucej signalizačný protokol SIP. Cieľom tejto práce bolo vytvoriť nástroj, ktorý bude sledovať stav prvkov siete VoIP (telefóny, ústredne) a následne získané dáta zaznamenávať. Pri klientských staniciach bude určená ich geografická poloha. Ďalším požiadavkom na vyvíjaný systém je vizualizovanie dát v prehľadnej podobe.

Text práce sa celkovo člení na 5 kapitol. Kapitola 1 obsahuje úvod do samotnej práce.

Kapitola 2 je zameraná na popis signalizačného protokolu SIP, ktorého správy bude vyvíjaný systém analyzovať a na základe ich obsahu vytvárať obraz o aktuálnej situácií na sieti.

Kapitola 3 sa zameriava na návrh a samotnú implementáciu systému. V úvode kapitoly je predstavená aktuálna situácia v oblasti monitorovania sietí VoIP. V ďalšej časti kapitoly je predstavený logický koncept systému, ktorý je tvorený niekoľkými celkami, kde každá z komponent je zodpovedná za vykonávanie špecifických činností. Postupne je popísaný ERD - diagram pre navrhovaný systém. Kapitola sa následne venuje popisu implementácie systému.

Kapitola 4.2 sa zaoberá testovaním implementovaného systému a zhodnoteniu jeho činnosti.

Posledná kapitola 5 obsahuje zhrnutie celej práce no predovšetkým predstavené výsledky práce a smer, ktorým by sa mohol vyvíjať monitorovací systém v budúcnosti.

Kapitola 2

Protokol SIP

SIP(Session initiation protocol) [2] je aplikačný protokol pre inicializáciu, modifikáciu a ukončovanie multimediálnych spojení. Najčastejšie sa s ním stretávame v spojení so službou VoIP. Protokol SIP je možné použiť taktiež pre instant messaging alebo videokonferencie. Prenos správ protokolu je realizovaný jedným z transportných protokolov (UDP alebo TCP). Štandardným portom protokolu SIP je port 5060. Protokol SIP je typu out-of-band a tak sa technológia SIP používá v spojení s ďalšími protokolmi, ktroré sa podieľajú na prenose multimediálnych dát. Pre renos multimedialných dát sa používajú protokoly RTP/RTCP. Správy protokolu SIP sú textovej podoby.

2.1 Formát správ

Pre pochopenie činnosti protokolu SIP je klúčové poznať formát jednotlivých správ protokolu. Ako už bolo spomínané správy protokolu SIP sú textovej podoby, ktoré sú kódované v UTF-8. Formát jednotlivých hlavičiek je veľmi podobný s protokolom HTTP svojou syntaxiou ako aj sémantikou.

Gramatika popisujúca format správ protokolu SIP

```
generic-message = start-line *message-header CRLF [ message-body ]
start-line = Request-Line / Status-Line
Request-Line = Method SP Request-URI SP SIP-Version CRLF
Method = REGISTER | INVITE | ACK | CANCEL | BYE | OPTIONS
Request-URI = sip:user:password@host:port;uri-parameters?headers
SIP-Version = SIP/x.x
Status-Line = SIP-Version SP Status-Code SP Reason-Phrase CRLF
Status-Code = 1xx | 2xx | 3xx | 4xx | 5xx | 6xx
message-header = "header-name"HCOLON header-value *(COMMA header-value)
```

Na základe prvého riadku správy je rozlišované, či sa jedná o požiadavok (Request-Line) alebo odpoveď na požiadavok (Status-Line).

V prípade požiadavku je správa uvedená názvom metódy (REGISTER,INVITE...), za ktorým následuje SIP adresa zariadenia, pre ktoré je požiadavok určený. Riadok je ukončený reťazcom identifikujúcim verziu protokolu SIP (SIP/x.x).

Keď sa jedná o odpoveď na požiadavok tak je správa uvedená reťazcom identifikujúcim verziu protokolu SIP (SIP/x.x) , za ktorým následuje Status-Code, ktorý informuje o stave

porozumenia a následného uspokojenia požiadavku. Telo správy je využívané len v prípade metódy INVITE, kde obsah je tvorený protokolom SDP.

Príklad správy protokolu SIP

```
INVITE sip:test193@example SIP/2.0
CSeq: 1 INVITE
Via: SIP/2.0/UDP 147.229.192.182:5012;branch=z9hG4bK0692a1;rport
User-Agent: XMeeting/0.3.4a
From: "Tomas"<sip:tomas@example>;tag=a41ca1ee
Call-ID: 4819a1ee-cd2e-3b6f2250e1@b04-204b.kn.vutbr.cz
To: <sip:test193@example>
Contact: <sip:tomas@147.229.192.182:5012;transport=udp>
```

2.2 Stavový kód (Status code)

Stavový kód je 3 číslicový kód, ktorý informuje o stave porozumenia a následného uspokojenia požiadavky zo strany zariadenia, ktoré dostalo požiadavok. Číselná hodnota môže byť doplnená o stručný textový popis stavu, avšak tento reťazec nemá štandartizovanú podobu. Stavový kód definuje celkovo 6 tried kódov pričom každá trieda má špecifický význam. Triedy kódov a aj sémantika niektorých kódov (napr. 200 - OK) je zhodná s protokolom HTTP. Význam týchto kódov je popísaný za pomoci tabuľky 2.1.

Kód	Názov	Význam
1xx	**Provisional**	**Prijatý a čaká sa na spracovanie.**
180	Ringing	Telefón na požadovanej stanici zvoní.
2xx	**Success**	**Prijatý a úspešne uspokojený.**
200	OK	Požiadavok prijatý a uspokojený.
3xx	**Redirection**	**Nutnosť vykonať ďalšiu akciu pre uspokojenie požiadavku**
301	Moved Permanently	Užívateľ nie je dostupný na požadovanom Request-URI.
4xx	**Client Error**	**Chyba na strane klienta**
400	Bad Request	Chýbajúca povinná hlavička.
401	Unauthorized	Nutnosť autorizácie užívateľa.
5xx	**Server Error**	**Chyba na strane serveru**
501	Not implemented	Neimplementovaná podpora pre požiadavok.
502	Bad Gateway	Gateway alebo SIP Proxy prijala nesprávnu odpoveď od ďalšieho serveru.
6xx	**Global Failure**	**Žiadný servere nemôže uspokojiť požiadavok**
603	Decline	Volajúca stanica kontaktovaná ale odmietla požiadavok na hovor.

Tabulka 2.1: Popis významu tried stavových kódov doplnený o príklady stavových kódov.

2.3 Spôsob adresovania staníc

Dôležitú úlohu v protokole SIP zohráva adresácia staníc komunikujúcich protokolom. Klientom je pridelená adresa ktorá sa nazýva SIP URI reprezentujúca telefónne číslo.

Formát SIP URI je popísaný nasledujúcou gramatikou: `sip:user@domain:port`. Význam jednotlivých častí je: user - identifikácia užívateľa , domain - doména, ktorej súčasťou je daný užívateľ, port - port, na ktorom stanica príjma požiadavky.

2.4 Formát hlavičiek protokolu

Formát hlavičiek popisuje gramatika: `header-field = field-name: field-value`
Položka `field-name` reprezentuje názov hlavičky a `field-value` hodnotu danej hlavičky.

Pre zamedzenie vytváraniu príliš veľkých paketov, ktoré by boli následne fragmentované. Niektoré hlavičký môžu byť zapísané v krátkej forme. K jednej hlavičke môže náležať 1 - N hodnôt, pričom hodnoty sú oddelené znakom čiarka. Význam jednotlivých hlavičiek sa môže líšiť v závislosti od použitej metódy protokolu. Príkladom takejto hlavičky je hlavička From, ktorá v prípade metódy `INVITE` reprezentuje voljúcu stanicu no v prípade metódy `REGISTER` sa jedná o adresu osoby zodpovednej za registráciu.

Povinné hlavičky

Protokol SIP definuje niekoľko povinných hlavičiek. Znalosť týchto hlavičiek je kľúčová pre pochopenie spôsobu činnosti protokolu a následne vytvoreného monitorovacieho systému. Príklady hodnôt povinných hlavičiek sú ilustrované za pomoci tabuľky 2.2.

From
: Obsahuje logickú adresu volanej stanice v podobe SipUri. Volitelnou súčasťou tejto hlavičky je informácia, ktorá nesie zobrazované meno užívateľa, ktoré bude vidieť ďalší účastnik relácie v prípade príjmu hovoru.

To
: Obsahuje logickú adresu volanej stanice, pričom formát hodnoty je zhodný s formátom hlavičky From.

Call-Id
: Jedná sa o jednoznačný identifikátor všetkých správ v rámci dialógu.

CSeq
: Obsahuje informáciu o poradovom čísle tranzakcie a názve metódy ku, ktorej sa tranzakcia viaže.

Max-Forwards

: Definuje maximálny počet serverov proxy/brán cez ktoré môže SIP paket prejsť, pokedy nie je správa zahodená. Je tým zamedzené nekonečnému putovaniu správy po sieti.

Via
: Obsahuje sadu zariadení cez, ktoré správa putovala smerom od klienta k cieľovej stanici. Obsah tejto hlavičky je používaný pre smerovanie odpovedi smerom ku klientskej stanici, ktorá vytvorila požiadavok.

Názov	Príklady
Call-ID	Call-ID: f81d4faeddasda12191e6bf6@example.com i:f81d4fae-@example.com
From	From: ”maTomas”<sip:tomas@example.com> f: <sip:tomas@example.com>
To	To: ”Tomas”<sip:tomas@example.com> t: <sip:tomas@example.com>
CSeq	CSeq: 1 INVITE
Max-Forwards	Max-Forwards: 7

Tabulka 2.2: Príklady hodnôt hlavičiek

2.5 Architektúra protokolu

Architektúru protokolu SIP môžme definovať za pomoci 2 základných typov prvkov a to klientská stanica označovaná ako UAC a stanic typu server označovaných ako UAS. UAC sú klientské stanice, či už softvérové alebo hardvérové telefóny, ktoré generujú požiadavky na základe činnosti užívateľa. Vytvorené požiadavky sú smerované k staniciam typu UAS, ktoré sa snažia uspokojiť tieto požiadvavky buď priamo nimi alebo ich smerujú na ďašie zariadenia.

Stanice typu UAS poznáme niekoľkých typov

Proxy server
Úlohou proxy serveru je analyzovať správy a následne ich na základe informácií o lokácií cieľovej stanici správy smerovať smerom ku koncovej stanici.

Lokalizačný server
Lokalizačný server zaznamenáva sieťové adresy a porty klientských staníc, jedná sa o databázu, ktorá obsahuje mapovanie SIP URI na logické sieťové adresy a porty. Činnosť tohto typu je často zapuzdrená priamo v registračnom serveru.

Server pre smerovanie
Jedná sa o nasledujúci bod pre smerovanie správy smer cieľovej stanici.

Registračný server
Tento typ serveru príjma požiadavky typu `REGISTER`. Na základe obsahu týchto správ server aktualizuje alebo pridáva záznamy do lokalizačného serveru.

2.6 Typy požiadavkov

Protokol SIP definuje niekoľko metód za pomoci, ktorých je realizovaná komunikácia medzi jednotlivými zariadeniami. Typ požiavku je určený na základe hodnoty prvého slova správy. SIP stanice odpovedajú na tieto požiadavky za pomoci takzvaných status správ.

Podpora spracovávania, niektorých požiadavkov musí byť podporovaná stanicami komunikujúcimi protokolom SIP. Význam týchto typov požiadavkov je ilustrovaný za pomoci tabuľky 2.3.

Názov požiadavku	Použitie
REGISTER	Registrácia IP telefónu ku sieti.
INVITE	Požiadavok pre vytvorenie telefónneho hovoru.
ACK	Potvrdenie zavedeného spojenia.
CANCEL	Zrušenie nezavedeného spojenia.
BYE	Ukončenie telefónneho hovoru
OPTIONS	Získanie informácií o možnostiach prenosu stanice SIP.

Tabulka 2.3: Povinné metódy

2.7 Registrácia klientských staníc

Za jednú z podstatných súčastí protokolu SIP považujeme mechanizmus registrácie klientských staníc komunikujúcich protokolom SIP. Za pomoci tejto činnosti dokáže UAS zistiť parametre (adresáciu) pre zasielanie správ tak aby bolo možné dosiahnuť požadovanú cieľovú stanicu. Na základe týchto informácií je realizované smerovanie požiadavkov. S vykonávaním registrácií úzko súvisí činnosť lokalizačného serveru, ktorý vo svojej databázy uchováva užívateľské mená staníc a údaje pre ich identifikáciu (ip adresy, porty). Ďalšiou veľmi dôležitou funkciou registrácií je zabezpečenie toho aby mohli komunikovať len autorizované stanice, čím sa zamedzí možnosť komunikácie neautorizovaných staníc.

Princíp registrácie klientov

Pre zjednodušenie popisu si najskôr popíšeme spôsob registrácie klientov bez potreby autorizovať sa. Obrázok 2.1 znázorňuje spôsob registrovania klientských staníc. Spôsob akým sa stanica registruje môžme rozdeliť do 4 krokov.

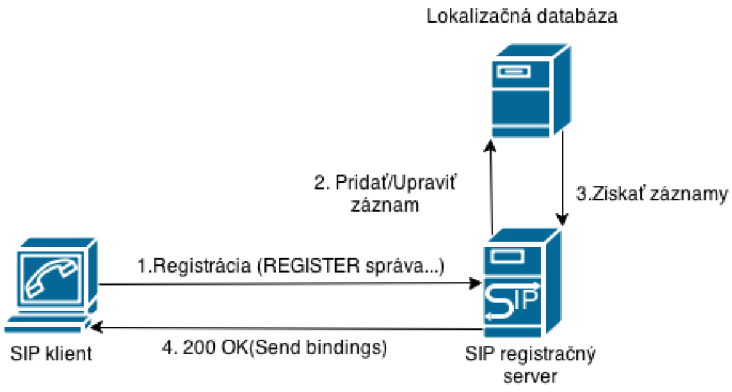
1.krok
Klientská stanica inicializuje žiadosť o registráciu za pomoci požiadavku na metódu `REGISTER`.Na základe obsahu prvého riadku je určená adresa registračného serveru (Registrar), ktorému bude požiadavok doručovaný. V tejto správe klientská stanica zadá informácie o osobe zodpovednej za registráciu hlavičkou(`From`), logickej adrese Address of Record (AOR), ktorá sa chce registrovať hlavičkou(`To`). Všetky registrácie z jedného telefónu by mali obsahovať zhodnu hodnotu hlavičky `Call-ID`. Informácie o IP adrese zariadenia a portu klientskej aplikácie sú obsiahnuté v hlavičke `Contact`.

2.krok
Keď registračný server prijme správu s požiadavkou na registráciu stanice a v správe sú obsiahnute všetký požadované hlavičky tak server aktualizuje alebo pridá záznam v lokalizačnej databáze kde sú zaznamenané informácie o registrovanej stanici, čiže o jeho logickej adrese, užívateľskom mene a porte, na ktorom stanica príjma správy protokolu SIP. K jednému užívateľskému menu môže náležať viacero záznamov v lokalizačnej databáze. Tento počet je možné špecifikovať za pomoci nastavenia serveru.

3.krok
Po úspešnom vykonaní predchádzajúceho kroku sú z lokalizačnej databáze vybrané všetky záznamy o prebiehajúcich registráciach daného užívateľa.

4.krok
Zoznam registrácií je následné zaslaný stanici, ktorá požadovala registráciu v hlavičke

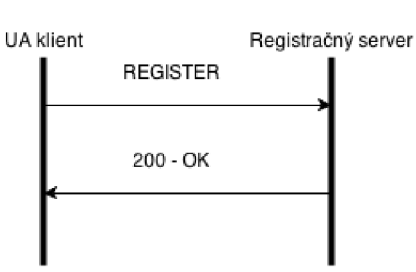
`Contact`. Pri každom zázname v hlavičke kontakt je pripojený parameter `expires`, ktorý informuje klientskú stanicu o zostavajúcom čase počas, ktorého bude stanica považovaná za registrovanú.



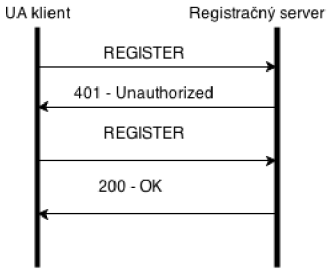
Obrázek 2.1: Schéma registrácie klienta protokolu SIP

Registrácia s využitím autentizácie

V prípade, že UAS vyžaduje autorizáciu tak klientskej stanici odpovedá správou s jedným zo stavových kódov, ktorý informuje o nutnosti vykonať autentizáciu užívateľa. Preto musí klient zaslať nový požiadavok pre registráciu pričom obsahuje vyplnené požadované hlavičky pre autorizáciu. Autentizácia je často realizovaná za pomoci bezstavovej HTTP autentizácie typu realm. Autorizáciu je možné vykonávať pomocou jednej z metód texttINVITE,REGISTER. Obrázok 2.2 ilustruje oba prípady registrácie klientov s využitím autentifikácie tak bez nej.



(a) Sled správ protokolu SIP pre registráciu bez nutnosti autorizácie



(b) Sled správ protokolu SIP s potrebou autentizácie

Obrázek 2.2: Príklady sledov správ protokolu SIP potrebných pre registráciu klienta

Odregistrovanie stanice

UAC sa môže odregistrovať z UAS za pomoci jedného z nasledujúcich 3 spôsobov

1. Spočíva v zaslaní požiadavku typu `REGISTER`, pričom hodnota voliteľnej hlavičky `Expires` je nastavená na hodnotu 0.

2. Druhou možnosťou je vytvorenie požiadavku s metódou `REGISTER` pričom hodnota parametru `expires` pri hodnote v hlavičke `Contact` je nastavená na hodnotu 0.

3. Vytvorenie požiadavku typu `REGISTER`, pričom hodnota hlavičky `Contact` je nastavená na hodnotu „* ”, v tom prípade su zmazané všetky aktuálne aktívne registrácie daného užívateľa.

2.8 Vytváranie a správa telefónných hovorov

Najdôležitejšiou súčasťou protokolu SIP je vytváranie hovorov a ich správa. Pre pochopenie spôsobu vytvárania hovorov za pomoci protokolu SIP je nutné poznať správu metódy `INVITE` a význam jednotlivých hlavičiek, ktoré musia byť v tejto správe obsiahnuté.

Správa metódy INVITE

Tento typ správy je používaný pre vytváranie požiadavkov na uskutočňovanie hovorov medzi 2 stanicami, ktoré implementujú protokol SIP. Správa metódy `INVITE` je jediný typ správy, v ktorom je obsiahnuté telo. Telo správy je tvorené protokolom SDP. Pre jednoznačnú identifikáciu hovoru je využívaný obsah hlavičky `Call-Id`. Hlavička `From` obsahuje telefónne číslo volajúcej stanice a hlavička `To` tel. číslo volanej stanice. Hlavička `Contact` je voliteľná, v prípade jej existencie obsahuje informáciu o logickej adrese a porte na, ktorom vysielaca stanica príjma odpovede na požiadavky, ktoré generovala.

Význam protokolu SDP v spojení s protokolom SIP

Protokol SDP (Session description protocol) [1] je určený k popisu vlastnosti multimediálneho spojenia, protokol sám o sebe neprenáša dáta iba informácie o vlastonstiach prenosu. Protokol zasiela informácie o použitých adresách na ktorých budú vysielané/príjmané multimedialné dáta, portoch pre prenos dát a type multimediálneho kódeku. Okrem týchto položiek je v protokole SDP obsiahnutých niekoľko ďalších informácií, ktoré však pre internetovu telefóniu nie sú tak podstatné.

Protkol má textovú podobu, pričom jednotlivé hlavičky sú obsiahnuté na samostatných riadkoch. Poradie jednotlivých hlavičiek musí byť dodržané pričom niektoré možu byť vynechané.

Formát hlavičiek je nasledovný: `<type>=<value>`, kde type reprezentuje jednoznakový identifikátor typu hlavičky a value uchováva hodnotu danej hlavičky.

Príklad obsahu správy protokolu SDP

```
v=0
o=- 1381267575 1381267575 IN IP4 147.229.192.182
s=Opal SIP Session
c=IN IP4 147.229.192.182
b=AS:1100
t=0 0
m=audio 5048 RTP/AVP 0 8 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
```

```
a=fmtp:101 0-15
```

Medzi ďoležité hlavičky protokolu SDP patria hlavičky `m` a `c`.Obsah týchto hlavičiek je využívaný protokolom SIP pre nastavenie potrebných parametrov pre prenos multimedialných dát.

Hlavička c

Obsahuje informácie o type siete, na ktorej budu dáta prenášané (v súčasnosti je používaný typ IN - internet no je možné, že budú definované ďalšie v budúcnosti), následne je obsiahnutá informácia o type použitého protokolu 3.vrstvy (IPv4,IPv6) a logická adresa používaná pre adresovanie prenášaných dát. Táto hlavička nie je označená ako povinná, v prípade keď jej obsah je zahrnutý v každom definovanom médiu.

Hlavičky m

Obsahujú informácie o popise médií, a tak tu nachádzame informácie o type média(audio/video ...),porte, na ktorom budú dáta vysielané/príjmané , identifikátor prenosového protokolou(napr. RTP) a kódové označenie typu kódeku.

Prenos multimediálnych dát a použitie protokolu RTP

Keďže protokol SIP je typu Out-Of-Band tym pádom je potrebné pre prenos multimediálnych dát použiť iný protokol. Týmto protokolom je RTP (Real-time Transport Protocol).[3]

RTP je binárny protokol ktorý zaručuje prenos multimediálnych dát s real-time charakteristikami po sieti (streaming, interaktívne audio ...). Prenos dát týchto služieb vyžaduje označovanie správ sekvenčnými číslami, časovanie, monitorovanie doručovania, špecifikovanie typu kódeku. Správy protokolu sú zasielané za pomoci transportného protokolu UDP. RTP žiadným spôsobom neobsahuje mechanizmy pre zaručenie QoS. Protokol RTP sa používa v spojení s protokolom RTCP[3], ktorý rieši situácie spojené s nespoľahlivým prenosom realizovaným transportným protokolom UDP (príchod paketov v nesprávnom poradí, monitorovanie doručovania paketov, zmenu použitého kódeku...).

Vytváranie hovoru

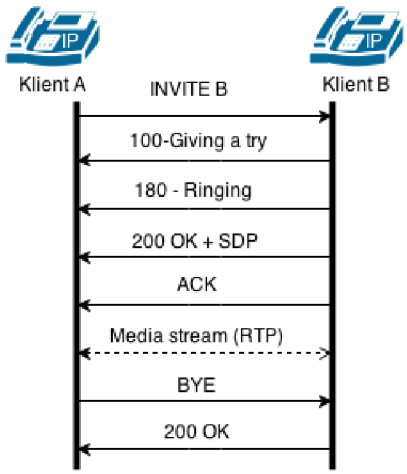
Na začiatku volajúca stanica vygeneruje správu metódy `INVITE` na základe, ktorej bude oboznámená volaná stanica, že jeden z klientov s danou stanicou chce vykonať telefónny hovor.Správa obsahuje vo svojom tele obsah protokolu SDP, kde volajúca stanica poskytne informácie o možnostiach prenosu multimediálnych dát.

Potom ako je požiadavok pre zavedenie hovoru vytvorený je odoslaný na SIP server, ktorý správu smeruje ku volanej stanici. Pre správne smerovanie je využívaný obsah lokalizačnej databázy, keďže obsahuje informácie o logických adresách jednotlivých staníc a záznamov potrebných pre smerovanie do iných logických celkov (domén) protokolu SIP. V prípade chyby, ako napr. nedosiahnutie stanice, je volajúca stanica oboznámena odpoveďou s odpovedajúcim stavovým kódom. Počas procesu spracovávania požiadavku (putovanie paketu po sieti, zvonenie telefónu ...) je volajúca stanica oboznámena o tejto situácií priebežnou odpoveďou so stavovým kódom z triedy 1xx. Na základe reakcie cieľovej stanice (zdvyhnutie alebo odmietnutie hovoru) je volajúca stanica informovaná o tejto skutočnosti správou so špecifickým kódom. V prípade, že sa volaná stanica rozhodla hovor zdvyhnuť tak odpovedá správou so stavovým kódom 200 pričom v tele je obsiahnutý obsah protokolu

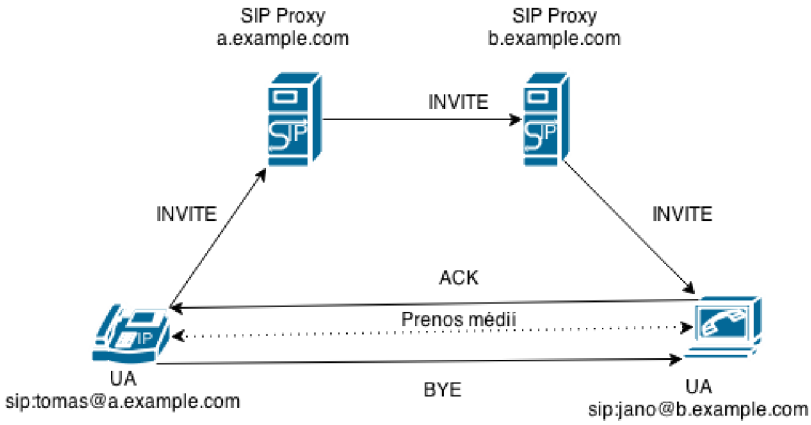
SDP. Keď volajúca stanica príjme túto správu a vyhovujú jej parametre pre prenos dát tak reaguje požiadavkom typu `ACK`, čím je potvrdené zavedenie hovoru v inom prípade je zaslaná správa metódy `CANCEL`, ktorá ruší ešte nezavedený telefónny hovor.

Priebeh vytvárania telefónneho hovoru je ilustrovaný za pomoci Obrázku 2.4. V momente keď si stanice vymenili obsah hlavičky `Contact` nastáva preposielanie správ priamo medzi klientskými stanicami bez nutnosti používania SIP serverov. Táto situácia je názorné zobrazená na obrázku 2.4. Klient so SIP URI *tomas@a.example.com* chce vykonať telefónny hovor so stanicou s telefónným číslom *jano@b.example.com*. Správa najskôr putuje na prvý proxy server, ktorý správu analyzuje následne na základe informácií z lokalizačnej databáze server preposiela správu na následujúci SIP Proxy s doménovým menom b.example.com . Server *b.example.com* obsahuje záznam o lokalite telefónu na tel. čísle *jano@b.example.com* a tak je správa zaslaná ku koncovej stanici. Po zdvyhnutí hovoru si zariadenia vymenia obsah hlavičky `Contact` a tak začína priama komunikácia medzi klientami. V tomto okamihu su telefónne dáta prenášané formou správ protokolu RTP. V poslednom kroku stanica s tel. číslom *tomas@a.example.com* zašle požiadavok typu `BYE` pre ukončenie hovoru stanici *jano@b.example.com* a hovor je následne ukončený.

Sled správ zasielaných pri vytváraní hovoru je zobrazený na Obrázku 2.3.



Obrázek 2.3: Zavedenie hovoru v prostredí protokolu SIP



Obrázek 2.4: Lichobežnikový diagram popisujúci spôsob vytvárania spojenia medzi 2 klientskými stanicami.

2.9 Zhrnutie

Cieľom tejto kapitoly bolo popísať vlastnosti protokolu SIP (formát správ, hlavičky protokolu a jeho architektúru). Zamerali sme sa na vysvetlenie základných vlastností protokolu, ktorých znalosť následné využijeme pri implementácií monitorovacieho systému. Protokol je textovej podoba a má niektoré spoločné vlastnosti s protokolom HTTP. V kapitole následne boli vysvetlené podstatné hlavičky ako napríklad `From, To, Contact, Call-Id` a iné, ktorých znalosť je nutná. Boli popísané metody protokolu predovštekým sa jednalo o metódy `REGISTER` - registrácia účastnikov a `INVITE` - vytváranie hovorov. Popísali sme si spôsoby registrácie klientských staníc a činnosti potrebné pre zavedenie telefónného hovoru. V spojení s vytvorením telefónných hovorov boli spomenuté protokoly SDP a RTP/RTCP a ich význam pre IP telefóniu.

Kapitola 3

Návrh a implementácia monitorovacieho systému

Kapitola sa zaoberá návrhom a samotnou implementáciou navrhovaného monitorovacieho systému. Prvá podkapitola obsahuje popis existujúcich riešení pre monitorovanie IP telefónných sietí. V druhej podkapitole si popíšeme návrh monitorovacieho systému. Definujeme ho ako sadu niekoľkých logických celkov, ktoré následne budú implementované. Popíšeme si databázové schéma a spôsob komunikácie medzi systémom a databázou. Zameriame sa na význam jednotlivých tried, ktoré sa podieľajú na činnosti systému. V závere prejdeme ku princípom na základe, ktorých sú implementované jednotlivé logické bloky.

3.1 Aktuálna situácia v oblasti monitorovania VoIP sietí

V súčasnej dobe sa stretávame s viacerými riešeniami pre monitorovanie VoIP sietí. Existujú riešenia komerčné. ktoré často dosahujú vačšiu kvalitu ale zároveň sa stretávame tiež s bezplatnými open source riešeniami.

Veľmi zaujímavým open source riešením pre monitorovanie VoIP je nástroj VoIPmonitor [1], ktorý umožnuje monitorovanie komunikácie pomocou protokolov SIP, SCCP, RTP/RTCP. Výhodou tohto systému je, že sa zameriava zároveň na určovanie kvality telefónných spojení, zároveň zaznamenáva obsah telefónných hovorov pričom niektoré je možné aj prehrať prijamo vo vizualizačnej aplikácií (táto skutočnosť závysí na použitom kódeku počas telefónneho hovoru). Technickú podporu pre systém je možné si zaplatiť.

Ďalším zaujímavým riešením je komerčné riešenie VoIP & Network Quality Manager[2] od spoločnosti SolarWinds, ktoré poskytuje celú škálu funkcií pre monitorovanie VoIP vrátane určovania kvality telefónných hovorov. Systém využíva technológiu NetFlow. Hlavnou nevýhodou tohto systému je jeho cena.

Ako som postrehol vačšina monitorovacích systémov pre VoIP neposkytuje možnosť geolokácie jednotlivých staníc, zároveň vizualizačné rozhranie je často veľmi komplikované. Jedným z dôvodom pre vývoj vlastného VoIP monitorovacieho systému je vytvorenie otvoreného riešenia, ktoré by mohlo byť v budúcnosti doplnené o podporu ďalších protokol ako napríklad H.323 alebo SCCP. Ďalšiou kľúčovou vlastnosťou vyvíjaného systému bude zaznamenávanie údajov o dostupností klientských staníc. Vyvíjaný monitorovací systém na

[1] Viac informácií na: http://www.voipmonitor.org/

[2] Viac informácií na: http://www.solarwinds.com/voip-network-quality-manager.aspx

rozdiel od spomínaných nebude obsahovať modul pre určovanie kvality telefónných hovorov, čo považujem ako nevýhodu oproti spomínaným riešeniam.

3.2 Návrh systému

Podkapitola sa zaoberá návrhom monitorovacieho systému. Pred samotným návrhom systému bola vykonaná analýza požiadavkov pre monitorovací systém.

Medzi kľúčové požiadavky kladené na systém patria: zaznamenávanie informácií o prebiehajúcich telefónných hovorov (volajúca/volaná stanica, dĺžka hovor,stav hovoru, typy použitých kodekov...), registráciach užívateľov (ip adresa a port klientskej stanice, user agent, geografická poloha...) a získavanie informácií o dostupnosti klientskej stanice protkolom SIP. Nie menej dôležitým požiadavkom na systém bolo vhodné data vizualizovať tak aby boli v prehľadnej podobe a ľahko sa v nich vyhľadávalo.

Podkapitola je rozdelená do 2 celkov, kde v 1. celku je umiestnený popis logickej štruktúry monitorvacieho systému a 2. celok je venovaný popisu konceptuálneho modelu (ERD-diagram) pre uvedený systém.

Schéma systému

Systém bol navrhnutý ako sada 6 logických komponent pričom každá z nich vykonáva určité činnoti, za ktoré je zodpovedná.Schéma systému je reprezentovaná za pomoci diagramu 3.1. Význam jednotlivých navrhnutých blokov je nasledovný:

VoIP Database
: Jedná sa o databázu, ktorá obsahuje údaje o telefónných hovoroch, registráciach klientov, aktivite staníc, geografickej lokalite....

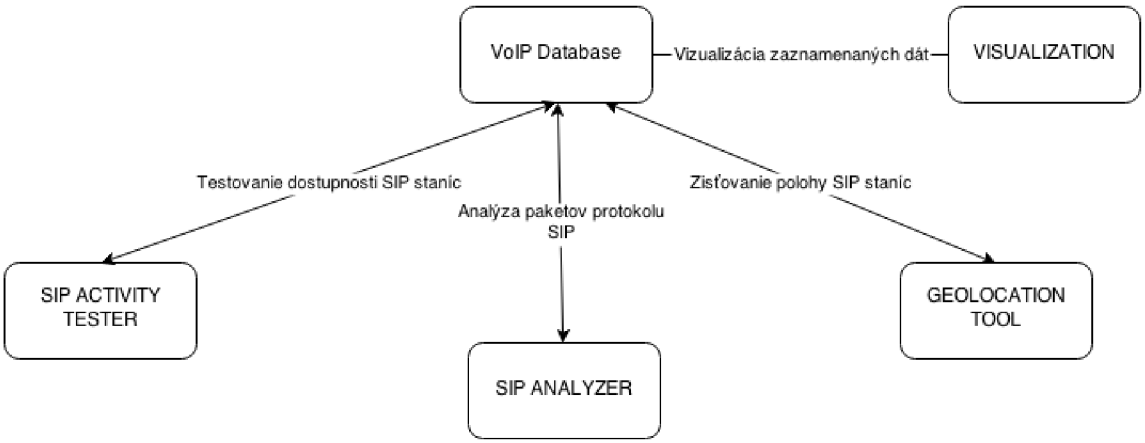
SIP ANALYZER
: Je blok zodpovedný za spracovávanie prijatých paketov protokolu SIP a ich následnu analýzu. Tento blok bude poskytovať funkcionalitu pre spracovávanie paketov protokolu SIP predovšetkým analýzu jeho hlavičiek na základe, ktorých bude urôvaný stav telefónnej siete (prebiehajúce hovory, požiadavky na registrácie...). Získané informácie budu zaznamenavané vo vhodnej podobe do databázy. Blok bude podporovať 2 typy analýzy sieťovej prevádzky a to nasnímanej, ktorá je uložená v súbore s formátom pcap a real-time analýzu. Blok `SIP ANALYZER` predstavuje pasívnu časť monitorovacieho systému, keďže negeneruje žiadne správy ale stav siete len odvodzuje od prebiehajúcej sieťovej komunikácie.

SIP ACTIVITY TESTER
: Predstavuje aktívny monitorovací blok keďže v pravidelných časových intervalov zasiela správy pre overenie dostupnosti staníc, ktoré by mali byť schopné komunikovať za pomoci protokolu SIP. Informácie o dostupnosti staníc budu zaznamenávané do databázy.

GEOLOCATION TOOL
: Je blok, ktorý je využívaný pre získavanie informácií o geografickej polohe zariadenia, komunikujúceho protokolom SIP.

VISUALIZATION
: Predstavuje blok, ktorý realizuje vizualizovanie zaznamených dát (výpis hovorov,

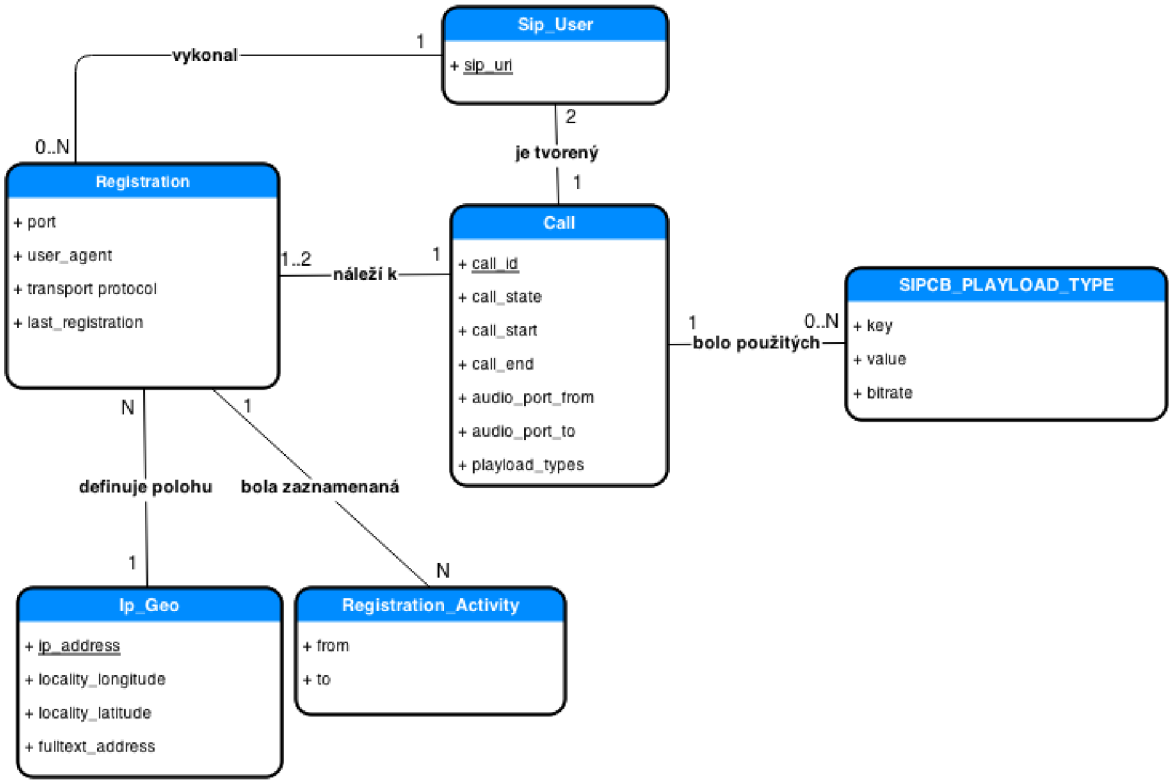
registrácie klientov,štatistiky ...) v prehľadnej podobe, pričom umožnuje pohodlné filtrovanie výsledkov. Tento blok realizuje užívateľské prostredie za pomoci ktorého sieťový administrátor komunikuje so systémom.



Obrázek 3.1: Logické bloky reprezentujúce navrhovaný systém.

Konceptuálny model

Počas navrhovania systému som vytvoril ERD diagram, ktorý zachycuje dátové entity a vzťahy medzi nimi. Medzi základné entity patria: `Sip_User` - reprezentuje užívateľa protokolu SIP, `Registration` - reprezentuje registráciu klienta z konkrétnej IP adresy a portu. `Call` - popisuje obsah informácií o hovore, pričom hovor je definovaný 2 užívateľmi medzi, ktorými prebieha komunikácia a zároveň sa vzťahuje ku 2 registráciam. Na základe vzťahu medzi entitami `Registration` a `Sip_Call` je možné pri telefónnom hovore určiť konkrétne IP adresy zariadení medzi, ktorými prebiehal hovor. Zároveň je možné zaznamenávať hovor typu kedy volaná stanica nie je registrovaná v tomto prípade vzťah medzi entitami `Sip_Call` a `Registration` je 1:1 keďže registrovaná je len volajúca stanica. V rámci hovoru mohlo byť použitých 0..N kódekov pričom popis kódeku je definovaný entitou `SIPCB PLAYLOAD_TYPE`. Entita `Ip_Geo` definuje údaje o geografickej polohe vzhľadom ku IP adrese zariadenia a tak k registrácií sa vzťahuje práve 1 takýto záznam. K jednej registrácií sa vzťahuje N záznamov o aktivite zariadenia, táto skutočnosť je modelovaná za pomoci entity `Registration_Activity` a jej vzťahu k entite `Registration`. ERD Diagram pre monitorovací systém je zachytený na obrázku 3.2.



Obrázek 3.2: Konceptuálny model - ERD diagram.

3.3 Použité technológie

Na základe doterajších skúsenosti bola vybraná databáza MySQL.

Pre samotnú implementáciu bolo potrebné zvoliť vhodnú implementačnú platformu. Pre implementáciu blokov `SIP ANALYZER`, `GEOLOCATION TOOL` a `SIP ACTIVITY TESTER` bol vybraný skriptovací jazyk Python verzie 2.7.6 [3]. V prostredí jazyka Python boli použité štandardné knižnice dodávane s inštaláciou interpreta jazyka, ktoré boli doplnené knižnicou Scapy[4], ktorá je používaná pre spracovávanie paketov až po transportnú vrstvu. Obsah transportnej vrstvy je analyzovaný a následne spracovávaný za pomoci vlastne implementovaných skriptov.

Pre implementáciu bloku `VISUALIZATION` bol zvolený skriptovací jazyk PHP s použitím frameworku CodeIgniter[5].Pre riešenie vzhľadu webovej aplikácie bol použitý CSS/JS framework Bootstrap[6]. Javascriptová funkcionalita bola riešená za pomoci frameworku jQuery[7]. Pre tvorbu grafov boli použité API od spoločnosti Google pod názvom Google Charts API[8] doplnené o API vis.js[9], ktoré bolo použité pre tvorbu časových a sieťových diagramov.

[3]Viac informácií na: http://docs.python.org/2/
[4]Viac informácií na: http://www.secdev.org/projects/scapy/
[5]Viac informácií na: http://ellislab.com/codeigniter
[6]Viac informácií na: http://getbootstrap.com
[7]Viac informácií na: http://jquery.com
[8]Viac informácií na:https://developers.google.com/chart/?hl=cs
[9]Viac informácií na: http://visjs.org

Pre získavanie dát popisujúcich geografickú polohu (zemepisná šírka a dĺžka) bolo použité IP location JSON API poskytované serverom IPInfoDb [10].

3.4 Implementácia dátového modelu

Podkapitola sa zaoberá na začiatku vytvorením odpovedajúcej databázovej schémy na základe ERD-diagramu 3.2 a je zakončená popisom vytvoreného rozhrania pre zaznamenávanie objektov systému do databáze.

Databázová schéma

Samotná databáza je tvorená celkovo 12 tabuľkami. Tieto tabuľky môžme rozdeliť do 2 skupín a to na tabuľky obsahujúce číselnikové hodnoty a tabuľky, ktorých obsah sa často mení. Význam jednotlivých tabuliek je uvedený v tabuľke 3.1.

Tabuľky číselníkových hodnôt obsahujú stĺpce `id,key,value` - pričom stĺpec `id` je primárnym kľúčom a je typu integer, `key` - unikátna textová hodnota vhodná pre vyhľádavanie dátového typu `varchar` a `value` - textová hodnota číselniku dátoveho typu `varchar`. Jedinou výnimkou z týchto tabuliek je tabuľka `sipcb_playload_type`, ktorá navyše obsahuje stĺpce `bitrate, description,sipcb_playload_type_source_id`. Stĺpec bitrate obsahuje informáciu o počte bitov, ktoré je potrebné zakódovať za jednotku času, stĺpec je typu `integer(11)`. Stĺpec sipcb_playload_type_source_id obsahuje cudzí kľúč do tabuľky `sipcb_playload_type_source` obsahujúcej informáciu o zdroji tejto číselnikovej hodnoty.

V tabuľkách s premenlivým obsahom stĺpce ktorých názov končí reťazcom _id obsahuje cudzi kľúč do odpovedajúcej tabuľky bez tejto prípony.

Tabuľka `auth` obsahuje informácie o užívateľoch systému(administrátoroch), ktoré sú využívané pre potreby zabezpečenia vizualizačnej časti systému. Tabuľka obsahuje stĺpce *username* - užívateľské meno, *password* - heslo, *email* - emailová adresa užívateľa.

Tabuľka `ip_geo` obsahuje záznamy o geografickej polohe vzťahujúcej sa k IP adresám zaznamenaným zo SIP komunikácie a tak obsahuje stĺpce *locality_latitude, locality_longitude* - zemepisné súradnice, *fulltext_address* - geografická adresa a *ip_address* - logická adresa.

Tabuľka `sip_user` obsahuje zoznam `sip_uri` užívateľov komunikujúcich protokolom SIP.

Tabuľka `registration` obsauje informácie o registráciach užívateľov pričom primárnym kľúčom v tejto tabuľke je kombinácia hodnôt v stĺpcoch *ip_geo_id*, *sip_user_id*. Medzi ďalšie údaje patria *sip_ports* - port ktorý identifikuje klientskú SIP stanicu,*user_agent*, *display_name* - zobrazované meno užívateľa, ,*is_udp* - príznak s hodnotou(0/1) informujúci či stanica komunikuje za pomoci protokolu UDP alebo iným transportným protokolom,*sipcb_registration_state_id*, *last_registration* - dátum a čas posledného požiadavku na registráciu a *sipcb_registration_state_id* .

Tabuľka `sip_call` obsahuje informácie o prebiehajúcich a uskutočnených telefónnych hovoroch pričom obsahuje informácie : *sip_callid* -identifikátor tel. hovoru, *sip_user_from_id, sip_user_to_id* - volajúci a volaný užívateľ, *registration_from_id, registration_to_id* - konkrétne registrácie volajúcej a volanej stanice,*sipcb_call_state_id* - stav hovoru, *sipcb_playload_type* - množina hodnôt reprezentujúca multimediálne kódeky, ktoré môžu byť použité počas hovoru, *call_start, call_end* - začiatok a koniec hovoru a *audio_port_from, audio_port_to* - porty používane pre adresovanie multimediálných tokov transportnou vrstvou.

[10] Viac informácií na:http://www.ipinfodb.com/ip_location_api_json.php

Názov tabuľky	Význam	Číselník
auth	Systémový administrátori	nie
sipcb_call_state	Stav hovoru	áno
sipcb_playload_type	Audio profily	áno
sipcb_playload_type_source	Zdroj názvu kodeku	áno
sip_user	Užívateľske účty	nie
sip_call	Záznamy o hovoroch	nie
registration	Registrácie SIP užívateľov	nie
registration_activity	Aktivita SIP užívateľov	nie
ip_geo	Preklad IP adres na geografickú polohu	nie

Tabulka 3.1: Význam tabuliek

Realizácia komunikácie medzi DB a systémom

Pre komunikáciu monitorovacieho systému s databázou bola navrhnutá a následne implementovaná abstraktná trieda `Model`, ktorá predstavuje rodičovskú tiredu pre ďalšie triedy, ktorých objekty budú ukladané do databáze napr. telefónne hovory, registrácie....

Trieda `Model` ku svojej činnosti využíva triedu `Db`, ktorej objekt je jednou z členských premenných. V rámci konštruktoru triedy `Db` je vytvorené spojenie s databázou, pričom pre nastavenie parametrov spojenia(názov zariadenia s Db, prihlasovacie meno a heslo, názov databáze) je využívaný konfiguračný súbor vo formáte JSON.V prípade, že nebolo úspešne vytvorené spojenie s databázou prípadne konfiguračný súbor neexistuje, tak je monitorovací systém ukončený s chybou. Po vytvorení objektu triedy `Db` je možné za pomoci metód pracujúcich s objektom triedy `Db` získať databázový kurzor pre realizáciu databázových operácií.

Pre umožnenie ukladania objektov tried dediacich od triedy musia mať v konštruktore definovanú sadu členských premenných na základe, ktorých je určená príslušnosť ku databázovej tabuľke, informácia o stĺpci, ktorý predstavuje primárny kľúč tabuľky a samotné priradenie hodnôt atribútov objektu ku konkretným stĺpcom tabuľky.

Príklad časti konštruktora triedy `Registration` ilustruje spôsob vytvorenia vazby medzi objektom triedy a databázovou tabuľkou.

```
#názov tabuľky, ku ktorej sa vzťahuje model
self._table_name = "registration"
#názov stĺpca definujúci primárny kľúč
self._table_pk = "id"

#definovanie priradenia atributov objektu ku stĺpcom tabuľky
self._db_columns = {
#názov stĺpca : type - dátový typ , variable - názov členskej premennej
"sip_user_id": { "type": "int", "variable": "_sip_user_id"},
"ip_geo_id": {"type": "int", "variable": "_ip_geo_id"},
"is_udp": {"type": "int", "variable": "_is_udp"},
"sip_ports": {"type": "str", "variable": "_sip_ports"}
}
```

Na základe definície týchto premenných je možné realizovať operácie insert,update,delete nad daným objektom v prostredí databáze. Implementácia tejto triedy predstavuje výcho-

diskový bod pre implementáciu tried `SipUser`,`SipCall`,`Registration...`, ktoré poskytujú funkcionalitu pre správu daných objektov v prostredí databáze a samotného analyzátora protokolu SIP.

Pre zamedzenie vytváraniu duplicitných záznamov v databázy, trieda `Model` obsahuje metódu `find`, za pomoci ktorej je vyhľadavaný databázový záznam obsahujúci atribúty objektu, ktorý chceme uložiť. Databázový záznam je vyhľadávaný na základe obsahu parametru (WHERE podmienka SQL dotazu) metódy `find`. V prípade, že záznam bol úspešne vyhľadaný tak je nastavený atribut objektu, ktorý obsahuje id záznamu v databázovej tabuľke.

Triedy definujúce objekty databáze

Pre zabezpečenie pohodlnej práce s dátovými objektami systému bola implementovaná sada tried: `SipUser, Registration, RegistrationActivity, SipCall, IpGeo`. Ttieto triedy dedia funkcionalitu od triedy `Model`, čo umožnuje nad týmyto objektami realizovať operácie update a delete. V každej z týchto tried je implementovaná metóda *save* - ktorá zapuzdruje kontrolu existencie záznamu v databázy na základe primárneho kľúča, čím sa zamedzí ukladaniu duplicitných záznamov do databáze.

Zároveň v jednotlivých triedach sú obsiahnuté špecifické metódy napr. v prípade triedy `SipCall` metóda pre výpočet dĺžky telefónneho hovoru.

Pri triedach `SipCall a Registration` sa stretávame s členskými premennými, ktoré hovoria o stave telefónneho hovoru, či registrácie. Množina hodnôt, ktorú môže nadobúdať je definovaná za pomoci výčtu, ktorý je realizovaný ako sada premenných triedy, pričom ich hodnota odpovedá číselníkovým hodnotám uloženým v databázy. Týmto krokom sa zamedzilo častému pristupovaniu k databázy.

3.5 Implementácia bloku SIP Analyzer

V tejto podkapitole si popíšeme implementáciu bloku `SIP Analyzer`, ktorý je realizovaný ako množina tried. Trieda `SipAnalyzer`(sip_analyzer.py) predstavuje radič pre analýzu komunikácie protokolu SIP. Na základe informácií získaných zo správ protokolu, trieda vytvára objekty tried `SipUser, SipCall, IpGeo, Registration` a následné s nimi pracuje.

Trieda `SipAnalyzer` je navrhnutá tak aby uspokojila požiadavky pre analýzu komunikácie zaznamenanej a následne uloženej v súbore s príponou `pcap` a zároveň real-time analýzy sieťovej prevádzky. Blok môže fungovať v logovaciom režime, kedy sú na štandardný výstup zobrazované výpisy o prebiehajúcej činnosti. V prípade závažnej chyby (nemožnosť pripojiť sa k databázy ...) je zobrazená chybová hláška na štandardný chybový výstup a následne činnosť bloku je ukončená. Vlastnosti analyzátoru (VoIP doména, logovací režim, typ monitorovania) su určené za pomoci parametrov konštruktoru triedy `SipAnalyzer`.

Trieda `SipAnalyzer` obsahuje členské premenné pre uchovanie množiny portov, na ktorých prebieha SIP komunikácia a 2 dátove štruktúry slovníkového typu pre uchovanie telefónných hovorov (objekty triedy `SipCall`) a registrácií užívateľov (objekty triedy `Registration`). V prípade dátovej štruktúry pre uchovanie telefónnych hovorov je kľúčom pre prístup k hovoru, hodnota SIP hlavičky `Call-Id`. Prvky v dátovej štruktúre pre registrácie sú jednoznačne identifikované za pomoci kombinácie SIP uri a ip adresy, z ktorej prichádza požiadavka na registráciu stanice. Na základe špecifikácie protokolu SIP môže byť registrácia jednoznačne identifikovaná na základe obsahu hlavičky `Call-id` avšak na

základe testov som sa rozhodol, že registrácie nebudu takto identifikované keďže niekoľkonásobne som sa stretol s faktom, že požiadavky pre registráciu zo zhodnej IP adresy a klientskej aplikácie obsahujú rozdielne hodnoty hlavičiek `Call-id`.

Samotné spustenie činnosti `SIP ANALYZER` je realizované za pomoci spustenia skriptu `main.py` s vhodnými parametrami príkazového riadku, ktorý sa postará o vytvorenie objektu triedy `SipAnalyzer` a následné spustenie analyzátora.

Identifikácia SIP komunikácie

Pre identifikáciu SIP komunikácie bola v triede `SipAnalyzer` definovaná členská premenná, dátoveho typu set[11],(množina), ktorá uchováva množinu TCP/UDP portov, na ktorých prebieha SIP komunikácia. Počas štartu analyzátoru je množina inicializovaná hodnotou 5060, ktorá predstavuje štandardný SIP port. Následne počas doby spustenia analyzátoru je obsah dátovej štruktúry aktualizovaný o ďalšie porty, ktoré boli získané z požiadavkov *INVITE* a *REGISTER*.

V prípade, že cieľový/zdrojový port práve spracovaváneho paketu sa nachádza v tejto množine a zároveň obsah správy začína reťazcom SIP alebo jedným z názvov metód protokolu SIP tak je rozhodnuté, že sa jedná o komunikáciu protokolom SIP. Tieto pakety následne budú spracovávané metódou `parseSipPacket` triedy `SipAnalyzer`.

Získavanie hodnôt hlavičiek protokolov SIP a SDP

Pre získanie hodnôt niektorých povinných hlavičiek protokolu SIP boli navrhnuté regulárne výrazy s podporou spracovávania krátkych názvov hlavičiek. Podoba týchto výrazov je uložená v premenných triedy `SipAnalyzer`.

```
Získanie zobrazovaného mena a SIP URI z hlavičiek From a To.
Voliteľné časti sú ignorované.
"(?i)\r\n(From|f)\s*:\s*(.*)<([^>;]*).*>;?.*\r\n"
"(?i)\r\n(To|t)\s*:\s*(.*)<([^>;]*).*>;?.*\r\n"
Získanie ip adresy zariadenia a portu z hlavičky Contact.
"(?i)\r\n(Contact|m)\s*:\s*[^<]*<.*@(?=)([^:>]*):?([^;>]*);?.*>\r\n"
Získanie Call-ID identifikátora
"(?i)\r\n(Call-ID|i)\s*:\s*(.*)\s*\r\n"
Získanie sekvenčného čísla v rámci transakcie a názvu metódy z hlavičky CSeq.
"(?i)\r\nCSeq\s*:\s*([0-9]+)\s*(.*)\r\n"
Získanie reťazc user agent.
"(?i)\r\nUser-Agent\s*:\s*(.*)\r\n"
```

Ďalším dôležitým regulárnym výrazom používaným v monitorovaciom systéme je výraz pre získavanie informácií o multimediálných kódekoch. Tieto informácie sú uložené v správe protokolu SDP, ktorá je zapuzdrená v požiadavku INVITE. Pre získavanie týchto údajov bol navrhnutý nasledovný regulárny výraz:

```
Media_type reprezentuje množinu hodnôt RTP AVP profilov.
(?i)a=rtpmap:({media_type})\s*([^//]*)//?([0-9]*).*\r\n
```

Podpora pre získavanie hlavičiek protokolov SIP a SDP je realizovaná v rámci triedy `SipAnalyzer` prostredníctvom metód ako napríklad *geCallId, getUserAgent...*, ktoré vracajú hodnoty požadovaných hlaviečiek.

[11] Viac informácií na: https://docs.python.org/2/library/stdtypes.html

Spracovávanie správ protokolu SIP

Spracovávanie správ je realizované metódou `parseSipPacket` triedy `SipAnalyzer`. Činnosť tejto metódy môžme rozdeliť do niekoľkých krokov.

1. Určenie či sa jedná o požiadavok alebo odpoveď na základe prvého riadku správy
2. Keď sa jedná o požiadavok a zároveň názov metódy je INVITE tak je zaznamenaný požiadavok o telefónny hovor a spustená činnosť spojená so spracovávaním údajov o telefónných hovoroch prostredníctvom metódy `parseInviteMessage`. Keď sa jedná o metódu REGISTER je spustená obsluha pre spracovávanie požiadavkov pre registráciu klienta metódou `parseRegisterMessage`. V prípade správ ACK, BYE a CANCEL je v dátovej štruktúre s telefónnymi hovormi triedy `SipAnalyzer` vybraný telefónny hovor.Následne na základe typu požiadavku je stav hovoru zmenený nasledovne ACK - úspešne začatý, BYE - ukončený, CANCEL - odmietnutý. Podpora ďalších metód protokolu nie je implementovaná.
3. V prípade odpovede na požiadavok je názov metódy určený na záklde obsahu hlavičky CSeq. Pri odpovediach na metódy INVITE, REGISTER sú spustené ich obsluhy pre spracóvavanie odpovedí na tieto požiadavky. V oboch prípadoch avšak musí existovať záznam uchovavajúci informáciu o požiadavku, ku ktorému sa odpoveď vzťahuje.

Spracovávanie registračných požiadavkov

Je zapuzdrené 2 metódami a to `parseRegisterMessage` a `parseRegisterResponse` triedy `SipAnalyzer`.

Požiadavok na registráciu klienta je klasifikovaný na základe SIP správy s metódou *REGISTER*, ktorý generovala klientská stanica. Registračný požiadavok obsahuje dôležité informácie pre náš systém a to: názov užívateľa(SIP Uri), ip adresa a port klientskej stanice.

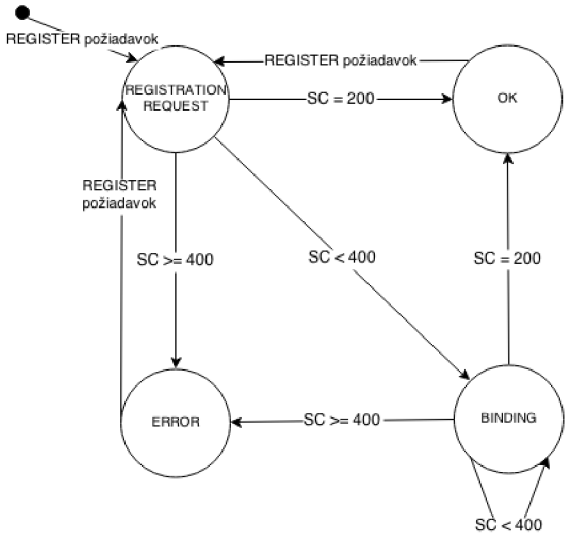
Požiadavok *REGISTER* je spracovávaný metódou `parseRegisterMessage`. Činnosť implementovanej metódy je možné popísať nasledovne:

- Určenie transportného protokolu, ktorým bol registračný požiadavok zaslaný, na základe existencie jedného z podreťazcov "TCP" alebo "UDP" v prvom riadku správy protokolu. Informácia o použitom transportnom protokole bude neskôr používaná blokom `SIP ACTIVITY TESTER` pre zaslanie správ *OPTIONS* za pomoci ktorých bude testovaná dostupnosť danej stanice.
- Získanie SIP URI užívateľa, ktorý sa pokúša registrovať z hlavičky *To* a ďalších atributov definujúcich registráciu ako napríklad názov klientskej aplikácie, ktorá generovala požiadavok(hlavička *UserAgent*) a veľkosť časového intervalu v sekundách počas, ktorých bude užívateľ po úspešnej registrácií považovaný za aktívneho(hlavička *Expires*). Na základe obsahu hlavičky `To` bude vytvorený databázový objekt definovaný triedou `SipUser`.
- V prípade, že hodnota hlavičky *Expires* je 0 tak sa jedná o odregistrovanie užívateľa a tak je objekt triedy `Registration` odstránený z dátovej štruktúry uchovávajucej registračné požiadavky umiestnenej v objekte triedy `SipAnalyzer`. Následne je činnosť metódy ukončená.
- Získanie ip adresy /doménového mena a portu z hlavičky *Contact*. Získaná IP adresa bude zároveň použitá pre potreby geolokácie.

- Do dátovej štruktúry, obsahujúcej SIP porty je pridaný port ktorý určuje stanicu, ktorá sa pokúša registrovať. V prípade použitia transportného protokolu TCP sa jedná o 2 porty.

- Vytvorenie objektu triedy `Registration` reprezentujúci registračný požiadavok,nastavenie všetkých získaných parametrov a následné jeho uloženie do dátovej štruktúry pre registračné požiadavky a do databáze.

Stav registrácie je zmenený na základe požiadavkov `REGISTER` a stavového kódu odpovedí na tieto požiadavky. Spôsob akým je stav registrácie menený je popísaný za pomoci automatu na obrázku 3.3. V prípade registrácií je do databáze zaznamenávaný stav posledného registračného požiadavku. Príklad výstupu monitorovacieho systému, ktorý je spustený v logovaciom režime Blok `SipAnalyzer` spustený v logovaciam režime zobrazuje na štandardný výstup informáciu o vzniknutom registračnom požiadavku a o zmenách jeho stavu. Príklad výstupu je možné vidieť na obrázku 3.4.



Obrázek 3.3: Konečný automat pre zmenu stavu registrácie klientskej stanice

```
2014-04-29 12:18:10 -- Registration: sip:user1@Test1/192.168.10.8 Error SC:403
2014-04-29 12:18:40 -- Registration: sip:user1@Test1/192.168.10.8 Want register
2014-04-29 12:18:40 -- Registration: sip:user1@Test1/192.168.10.8 Error SC:403
2014-04-29 12:19:10 -- Registration: sip:user1@Test1/192.168.10.8 Want register
2014-04-29 12:19:10 -- Registration: sip:user1@Test1/192.168.10.8 Successfull
```

Obrázek 3.4: Príklad výstupu monitorovacieho systému

Získavanie informácií o prebiehajúcich telefónných hovoroch

Pre získavanie informácií o prebiehajúcich telefónných hovoroch sú využívané SIP správy s požiadavkami *INVITE, BYE, CANCEL* a ich odpoveďami. Príslušnosť správ k telefónnemu hovoru je určená na základe obsahu hlavičky *Call-Id*.

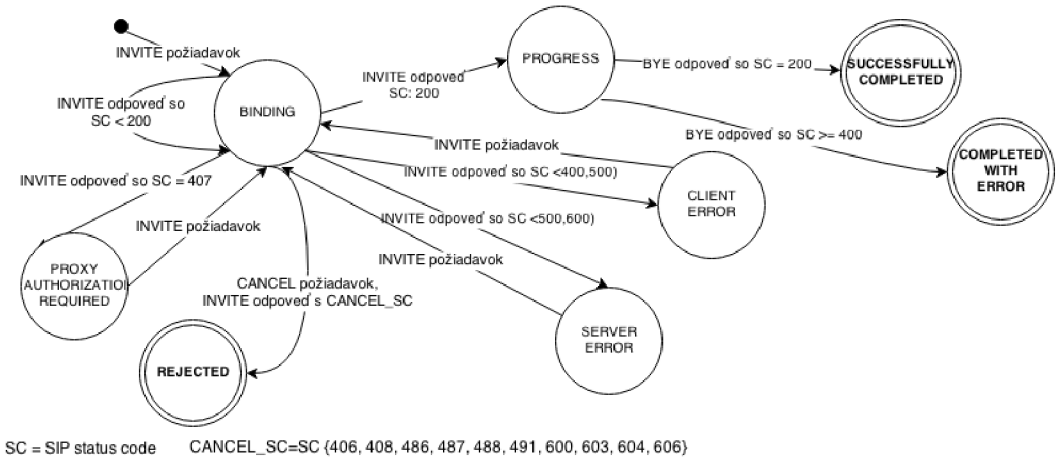
Požiadavok na telefónny hovor je analyzovaný za pomoci metódy `parseInviteMessage`. Zoznam činnosti realizovaných spomenutou metódou je možné definovať nasledovne:

- Vytvorenie objektu triedy `SipCall` reprezentujúci telefónny hovor a nastavenie jeho stavu na hodnotu *BINDING_STATE* - prebieha vytváranie spojenia
- Nastavenie atribútov pre identifikáciu volajúcej a volanej stanice, vrátane informácií o registrácií, z ktorej bol vygenerovaný požiadavok.
- Na základe tela správy, ktoré je tvorené protokolom SDP prebieha získanie parametrov pre multimediálny prenos (kódeky, ktoré akceptuje volajúca stanica, zdrojové číslo portu...).
- V poslednom kroku je objekt uložený do databáze a do dátovej štruktúry uloženej v triede `SipAnalyzer`, ktorá uchováva zoznam práve prebiehajúcich telefónnych hovorov reprezentovaných objektami triedy `SipCall`.

Stav hovoru je menený na základe ďalších správ a ich odpovedí, ktoré sa vzťahuju k uvedenému telefónnemu hovoru. Medzi tieto správy patria *INVITE, CANCEL, BYE*. Spôsob akým sa mení stav hovoru je definovaný konečným automatom na obrázku 3.5. V prípade, že bol prijatý požiadavok *CANCEL* tak je hovor označený ako odmietnutý. Keď sa jedná o požiadavok *BYE* je hovor ukončený úspešne alebo chybou. Pri odpovediach na požiadavky *INVITE a BYE* je stav hovoru menený na základe stavového kódu odpovedí. Hodnota status kódu, poslednej odpovede, je uložená v objekte reprezentujúcom telefónny hovor. Odpovede viažúce sa k metódam *INVITE a BYE* sú spracovávané metódami `parseInviteResponse a parseByeResponse`.

V prípade odpovede na požiadavok *INVITE* so stavovým kódom 200 (telefónny hovor začal) objekt reprezentujúci tel. hovor má nastavené atribúty: multimediálny port volanej stanice, registráciu volanej stanice.Na základe testov a obsahu špecifikácie protokolu SIP bola zostavená množina stavových kódov pre odmietnutie hovoru.

Blok `SipAnalyzer` spustený v logovaciam režime zobrazuje na štandardný výstup informáciu o vzniknutom požiadavku na telefónny hovor a o zmenách jeho stavu. Príklad výstupu je možné vidieť na obrázku 3.6.



Obrázek 3.5: Konečný automat pre zmenu stavu telefónneho hovoru

```
2014-04-29 12:26:54 -- Call: sip:user1@Test1 ---> sip:123@Test1 Request
2014-04-29 12:26:54 -- Call: sip:user1@Test1 ---> sip:123@Test1 Proceeding
2014-04-29 12:26:59 -- Call: sip:user1@Test1 ---> sip:123@Test1 Completed
```

Obrázek 3.6: Príklad výstupu monitorovacieho systému v prípade detekcie telefónneho hovoru

3.6 Implementácia blokov `SIP ACTIVITY TESTER, GEOLOCATION TOOL`

V tejto podkpapitole si popíšeme 2 logické celky monitorovacieho systému a to `SIP ACTIVITY TESTER` a `GEOLOACTION TOOL`. Sústredime sa predovšetkým na princípy na základe, ktorých boli tieto logické celky implementované.

Implementácia bloku `SIP ACTIVITY TESTER`

Spôsob overovania aktivity klientskej stanice spočíva v odosielaní správ s požiadavkom *OPTIONS* klientským staniciam v pravidelných časových intervaloch. Požiadavok typu *OPTIONS* bol vybraný keďže sa jedná o povinný typ správy, ktorého spracovanie musí podporovať každá stanica komunikujúca protokolom SIP. Požiadavok *OPTIONS* slúži pre získavanie informácií o možnostiach prenosu konkrétneho klienta. Ked je stanica aktívna odpovedá na tento požiadavok odpoveďou so stavovým kódom 200.

Dôležitým aspektom, ktorý treba rešpektovať je transportný protokol, za pomoci, ktorého klienstká stanica zaslala registračný požiadavok.

Blok je tvorený triedami `TestActivity, Worker a RegistrationActivity`.
Trieda `TestActivity` predstavuje radič samotného bloku. Táto trieda obsahuje členské premenné uchovávajúce zoznam aktívnych staníc a spoločnú frontu pre objekty triedy `Worker`. Spoločná fronta obsahuje zoznam rgistrácií, ktorých aktivitu je potrebné overiť.

Aktivita na registrácií je reprezentovaná ako objekt triedy `RegistrationActivity`, ktorý obsahuje informácie o registrácí ,ku ktorej sa viaže záznam, začiatku a konci aktivity. Trieda obsahuje funkcionalitu pre zaznamenanie záznamu do databázovej tabuľky registration_activity a umožnuje zvyšovať dobu aktivity danej registrácie.

Zasielanie správ je realizované za pomoci niekoľkých vlákien, ktoré pracujú so spoločnou frontou z triedy `TestActivity`. Vlákna sú typu deamon a tak ich činnosť je spustená v prípade, že sa nachádza požiadavok vo fronte. Obsah tejto fronty je doplňaný v pravidelných časových intervaloch triedou `TestActivity`. Činnosť vlákien bola implementovaná vytvorením triedy `Worker`, ktorá dedí od triedy `Thread` z modulu *threading*[12].

Trieda `Worker` poskytuje nasledujúcu funkcionalitu:

- Vytvorenie schránky (socket) pre správny transportný protokol (na základe obsahu stĺpca *is_udp* v tabuľke *registration*).
- Zostavenie požiadavku *OPTIONS* protokolu SIP. Hlavička `To` obsahuje SIP URI stanice, ktorej aktivitu chceme overiť.Obsah hlavičky `Via` je tvorený informáciami o transportnom protokole, ktorým je požiadavok zaslaný doplnené o údaje obsahujúce IP adresu a aplikačný port identifikujúci socket triedy `Worker`, ktorým bol požiadavok

[12]Viac informácií na https://docs.python.org/2/library/threading.html

zaslaný. Parameter `branch` hlavičky `Via` a hodnota hlavičky `Call-id` je náhodne generovaný reťazec o dĺžke 14 znakov. Príkladom požiadavku OPTIONS generovaného triedou `Worker` je nasledujúci požiadavok:

```
OPTIONS sip:sip:tomas@my-lab SIP/2.0
Via: SIP/2.0/UDP 147.229.192.182:52535;branch=z9hG4000740ec0
Max-Forwards: 70
To: <sip:sip:tomas@my-lab>
From: <sip:tester@sipanalyzer.com>;tag=1928301774
Call-ID: 000b7d0f8@sipmonitor
CSeq: 1 OPTIONS
Accept: application/sdp
Content-Length: 0
```

- Zaslanie požiadavku za pomoci BSD schráky.
- V prípade, že odpoveď nedorazí v intervale 3 sekundy alebo intervale definovanom parametrom skriptu *--timeout* je stanica považovaná za neaktívnu. V prípade príjmu odpovedi je stav registrácie odvodený na základe stavového kódu prijatej/tých odpovedí. V prípade stavového kódu nižsieho ako je hodnota 200 tak sa čaká na príchod ďalších odpvodí. Keď hodnota stavového kódu je v intervale $\langle 200,300)$ tak je stanica považovaná za aktívnu. V prípade aktivity stanici je buď vytvorený nový objekt triedy `RegistrationActivity` alebo je vyhľadaný v dátovej štruktúre obsahujúcej zoznam aktívnych staníc a hodnota veľkosť časového intervalu reprezentujúceho aktivitu je navýšená o hodnotu predstavujúcu interval pre testovanie aktivity stanice.
  Príklad odpovede na požiadavok OPTIONS informujúci o aktivite SIP stanice:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 147.229.192.182:52535;branch=z9hG4000740ec0
From: <sip:tester@sipanalyzer.com>;tag=1928301774
To: <sip:sip:tomas@my-lab>;tag=1291809152
Call-ID: 000b7d0f8@sipmonitor
CSeq: 1 OPTIONS
Server: YATE/4.0.1
Allow: ACK, INVITE, BYE, CANCEL, OPTIONS, INFO
Content-Length: 0
```

Veľkosť intervalov pre overovanie aktivity je štandardne nastavený na 5minút a časový limit pre príjem odpovedí na požiadavky je nastavený na 3s. Veľkosť týchto intervalov je možné zmeniť za pomoci spustenia skriptu s parametrami príkazového riadku *--interval a --timeout*.

Implementácia bloku `GEOLOCATION TOOL`

Pod pojmom IP geolokácia v prostredí počítačových sietí rozumieme proces, počas ktorého je získavaná geografická poloha zariadenia pripojeného k počítačovej sieti s dosiahnutím určitej presnosti.

Problematika geolokácie sieťového zariadenia na základe IP adresy je značne zložitá. Je to spôsobené hlavne povahou počítačových sietí a tak nie je isté, že zariadenie s jednou IP adresou sa vždy nachádza na zhodnom mieste. V prípade sietí pracujúcich s prootokolom IPv4 je tento problém umocnený malým množstvom IP adries a tak sa často stretávame so situáciami kedy adresy jedného adresného bloku protokolu IP sú pridelované zaiadeniam naprieč rôznymi kontinentami.

Tvorba geolokačných databáz

Pre potreby IP geolokácie sú vytvárané geolokačné databáze, v ktorých nachádzame záznamy o IP adresách a ich geografickom začlenení (zemepisná šírka a dĺžka). Obsah týchto databáz je možné vytvárať za pomoci rôznych prístupov. Medzi základné prístupy ku IP geolokácií patria nasledovné:

- Informácie o geografickej polohe sú uložené v záznamoch DNS - V tomto prípade často záznamy neexistujú a zároveň nie je zaručená aktuálnosť záznamov.
- Využitie Traceroute - geografická poloha je určená na základe známej polohy najbližšieho smerovača na ceste k danému zariadeniu.
- Získanie geografickej polohy zariadenia na základe dotazovania na službu `whois`. Z odpovedi služby `whois` je možné získať informácie minimálne o registrátorovi uvedeného adresného priestoru, čísle AS a samotnej geografickej adresy prípadne informácie o koncovom užívateľovi. Hlavným problémom je možnosť existencie neaktuálných záznamov.
- Metódy založené na výpočtoch a meraniach. Jedná sa hlavne o merania latencie. Latencia prípadne iné merania sú realizované za pomoci zasielania „meracích správ(často ICMP správy) medzi sieťovými zariadeniami so známou geografickou polohou. Následne na základe vzťahu medzi nameranou hodnotou a vzdialenosťou je geografická poloha aproximovaná. Všeobecne je presnosť tejto metódy najvyššia ale zároveň veľmi náročná na realizáciu. Príkladom je komerčná služba `IP2Location`[13].
- Kombináciou vyššie spomenutých techník.

Použitie IP geolokačných služieb, ktorých databáza obsahuje geolokačné záznamy s vyššiou presnosťou sú veľmi často spoplatňované. Stretávame sa aj s bezplatnými IP geolokačnými riešeniami ako napríklad služby `freegeoip.net` alebo `ipinfodb.com`, ktoré dosahujú realatívne dobrú presnosť. Riešenia často poskytujú API, ktoré je možné priamo použiť vo vyvíjaných aplikáciach alebo je možné priamo stiahnuť geolokačnú databázu.

Realizácia IP geolokácie v prostredí monitorovacieho systému

Pri realizácií `GEOLOCATION TOOL` sme sa rozhodovali pre použitie API jednej zo služieb `ipinfodb.com` alebo `freegeoip.net`. Každá z týchto služieb má ako výhody tak ja nevýhody. Hlavnou výhodou služby `freegeopip.net` je to, že služba neobmedzuje počet požiadavkov na geolokáciu za jednotku času ako je v prípade služby `ipinfodb.com`, kde je limit 2 požiadavky/sekundu. Databáza služby `ipinfodb.com` je tvorená dátami komerčného riešenia `http://www.ip2location.com/` avšak s nižšiou presnosťou. Hlavne na základe

[13]Viac informácií na: http://ip2location.com/

tohto faktu sme sa rozhodli pre použitie webové `IP Address Geolocation JSON API` poskytované serverom www.ipinfodb.com[14]. Pre získanie kľúča pre prácu s geolokačným API služby `ipinfodb.com` bolo potrebné sa zaregistrovať na internetovej stránke služby.

Blok `GEOLOCATION TOOL` je tvorený predovšetkým triedou `IpGeolocationTool`. Pre spustenie činnosti `GEOLOCATION TOOL` je spustený skript *ip_geolocation_tool.py*, ktorý vytvorý objekt triedy `IpGeolocationTool`, nad ktorým zavolá metódu `run`. Po zavolaní metódy sa spusti činnosť geolokačného bloku.

V určitých časových intervaloch sú načítavané záznamy z tabuľky *ip_geo*, ktoré nemajú zadané informácie o geografickej polohe a zároveň obsahujú platné IP adresy. Veľkosť intervalu pre načítanie záznamov z databáze pre geolokáciu je v intervale ⟨1hodina,1deň⟩,pričom jeho hodnota je inicializovaná na 1 hodinu a následné je tento interval dynamicky menený v závislosti od počtu načítaných záznamov. V prípade, že ich počet nie je nulový tak sa hodnota intervalu zníži o hodnotu 1 hodina v opačnom prípade je táto hodnota navýšená o 1 hodinu.

Jednotlivé záznamy sú spracovávané sekvnečne, keďže pre bezplatné používanie služby serveru IPInfoDB je možné zaslať maximálne 2 požiadavky počas 1sekundy v inom prípade sú odpovede na požiadavky spomalované za pomoci front.

Následné je zaslaný HTTP požiadavok uvedenej službe pre preklad IP adresy na geografickú polohu. Formát požiadavku pre webovú službu je nasledovný:

```
http://api.ipinfodb.com/v3/ip-city/?key=<api-key>&ip=<ip>&format=json
```

Položka <api-key> predstavuje API kľúč získaný po registrácií a položka <ip> predstavuje IP adresu, na základe ktorej chceme určiť geografickú polohu.

Server odpovedá reťazcom vo formáte JSON, ktorého podoba môže byť nasledovná:

```
{
  statusCode: "OK",
  statusMessage: "",
  ipAddress: "147.229.192.182",
  countryCode: "CZ",
  countryName: "CZECH REPUBLIC",
  regionName: "JIHOMORAVSKY KRAJ",
  cityName: "BRNO",
  zipCode: "614 00",
  latitude: "49.1952",
  longitude: "16.608",
  timeZone: "+02:00"
}
```

Z geografických údajov sú zaznamenaváne údaje o zemepisnej šírke, dĺžke a textovej podoby adresy, ktorá pozostáva z údajov o meste, kraji a krajine. Po úspešnom preklade sú údaje zaznamenané do databázovej tabuľky *ip_geo* v inom prípade je IP adresa označená ako nevalidná príznakom 0 uloženom v stĺpci *is_valid*.

3.7 Implementácia bloku VISUALIZATION

Kapitola je orientovaná na implementáciu bloku `VISUALIZATION`, ktorý zodpovedá za zobrazovanie zaznamenaných dát.

[14] Viac informácií: http://www.ipinfodb.com/ip_location_api_json.php

Na základe dohody sme sa rozhodli, že vizualizácia bude implementovaná vo forme webovej aplikácie. Samotná webová aplikácia je vytvorená pomocou jazykov PHP a Javascript. Serverová časť aplikácie bola vytvorená za pomoci PHP frameworku CodeIgniter. Aplikácie vytvárané pomocou spmenutého frameworku sú vyvijané za pomoci návrhového vzoru MVC (Model-View-Controller). Uživateľské rozhranie bolo vytvorené pomocou frameworku Bootstrap (CSS + Javascript framwork) a Javascript komponenty jQuery. Komponenta jQuery je využívaná pre prácu s AJAXom. Zobrazovanie geografických dát je realizované pomocou Google Maps API[15], ktoré je poskytované spoločnosťou Google.

Impelmentáciu vizualizačnej časti systému môžme rozdeliť do niekoľkých celkov a to implementáciu modelov, ktoré pracujú priamo s dátami v databázových tabuľkách, radičov, ktoré riadia chod aplikácie a samotných zobrazovačov, pomocou ktorých je zobrazený výstup. Toto implementačné členenie je spôsobené použitím PHP frameworku s architektúrou MVC.

Spoločné implementačné rysy komponentov tvoriacich MVC

Bázovou triedou všetkých radičov je trieda `Main` (rodičovská trieda `CI_Controller`), ktorá poskytuje funkcionalitu pre overenie stavu prihlásenia do systému a poskytuje premennú, ktorá obsahuje dáta, ktoré budu zobrazované. V konštruktoroch radičov sú načítavané modely, ktoré sú používaé na všetkých akciách (podstránkach). Pod pojom „akciaÿ oblasti frameworku typu MVC rozumieme podstránku. Ttieto akcie sú realizované ako metódy radičov s modifikátorom prístupu `public`. Napr. po zadaní URL *http:/adresa_app/registration/show* je požiadavok smerovaný na radič triedy `Registration`. Následne je vybraná metóda `show`, ktorá zabezpečí zobrazenie obsahu na požadovanej podstránke.

Bázovou triedou vytvorených modelov je trieda `Base_model` (rodičovská trieda `CI_Model`), ktorá poskytuje funkcionalitu pre nastavenie názvu tabuľky, ku ktorej sa model vzťahuje, predvolený spôsob radenia dát pri SQL dotazoch, prevod časových údajov do formátu MySQL a samotné aplikovanie radenia záznamov. Tvorba databázovych dotazov SQL je realizovaná za pomoci databazóveho vzoru `Active Record`, ktorým je táto činnosť značne ulahčená, keďže z databázovými požiadavkami pracujeme ako s objektami.

Prihlásenie do web aplikácie

Keďže sú zobrazované citlivé dáta tak je potrebné aby sa užívateľ musel prihlásiť do vizualizačného podsystému. Pre prihlasovanie bola vytvorená tabuľka `auth`, v ktorej sú uchované užívateľské účty. Heslo pre prihlasovanie je uložené v šifrovanej podobe za pomoci SHA1 algoritmusu. Po úspešnom prihlásení je nastavená hodnota globalnej premennej `$SESSION["auth_id"]` na hodnotu primárneho kľúča prihlaseného užívateľa. Činnosť spojená s prihlasovaním je zapuzdrená v 2 triedach: `Auth` - predstavujúca radič a `Auth_model` - model. Na každej podstránke je kontrolovaný stav prihlásenia za pomoci metódy `checkLogin` v konštruktore bázovej triedy `Main` pre jednotlivé radiče. Keď užívateľ nie je prihlásený je vykonané presmerovanie na prihlasovaciu stránku.

Samotné prihlasovanie užívateľ vykoná zadaním odpovedajúcich prihlasovacích údajov do prihlasovacieho formuláru zobrazeného po zadaní URL adresy webovej aplikácie do webového prehliadača. Po úspešnom prihlásení je užívateľ automatický presmerovaný na podstránku obsahujúcu výpis telefónných hovorov.

[15] Viac informácií: https://developers.google.com/maps/?hl=sk

Pre pohyb v aplikácií bolo vytvorené horizontálne menu uvedené na obrázku 3.7, ktoré zároveň hovorí o členení webovej aplikácie.

Obrázek 3.7: Hlavné menu

Tabuľkové výpisy

Tabuľkové výpisy sú určené pre výpis tabuľkových hodnôt tabuliek `sip_user`, `sip_call`, `registration` a `registration_activity`. V niektorých prípadoch je možné zobraziť detailnejšie informácie vzťahujúce sa k záznamu za pomoci kliknutia na tlačítko s popiskom „Zobraziťpřípadne na identifikátor záznamu. V prípade niektorých záznamov je možné sa prekliknuť na detailne stránky záznamov, ktoré sa k vybranému záznamu vzťahujú. Napr. keď sa jedná o telefónny hovor tak je možné sa prekliknuť priamo na profilové stránky SIP užívateľov, ktorý sa podieľali na hovore.

Dáta je možné zoraďovať podľa hodnoty v jednom zo stĺpcov. Požiadavok pre zoradenie dát je realizovaný kliknutím na ikonu informujúcu o aktuálnom spôsobe zoraďovania. Ikona je umiestnená vedľa názvu stĺpcov podľa ktorých je možné dáta zoraďovať. Na jednej stránke je zobrazených maximálne 15 záznamov v inom prípade je zobrazený ovladací prvok pre pohyb po stránkach. Príklad tabuľkového výpisu je možné vidieť na obrázku 3.10.

Pre zobrazenie tabuľkového výpisu boli vytvorené akcie (metódy) `index` radičov, ktoré zabezpečujú zobrazenie požadovaných dát. Po spustení tejto metódy je načítaný obsah filtra, ktorý je predávaný za pomoci GET parametrov. Potom pomocou odpovedajúceho modelu sú načítané data z databáze, pri ktoróm je rešpektovaný obsah filtra. Napr. keď sa jedná o radič definovaný triedou `Registration` tak je pre načítanie dát pre tabuľkový výpis použitý model def. triedou `Registration_model`. Samotná realizácia filtrovania dát je vykonaná za pomoci AJAXu tak ako aj obsluha ďalších prvkov súvisiacich s tabuľkovými výpismi.

Obsluha zobrazovania detailu záznamu je zapuzdrená v akciách (metódach) `show` radičov, ktoré umožnujú uspokojenie tohto požiadavku.

Sekcia SIP užívatelia

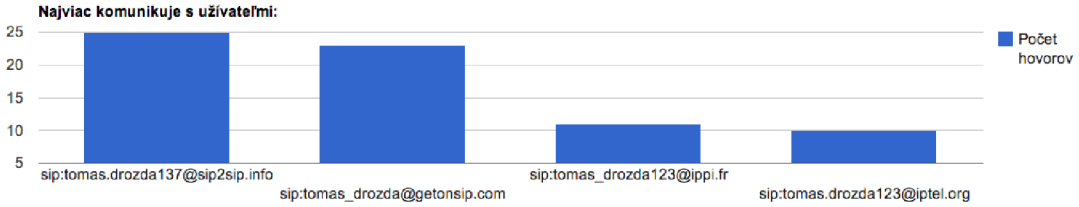
Sekcia slúži pre zobrazovanie dát obsahujúcich informácie o užívaľských účtoch získaných z registračných požiadavkov a telefónnych hovorov (dáta z tabuľky sip_user). Spôsob tabuľkového zobrazenia dát je možné vidieť na obrázku 3.8.

Profilová stránka obsahuje informácie o registráciach, pomocou ktorých sa pokúšal o pripojenie do VoIP siete, a telefónných hovoroch vo forme tabuľkových výpisov.

Stretávame sa tu s 2 štatistikami. Jedná sa o histogram množstva hovorov v jednotlivých hodinách dňa. Podobu štatistiky je možné vidieť na obrázku 3.19. Ďalšia štatistika hovorí o uživateľoch, s ktorými vybraný uživateľ uskutočňuje telefónne hovory najčastejšie. Jej podobu je možné vidieť na obrázku 3.9. Do štatistik sú zahrnuté hovory, ktoré boli úspešne ukončené alebo ukončené s chybou.

Užívateľ	Počet hovorov	Priemerná dĺžka hovorov	Posledná registrácia	Akcie
sip:123@Test1	8	00:00:03	5.5.2014 16:18:19	Zobraziť

Obrázek 3.8: Tabuľkový výpis - SIP užívatelia.



Obrázek 3.9: Graf zobrazujúci s kým užívateľ najčastejšie komunikuje.

Sekcia Výpis hovorov

Slúži pre zobrazenie informácií o telefónnych hovoroch. Obsahuje tabuľkový výpis, ktorého podoba je ilustrovaná za pomoci obrázku 3.10. Pri jednotlivých telefónnych hovoroch je možné sa prekliknuť na profily SIP užívateľov, ktorý sa podieľali na telefónnom hovore.

Čas vytvorenia požiadavku	Stav požiadavku	SIP status code	Volajúca stanica	Volaná stanica	Začiatok hovoru	Koniec hovoru	Dĺžka hovoru	Možné kódeky	Audio porty(zdroj - cieľ)	Akcie
5.5.2014 16:16:19	Hovor ukončený	200 OK	sip:user1@Test1	sip:123@Test1	29.4.2014 12:23:39	29.4.2014 12:23:44	00:00:05		11780-15524	Zobraziť

Obrázek 3.10: Tabuľkový výpis - Výpis hovorov

Po rozkliknutí záznamu je zobrazený detail telefónneho hovoru, ktorý na rozdiel od tabuľkového výpisu obsahuje ďalšie informácie. Jedná sa o položky: *Call-Id* telefónneho hovoru a údaje o geografickej polohe volajúcej a volanej stanice. Údaje o geografickej polohe sú zobrazené len v tom prípade keď bol úspešne vykonaný proces geolokácie IP adries identifikujúcich volajúcu a volanú stanicu.

Sekcia Registrácie

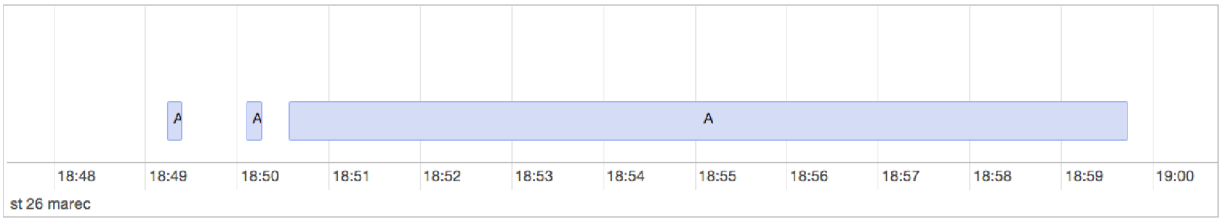
Je určená primárne pre zobrazovanie dát uložených v tabuľkách `registration`, `ip_geo` a `registration_activity` . Samotná sekcia sa člení na ďalšie 3 podsekcie.

Podsekcia „tabuľkový výpis"

Táto podsekcia zobrazuje zoznam registrácií SIP užívateľov vo forme tabuľky, ktorej podoba je na obrázku 3.11. Po rozkliknuti záznamu je zároveň zobrazená geografická poloha, z ktorej sa uživateľ registroval za pomoci Google Maps API. V prípade, že k registrácií bola zaznamenaná dostupnosť stanice protokolom SIP, tak sú tieto údaje zobrazované za pomoci časovej osy ako na obrázku 3.12. Popisok „Asignalizuje aktivitu stanice v inom prípade stanica bola neaktívna.

Užívateľ	Stav poslednej registrácie	SIP status code pos. registrácie	Zobrazované meno	Adresa	User agent	IP adresa	Port	Vytvorené	Posledná registrácia	Akcie
sip:tomas@my-lab	Úspešná	200 OK		Brno,Czech republic	YATE/4.0.1	147.229.192.182	5060	6.5.2014 11:20:01	6.5.2014 11:20:02	Zobraziť

Obrázek 3.11: Tabuľkový výpis - Registrácie



Obrázek 3.12: Časová osa zobrazujúca aktivitu SIP stanice

Podsekcia „aktivita SIP staníc"

Obsahuje tabuľkový výpis ako na obrázku 3.13 zobrazujúci informácie o tom, ktoré stanice sú/boli pripojené do VoIP siete a mohli za pomoci protokolu SIP komunikovať.

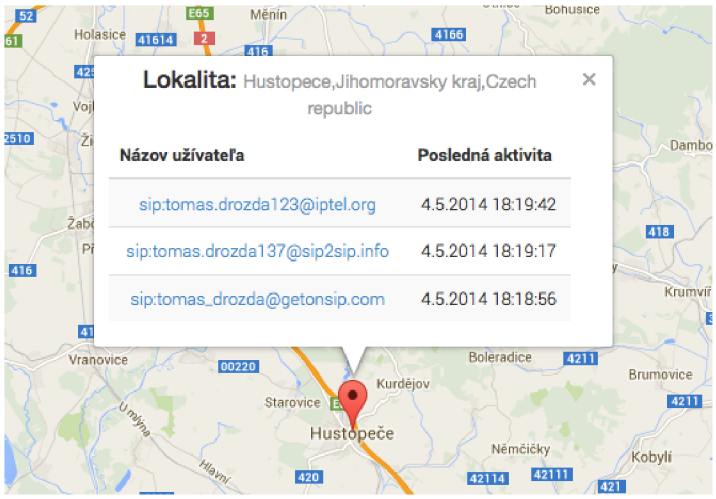
Registrácia	Od	Do
sip:tomas_drozda123@ippi.fr/147.229.192.182	8.4.2014 16:01:38	8.4.2014 16:07:38
sip:tomas_drozda1@getonsip.com/147.229.192.182	8.4.2014 16:01:38	8.4.2014 16:07:38

Obrázek 3.13: Tabuľkový výpis - Aktívne SIP stanice

Podsekcia „zobrazenie na mape"

V tejto sekcii je zobrazená mapa, na ktorej sa nachádzajú značky reprezentujúce rozpoznané polohy za pomoci IP geolokácie. Po kliknutí na značku je zobrazené dialógové okno, ktorého obsahom je tabuľkový výpis SIP užívateľov, ktorý komunikovali z danej polohy. Pri užívateľoch je zobrazený čas ich poslednej aktivity z danej polohy. Príklad zobrazovaného dialógového okna je možné vidieť na obrázku 3.14.

Obsluha tejto sekcie je realizovaná za pomoci kontroléru `Registration` a jej metódy `map`. Po načítaní stránky sú za pomoci Google Maps API zobrazené značky reprezentujúce rozpoznané polohy. Po kliknutí na značku je pomocou AJAXu načítaný zoznam užívateľov. Týmto riešením sa znížila doba odozvy po zadaní URL adresy na danej podstránke.

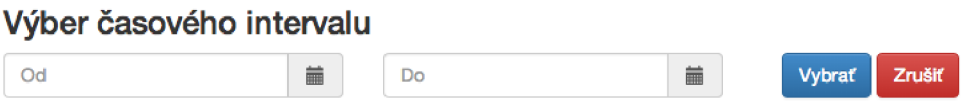


Obrázek 3.14: Zobrazenie staníc, ktoré komunikovali z uvedenej geografickej polohy.

Štatistiky

Sekcia je orientovaná na zobrazovanie štatistík niekoľkých druhov. Štatistiky sú vytvorené na základe informácií o zaznamenaných telefónnych hovorov. Môžme ich rozdeliť do 3 skupín a to: časové diagramy, grafy a sieťová mapa. Každá podsekcia obsahuje štatistiky jedného typu.

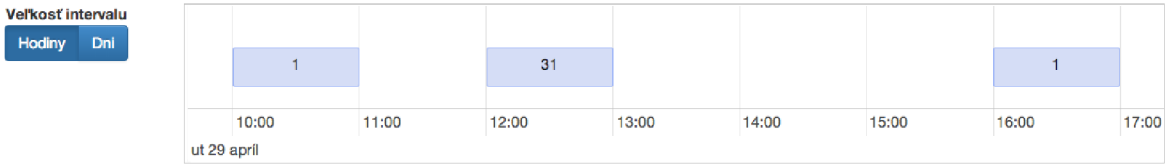
Spoločnou vlastnosťou podstránok štatistík je možnosť vybrať časový interval, pre ktorý budu štatistiky zobrazované. Výber časového intervalu je realizovaný za pomoci formuláru, ktorý je zobrazený na obrázku 3.15. Obsluha výberu časového intervalu je implementovaná pomocou AJAXu. Po kliknutí na tlačítko „Vybrať̈je zaslaný požiadavok za účelom získania dát pre štatistiku/štatistiky zobrazované na podstránke. Tento požiadavok je zaslaný radiču, ktorý je implementovaný triedou `Statistic` následne je obsluhovaný jednou z metód s prefixom `get_ajax_`. Následne sú dáta vrátene vo formáte JSON, ktorý obsahuje dáta, ktoré budu zobrazované a zároveň o stave uspokojenia požiadavku. V prípade, že požiadavok bol úspešne vykonaný tak je zavolaná jedna z implementovaných Javascript metód pre vykreslenie štatistík. Keď nie je vybraný časový interval tak štatistiky sú vytvárané zo všetkých údajov o úspešne ukončených telefónnych hovoroch.



Obrázek 3.15: Formulár - výber časového intervalu pre štatistiky

Podsekcia „množstvo telefónnych hovorov v čase”

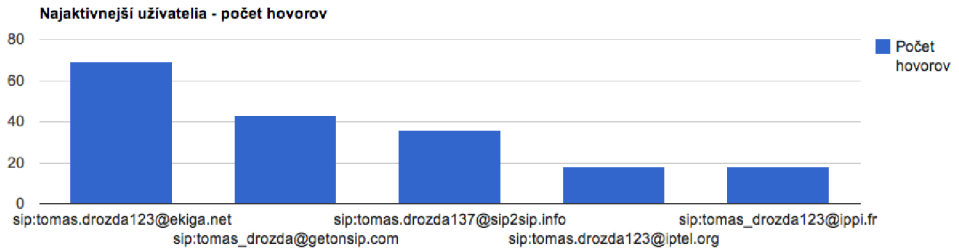
Zobrazuje časový diagram, na ktorom je zobrazený počet hovorov v jednotlivých časvých intervaloch. Veľkosť časových intervalov môže byť hodina alebo deň (predzvolené). Tento interval je možné meniť za pomoci 2 tlačitiek. Rekacia na zmenu intervalu je implementovaná pomocou AJAXu. Časový diagram je možné vidieť na obrázku 3.16.



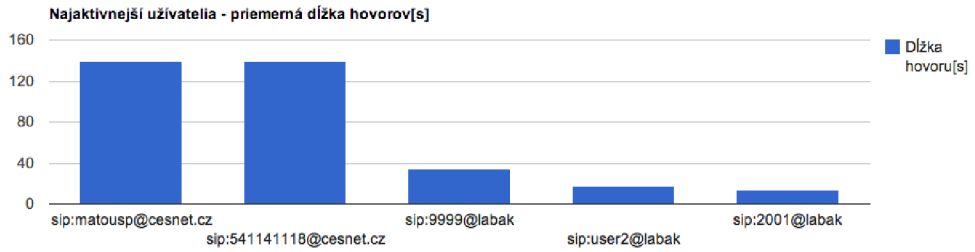
Obrázek 3.16: Časová osa zobrazujúca počet hovorov v čase

Podsekcia „Grafy"

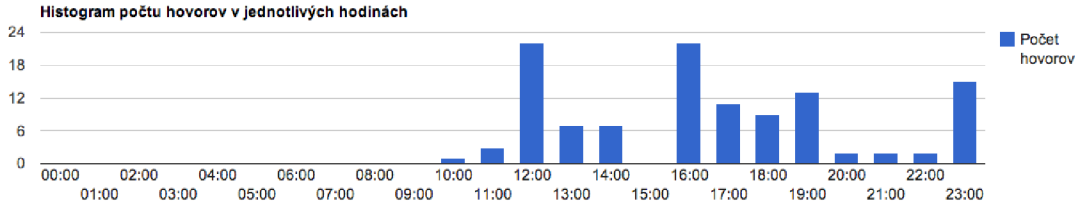
Táto podstránka sa zameriava na zobrazovanie celkovo 3 štatistík. Jedná sa o gafy hovoriace o najaktívnejších užívateľoch z pohľadu počtu hovorov (viď. obrázok 3.17), priemernej dĺžky hovorov (obr. 3.18). Posledná štatistika je tvorená histogramom, ktorý hovorí o počte hovorov v jednotlivých hodinách (obr. 3.19). Na základe posledne spomenutého diagramu je možné vidieť v akých hodinách užívatelia najviac vykonávajú telefónne hovory.



Obrázek 3.17: Najaktívnejší užívateľia z pohľadu počtu hovorov



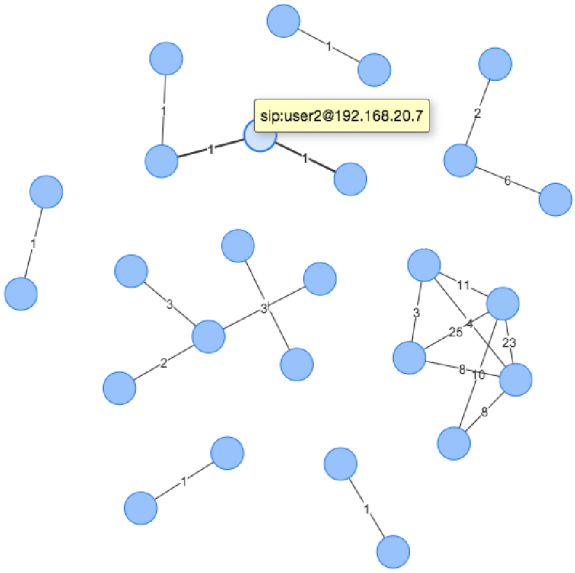
Obrázek 3.18: Najaktívnejší užívateľia na základe priemernej dĺžky hovorov



Obrázek 3.19: Histogram množstva hovorov v jednotlivých hodinách

Podsekcia „mapa telefónnych hovorov"

Na tejto podstránke je zobrazená sieťový graf (mapa), ktorej uzly predstavujú SIP užívateľov. Hrany predstavujú telefónne spojenia medzi užívateľmi a ich ohodnotenie informuje o počte vykonaných hovorov. Po kliknuti na jedného z užívateľov (uzol grafu) je zobrazený jeho popisok a su zvýraznené hrany smerujúce k užívateľom, s ktorými komunikoval. Túto mapu je možné vidieť na obrázku 3.20.



Obrázek 3.20: Sieťová mapa zbrazujúca hovory medzi užívateľmi

3.8 Spôsob nasadenia monitorvacieho systému

Blok `VoIP Database` bude realizovaný ako spoločné dátové uložisko MySQL pre ďalšie bloky.

Blok `SIP ANALYLEZER` bude nasadený do prevádzky priamo na servery, ktorý vykonáva funkciu telefónnej ústredne, pričom musí obsahovať vhodne nastavené parametre pre komunikáciu s databázovým serverom. Tato aplikácia bude spustená vo forme démona.

Bloky `SIP ACTIVITY TESTER a VISUALIZATION` je možné umiestnit na ľubovolné zariadenie pričom musí byť správne nastavené parametre pre pripojenie ku sploločnému databázovemu serveru.

3.9 Zhrnutie

Kapitola sa zaoberala návrhom a implementáciou monitorovacieho systému. V prvom kroku bola vykonaná analýza na základe, ktorej bol vytvorený ERD diagram pre popis systému ako sady dátových entit a vzťahov medzi nimi. Následne som dospel k záveru, že systém bude implementovaný ako sada celkovo 5 logických blokov označovaných ako `VoIP Database`, `SIP ANALYZER`, `SIP ACTIVITY TESTER`, `GEOLOCATION TOOL` a `VISUALIZATION`. Blok `VoIP` predstavuje MySQL databázu. Blok `SIP ANALYZER` je pasívny monitorovací blok, ktorýého úlohou je analýza správ signalizačného protokolu SIP. Na základe obsahu SIP správ blok `SIP ANALYZER` vytvára a upravuje databázové záznamy informujúce o činnosti užívateľov v

prostredí siete VoIP (registrácie, telefónne hovory...). Blok `SIP ACTIVITY TESTER` je aktívna monitorovacia komponenta, ktorá overuje či klientské stanice su schopné komunikovať protokolom SIP. `GEOLOCATION TOOL` blok vytvára požiadavky pre určenie geografickej polohy sieťových zariadení na základe ich IP adresy.V závere je predstavený spôsob akým bola implementovaná komponenta `VISUALIZATION`, ktorá je zodpovedná za zobrazovanie získaných dát. Vizualizačný blok je realizovaný ako webová aplikácia.

Kapitola 4

Testovanie systému

Kapitola je zameraná na testovania implementovaného systému. V prvom rade sa zameriame na testovanie spracovávania registračných požiadavkov a hovorov. Kapitolu zakončíme testovaním IP geolokačného bloku.

4.1 Spôsob testovania počas vývoja

Samotný systém bol priebežne testovaný počas celého vývoja a to tak, že na počítači, na ktorom prebiehal vývoj boli spustené jednotlivé bloky systému a zároveň niekoľko klientov komunikujúcich protokolom SIP. Jednalo sa konkrétne o softvérové telefóny X-Lite[1], YateClient[2], Jitsi[3] a Zoiper[4]. Následne boli vykonávané registrácie klientov a vytvárané telefónne hovory. Pri registráciach som sa zameral na overenie či systém dokáže zaznamenať registračné požiadavky (úspešna registrácia, potreba zadať prihlasovacie údaje...). V spojení s telefónnymi hovormi sa testovali reakcie systému v prípade zavedenia hovoru, jeho zloženia, odmietnutia a časté chybové stavy počas vytvárania hovoru (nepríhlasený klient, nesprávna konfigurácia smerovania hovorov).

4.2 Testovanie v prostredí sieťového laboratória

Pre potreby testovania v sieťovom laboratórií bolo vytvorené sieťové zapojenie podla topológie uvedenej na obrázku 4.1. Sieťová topológia obsahuje celkovo dve siete LAN s adresnými rozsahmi 192.168.10.0/24 a 192.168.20.0/24. Každá z týchto sieti predstavuje samostatnú VoIP sieť, ktorej činnosť je riadená za pomoci ústredne Asterisk zapojenej do danej siete. Ústredňa Asterisk bola vybraná na základe jej popularity v oblasti VoIP a jednoduchej konfigurácií. Klientské stanice reprezentovali oba typy telefónov (hardvérové, softvérové).

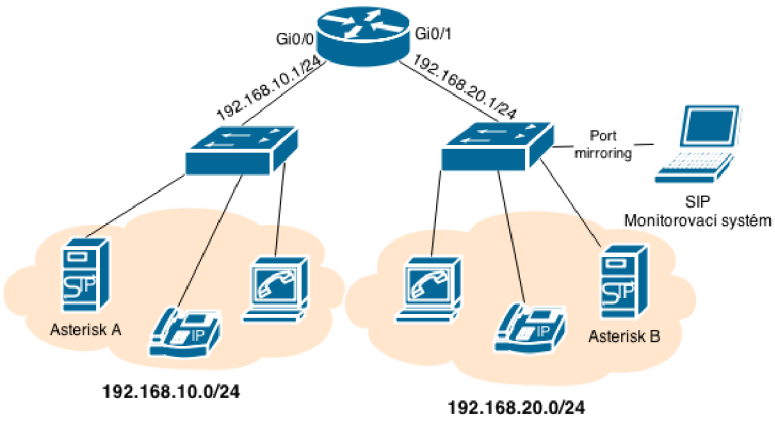
Monitorovací systém bol spustený na počítači, ktorý bol pripojený do jednej zo siete. Na sieťovom prepínači danej siete bol nastavený port mirroring za účelom preposielania sieťovej komunikácie monitorovacej stanici.

Blok `SIP ANALYZER` monitorovacieho systému bol spustený v logovaciom režime. Časti logovacích výstupov budu použité v tejto kapitole pre ilustráciu toho ako sa systém zachoval

[1] Viac informácií na: http://www.counterpath.com/x-lite.html
[2] Viac informácií na: http://yateclient.yate.ro/
[3] Viac informácií na: https://jitsi.org/
[4] Viac informácií na: http://www.zoiper.com/en

v skúmaných situáciach. Zároveň pri jednotlivých udalostiach sú doplnené ukážku z webovej aplikácie zobrazujúce informácie o týchto udalostiach.

V priebehu celej doby testovania bola sieťová komunikácia zaznamenávaná do súboru vo formáte pcap. Pre urýchlenie vykonávania testov boli v danom súbore ponechané len správy protokolu SIP.



Obrázek 4.1: Sieťová topológia pre testovanie

Registračné požiadavky

Pri testovaní reakcií monitorovacieho systému na príchod registračných požiadavkov a ich odpovedí som sa prednostne zameral na overenie či je systém dokáže informovať administrátora o nesprávne zadaných prihlasovacích údajoch používaných pre registrovanie klienta. Obrázok 4.2 obsahuje časť výpisu bloku `SIP ANALYZER` spustenom v logovaciom režime. Na uvedenom obrázku je možné vidieť, že užívateľ sa pokúšal o registráciu a následne mu bolo oznamené, že musí vyplniť prihlasovacie údaje. Potom užívateľ zaslal nový požiadavok pre registráciu so správnymi údajmi a bol úspešne registrovaný. Obrázok4.3 obsahuje tabuľkový výpis zobrazujúci údaje o spomínanej registrácií.

```
2014-04-29 10:47:44 -- Registration: sip:user1@labak/192.168.20.5 Want register
2014-04-29 10:47:44 -- Registration: sip:user1@labak/192.168.20.5 Authorization is required
2014-04-29 10:47:44 -- Registration: sip:user1@labak/192.168.20.5 Want register
2014-04-29 10:47:44 -- Registration: sip:user1@labak/192.168.20.5 Successfull
```

Obrázek 4.2: `SIP ANALYZER`- úspešná registrácia

Užívateľ	Stav poslednej registrácie	SIP status code pos. registrácie	Zobrazované meno	Adresa	User agent	IP adresa	Port	Vytvorené	Posledná registrácia	Akcie
sip:user1@labak	Úspešná	200 OK			SJphone/1.65.377a (SJ Labs)	192.168.20.5	5060	29.4.2014 11:01:44	18.5.2014 15:21:25	Zobraziť

Obrázek 4.3: `VISUALIZATION`-úspešná registrácia

Telefónne hovory

Pri testovani reakcií monitorovacieho systému na požiadavky pre vytvorenie telefónneho hovoru som sa zameral na nasledovné situácie:

- Hovor ktorý bol úspešne ukončený.
- Odmietnutý hovor.
- Chyba v smerovaní telefónnych hovorov.
- Neregistrovaný užívateľ sa pokúšal o vytvorenie telefónneho hovoru.

Obrázok 4.4 zobrazuje časť výpisu bkloku `SIP ANALYZER` v logovaciom režime kde je zachytená situácia keď užívateľovi *user1@192.168.10.6* sa podarilo úspešne vytvoriť telefónne spojenie s užívateľom *user2@192.168.20.7:5062*. Hovor bol úspešne ukončený o čom hovorí posledný riadok výpisu. Obrázok 4.5 ilustruje spôsob akým blok `VISUALIZATION` zobrazuje danú skutočnosť.

```
2014-04-29 12:45:18 -- Call: sip:user1@192.168.10.6 ---> sip:user2@192.168.20.7:5062 Request
2014-04-29 12:45:18 -- Call: sip:user1@192.168.10.6 ---> sip:user2@192.168.20.7:5062 Proceeding
2014-04-29 12:45:19 -- Call: sip:user1@192.168.10.6 ---> sip:user2@192.168.20.7:5062 Proceeding
2014-04-29 12:45:23 -- Call: sip:user1@192.168.10.6 ---> sip:user2@192.168.20.7:5062 In progress
2014-04-29 12:45:28 -- Call: sip:user1@192.168.10.6 ---> sip:user2@192.168.20.7:5062 Completed
```

Obrázek 4.4: `SIP ANALYZER`-hovor úspešne ukončený

Čas vytvorenia požiadavku	Stav požiadavku	SIP status code	Volajúca stanica	Volaná stanica	Začiatok hovoru	Koniec hovoru	Dĺžka hovoru	Možné kódeky	Audio porty(zdroj - cieľ)	Akcie
29.4.2014 12:45:18	Hovor ukončený	200 OK	sip:user1@192.168.10.6	sip:user2@192.168.20.7	29.4.2014 12:45:23	29.4.2014 12:45:28	00:00:05	PCMA	16580-11784	Zobraziť

Obrázek 4.5: `SVISUALIZATION`-hovor úspešne ukončený

Obrázok 4.6 zobrazuje situáciu kedy užívateľ *user2@labak* sa pokúšal uskutočniť telefónny hovor so stanicou s telefónným číslom *1001@labak*. V tomto prípade bol telefónny hovor odmietnutý.

```
2014-04-29 10:49:03 -- Call: sip:user2@192.168.20.3 ---> sip:user1@192.168.20.5 Request
2014-04-29 10:49:03 -- Call: sip:user2@192.168.20.3 ---> sip:user1@192.168.20.5 Proceeding
2014-04-29 10:49:03 -- Call: sip:user2@192.168.20.3 ---> sip:user1@192.168.20.5 Proceeding
2014-04-29 10:49:03 -- Call: sip:user2@labak ---> sip:2001@labak Proceeding
2014-04-29 10:49:13 -- Call: sip:user2@192.168.20.3 ---> sip:user1@192.168.20.5 Rejected
```

Obrázek 4.6: `SIP ANALYZER`-odmietnutý hovor

Čas vytvorenia požiadavku	Stav požiadavku	SIP status code	Volajúca stanica	Volaná stanica	Začiatok hovoru	Koniec hovoru	Dĺžka hovoru	Možné kódeky	Audio porty(zdroj - cieľ)	Akcie
29.4.2014 10:49:03	Hovor odmietnutý	487 Request Terminated	sip:user2@192.168.20.3	sip:user1@192.168.20.5				PCMA		Zobraziť

Obrázek 4.7: `VISUALIZATION`-odmietnutý hovor

Obrázok 4.8 zobrazuje situáciu keď užívateľ *user2@labak* sa pokúšal uskutočniť telefónny hovor s užívateľom *1001@labak*. Na základe vypisu je jasné, že nastala chyba v smerovaní požiadavku cieľovému užívateľovi.Hovorí o tom časť výpisu: „Client error SC:404:

```
2014-04-29 11:02:25 -- Call: sip:user2@labak ---> sip:1001@labak Request
2014-04-29 11:02:25 -- Call: sip:user2@labak ---> sip:1001@labak Client error SC:404
```

Obrázek 4.8: SIP ANALYZER-nesprávna konfigurácia smerovania

Čas vytvorenia požiadavku	Stav požiadavku	SIP status code	Volajúca stanica	Volaná stanica	Začiatok hovoru	Koniec hovoru	Dĺžka hovoru	Možné kódeky	Audio porty(zdroj - cieľ)	Akcie
29.4.2014 11:02:25	Chyba na strane klienta	404 Not Found	sip:user2@labak	sip:1001@labak				PCMA		Zobraziť

Obrázek 4.9: VISUALIZATION-nesprávna konfigurácia smerovania

Na obrázku 4.10 je zobrazená situácia keď užívateľ *user2@labak* sa pokúšal uskutočniť telefónny hovor s užívateľom *2001@labak* avšak užívateľ *user2@labak* nebol registrovaný do siete VoIP. Informuje o tom časť výpisu s obsahom „Client error SC:401“

```
2014-04-29 10:24:16 -- Call: sip:user2@labak ---> sip:2001@labak Request
2014-04-29 10:24:16 -- Call: sip:user2@labak ---> sip:2001@labak Client error SC:401
```

Obrázek 4.10: SIP ANALYZER-užívateľ nie je prihlásený a vytvára požiadavok pre telefónny hovor

Geolokácia

Testovanie činnosti geolokačného podsystému bolo realizované za pomoci vloženia záznamov s verejnými IP adresami, ktorých skutočnú polohu sme poznali, do tabuľky *ip_geo*. Následne bola spustená činnosť geolokačného podsystému.

Pre vytvorenie záznamov s IP adresami, ktorých skutočnú polohu poznáme môžme použiť nasledovný SQL dotaz.

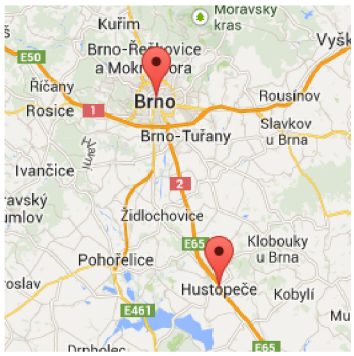
```
INSERT INTO ip_geo (ip_address) VALUES
                                ('80.90.130.206'), ('147.229.192.182');
```

Po zadaní uvedeného dotazu sú vytvorené 2 záznamy obsahujúce IP adresy, ktorých geografická poloha bude určená. Tabuľka 4.1 zobrazuje výsledky činnosti IP geolokačného podsystému.

IP adresa	Skutočná poloha	Zistená poloha
80.90.130.20	Horni Bojanovice, Jihomoravsky kraj, Czech republic	Hustopece, Jihomoravsky kraj,Czech republic
147.229.192.182	Purkynova 93, Brno, Jihomoravsky kraj, Czech republic	Brno, Jihomoravsky kraj, Czech republic

Tabulka 4.1: Výsledky testovania IP geolokácie na IP adresách so známou geografickou polohou.

Na základe uvedených údajov je vidieť, že geolokácia je pomerne presná pri uvedených adresách. Jedna sa o rozdiel o veľkosti pod hranicu 10km medzi skutočnou a zistenou polohou. Na obrázku 4.11 je uvedený spôsob akým sú tieto informácie vizualizované v prostredí webovej aplikácie.



Obrázek 4.11: `VISUALIZATION`-zobrazenie geolokačných informácií.

4.3 Zhrnutie

Kapitola sa venovala testovaniu implementovaného monitorovacieho systému. V prvej časti kapitoly bol predstavený spôsob akým systém bol testovaný počas vývoja.

Potom sú popísane situácie, na ktoré musel systém korektne zareagovať aby bolo možné system prehlásiť, že funguje podľa požiadavkov. V prípade registrácií sa testovala reakcia systému na to, že užívateľ zadal nesprávne prihlasovacie údaje. Pri telefónných hovoroch sa testovali situácie ako: úspešne ukončený, odmietnutý, chyba smerovania, snaha o vytvorenie hovoru neautorizovaným užívateľom.

Funkčnosť IP geolokácie bola testovaná na niekoľkých verejných IP adresách, ktorých skutočnú polohu dobre poznáme. Jednalo sa konkrétne o IP adresy z mesta Brno a obce nachádzajúcej sa približne 40km na juhovýchod od spomínaného mesta. Výsledky geolokácie hovoria o tom, že geolokácie je primerane presná, keďže rozdiel medzi skutočnou a zistenou polohou bol malý (pod vzdialenosť 10km).

Na základe vykonania uvedených testov môžme považovať systém za funkčný. Avšak je potrebné dodať, že systém bol testovaný len v prostredí sieťového laboratóriá a tak je v budúcnosti nutné systém otestovať pred ostrým nasadením na skutočnej sieti.

Kapitola 5

Záver

V úvode práce je predstavený signalizačný protokol SIP, ktorý je jedným z najpoužívanejších v oblasti VoIP. Boli tu popísané vlastnosti protokolu: formát správ, architektúra a spôsob akým sú vytvárané telefónne hovory a registrácie. Pri popise formátu správ protokolu sme sa zamerali na povinné hlavičiky, ktorých znalosť bola nutná pre samotnú implementáciu.

Jadro práce sa zameriava na návrh a implementáciu monitorovacieho systému. System bol navrhnutý ako celkovo 6 logických celkov: `VoIP Database` - MySQL databáza, `SIP ANALYZER` - pasívne monitorovanie, `SIP ACTIVITY TESTER` - aktívne monitorovanie (testovanie dostupnosti klientských staníc), `GEOLOCATION TOOL` - IP geolokácia a `VISUALIZATION` - zobrazenie zaznamenaných dát. Bol tu predstavený ERD diagram, ktorý zachycuje dátové entitiy vystupujúce v systéme a vzťahy medzi nimi. Medzi zásadné entity systému považujeme entity `Sip_User` - užívateľia, `Registration` - registračné požiadavky, `Call` - telefónne hovory a `IP_Geo` - geolokačné informácie. Na základe znalostí o protokole SIP bol systém implementovaný. Komponenty zodpovedné za monitorvanie boli naprogramované ako konzolové aplikácie, ktoré je možné spustiť v logovaciom režime, kedy sú na štandardný výstup zobrazované informácie o prebiehajúcej činnosti. Tieto záznamy je možné ďalej spracovávať. Vizualizačná časť systému je implementovaná vo forme webovej aplikácie s prehľadným užívateľským rozhraním, ktoré umožnuje vykonávať všetky potrebné úkony. Samotná vizualizácia je tvorená tabuľkovými výpismi a detailným zobrazením záznamov doplnená o zobrazenie geolokačných dát pomocou Google Maps API a štatistik.

Testovanie a overovanie funkčnosti implementovaného monitorovacieho systému prebiehalo počas samotného vývoja. V závere vývoja bol systém potom otestovaný v sieťovom laboratóriu na vopred pripravenej topológií. Na základe vykonaných testov je možné, že systém úspešne splňuje požiadavky. V budúcnosti je nutné aby systém bol spustený na reálnej sieti v testovacej prevádze, čím bude overená činnosť systému pri omnoho vyššej sieťovej zátaži ako v prípade sieťového laboratória.

Dôležitou výhodou implementovaného systému oproti konkurencii je jednoduchosť grafického užívateľského rozhrania, ktoré umožnuje pohodlne pracovať s dátami. Na rozdiel od iných systémov, vytvorený systém uchováva informácie o tom v akých časových intervaloch boli klientské stanice pripojené do siete. Impelementácia IP geolokácie je ďalšiou výhodou oproti konkurenčným riešeniam. Naopak medzi nedostatky implementovaného systému možno považovať narastajúcu veľkosť dátových štruktúr triedy `SipAnalyzer`, ktoré obsahujú objekty tried `Registration` a `SipCall`. Jedným z možných riešení je použitie NoSQL databáze za pomoci čoho sa aplikácia odľahčí. V budúcnosti by bolo vhodné implementovať rozšírenie, ktoré by ku prebiehajúcim hovorom počítalo ukazateľe kvality telefónneho hovoru.

Literatura

[1] Handley, M.; Jacobson, V.: RFC 4566: SDP: Session Description Protocol. RFC 4566, Červenec 2006.

[2] Rosenberg, J.; Schulzrinne, H.; Camarillo, G.; aj.: *RFC 3261: SIP: Session Initiation Protocol*. IETF, Červen 2002.

[3] Schulzrinne, H.; Casner, S.; Frederick, R.; aj.: RFC 3550: RTP: A Transport Protocol for Real-Time Applications. RFC 3550, Červenec 2003.

Příloha A

Obsah CD

- /system/-zdrojové súbory monitorovacieho systému
- /system/SIP ANALYZER/- zdrojové súbory bloku `SIP ANALYZER`
- /system/SIP ACTIVITY TESTER/- zdrojové súbory bloku `SIP ACTIVITY TESTER`
- /system/GEOLOCATION TOOL/- zdrojové súbory bloku `GEOLOCATION TOOL`
- /system/VISUALIZATION/- zdrojové súbory bloku `VISUALIZATION`
- /system/voip_d.sql- SQL skript pre vytvorenie databáze
- /pcaps/- pcap súbory pre testovanie
- /doc/-zdrojové kódy technickej správy v jazyku LaTex
- /doc/projekt.pdf- text technickej správy vo formáte PDF

Příloha B

Softvérové požiadavky na systém

V tejto prílohe sú uvedené softvérové požiadavky pre spustenie monitorovacieho systému.

- Operačný systém- ľubovolný unixový systém (vyvíjané a testované na OS X 10.9 a Debian).
- Webový server Apache s aktivovaným modulom `mod_rewrite` a vypnutý `safe_mode`.
- PHP interpret vo verzii 5.3 a vyššie.
- Databáza MySQL verzie 5.1 a vyššie.
- libmysql-dev (knižnica pre C/C++, potrebná pre inštaláciu MySQLDb).
- Interpret jazyka Python verzie 2.7. s nainštalovanými knižnicami Scapy[1] a MySQLDb[2].

[1] Návod: http://www.secdev.org/projects/scapy/doc/installation.html#installing-scapy-v2-x
[2] Návod: http://sourceforge.net/projects/mysql-python/files/mysql-python/1.2.3/

Příloha C

Užívateľská príručka

Kapitola obsahuje popis konfigurácií, ktoré je potrebné vykonať pred spustením jednotlivých blokov. Následne kapitola popisuje spôsob ako spustiť jednotlivé bloky systému.

C.1 Konfigurácia blokov systému

Pre spustenie monitorovacieho systému je potrebné vykonať niekoľko nastavení.

Prístup k DB

V prípade blokov reprezentujúcich konzolové aplikácie je táto operácia vykonávaná zadaním správnych údajov v konfiguračnom súbore `database.json`, ktorý je obsiahnutý v adresáry `config` každého z blokov. Príklad obsahu konfiguračného súboru:

```
{
"hostname"  : "localhost",
"username"  : "root",
"password"  : "",
"database"   : "voip_db2"
}
```

V prípade bloku `VISUALIZATION` je táto operácia realizovaná za pomoci zadania správnych parametrov v konfiguračnom súbore na ceste: /system/VISUALIZATION/application/config/database.ph
Význam jednotlivých konfiguračných položiek je uvedený v danom súbore.

Blok `VISUALIZATION` - obsah súboru .htaccess

Pred prístupom k webovej aplikácií je potrebné zmeniť obsah súboru `.htaccess`, ktorý je umiestnený v koreni vizualizčnej časti systému. Konkrétne je potrebné zmeniť hodnotu riadku s obsahom `RewriteBase`.

C.2 Spustenie blokov systému

Konzolové aplikácie majú niekoľko spoločných parametrov príkazového riadku:

- `-h` alebo `--help` : parameter pre zobrazenie nápovedy
- `-l` : spustenie bloku v logovaciom režime. Tento parameter je možné kombinovať s ďalšími parametrami.

Blok `SIP ANALYZER`

- `sudo python2.7 ./main.py`
  Spustenie v režime real-time analýzy.
- `sudo python2.7 ./main.py --input=PCAP_FILENAME`
  Analýza obsahu súbora s názvom *PCAP_FILENAME*.

Blok `SIP ACTIVITY TESTER`

- `python2.7 ./sip_activity_tester.py`
- `python2.7 ./sip_activity_tester.py --timeout=VALUE`
  VALUE- časový interval pre kontrolu dostupnosti stanice
- `python2.7 ./sip_activity_tester.py --interval=VALUE`
  VALUE- časový interval pre príjem odpovedi na správu pre kontrolu dostupnosti
- Parametre `--interval` a `--timeout` je možné kombinovať.

Blok `GEOLOCATION TOOL`

- `python2.7 ./ip_geolocation_tool.py`
  Spustenie IP geolokačného bloku.

Blok `VISUALIZATION`

Systém obsahuje predvolený užívateľský účet:

- `prihlasovacie meno`: admin
- `prihlasovacie heslo`: admin_monitor_123