

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2016

Bc. Nikola Papež



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

**IMPLEMENTACE PROTOKOLU SIP V OPEN SOURCE PBX  
A JEJICH TESTOVÁNÍ**

TESTING OF SIP IMPLEMENTATIONS IN OPEN SOURCE PBX'S

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. Nikola Papež**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Pavel Šilhavý, Ph.D.**

**BRNO 2016**



# Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

**Student:** Bc. Nikola Papež

**ID:** 129359

**Ročník:** 2

**Akademický rok:** 2015/16

**NÁZEV TÉMATU:**

## Implementace protokolu SIP v open Source PBX a jejich testování

### POKYNY PRO VYPRACOVÁNÍ:

Realizujte porovnání implementací protokolu SIP (SIP stacks). Zaměřte se, mimo jiné, na nativní implementaci v PBX Asterisk, projekty Sofia-sip, PJSIP a SIP stack v PBX Kamailio a OpenSER. Realizujte testování výkonosti, stability a odolnosti vůči útokům. Porovnejte výkonnostní, paměťové nároky a nároky na další prostředky systému při zátěži. Srovnajte rovněž různé konfigurace, u nichž lze předpokládat vliv na výkonnost.

### DOPORUČENÁ LITERATURA:

[1] BOSSE, J.G.. Signaling in telecommunication networks. John Wiley & Sons, Ltd. En-gland 2002 , ISBN 0-47-66288-7.

[2] COLLINS, D.. Carrier grade voice over IP / 2nd ed. New York : McGraw-Hill, 2003. ISBN 0-07-140634-4.

[3] DWIVEDI, HIMANSHU. Hacking VoIP :protocols, attacks, and countermeasures / San Francisco : No Starch Press, 2009. ISBN 978-1-59327-163-3.

[4] ENDLER, D.. Hacking exposed VoIP :voice over IP security secrets & solutions / New York : McGraw-Hill, 2007. ISBN 978-0-07-226364-0.

**Termín zadání:** 1.2.2016

**Termín odevzdání:** 25.5.2016

**Vedoucí práce:** Ing. Pavel Šilhavý, Ph.D.

**Konzultant diplomové práce:**

**doc. Ing. Jiří Mišurec, CSc., předseda oborové rady**

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Tato práce zkoumá a porovnává několik vybraných knihoven protokolu SIP, výkon, stabilitu, bezpečnost a vliv na jejich konfiguraci. Na začátku jsou stručně vyjmenovány hlavní funkce signalizačního protokolu. Následují kapitoly popisující testované ústředny a je teoreticky srovnáno několik stacků implementujících SIP protokol. V praktické části probíhalo měření na zátěžovém generátoru Spirent TestCenter C1, pomocí kterého se vykonávaly veškeré testy na ústřednách. Zmiňované SIP knihovny, PBX i operační systém, na kterém byly ústředny provozovány jsou open source.

## **KLÍČOVÁ SLOVA**

SIP, PBX, Asterisk, Freeswitch, Kamailio, Spirent, testování, porovnání, výkon

## **ABSTRACT**

This diploma thesis examines and compares several selected libraries of SIP protocol, performance, stability, security and impact of their configuration. The main functions of the signalling protocol are briefly named at the beginning. The following chapters describe the tested PBXs and several stacks for SIP protocol are theoretically compared. The practical part deals with measurements conducted on the load generator Spirent TestCenter C1 which is used for all the performed tests on exchanges. All the mentioned SIP libraries, PBXs and the operating system on which the PBXs were running are open source software.

## **KEYWORDS**

SIP, PBX, Asterisk, Freeswitch, Kamailio, Spirent, testing, comparison, performance

PAPEŽ, Nikola *Implementace protokolu SIP v open source PBX a jejich testování*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 99 s. Vedoucí práce byl Ing. Pavel Šilhavý, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Implementace protokolu SIP v open source PBX a jejich testování“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Pavlovi Šilhavému, Ph.D. za odborné vedení, konzultace, podnětné návrhy k práci a také za jeho trpělivost a ochotu. Dále bych chtěl poděkovat svému synu Eliasovi za to, že mě vždy při práci dokázal povzbudit, i když ještě pořádně ani neumí mluvit. Neposledně také překrásné Lucii za její psychickou i fyzickou podporu.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno .....

.....

podpis autora

# OBSAH

|  |           |
|--|-----------|
| <b>Úvod</b>  | <b>15</b> |
| <b>1 Obecné vlastnosti protokolu pro inicializaci relací SIP</b> | <b>17</b> |
| <b>2 Open source pobočkové ústředny PBX</b>                      | <b>21</b> |
| 2.1 Pobočková ústředna Asterisk . . . . .                        | 21        |
| 2.2 Pobočková ústředna FreeSWITCH . . . . .                      | 21        |
| 2.3 Proxy ústředna OpenSER a Kamailio . . . . .                  | 22        |
| <b>3 Knihovny implementující protokol SIP</b>                    | <b>23</b> |
| 3.1 Výčet nejpoužívanějších SIP stacků . . . . .                 | 23        |
| 3.1.1 PJSIP stack . . . . .                                      | 23        |
| 3.1.2 Sofia-sip stack . . . . .                                  | 24        |
| 3.1.3 ReSIProcate stack . . . . .                                | 24        |
| 3.1.4 oSIP stack . . . . .                                       | 24        |
| 3.1.5 OPAL stack . . . . .                                       | 25        |
| 3.1.6 VOCAL stack . . . . .                                      | 25        |
| 3.2 Porovnání SIP stacků . . . . .                               | 25        |
| <b>4 Praktické testy</b>   | <b>26</b> |
| 4.1 Systém a jeho příprava . . . . .                             | 28        |
| 4.1.1 Výběr a konfigurace operačního systému . . . . .           | 29        |
| 4.1.2 Skript pro shromažďování dat . . . . .                     | 31        |
| 4.1.3 Konfigurace generátoru zátěže . . . . .                    | 32        |
| 4.2 Instalace a konfigurace PBX ústředny . . . . .               | 34        |
| 4.2.1 Instalace ústředny Asterisk . . . . .                      | 34        |



|       |   |    |
|-------|---|----|
| 4.2.2 | Instalace ústředny Freeswitch . . . . .                   | 38 |
| 4.2.3 | Instalace ústředny OpenSER . . . . .                      | 38 |
| 4.2.4 | Instalace ústředny Kamailio . . . . .                     | 40 |
| 4.3   | Měření maximálního počtu relací . . . . .                 | 42 |
| 4.3.1 | Výkon a stabilita ústředny Asterisk 11 . . . . .          | 42 |
| 4.3.2 | Výkon a stabilita ústředny Asterisk 13 . . . . .          | 46 |
| 4.3.3 | Výkon a stabilita ústředny Freeswitch . . . . .           | 47 |
| 4.3.4 | Výkon a stabilita ústředny OpenSER . . . . .              | 47 |
| 4.3.5 | Výkon a stabilita ústředny Kamailio . . . . .             | 48 |
| 4.4   | Měření maximálního počtu transakcí . . . . .              | 49 |
| 4.4.1 | Měření transakcí ústředny Asterisk 11 . . . . .           | 49 |
| 4.4.2 | Měření transakcí ústředny Asterisk 13 . . . . .           | 53 |
| 4.4.3 | Měření transakcí ústředny FreeSWITCH . . . . .            | 53 |
| 4.4.4 | Měření transakcí ústředny OpenSER . . . . .               | 53 |
| 4.4.5 | Měření transakcí ústředny Kamailio . . . . .              | 54 |
| 4.5   | Měření s jinými kodeky . . . . .                          | 55 |
| 4.5.1 | Chování ústředny Asterisk 11 s odlišnými kodeky . . . . . | 55 |
| 4.5.2 | Chování ústředny Asterisk 13 s odlišnými kodeky . . . . . | 55 |
| 4.5.3 | Chování ústředny FreeSWITCH s odlišnými kodeky . . . . .  | 56 |
| 4.6   | Měření s modifikovanými zprávami . . . . .                | 58 |
| 4.6.1 | Starší INVITE žádost . . . . .                            | 58 |
| 4.6.2 | Více hodnot v položce Content-Length . . . . .            | 59 |
| 4.6.3 | Odpověď neznámého typu . . . . .                          | 59 |
| 4.6.4 | Mezery v poli příjemce . . . . .                          | 60 |
| 4.6.5 | Neznámá metoda . . . . .                                  | 60 |

|          |  |           |
|----------|--|-----------|
| 4.6.6    | Různé typy transportu zprávy . . . . .                     | 61        |
| 4.6.7    | Nulová hodnota položky Max-Forwards . . . . .              | 62        |
| 4.6.8    | Špatně interpretovaný parametr v poli Contact . . . . .    | 62        |
| 4.6.9    | Chybějící identifikátor transakce . . . . .                | 62        |
| 4.6.10   | Žádost se špatnou adresou . . . . .                        | 63        |
| 4.6.11   | Nesprávná hodnota velikosti těla zprávy . . . . .          | 64        |
| 4.6.12   | Odstraněná mezera mezi jménem a adresou . . . . .          | 64        |
| 4.6.13   | Parametry oddělené středníkem v poli Request-URI . . . . . | 65        |
| 4.6.14   | Neuzavřený parametr . . . . .                              | 65        |
| 4.6.15   | Chybné pole Accept . . . . .                               | 66        |
| 4.7      | Celkové srovnání výsledků . . . . .                        | 67        |
| <b>5</b> | <b>Závěr</b>   | <b>70</b> |
|          | <b>Literatura</b>  | <b>72</b> |
|          | <b>Seznam zkratk</b>                                       | <b>75</b> |
|          | <b>Seznam příloh</b>                                       | <b>78</b> |
| <b>A</b> | <b>Přílohy</b>   | <b>79</b> |
| A.1      | Konfigurační soubory ústředn . . . . .                     | 79        |
| A.1.1    | Konfigurační soubory ústředny Asterisk 11 . . . . .        | 79        |
| A.1.2    | Konfigurační soubory ústředny Asterisk 13 . . . . .        | 79        |
| A.1.3    | Konfigurační soubory ústředny Freeswitch . . . . .         | 81        |
| A.1.4    | Konfigurační soubory ústředny OpenSER . . . . .            | 82        |
| A.2      | Skript pro měření výkonu . . . . .                         | 83        |
| A.3      | Grafy . . . . .  | 84        |

|          |                                     |           |
|----------|-------------------------------------|-----------|
| A.3.1    | Měření relací . . . . .             | 84        |
| A.3.2    | Měření transakcí . . . . .          | 89        |
| A.3.3    | Měření s odlišnými kodeky . . . . . | 92        |
| A.4      | Tabulky . . . . .                   | 93        |
| <b>B</b> | <b>Obsah přiloženého CD</b>         | <b>99</b> |

## SEZNAM OBRÁZKŮ

|      |  |    |
|------|--|----|
| 4.1  | Komunikace v proxy režimu . . . . .  | 27 |
| 4.2  | Analýza proxy režimu . . . . .   | 27 |
| 4.3  | Komunikace jako B2BUA . . . . .  | 28 |
| 4.4  | Analýza B2BUA režimu . . . . .   | 28 |
| 4.5  | Ukázka spuštěného měřicího skriptu . . . . .   | 31 |
| 4.6  | Generátor zátěže Spirent . . . . .   | 32 |
| 4.7  | Ukázka rozhraní aplikace Avalanche Commander . . . . .   | 32 |
| 4.8  | Požadované vytížení při simulaci . . . . .   | 33 |
| 4.9  | Změřené RTP toky ústředny Asterisk 11 v režimu B2BUA s 1536 MB<br>RAM . . . . .  | 43 |
| 4.10 | Změřený počet relací ústředny Asterisk 11 v režimu B2BUA s 1536<br>MB RAM . . . . .  | 43 |
| 4.11 | Změřené RTP toky ústředny Asterisk 11 v režimu proxy s 1536 MB<br>RAM . . . . .  | 44 |
| 4.12 | Změřený počet relací ústředny Asterisk 11 v režimu proxy s 1536 MB<br>RAM . . . . .  | 44 |
| 4.13 | Vytížení procesoru při simulaci počtu relací u ústředny Asterisk 11,<br>v režimu B2BUA a proxy s 1536 MB RAM . . . . .       | 45 |
| 4.14 | Zaplnění operační paměti při simulaci počtu relací u ústředny Asterisk<br>11, v režimu B2BUA a proxy s 1536 MB RAM . . . . . | 45 |
| 4.15 | Využití souborů při simulaci počtu relací u ústředny Asterisk 11, v re-<br>žimu B2BUA a proxy s 1536 MB RAM . . . . .        | 46 |
| 4.16 | Změřené transakce ústředny Asterisk 11 v režimu B2BUA s 1536 MB<br>RAM . . . . .   | 50 |
| 4.17 | Výpis z konzole Asterisk 11 . . . . .  | 50 |
| 4.18 | Změřené transakce ústředny Asterisk 11 v režimu proxy s 1536 MB<br>RAM . . . . .   | 51 |

|      |  |    |
|------|--|----|
| 4.19 | Vytížení procesoru při simulaci počtu transakcí u ústředny Asterisk 11, v režimu B2BUA a proxy s 1536 MB RAM . . . . .       | 51 |
| 4.20 | Zaplnění operační paměti při simulaci počtu transakcí u ústředny Asterisk 11, v režimu B2BUA a proxy s 1536 MB RAM . . . . . | 52 |
| 4.21 | Využití souborů při simulaci počtu transakcí u ústředny Asterisk 11, v režimu B2BUA a proxy s 1536 MB RAM . . . . .          | 52 |
| 4.22 | Maximální dosažené relace v porovnání s různými kodeky s 1536MB operační pamětí . . . . .                                    | 56 |
| 4.23 | Maximální toky dat přes ústředny s 1536MB operační pamětí a různými kodeky . . . . .   | 57 |
| 4.24 | Maximální dosažené transakce ústředen v režimu proxy . . . . .   | 67 |
| 4.25 | Maximální dosažené transakce ústředen v režimu B2BUA . . . . .   | 68 |
| 4.26 | Maximální dosažené relace ústředen v režimu B2BUA . . . . .  | 68 |
| 4.27 | Maximální dosažené relace ústředen v režimu proxy . . . . .  | 69 |
| A.1  | Změřené RTP toky ústředny Asterisk 13 v režimu B2BUA s 1536 MB RAM . . . . .   | 84 |
| A.2  | Změřený počet relací ústředny Asterisk 13 v režimu B2BUA s 1536 MB RAM . . . . .   | 85 |
| A.3  | Změřené RTP toky ústředny FreeSWITCH v režimu B2BUA s 1536 MB RAM . . . . .  | 85 |
| A.4  | Změřený počet relací ústředny FreeSWITCH v režimu B2BUA s 1536 MB RAM . . . . .  | 86 |
| A.5  | Změřené RTP toky ústředny FreeSWITCH v režimu proxy s 1536 MB RAM . . . . .  | 86 |
| A.6  | Změřený počet relací ústředny FreeSWITCH v režimu proxy s 1536 MB RAM . . . . .  | 87 |
| A.7  | Změřené RTP toky ústředny OpenSER v režimu proxy s 1536 MB RAM . . . . .   | 87 |
| A.8  | Změřený počet relací ústředny OpenSER v režimu proxy s 1536 MB RAM . . . . .   | 88 |

|      |   |    |
|------|---|----|
| A.9  | Změřené RTP toky ústředny Kamailio v režimu proxy s 1536 MB RAM                             | 88 |
| A.10 | Změřený počet relací ústředny Kamailio v režimu proxy s 1536 MB<br>RAM . . . . .            | 89 |
| A.11 | Změřené transakce ústředny Asterisk 13 v režimu B2BUA s 1536 MB<br>RAM . . . . .            | 89 |
| A.12 | Změřené transakce ústředny FreeSWITCH v režimu B2BUA s 1536<br>MB RAM . . . . .             | 90 |
| A.13 | Změřené transakce ústředny FreeSWITCH v režimu proxy s 1536 MB<br>RAM . . . . .             | 90 |
| A.14 | Změřené transakce ústředny OpenSER v režimu proxy s 1536 MB RAM                             | 91 |
| A.15 | Změřené transakce ústředny Kamailio v režimu proxy s 768 MB RAM                             | 91 |
| A.16 | Maximální dosažené relace v porovnání s různými kodeky s 768MB<br>operační pamětí . . . . . | 92 |
| A.17 | Maximální toky dat přes ústředny s 768MB operační pamětí a různými<br>kodeky . . . . .      | 92 |

# SEZNAM TABULEK

|     |  |    |
|-----|--|----|
| A.1 | Srovnání nejpodstatnějších vlastností stacků SIP . . . . .             | 93 |
| A.2 | Shrnutí měření dosažených relací a transakcí . . . . .                 | 94 |
| A.3 | Shrnutí naměřeného výkonu na ústřednách při měření relací . . . . .    | 95 |
| A.4 | Shrnutí naměřeného výkonu na ústřednách při měření transakcí . . . . . | 96 |
| A.5 | Shrnutí naměřeného výkonu na ústřednách při měření kodeků . . . . .    | 97 |
| A.6 | Seznam všech odpovědí na modifikované zprávy . . . . .                 | 98 |

# ÚVOD

Koncem minulého století se v Internetu a podnikových sítích začalo velmi úspěšně využívat protokolu IP. Následkem toho došlo ke konvergenci datových a jiných komunikačních sítí do jedné univerzální infrastruktury. Vzhledem k jednoduchosti a flexibilitě této konvergované sítě vzrostla i oblíbenost ústředen 5. generace používajících technologii VoIP. Pobočkové ústředny PBX této generace využívají komutaci hovorového signálu pomocí paketů. Velkou výhodou daného řešení oproti klasické telefonní síti je právě finanční a ekonomická nenáročnost a ve většině případů možnost využití stávajících prvků v datových sítích. Pro komunikaci mezi dvěma účastníky se zde používá řada protokolů. Jeden z nejčastějších protokolů v internetové telefonii je SIP.

U takových ústředen z hlediska žádanosti rostou také nároky a požadavky na jejich výkon. Ten se dá ovlivnit nejen správnou aplikací ústředny do konkrétního prostředí či výkonými hardwarovými komponenty, ale i výběrem vhodné knihovny pro danou ústřednu. Cílem této diplomové práce je tedy srovnání výkonosti a odolnosti knihoven právě pro protokol SIP, jinak také řečeno stacků.

Práce srovnává stacky, které jsou již implementovány v ústřednách anebo které je možné do ústředny doinstalovat jako zásuvný modul a to při různých konfiguracích. Jak stacky, tak i ústředny byly vybrány pouze s otevřenou, volně šiřitelnou licencí. Pro lepší orientaci v práci jsou ze začátku zmíněny obecné vlastnosti protokolu SIP a metody jeho komunikace. Další kapitola popisuje open source ústředny, které byly použité pro testování jejich SIP knihoven. Konkrétně se tedy jedná o testování nativního stacku *chan\_sip* nacházejícího se v ústředně Asterisk 11, v její vyšší verzi Asterisk 13 to byl pak *PJSIP* doinstalovaný jako zásuvný modul. Dále pak byla testovaná knihovna *sofia-sip* implementovaná v ústředně FreeSWITCH a stacky ústředen OpenSER a jejího nástupce Kamailio. Dle dalších vlastností pak byly teoreticky srovnány i knihovny nebo sady knihoven, které nejsou přímo obsaženy v ústřednách. Jsou to například *reSIP*, *oSIP*, *OPAL* či *VOCAL*.

Praktická část popisuje podrobnou přípravu operačního systému, instalaci jednotlivých ústředen, jejich stacků, popřípadě i instalaci kodeku a vlastní konfiguraci. Je zde také zmíněn vlastní jednoduchý skript, co dopomáhal při měření některých parametrů na systému, který byl testován. Následuje pak měření s použitím testeru Spirent TestCenter C1. Pomocí testeru byly provedeny zátěžové testy, které dle nastavené charakteristiky simulovaly rostoucí počet hovorů tekoucích přes testovanou ústřednu. Tyto relace vyřizovala právě daná knihovna SIP nacházející se v ústředně.



Porovnávalo se také, jak se chovají ústředny s odlišnými audio kodeky v režimu *pass-through*<sup>1</sup> a jestli má jejich změna na ně nějaký vliv. Měřilo se na třech typech kodeků – G.711, G723 a G.729.

S pomocí testeru byly vytvářeny i upravené zprávy protokolu SIP, které měly ověřit, jak se daný stack chová, když dostane nestandardní požadavek ze sítě. Struktura modifikovaných zpráv se držela dle doporučení RFC 4475. Takové upravené požadavky totiž mohou v některých případech i výrazně ovlivnit běh ústředny. Dá se tedy mluvit o útocích na bezpečnost. Bylo vybráno patnáct odlišných útoků pro ověření stability a odolnosti ústředny. To, jak určitá ústředna bude reagovat má na starosti právě konkrétní SIP stack.

Na základě naměřených hodnot a nasbíraných dat pak bylo každé měření náležitě okomentováno. Následně i pro větší přehlednost byly výsledky z měření mezi sebou z hlediska výkonu srovnány a zdůrazněny jejich výhody a nevýhody, popřípadě doporučené využití a konfigurace těchto ústředen.

---

<sup>1</sup>pass-through – režim, kdy volaný i volající využívají stejného kodeku, tento kodek je nastaven i v ústředně. Nedochází tedy k transkódování a dalšímu vytížení ústředny, co navíc může vyžadovat i dodatečné licence ke kodeku. Označováno také i jako *pass-thru*.

# 1 OBECNÉ VLASTNOSTI PROTOKOLU PRO INICIALIZACI RELACÍ SIP

Protokol SIP je poměrně často užívaným protokolem v oblasti VoIP telefonie. Definován IETF a popsán dle RFC, jeho první verze SIP 1.0 byla standardizovaná roku 1999 dle RFC 2543. Aktuálně od roku 2002 je ve verzi SIP 2.0 dle RFC 3261. Svou oblíbenost si získal hlavně díky jeho jednoduchosti, například oproti také velice populárnímu, avšak komplexnějšímu protokolu H.323. Ten totiž využívá signalizace v binární formě a na rozdíl od SIP protokolu je pro člověka hůře čitelná. Může být využit jak v klasickém hardwarovém telefonu, tak i v softwarovém, jako je například Skype, Google Talk nebo jako nativní telefonní klient v OS Android [3, 5, 6, 7, 22, 23].

Typické řešení VoIP sítě, která využívá signalizačního protokolu SIP se skládá z klienta User Agent a tří typů serveru - registrar, redirect a proxy server, kde tento protokol obvykle naslouchá na portu 5060 TCP nebo 5060 UDP [3, 18, 25].

- **User Agent** – je softwarový nebo hardwarový klient, který se stará o iniciaci a příjem hovorů. V praxi jde o fyzický telefon nebo softwarovou aplikaci. User Agent se dělí také na UAC, což je klient, který vytváří spojení s druhou stranou a UAS, server, který na dané příchozí požadavky reaguje.
- **Registrar server** – jak název napovídá, umožňuje registraci klienta a také slouží i k jejich autentizaci a adresaci.
- **Redirect server** – protokol po něm vyžaduje či na něj zasílá adresy, které by měly být kontaktovány pro dokončení počáteční žádosti (v případě, že se klienti nachází v jiné síti)
- **Proxy server** – přeposílá a směřuje klientská data a žádosti na základě informací ze serveru registrar. Může se také podílet na autentizaci a je často využíván pro centralizaci VoIP paketů na síti. Velmi často se stává, že registrar a proxy server jsou integrovány do jednoho zařízení. Pro navázání komunikace proxy server není nutný.

Protokol využívá spojení typu klient-server. Jeho stavba SIP je podobná jako u HTTP. Oba se skládají z různých požadavků pro zavolání specifických akcí. Tyto požadavky se jinak také nazývají metody. Prvních šest metod je základních a počátečně definovaných dle doporučení RFC 3261, další zbylé metody byly později přidávány. Jejich výčet je vypsán níže [5, 9, 22, 23].

- **ACK** – z anglického *acknowledge*<sup>1</sup> potvrzuje, že zpráva odeslaná druhou stra-

---

<sup>1</sup>acknowledge – potvrzení

nou byla úspěšně přijata. Obvykle přijetí této zprávy znamená dokončený *handshake*<sup>2</sup> mezi klienty a následnou inicializaci hovoru.

- **BYE** – metoda, která slouží pro ukončení relace. Může být vyslána jak volaným, tak volajícím.
- **CANCEL** – vysláním CANCEL klient ruší předchozí validní žádost INVITE.
- **INVITE** – metoda je odesílána jedním klientem druhému k přizvání do telefonní relace.
- **OPTIONS** – požadavek o výměnu informace požadovaných funkcí klienta nebo proxy serveru. Stejně jako u HTTP protokolu, pokud je vyslána žádost OPTIONS od klienta na proxy server, proxy server odpoví seznamem podporovaných funkcí.
- **REGISTER** – žádost o registraci klienta na registrar server, který jí následně předá lokalizační službě. Tato metoda je také využita pro směřování klientských hovorů u proxy serverů.
- **PRACK** – tento požadavek má stejnou funkci jako metoda ACK, avšak s rozdílem, že se jedná o provizorní potvrzení. Jelikož pro provizorní odezvy protokol SIP nezajišťuje spolehlivé koncové doručení, lze pro tento případ metodu využít. Více o odezvách v textu níže.
- **SUBSCRIBE** – zjišťuje stav vzdáleného prvku.
- **PUBLISH** – je podobná metodě REGISTER, umožňuje vytvoření, aktualizace či odstranění stavu u jiné entity, která spravuje daný stav jménem uživatele.
- **NOTIFY** – upozornění účastníka o vytvoření události, která byla požadovaná metodou SUBSCRIBE.
- **INFO** – slouží k přenosu informací během hovoru
- **REFER** – indikuje příjemci, že by měl kontaktovat třetí stranu za využití kontaktních informací v požadavku.
- **MESSAGE** – předání rychlé zprávy za použití protokolu SIP.
- **UPDATE** – umožňuje klientovi aktualizovat parametry spojení.

Vyslaná zpráva SIP zobrazená *paketovým snifferem*<sup>3</sup> pak může vypadat následovně. Jde o konkrétní příklad metody INVITE.

```
Via: SIP/2.0/UDP 192.168.20.200:45477;branch=z9hG4bK-d8754z-5
      f01a1bf9b47e35e-1---d8754z-
Max-Forwards: 70
Contact: <sip:alice@192.168.20.200:45477;transport=UDP>
To: <sip:1001@192.168.20.200;transport=UDP>
```

<sup>2</sup>handshake – proces pro ustanovení spojení mezi dvěma entitami

<sup>3</sup>paketový sniffer – protokolový analyzátor, který naslouchá a snímá data procházející v síti přes konkrétní rozhraní

```

From: <sip:alice@192.168.20.200;transport=UDP>;tag=7cffe81d
Call-ID: YzI5ZjI4NDU3Mzk5NzIwODZkNDAYYjZjZWQ2OWU1MDg.
CSeq: 2 INVITE
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE, OPTIONS
, INFO, SUBSCRIBE
Content-Type: application/sdp
Supported: replaces, norefersub, extended-refer, timer, X-cisco-
serviceuri
User-Agent: Z~3.3.25608 r25552
Authorization: Digest username="alice",realm="asterisk",nonce
="33b6a2e1",uri="sip:6002@192.168.20.200;transport=UDP",
response="d020a636276abe0c48c0ad166bbefc3a",algorithm=MD5
Allow-Events: presence, kpml
Content-Length: 243

```

V hlavičce zprávy se může nacházet několik položek, nejpodstatnější však pro nás jsou:

- **Via** – tato část hlavičky je přidána pouze po výběru transportního protokolu, který je zde uveden. Dle doporučení RFC je nutné, aby verze protokolu SIP byla 2.0. Nachází se zde také povinný identifikátor transakce přidělený k požadavku.
- **Contact** – klientská adresa IP, čili User Agenta.
- **To** – příjemce zprávy, v našem případě klapka 1001 registrovaná na IP adrese serveru.
- **From** – odesílatel zprávy.
- **Call-ID** – unikátní číslo, které identifikuje daný hovor mezi dvěma klienty a zaručuje tak správné přiřazení paketů pro konkrétní komunikaci.
- **CSeq** – pořadové číslo paketu, většinou je každý paket inkrementován o 1. Další část je název vyslaného požadavku, v této zprávě jde o INVITE metodu.
- **User-Agent** – verze a označení softwarového nebo hardwarového telefonu. Jedná se o telefon Zoiper 3.
- **Authorization** – obsahuje autorizační údaje o klientovi. Zajímavostí u této části hlavičky je, že se může ve zprávě nacházet mnohokrát. V našem případě tomu ale není třeba, jelikož položky, které obsahuje jsou oddělené čárkami.

V případě, že příjemce dostane takový požadavek, odpoví adresátovi jistým návratovým kódem. Tyto odpovědi se dělí do několika skupin. Písmeno *x* je nahrazeno číslem v závislosti na typu odpovědi [3, 5, 6, 7, 18, 22, 23].

- **1xx** – odpověď informativního charakteru, která udává, že server zpracovává akci a že odpověď poskytne obvykle za déle, jak 200 ms.

- **2xx** – odpovídá, že žádost byla úspěšně vykonána. Tato odpověď se značí nejčastěji jako 200 OK.
- **3xx** – informuje o nové poloze uživatele nebo o jiné službě, která je schopná obsloužit hovor.
- **4xx** – dává klientovi zprávu o tom, že došlo k chybě a že by měl modifikovat svou zprávu.
- **5xx** – chyba na serveru.
- **6xx** – odpověď, která značí obecnou chybu, že daný požadavek nelze splnit na žádném serveru.

## 2 OPEN SOURCE POBOČKOVÉ ÚSTŘEDNY PBX

Existuje poměrně velké množství open source pobočkových ústředí. Pro potřeby této práce však byly ústředny Asterisk a to ve dvou verzích, FreeSWITCH, Kamailio a jeho starší verze s označením OpenSER. Tyto ústředny byly zvoleny zejména z důvodu jejich odlišností, a to hlavně u knihoven SIP.

### 2.1 Pobočková ústředna Asterisk

V tuto chvíli jedno z nejpoužívanějších open source řešení s obrovskou komunitou. Jeho odlišnost od ostatních tradičních ústředí tkví v tom, že ve svém *dialplanu*<sup>1</sup> ošetřuje všechny příchozí kanály v podstatě stejným způsobem. Asterisk je založený na *modulech*, což jsou komponenty, které je možné do ústředny přidat a načíst, a které poskytují určitý typ funkcionality, jako třeba jsou knihovny kanálů. To je například knihovna *chan\_sip.so*, což je jinak řečeno *sip stack*. Asterisk lze tedy chápat jako modulární ústřednu [1, 2, 4].

Sponzor, vlastník a primární vývojář ústředny je společnost Digium. Nabízí také Asterisk jako komerční produkt s technickou podporou. Z pohledu kódu mezi komerční a volně šiřitelnou verzí nejsou žádné rozdíly [1].

Aktuálně poslední verze Asterisku je 13. Jedná se o verzi LTS a dělí se ještě jako klasická a *Certified*. Hlavní rozdíl u *Certified Asterisk 13 LTS* je její delší doba vydávání aktualizací a delší doba testování. Lze tedy říci, že je stabilnější a proto byla použita pro naše testování. V rámci testů byla použita také verze *Certified Asterisk 11 LTS*, ve které je použit ještě starší *chan\_sip stack*. Počínaje verzí 12 je možné u Asterisku používat nový SIP modul pod názvem *PJSIP* [1].

### 2.2 Pobočková ústředna FreeSWITCH

Jedná se o multiplatformní modulární ústřednu psanou v jazyku C, spadající pod licenci MPL 1.1. Vznikla roku 2005 jako reakce na neshody v dalším směřování PBX Asterisk. Její nevýhodou může být její kratší doba vývoje, což může znamenat menší

---

<sup>1</sup>dialplan – využívá se ke směřování po vytočení čísla na příslušný koncový bod, skládá se z tzv. kontextů

odladěnost a také menší velikost komunity. Přesto by ústředna měla být schopna inicializace daleko více hovorů, jak Asterisk, čemuž by měla dopomáhat také její knihovna *sofia-sip*, která řeší implementaci protokolu SIP [2, 8, 14].

## 2.3 Proxy ústředna OpenSER a Kamailio

Tyto dvě názvy se nachází pod stejnou kapitolou, jelikož se v podstatě jedná o jednu ústřednu – roku 2002 vznikl projekt SIP Express Router s cílem vytvořit jednu z nejvýkonnějších SIP proxy. Lidé tohoto projektu se rozdělili na dvě části a v roce 2005 byl vytvořen komerční projekt SER a open source projekt pod názvem OpenSER. Kvůli sporům o obchodní značku byla ústředna OpenSER 1.3 poslední verzí a následně se roku 2008 přejmenovala na Kamailio 1.4. OpenSER je tedy předchůdcem Kamailia. Vyznačují se hlavně ve zpracování SIP signalizace a měly by být schopné zvládnout až stovky tisíc požadavků za sekundu [10, 12, 20]. V tomto případě měření je jim tedy usuzován nejlepší výsledek z vybraných ústředen.

## 3 KNIHOVNY IMPLEMENTUJÍCÍ PROTOKOL SIP

Tyto knihovny jsou v různých pramenech odlišně nazývány a existuje pro ně označení jako ovladače, moduly nebo také jako stacky. Jejich úkol je tedy obstarávat veškeré potřeby protokolu SIP a řídit jeho signalizaci. Stack samozřejmě může být i knihovna, která implementuje jiný typ protokolu. V praxi tomu však bývá tak, že je implementováno více protokolů a záleží na vývojáři, jak tento balík ovladačů bude vypadat, jestli bude odlehčený a konstruovaný pro kompaktní aplikace anebo naopak bude komplexnějšího typu.

### 3.1 Výčet nejpoužívanějších SIP stacků

V současné době je dostupných mnoho typů těchto knihoven, avšak z hlediska nejpoužívanějších stabilních a open source SIP stacků lze zařadit následující.

#### 3.1.1 PJSIP stack

Knihovna PJSIP je společně s pomocí vývojářů po celém světě vyvíjena a udržována společností Teluu a to již od roku 2005. Implementuje protokoly, jako jsou SIP, SDP, RTP, STUN, TURN a ICE. Je velmi škálovatelná a multiplatformní. Podporuje operační systémy jako jsou Windows, Mac OS, Linux a mnoho jeho distribucí, Symbián 3. a 5. generace, Apple iOS, Android a spousty dalších [1, 16].

Její otisk v paměti, neboli *footprint*<sup>1</sup> je velice malý a v případě základních knihoven začíná na 150 KB. Toto se samozřejmě odvíjí ale od jeho použité aplikace. Například během videohovoru bude tento otisk daleko větší [1].

Podporuje také například IPv6 a dokáže provádět videohovory. Koncem roku 2013 je také zahrnutá jako podporovaný stack v ústředny Asterisk 12 [1].

#### Nativně podporované audio kodeky

- Speex 8KHz, 16KHz, 32KHz
- iLBC, GSM
- L16, G.711A/U (PCMA/PCMU)

---

<sup>1</sup>footprint – velikost knihovny v paměti v případě její implementace



- G.722
- G.722.1 16KHz/32KHz

S pomocí knihoven třetích stran pak lze i přidat další podporu pro jiné kodeky, to však může být vázáno licencí.

#### **Nativně podporované video kodeky**

- H.263
- H.264

**Aktuální verze stacku:** 2.4.5, srpen 2015

### **3.1.2 Sofia-sip stack**

Knihovna vyhovující dle doporučení RFC 3261. Může být využita jak pro VoIP, tak i pro IM. Je založena na stacku vyvíjeném výzkumnými centry společnosti Nokia. První její verze vyšla roku 2005 avšak počínaje rokem 2011 až do této chvíle již nebyla aktualizována. Dále je možné ji nasadit na různé platformy, jako je Windows, Mac OS, primárně však je vyvíjena pro Linux. Je součástí ústředny FreeSWITCH [20, 8].

**Aktuální verze stacku:** 1.12.11, březen 2011

### **3.1.3 ReSIPProcate stack**

Tento SIP stack je využit jak v open source, tak i v komerčních produktech. Založen roku 2002 a součástí organizace *SIPfoundry*. Využívá přenosu přes UDP, TCP, TLS, DTLS nebo i WebRTC. Vyznačuje se svou velkou stabilitou a širokým použitím, jako jsou třeba telefony, brány, proxy servery, IM klienti či B2BUA. Dle dokumentace je také schopen asi 500 transakcí za sekundu, umí pracovat v multivláknových aplikacích a využívá pouze transportní vrstvy. [17].

**Aktuální verze stacku:** 1.10, říjen 2015

### **3.1.4 oSIP stack**

Začátek vývoje této knihovny je roku 2000, v dubnu roku 2001 vyšla jeho první verze a roku 2002 se stal součástí projektu GNU. Je využíván například v klientské

aplikaci *SFLphone* nebo i v ústředně FreeSWITCH [13].

**Aktuální verze stacku:** 4.1.0, leden 2013

### 3.1.5 OPAL stack

Tvorba VoIP stacku OPAL neboli *Open Phone Abstraction Library* začala později roku 1999 jako nástupce knihovny OpenH323. Jedná se o velice komplexní open source stack pod licencí MPL s podporou nejen protokolu SIP, ale i H.323. Do roku 2004 se však jednalo spíše o experimentální verze. Protokol RTP má již implementovaný. Umožňuje také IM dle RFC 3428, RFC 4975 a RFC 4103. Pro SIP nativně umožňuje přenos hlasu s využitím kodeku G.711 aLaw a uLaw, dále pak je možný doinstalovat jako plugin například Speex kodex. Video je zde možné přenášet dle standardu H.263 a H264. OPAL může být nasazen jako *User Agent* čili koncové zařízení, tak i *proxy* nebo *registrar* server [15].

**Aktuální verze stacku:** 3.16, květen 2016

### 3.1.6 VOCAL stack

VOCAL je open source projekt skupiny *Vovida*, financovaný firmou Cisco. Nejedná se pouze o SIP stack, ale podobně jako OPAL obsahuje velké množství implementací a aplikačních knihoven. Jsou to například překladač SIP/H.323 nebo SIP/MGCP, může sloužit jako *proxy* nebo *registrar* server či *User Agent*. Bohužel stack je však špatně interoperabilní s jinými implementacemi protokolů a doporučení RFC 3261 splňuje jen částečně [26].

**Aktuální verze stacku:** 1.5.0, duben 2003

## 3.2 Porovnání SIP stacků

Pro lepší přehlednost zde uvádím také teoretické porovnání již zmiňovaných stacků, které lze nalézt v tabulce nacházející se v příloze, kapitola A.4 [13, 15, 16, 17, 20, 26].

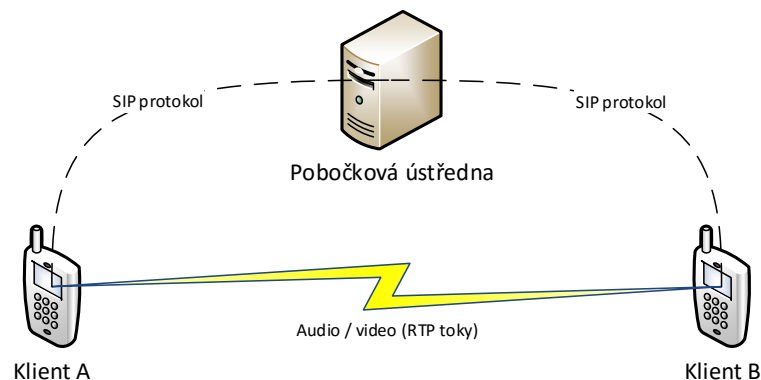
## 4 PRAKTICKÉ TESTY

Hlavní část této práce tvoří měření a testy ústředen a některých stacků zmiňovaných v předchozí teoretické části. Tato kapitola podrobně rozebírá veškerou přípravu testů a ústředen až po měření samotné.

### Stručný přehled měření

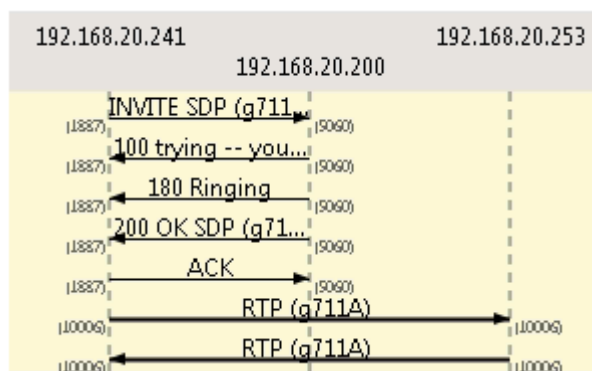
- Měření maximálního počtu relací
  - Konfigurace s 768 MB RAM
    - \* B2BUA režim
    - \* Proxy režim
  - Konfigurace s 1536 MB RAM
    - \* B2BUA režim
    - \* Proxy režim
- Měření maximálního počtu transakcí
  - Konfigurace s 768 MB RAM
    - \* B2BUA režim
    - \* Proxy režim
  - Konfigurace s 1536 MB RAM
    - \* B2BUA režim
    - \* Proxy režim
- Měření s jinými kodeky
  - Konfigurace s 768 MB RAM
    - \* B2BUA režim
  - Konfigurace s 1536 MB RAM
    - \* B2BUA režim
- Měření s modifikovanými zprávami
  - 15 typů vybraných zpráv dle RFC 4475

V případě konfigurace ústředny na proxy režim budou po sestavení hovoru procházet veškerá hlasová a jiná multimediální data přímo mezi dvěma klienty a ústředna je tedy nebude zpracovávat, jak je znázorněno na obrázku 4.1. Pro ústřednu to znamená menší zátěž, avšak již nemůže používat funkce, jako je například nahrávání hovorů.



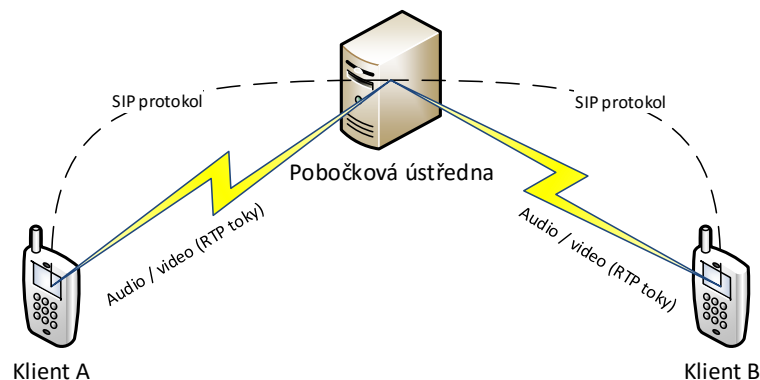
Obr. 4.1: Komunikace v proxy režimu

V praxi, kdybychom si poté odchytili pomocí paketového analyzátoru komunikaci, bude vypadat podobně, jak na obrázku 4.2, kde je 192.168.20.200 IP adresa ústředny, která zprostředkovává spojení. Zbylé dvě adresy jsou klientské stanice. Jakmile dojde k sestavení hovoru, tyto stanice si začnou vysílat již hlasová data pomocí kodeku G.711A již pouze mezi sebou.



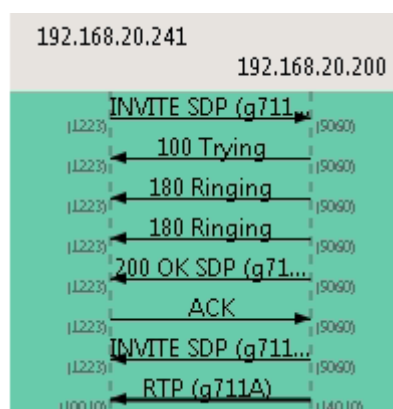
Obr. 4.2: Analýza proxy režimu

Opačná možnost, kdy veškerá audio a video data prochází přes ústřednu je režim B2BUA (obr. 4.3). Tímto režimem však nedisponují ústředny OpenSER a Kamailio, jelikož jejich hlavní funkcí je pracovat jako proxy server. Je třeba připomenout, že přenos multimediálních dat zprostředkovává RTP protokol. SIP protokol má na starost pouze signalizaci.



Obr. 4.3: Komunikace jako B2BUA

Na obrázku 4.4 je znázorněna zachycená komunikace v B2BUA režimu mezi serverem (vpravo) a klientem (vlevo).



Obr. 4.4: Analýza B2BUA režimu

## 4.1 Systém a jeho příprava

Před samotným započítím měření bylo nejdříve třeba si nakonfigurovat veškerá prostředí. Následující podkapitoly popisují výběr systému, konfiguraci ústředny, způsob měření a nastavení testeru Spirent TestCenter C1.

### 4.1.1 Výběr a konfigurace operačního systému

Veškeré testy a instalace probíhaly ve virtualizovaném 64bitovém operačním systému. Každá ústředna měla zvlášť svůj OS. Využíval se CentOS ve verzi 7, který je rovněž open source jako ústředny. Jedná se o náhradu za komerční distribuci Red Hat Enterprise Linux a byl vybrán zvláště proto, že je zaměřený na stabilitu a spolehlivost. Hodí se tedy do podnikového a produkčního prostředí, a tím pádem je pro nás, pro simulaci různých zátěží ideální. Jako alternativu systému CentOS 7 bych v opačném případě volil Debian ve verzi LTS<sup>1</sup>, a to z důvodu jeho skvělé kompatibility pro mnoho různých platforem.

Systém běžel virtualizovaně na této konkrétní hardwarové konfiguraci. Níže zmiňované operační paměti ve dvou verzích (větší a menší) značí, že se jednalo o dva typy nastavení systému.

```
Procesor: Intel i5 3570, 3,4 GHz, 4 jádra
Operační paměť: 768 MB a 1536 MB
Video paměť: 12 MB
Síťová karta: 1 Gbit
```

Před instalací ústředny také bývá dobrým zvykem provést aktualizaci systému a tím předejít možným problémům v budoucnu. Je také nutné uvést, že všechny příkazy byly zadávány jako *superuser*<sup>2</sup>.

```
[root@localhost ~]# yum update
```

Pro další potřeby, jako je například přístup do konzole ústředny je nutné zkontrolovat přítomnost modulu *SELinux*<sup>3</sup> a v případě běhu jej dočasně zakázat. Toto rozšíření je možné vypnout pomocí editace souboru `config` v adresáři `/etc/selinux/` anebo jednoduchým příkazem pomocí nástroje `sed`. Následně je také třeba systém restartovat.

```
[root@localhost ~]# sestatus
SELinux status:                enabled
[root@localhost ~]# sed -i s/SELINUX=enforcing/SELINUX=disabled/
g /etc/selinux/config
[root@localhost ~]# reboot
```

Nově také v systému CentOS 7 běží implicitně jiný firewall než *iptables* a to *firewalld*. Tento firewall blokuje komunikaci ústředny s ostatními entitami, a proto

<sup>1</sup>v případě Debianu zde jsou takové verze označovány celými čísly, například *Debian 8*

<sup>2</sup>superuser – systémový administrátor s veškerými právy, jinak také označován jako root. V CLI je indikován mřížkou #.

<sup>3</sup>SELinux – rozšíření jádra systému Linux, které slouží pro větší zabezpečení systému

byl v rámci testování zakázán a vypnut. Samozřejmě v podnikovém prostředí by se měla naopak přidat do firewallu výjimka popřípadě jej vyměnit za starší avšak stále osvědčený *iptables*.

```
[root@localhost ~]# systemctl status firewalld
Active: active (running)
[root@localhost ~]# systemctl disable firewalld
[root@localhost ~]# systemctl stop firewalld
```

Dále je také ještě třeba nastavit maximální povolený počet otevřených souborů v systému úpravou položky `limits.conf`.

```
[root@localhost ~]# vi /etc/security/limits.conf
```

Hodnoty byly nastaveny a zvýšeny.

|   |      |        |       |
|---|------|--------|-------|
| * | soft | nofile | 8192  |
| * | hard | nofile | 32768 |

Po zapsání hodnot výše je třeba restartovat systém. Provedené změny si můžeme ověřit příkazem `ulimit`.

```
[root@localhost ~]# ulimit -Sn
8192
[root@localhost ~]# ulimit -Hn
32768
```

Aby bylo měření zaplnění operační paměti lépe vypovídající, je vhodné vyprázdnit také před spuštěním ústředny cache paměť následujícím příkazem:

```
[root@localhost ~]# sync; echo 3 > /proc/sys/vm/drop_caches
```

Poslední úkon, který byl před spuštěním třeba udělat, bylo z bezpečnostních důvodů přiřadit procesu ústředny, popřípadě stromu procesů dvě jádra procesoru z celkových čtyř. Je totiž možné, že z hlediska vysoké zátěže ústředny během testování nezbudou již prostředky na vlastní chod systému a ten se může tak stát kompletně neovladatelným. Následující příkaz demonstruje přiřazení jádra číslo 2 a 3 pro ústřednu FreeSWITCH. To znamená, že FreeSWITCH bude moci využívat jen dvě tyto jádra, zato samotný systém není omezen a bude využívat všechny čtyři, ne, jak by se mohlo zdát zbývající jádra číslo 0 a 1.

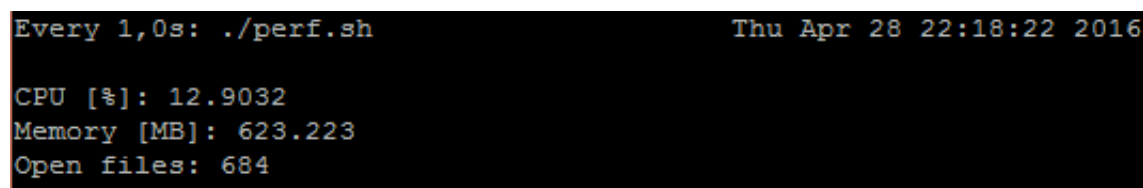
```
[root@localhost ~]# su -c 'taskset -c 2,3 freeswitch'
```

### 4.1.2 Skript pro shromažďování dat

Pro měření některých dalších parametrů bylo třeba vytvoření jednoduchého skriptu v jazyku Bash pod názvem `perf.sh`, který je také přiložen jako příloha č. A.2. Skript po každé sekundě zapisoval aktuální stav vytížení procesoru, zaplnění operační paměti a počet vytvořených souborů procesem či procesy ústředny. Nejprve po jeho zkopírování do systému bylo třeba povolení některých přístupových práv a poté se po sekundě opakovaně spouštěl, jak popisují následující dva řádky.

```
[root@localhost ~]# chmod 775 perf.sh
[root@localhost ~]# watch -n 1 ./perf.sh
```

Při spuštění skript začne zapisovat do souborů `cpu.txt` aktuální frekvenci procesoru, `mem.txt` aktuální stav zaplnění paměti RAM a do `files.txt` aktuální počet otevřených souborů procesem ústředny. Před každým měřením jiné ústředny však bylo potřeba přejmenovat její název ve zdrojovém kódu měřicího soubor kvůli správnému shromažďování dat a přiřazení PID. Skript také pro přehled aktuální informace vypisoval do terminálového okna, jak znázorňuje obr. 4.5 níže.



```
Every 1,0s: ./perf.sh Thu Apr 28 22:18:22 2016
CPU [%]: 12.9032
Memory [MB]: 623.223
Open files: 684
```

Obr. 4.5: Ukázka spuštěného měřicího skriptu



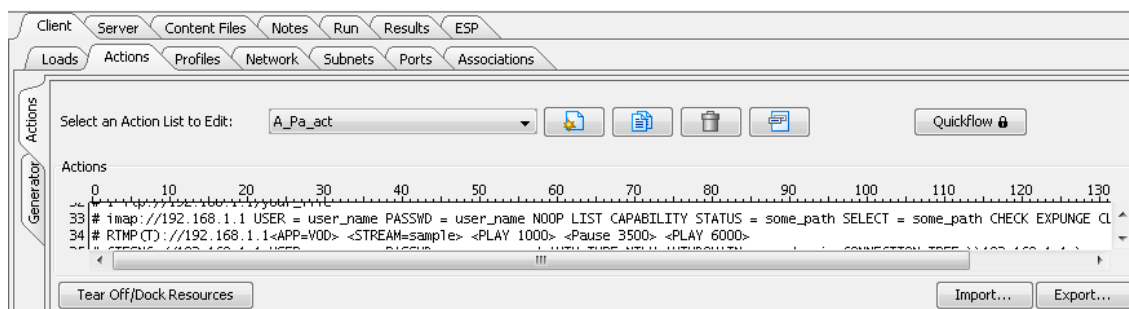
### 4.1.3 Konfigurace generátoru zátěže

Měření se provádělo za pomoci zátěžového generátoru Spirent TestCenter C1 (obr. 4.6) a programu Avalanche Commander. Tento tester dokáže generovat zátěž jak pro aplikace pro přenos datových souborů, tak i videa nebo hlasu. Mezi podporované protokoly například patří IPv4 a IPv6, SMTP, POP3, IMAP4, DNS, HTTP, HTTPS, FTP, SIP, RTSP, RTP, RTMP a další [21].



Obr. 4.6: Generátor zátěže Spirent

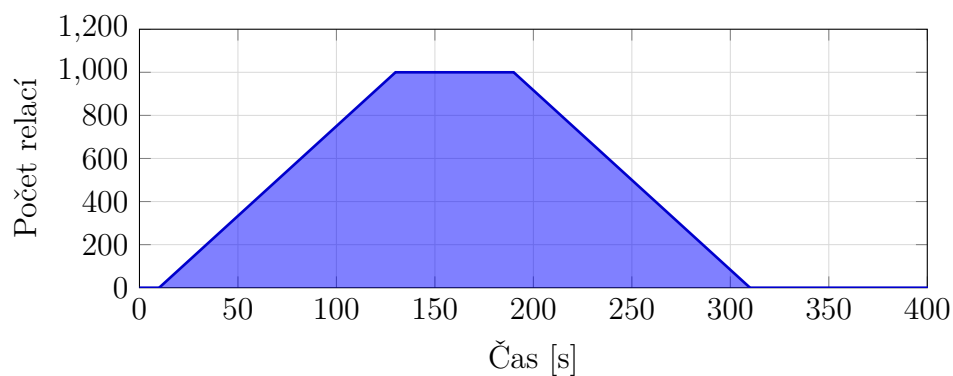
Jak je vidět z obrázku níže, který zobrazuje částečné rozhraní testeru. Obsah a nabídka parametrů pro provedení testu je opravdu komplexní. V našem případě byla zvolena aplikace SIPNG, kterou byly prováděny testovací hovory.



Obr. 4.7: Ukázka rozhraní aplikace Avalanche Commander

V záložce *Client*, která značí volajícího, byly nadefinovány klapky a číslo jak volajícího, tak i volaného. Bylo zde také definováno medium, zvukový soubor, který bude během hovoru přehráván. Pro tento soubor je možné zvolit i jeho výchozí kodek. Volaného, čili klienta na druhé straně hovoru značí záložka *Server*. Doba jednoho hovoru při měření relací byla nastavena na délku celého testu v době generování relací, tímto bylo zajištěno plné vytížení a také hladší průběh v charakteristice grafu.

Na obrázku 4.8 je zobrazeno, jak byl počet relací a hovorů navyšován. Celý proces tedy probíhal tak, že jakmile se klient přihlásil k ústředně, ihned inicializoval hovor, který trval během průběhu celého testu. Počet těchto klientů se následně zvyšoval, poté se ustálil a hovory začaly být ukončovány. Po konci testu je nastaven ještě tzv. *odpočinkový čas*, a to z toho důvodu, že mezi klienty stále ještě probíhal hovor a tak jeho ukončování a odregistrace klientů může trvat pozvolněji, než je definováno dle charakteristiky.



Obr. 4.8: Požadované vytížení při simulaci

## 4.2 Instalace a konfigurace PBX ústředen

Jelikož každá ústředna využívá odlišných knihoven pro implementaci protokolu SIP, bylo třeba si dané ústředny jednotlivě nahrát, nakonfigurovat a následně až potom provádět testování. Níže jsou popsány jejich instalace a konfigurace. Před započítím testování byly u každé ústředny vytvořeny dva účty s čísly 1000 a 1001 a nastaveno směrování hovorů. Účty byly nastaveny bez autorizace, jelikož byla na generátoru zátěže problémová. IP adresa serveru byla vždy 192.168.20.200.

### 4.2.1 Instalace ústředen Asterisk

Ústředna Asterisk byla instalována ve dvou verzích. První verze byla *Certified Asterisk 11*, druhá verze *Certified Asterisk 13*, instalována zejména z důvodů možnosti využít jiné knihovny SIP a to konkrétně *PJSIP stack*. V případě obou verzí jde o LTS podporu.

#### Instalace ústředny Asterisk 11

Nainstalujeme si nástroj `wget`, přesuneme se do pracovní složky a stáhneme a rozbálíme potřebné soubory Asterisku.

```
[root@localhost ~]# yum install wget sqlite-devel
[root@localhost ~]# cd /usr/src/
[root@localhost src]# wget http://downloads.asterisk.org/pub/
    telephony/certified-asterisk/asterisk-certified-11.6-current.
    tar.gz
[root@localhost src]# tar zxvf asterisk-certified*
```

Po rozbalení souborů se přesuneme do složky Asterisku, a poté do složky `contribscripts`, kde najdeme skript `install_prereq`, který za nás vybere a nainstaluje potřebné balíčky k chodu a konfiguraci ústředny.

```
[root@localhost src]# cd asterisk-certified*
[root@localhost asterisk-certified-11.6-cert11]# cd contrib/
    scripts
[root@localhost scripts]# ./install_prereq install
```

Vrátíme se zpět do kořene složky Asterisku a zahájíme kompilaci a instalaci. Z menu, které se objeví také můžeme zkontrolovat, jestli bude SIP modul opravdu zavedený.

```
[root@localhost scripts]# cd /usr/src/certified-asterisk*
[root@localhost asterisk-certified-11.6-cert11]# ./configure --
    libdir=/usr/lib64 && make menuselect && make && make install
```

Nastavíme, aby se Asterisk spustil po startu systému a můžeme si také vytvořit ukázkové konfigurační skripty.

```
[root@localhost certified-asterisk-11.6-cert11]# make config &&
    make samples
```

Nyní už jen zbývá si nakonfigurovat dialplan, SIP účty a následně můžeme spustit ústřednu a přihlásit se do konzole pro zobrazení výpisu událostí. Parametr `-rvvv` udává v jaké míře ústředna bude vypisovat události. Konfigurační soubory pro dialplan a účty SIP jsou přiložené k práci jako příloha č. A.1.1.

```
[root@localhost certified-asterisk-11.6-cert11]# service
    asterisk start && asterisk -rvvv
```

Nainstalovaná verze ústředny:

```
Asterisk 11.6-cert11
```

## Instalace ústředny Asterisk 13 a stacku PJSIP

Jak již bylo řečeno, Asterisk od verze 12 podporuje nový SIP stack, PJSIP, který je však třeba individuálně přidat. Pro začátek je ale třeba před instalaci nainstalovat potřebné balíčky. A poté se přesunout do pracovního adresáře.

```
[root@localhost ~]# install -y bzip2 epel-release dmidecode nano
    gcc-c++ ncurses-devel libxml2-devel make wget openssl-devel
    newt-devel kernel-devel sqlite-devel libuuid-devel gtk2-devel
    jansson-devel binutils-devel
[root@localhost ~]# cd /usr/src/
```

Začneme tedy stažením a rozbalením zdrojových souborů nejnovějšího stacku PJSIP.

```
[root@localhost src]# wget http://www.pjsip.org/release/2.4.5/
    pjproject-2.4.5.tar.bz2
[root@localhost src]# tar -xjvf pjproject*
[root@localhost src]# cd /usr/src/pjproject*
```

Po přesunutí do složky projektu PJSIP je nutné si soubory připravit pro kompilaci.

```
[root@localhost pjproject-2.4.5]# ./configure --prefix=/usr --
libdir=/usr/lib64 --enable-shared --disable-sound --disable-
resample --disable-video --disable-opencore-amr CFLAGS='-O2 -
DNDEBUG'
```

Následně provedeme instalaci a kompilaci balíčků.

```
[root@localhost pjproject-2.4.5]# make dep && make && make
install && ldconfig
```

Můžeme si také ještě ověřit, že knihovny byly správně do systému přiřazeny. V případě, že ano, máme tímto instalaci projektu PJSIP kompletní, a přesuneme se zpět do pracovní složky.

```
[root@localhost pjproject-2.4.5]# ldconfig -p | grep pj
[root@localhost pjproject-2.4.5]# cd /usr/src/
```

Nyní přichází na řadu stažení ústředny Asterisk 13. Stažený soubor si rozbalíme a přesuneme se do rozbalené složky.

```
[root@localhost src]# wget http://downloads.asterisk.org/pub/
telephony/certified-asterisk/asterisk-certified-13.1-current.
tar.gz
[root@localhost src]# tar zxvf certified-asterisk*
[root@localhost src]# cd /usr/src/certified-asterisk*
```

Zdrojové soubory připravíme ke kompilaci, nainstalujeme ústřednu a nastavíme jí start po spuštění PC. Můžeme také i vytvořit ukázkové konfigurační soubory.

```
[root@localhost certified-asterisk-13.1-current]# ./configure --
libdir=/usr/lib64
[root@localhost certified-asterisk-13.1-current]# make && make
install && make samples && make config
```

Pro kontrolu si můžeme v menu ověřit, že PJSIP moduly jsou k ústředně přiřazené.

```
[root@localhost certified-asterisk-13.1-current]# make
menuselect
```

Posledním krokem je spuštění ústředny. V příloze č. A.1.2 se nachází konfigurační soubory pro nastavení klapek přes PJSIP stack a dialplanu.

```
[root@localhost certified-asterisk-13.1-current]# service
asterisk start
```

Nainstalovaná verze ústředny:

```
Asterisk 13.1-cert2
```

Nainstalovaná verze PJSIP stacku:

```
PJSIP 2.4.5
```

## Instalace kodeku pro obě ústředny Asterisk

Jelikož se v práci budu zabývat i měřením přes další kodeky, je třeba si je doinstalovat. V tomto případě chyběl kodek G.729 v obou ústřednách Asterisk a nebyl jim přidělen již implicitně z licenčních důvodů. Kodek je dostupný přímo ze stránek společnosti Digium, která při zakoupení jejich licence umožní kodeku využívat funkce jako je konference, nahrávání hovorů a jiné. Nyní přejdeme k instalaci a implementaci kodeku do ústředny.

Přesuneme se do pracovní složky a stáhneme si kodek ze stránek společnosti Digium. Adresa pro stažení se odvíjí podle konfigurace PC a verze ústředny. Níže je uvedena adresa s kompatibilním modulem pro naši verzi ústředny Asterisk 11.

```
cd /usr/src
wget http://downloads.digium.com/pub/telephony/codec_g729/
    asterisk-11.0/x86-64/codec_g729a-11.0_3.1.7-generic_64.tar.gz
tar xzvf codec*
yes | rm codec*.tar.gz
cp codec*/codec_g729a.so /usr/lib64/asterisk/modules
```

Kodek rozbalíme a zkopírujeme do složky s moduly pro Asterisk.

```
tar xzvf codec*
cp codec*/codec_g729a.so /usr/lib64/asterisk/modules
```

Na konec již stačí pouze aktivovat modul a kodek nastavit v souboru `sip.conf`.

```
asterisk -rx "module load codec_g729a.so"
sed -i s/allow=alaw/allow=g729/g /etc/asterisk/sip.conf
asterisk -rx "sip reload"
```

Nainstalovaná verze kodeku:

```
G.729 3.1.7
```

## 4.2.2 Instalace ústředny Freeswitch

Instalace ústředny verze 1.6 nám vývojáři podstatně zjednodušili, a to tak, že jí lze nainstalovat jednoduše jako balíček pomocí nástroje a příkazu `yum`. Nejdříve je však třeba jej přidat do balíčkovacího systému. Dodatečně pak lze ještě doinstalovat různé zvukové soubory k ústředně. V některých případech se může stát, že instalace zahlásí chybu a je třeba doinstalovat ještě dodatečné balíčky, které si sama neobstará. V našem případě to byl `epel-release`.

```
[root@localhost ~]# yum install -y epel-release
```

Dále již nic nebrání v instalaci ústředny. Po provedení následujících kroků je ústředna plně funkční. Definice klapky a další nastavení FreeSWITCHe je popsána v příloze A.1.3.

```
[root@localhost ~]# rpm -Uvh http://files.freeswitch.org/
freeswitch-release-1-6.noarch.rpm
[root@localhost ~]# yum install freeswitch-config-vanilla
[root@localhost ~]# yum install sox freeswitch-sounds*
```

Nainstalovaná verze ústředny:

```
Freeswitch 1.6
```

## 4.2.3 Instalace ústředny OpenSER

I když je OpenSER předchůdcem ústředny Kamailio, jeho instalace je mírně odlišná, jelikož poslední verze 1.3.4 je stará již šest let a spoustu úkonů od té doby bylo zautomatizováno. Tradičně začneme instalací potřebných balíčků a přepnutím do pracovní složky [12, 10].

```
[root@localhost ~]# yum install -y wget bison flex subversion
gcc
[root@localhost ~]# cd /usr/src/
```

Nyní nainstalujeme spolu s účtem MySQL databázi, která je potřebná pro správnou funkci ústředny.

```
[root@localhost src]# wget http://repo.mysql.com/mysql-community
-release-el7-5.noarch.rpm
[root@localhost src]# rpm -ivh mysql-community-release-el7-5.
noarch.rpm
[root@localhost src]# yum install -y mysql-server mysql mysql-
devel
```

```
[root@localhost src]# sed -i s/sql_mode=NO_ENGINE_SUBSTITUTION,
    STRICT_TRANS_TABLES/sql_mode=NO_ENGINE_SUBSTITUTION/g /etc/my
.cnf
[root@localhost src]# service mysqld start
[root@localhost src]# mysqladmin -u root password 1234
```

Provedeme stažení zdrojových kódů ústředny. Poslední verze s tehdejším názvem OpenSER je 1.3.4.

```
[root@localhost src]# svn co http://openser.svn.sourceforge.net/
svnroot/openser/branches/1.3 openser
```

Přepneme se do složky nově stažených zdrojových kódů a provedeme jejich kompilaci a instalaci včetně modulu MySQL.

```
[root@localhost src]# cd openser/
[root@localhost openser]# make all include_modules="mysql"
[root@localhost openser]# make prefix=/usr/local install
include_modules="mysql"
```

Editujeme soubor `openserctlrc` a upravíme jeho parametry. Tyto změny můžeme najít v příloze A.1.4.

```
[root@localhost openser]# vi /usr/local/etc/openser/openserctlrc
```

Vytvoříme si databázi pro ústřednu a všechny následující kroky potvrdíme.

```
[root@localhost openser]# openserdbctl create
```

Před startem ústředny je ještě potřeba několik dalších kroků. Povolíme některé MySQL moduly v konfiguračním souboru `opernser.cfg`. Seznam povolených modulů a další změny v souboru jsou v příloze A.1.4..

```
[root@localhost openser]# vi /usr/local/etc/openser/openser.cfg
```

Nyní nastavíme, aby se ústředna spouštěla po startu serveru. K souboru rovněž nastavíme bezpečný přístup.

```
[root@localhost openser]# cp /usr/src/openser/packaging/rpm/
openser.init /etc/init.d/openser
[root@localhost openser]# chmod 755 /etc/init.d/openser
[root@localhost openser]# /sbin/chkconfig --add openser
[root@localhost openser]# systemctl daemon-reload
```

Dále je ještě třeba upravit cestu v konfiguraci spouštěcího souboru změnou položky z `oser=/usr/sbin/openser` na `oser=/usr/local/sbin/openser`.

```
[root@localhost openser]# vi /etc/init.d/openser
```



Klapky přidáme následovně:

```
[root@localhost openser]# openserctl add 1000 1234 1000@192
.168.20.200
[root@localhost openser]# openserctl add 1001 1234 1001@192
.168.20.200
```

Nainstalovaná verze ústředny:

```
OpenSER 1.3.4-notls
```

## 4.2.4 Instalace ústředny Kamailio

I pro ústřednu Kamailio, stejně jako pro FreeSWITCH existují repositáře, avšak z neznámých důvodů po instalaci ústředny z některých repositářů nelze inicializovat hovor. Jedná se o ústřednu verze 4.3.4. Proto je její instalace, i když je z repositáře, trochu komplikovanější, a to i z důvodu, že jí je třeba propojit s databází podobně, jak tomu bylo u OpenSER. Standardně tedy začneme přesunem do pracovního adresáře, kde si vytvoříme repositář, který nám je nabídnut ze stránek <http://rpm.kamailio.org/> pro náš systém CentOS. Z tohoto repositáře získáme funkční ústřednu ve verzi 4.2.3. Nutno také dodat, že spousta odkazů z domovských stránek ústředny Kamailio jsou nefunkční či zastaralé. Je tudíž těžké posoudit, jaká poslední verze ústředny je se systémem CentOS 7 opravdu kompatibilní [12, 10].

```
[root@localhost ~]# cd /usr/src/
[root@localhost src]# vi kamailio.repo
[root@localhost src]# mv kamailio.repo /etc/yum.repos.d/
```

Nainstalujeme ústřednu Kamailio v4.2.3.

```
[root@localhost src]# yum install -y kamailio kamailio-mysql
kamailio-debuginfo kamailio-unixodbc kamailio-utils
```

Následně nainstalujeme i databázi MySQL. Ještě před startem je však třeba upravit některé její parametry, aby s ní Kamailio mohla pracovat.

```
[root@localhost src]# wget http://repo.mysql.com/mysql-community
-release-el7-5.noarch.rpm
[root@localhost src]# rpm -ivh mysql-community-release-el7-5.
noarch.rpm
[root@localhost src]# yum install -y mysql-server mysql
[root@localhost src]# sed -i s/sql_mode=NO_ENGINE_SUBSTITUTION,
STRICT_TRANS_TABLES/sql_mode=NO_ENGINE_SUBSTITUTION/g /etc/my
.cnf
[root@localhost src]# service mysqld start
```

Poslední kroky je spárování ústředny s databází, kde je nejdříve třeba vytvořit přístup, následně upravit konfigurační soubor ústředny `kamctlrc` a `kamailio.cfg`. Tato konfigurace je stejná, jak v případě ústředny OpenSER v příloze A.1.4, jen názvy `openser` zaměníme za `kamailio`. Necháme ústřednu vytvořit databázi a spustíme ji. Po nadefinování nových účtů je ústředna plně funkční.

```
[root@localhost src]# mysqladmin -u root password 1234
[root@localhost src]# vi /etc/kamilio/kamctlrc
[root@localhost src]# vi /etc/kamailio/openser.cfg
[root@localhost src]# kamdbctl create
[root@localhost src]# service kamailio start
[root@localhost src]# kamctl add 1000 1234 1000@192.168.20.200
[root@localhost src]# kamctl add 1001 1234 1001@192.168.20.200
```

Nainstalovaná verze ústředny:

```
Kamailio 4.2.3
```

## 4.3 Měření maximálního počtu relací

Pro větší přehlednost v práci byly některé grafy z měření přesunuty do přílohy. Jedná se o grafy změřených RTP toků a počtu aktivních relací s nastavením operační paměti na 1536 MB. Jiné grafy, jako je měření vytížení procesoru, operační paměti a měření počtu otevřených souborů byly zase z práce vyjmuty (jsou přiloženy pouze jako elektronická příloha) a jejich charakteristika byla pouze slovně okomentována a to z důvodu jejich velkého množství. Pro některé ústředny se totiž měřily všechny tyto charakteristiky celkem ve čtyřech typech konfigurace (shrnutí na začátku kapitoly 4), čímž by počet těchto grafů zbytečně navyšoval obsah práce. Proto byly charakteristiky vloženy pouze jako vzor pro první ústřednu Asterisk 11 a to při konfiguraci s 1536 MB RAM.

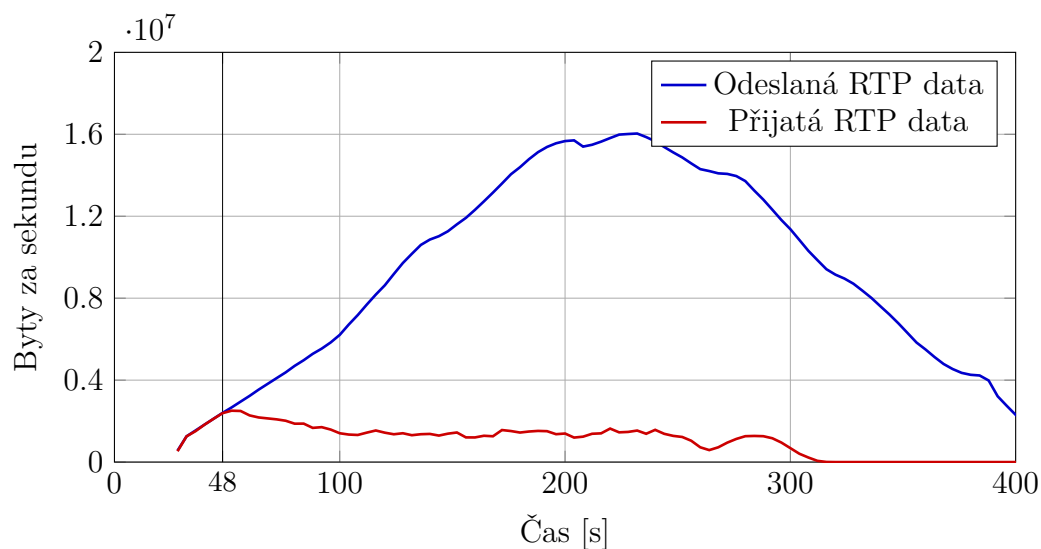
Měření maximálního počtu relací pro ústřednu probíhalo tím způsobem, že se v grafu našel bod, kde se RTP toky začínají rozdělovat. Z této doby rozdělení se pak v charakteristice s relacemi našel daný počet dosažených hovorů. Tyto hodnoty jsou v každém grafu vyznačené. Zároveň se zvyšujícím počtem relací rostl i počet souborů, které si ústředna vytvářela. Daný počet souborů byl rovněž zaznamenáván společně s rostoucí operační pamětí a výkonem procesoru pomocí již zmiňovaného skriptu v kapitole 4.1.2.

### 4.3.1 Výkon a stabilita ústředny Asterisk 11

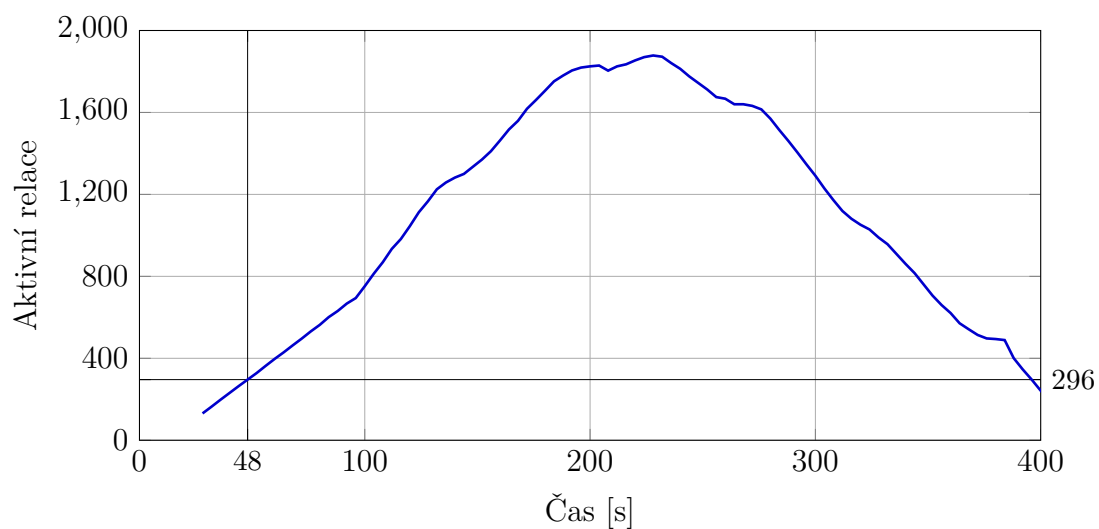
#### B2BUA režim

Rozdělení RTP toků z grafu 4.9 v režimu B2BUA došlo ve 48. sekundě, což odpovídá 296 stabilně dosaženým relacím viz obrázek 4.10. Zaplnění operační paměti společně s otevřenými soubory rostlo pozvolna až do maxima a přes průměrný výkon CPU okolo 33 % byla ústředna stále velmi dobře ovladatelná. V tomto režimu vyšla doba sestavení hovoru nejhůře ze všech ústředen s časem 690 ms.

S dvojnásobně menší operační pamětí se počet otevřených souborů snížil. Pravděpodobně, z důvodů, že již byla zaplněná. Zajímavé však je, že počet relací zůstal téměř identický a to 297. Hovor se v průměru i rychleji sestavil. Doba sestavení činila 241 ms a průměrné vytížení procesoru 26 %.



Obr. 4.9: Změřené RTP toky ústředny Asterisk 11 v režimu B2BUA s 1536 MB RAM

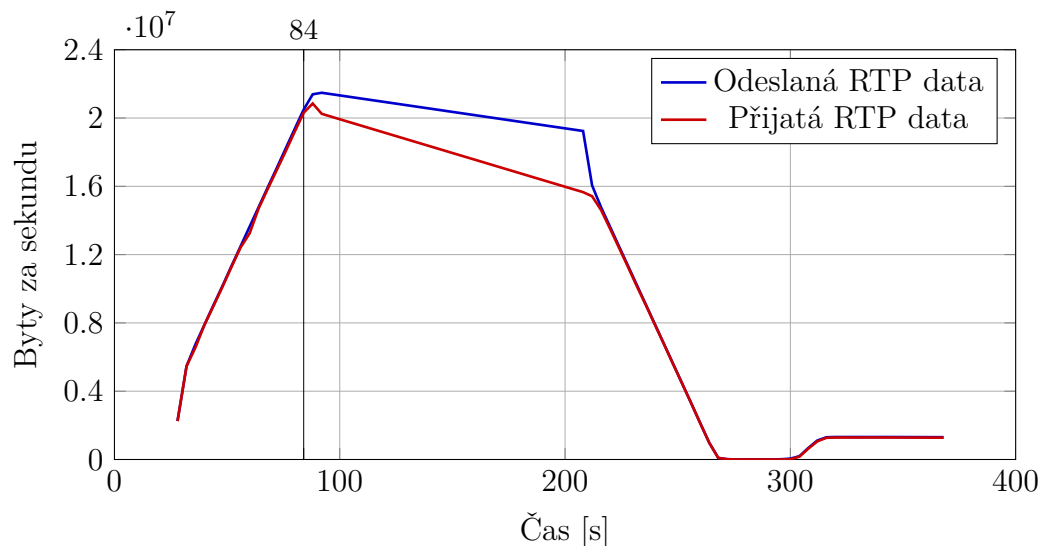


Obr. 4.10: Změřený počet relací ústředny Asterisk 11 v režimu B2BUA s 1536 MB RAM

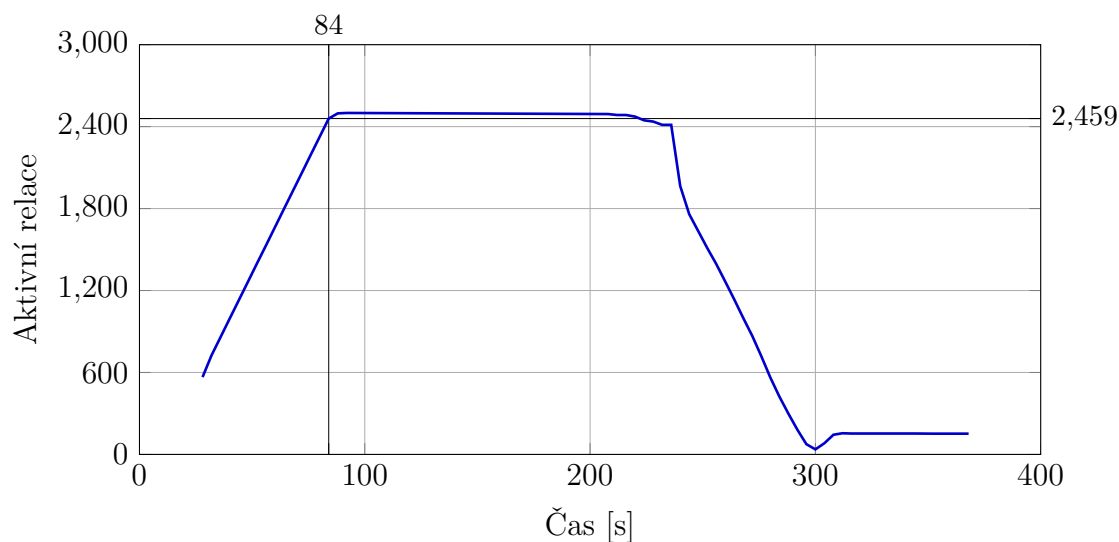
### Proxy režim

V režimu proxy se RTP toky dle obr. 4.11 roztrhly ve 84. sekundě, což činí při 1536 MB RAM 2450 současných relací (obr. 4.11) a při 768 MB RAM ve 80. sekundě tomu bylo 2326 relací. I zde tedy nemělo snížení operační paměti velký význam na

počet uskutečněných relací. Stejně tak i hodnoty vytížení CPU se v obou konfiguracích paměti pohybovaly na nízkých hodnotách, v průměru 7-13 %. Asterisk však v proxy režimu otevřel přes 20000 souborů, což je maximum ze všech měřených ústředen.

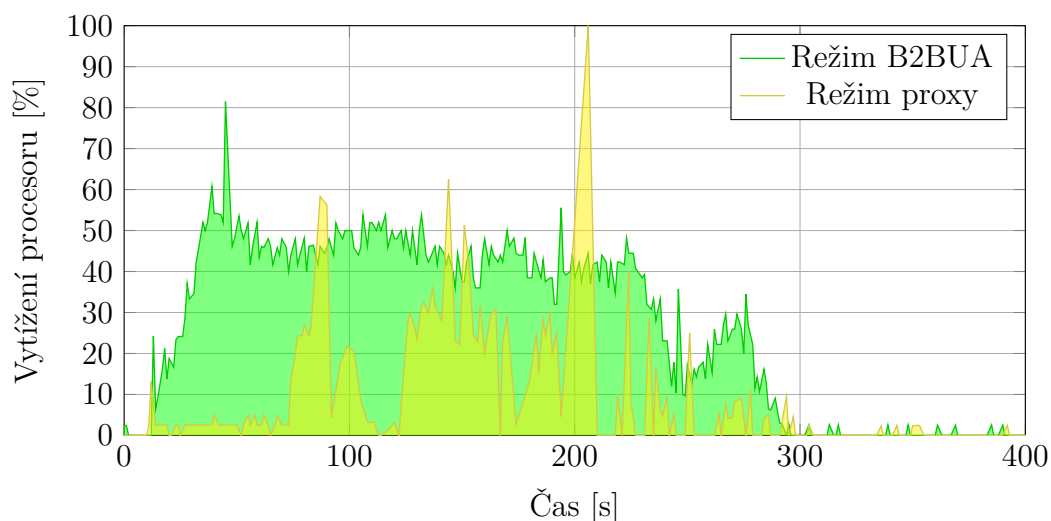


Obr. 4.11: Změřené RTP toky ústředny Asterisk 11 v režimu proxy s 1536 MB RAM



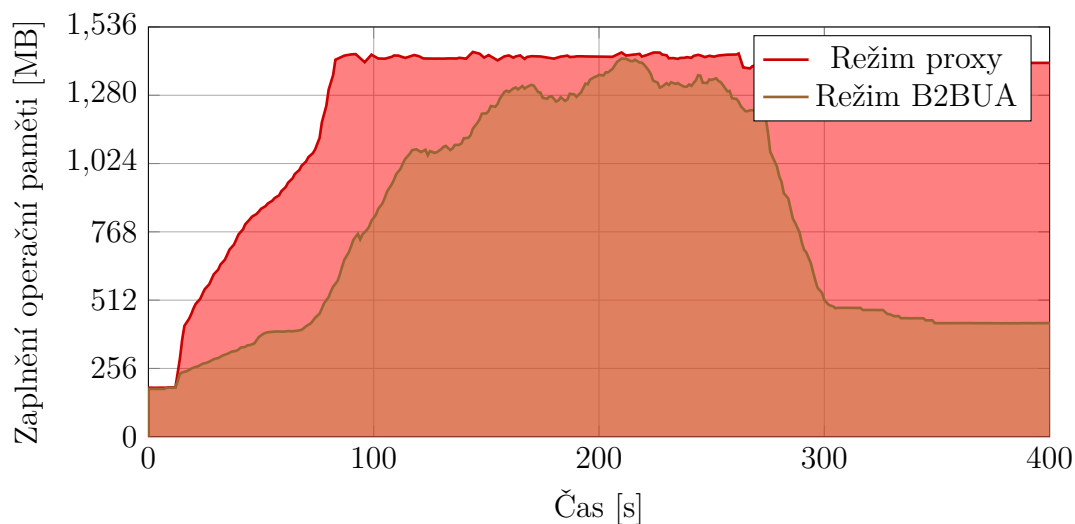
Obr. 4.12: Změřený počet relací ústředny Asterisk 11 v režimu proxy s 1536 MB RAM

Níže z obrázku 4.13 lze zpozorovat, že při vytížení procesoru u režimu proxy a B2BUA jsou značné rozdíly. Je to dáno zejména z toho důvodu, že v proxy režimu neprochází RTP toky přímo přes ústřednu. Proto také generuje menší zatížení.

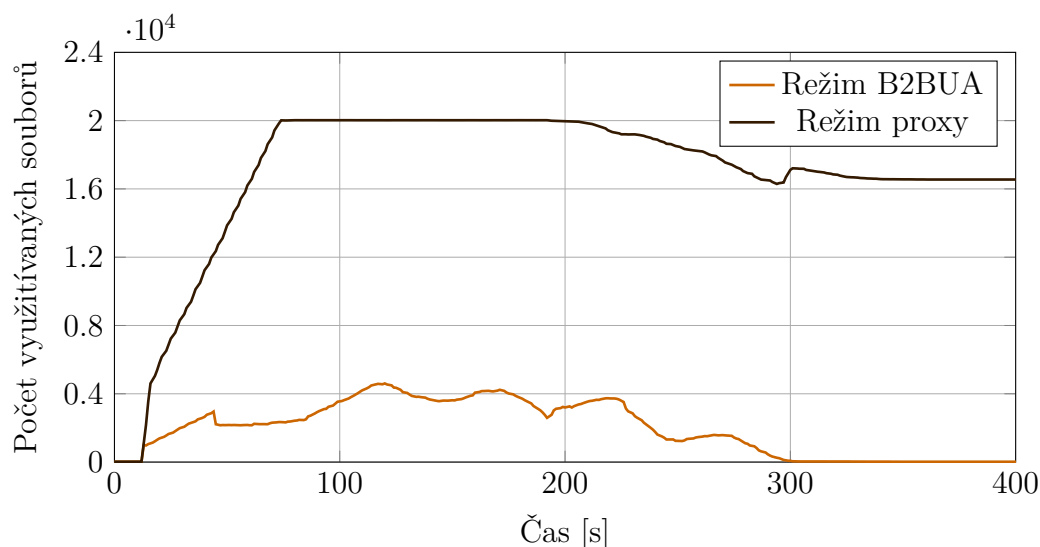


Obr. 4.13: Vytížení procesoru při simulaci počtu relací u ústředny Asterisk 11, v režimu B2BUA a proxy s 1536 MB RAM

K rychlejšímu zaplnění RAM podle obrázku 4.14 dochází u režimu proxy. Stejně tak je také u obrázku 4.15 pro proxy režim znatelný až čtyřnásobný nárůst otevřených souborů.



Obr. 4.14: Zaplnění operační paměti při simulaci počtu relací u ústředny Asterisk 11, v režimu B2BUA a proxy s 1536 MB RAM



Obr. 4.15: Využití souborů při simulaci počtu relací u ústředny Asterisk 11, v režimu B2BUA a proxy s 1536 MB RAM

### 4.3.2 Výkon a stabilita ústředny Asterisk 13

#### B2BUA režim

Výsledky měření na novější PBX Asterisk 13 s doinstalovaným modulem PJSIP byly dle očekávání lepší oproti *chan\_sip* modulu, jak je vidět na obrázku A.1 znázorňujícím toky dat a na obrázku A.2 znázorňujícím relace. Kde bylo při nastavení 1536 MB RAM dosaženo 331 aktivních hovorů. Při nižší konfiguraci paměti tomu bylo velmi podobně – 329 hovorů. Což je zhruba nárůst 30hovorů oproti ústředně Asterisk 11 a *chan\_sip* stacku. Vytížení CPU zde bylo podobné opět, jak v případě předchozí ústředny, ovšem s mírným zvýšením. Přesto však ústředna reagovala na příkazy a byla stále schopná provozu. Nárůst byl zaznamenán též u počtu otevřených souborů a to v případě obou konfigurací RAM paměti.

Režim proxy je možné u ústředny též nakonfigurovat v souboru `pjsip.conf` pomocí parametru `direct_media=yes` a následně by měly být hlasová i video data přenášeny přímo mezi klienty. Nicméně i přes tuto konfiguraci se mi daného režimu nepodařilo dosáhnout a ústředna se z neznámých důvodů stále chovala jako B2BUA. Proto v proxy režimu již měřeno nebylo.

### 4.3.3 Výkon a stabilita ústředny Freeswitch

#### B2BUA režim

Měření na ústředně Freeswitch bylo třeba několikrát zopakovat, jelikož při jejím maximálním vytížení přestal systém reagovat a stal se neovladatelným i v případě, že ústředně byly přiděleny pouze dvě jádra z celkových čtyř, jak bylo popsáno v kapitole 4.1.1. V některých případech ústředna i přestala ukončovat hovory a jelikož se po doběhnutí testu nevrátila do svého normálního pracovního stavu ani po delším doběhu, bylo nutné ručně restartovat celý server. Tento úkon však poškodil databázi ústředny, a ta již dále nebyla schopna správně fungovat. Nedá se tedy říct, že v době vysokého zatížení je ústředna stabilní a ovladatelná.

V případě úspěšného měření znázorněného na obrázku A.3 a A.4, kdy se byla ústředna schopná vrátit do svého normálního stavu, bylo dosaženo maxima 295 hovorů u nastavení s 1536 MB RAM, což je srovnatelné s ústřednou Asterisk 11, překvapivě však s nižší RAM pamětí to bylo 328 aktivních hovorů. Tento rozdíl lze vysvětlit nestabilitou ústředny, která se v průběhu testu jen velmi těžko dala ovládat. Počet otevřených souborů (385) zde byl rovněž vyšší než poloviční hodnota u konfigurace s 1536MB RAM pamětí, kde tomu bylo 145 souborů.

#### Proxy režim

Zde, co se týče vytížení CPU a stability OS nebyla situace o moc lepší oproti předchozí konfiguraci ústředny. Při nastavení s 768 MB RAM měla ústředna s naměřeným počtem 1512 relací nejnižší výsledek z ústřednen v proxy režimu a také velmi vysokou hodnotu průměrné doby sestavení spojení 2808 ms. Při 1536 MB RAM tomu bylo s touto dobou spojení přesně naopak s časem 36 ms. Zajímavé je, že podobné hodnoty ústředna vykazovala i při více měřeních. S vyšší pamětí byl počet relací (viz obr. A.5 a A.6) v průměru a s počtem 2391 srovnatelný s ústřednou Asterisk 11. Tento velký rozdíl může být ovlivněný právě velikostí operační paměti.

### 4.3.4 Výkon a stabilita ústředny OpenSER

#### Proxy režim

Jelikož OpenSER slouží hlavně jako páteří směřovač, proto lze na této ústředně realizovat pouze relace v režimu proxy. Opět, jak tomu bylo v předchozích případech



hlavně u ústředny Asterisk, u obou konfigurací operační paměti byly maximálních počty relací téměř stejné. U 1536 MB RAM se dle obrázku A.7 a A.8 naměřil počet 2795 relací, s nižší pamětí tomu bylo 2726 relací. Sestavení hovorů se zde dle očekávání pohybovalo velmi nízko a to okolo 110 ms a průměrné vytížení CPU bylo okolo 3 %, což je zanedbatelná hodnota. Využitá RAM paměť nebyla zcela zaplněná až do maxima. Její hodnoty se pohybovaly okolo 330MB.

### **4.3.5 Výkon a stabilita ústředny Kamailio**

#### **Proxy režim**

Výsledky dosažených relací ústředny Kamailio pro 1536 MB RAM znázorněny na obrázku A.9 a A.10 byly velice podobné (2806 aktivních relací), jakož tomu bylo i u OpenSER ústředny, stejně tak i pro nižší operační paměť (2768 aktivních relací).

## 4.4 Měření maximálního počtu transakcí

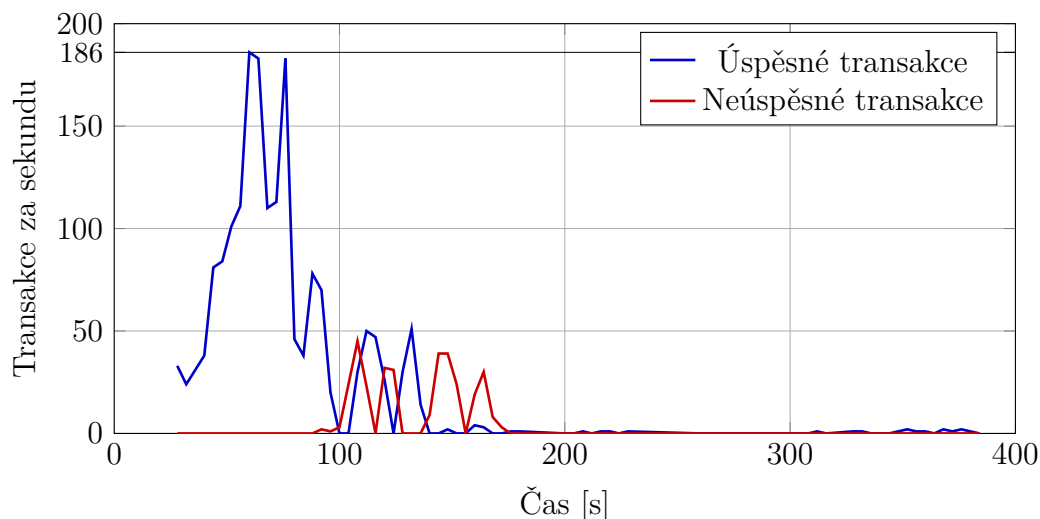
Pro větší přehlednost v práci byly některé grafy z měření přesunuty do přílohy. Jedná se o grafy změřených transakcí při nastavení operační paměti na 1536 MB. Jiné grafy, jako je měření vytížení procesoru, operační paměti a měření počtu otevřených souborů byly zase z práce vyjmuty (jsou přiloženy pouze jako elektronická příloha) a jejich charakteristika byla pouze slovně okomentována a to z důvodu jejich velkého množství. Pro některé ústředny se totiž všechny tyto charakteristiky měřily celkem ve čtyřech typech konfigurace, čímž by počet těchto grafů zbytečně navyšoval obsah práce. Proto byly charakteristiky vloženy pouze jako vzor pro první ústřednu Asterisk 11 a to s 1536 MB RAM. Doba sestavení relace se při tomto měření nesledovala, jelikož se měřil počet dosažených transakcí.

V každém grafu se změřenými transakcemi je vyznačeno maximum dosažený transakcí, které ústředna úspěšně vykonala. Takové maximum většinou nastává, když ještě ústředna není přehlcena požadavky a stíhá transakce vyřizovat.

### 4.4.1 Měření transakcí ústředny Asterisk 11

#### B2BUA režim

S možností větší operační paměti se podařilo naměřit, jak je vidět z obrázku 4.16, 186 transakcí. Vytížení procesoru bylo již poměrně velké a jeho průměr se pohyboval okolo 26-33 % a systém občas nereagoval. Znatelný byl též počet otevřených souborů, a to 10662. V režimu B2BUA je to tak nejvyšší dosažený počet ze všech ústředen. S menší operační pamětí tomu bylo daleko méně, okolo 3071 souborů.



Obr. 4.16: Změřené transakce ústředny Asterisk 11 v režimu B2BUA s 1536 MB RAM

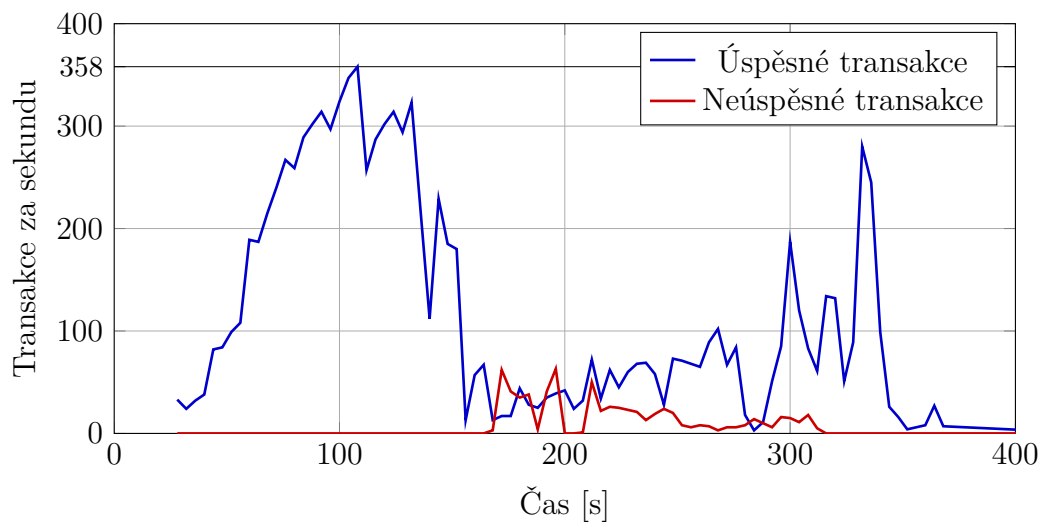
Na grafu je znatelné, že jakmile ústředna překročí svůj maximální možný výkon, začne docházet k takzvaným *retransmissionům*, což je přeposlání ztraceného paketu. Tyto pakety již ústředna nestačí vyřizovat a přetíží se. Takovýto stav může hlásit i v okně konzole na obrázku 4.17.

```
[Apr 28 22:16:47] WARNING[3478]: chan_sip.c:4050 retrans_pkt: Retransmission time
out reached on transmission 528db477c3cfd72@192.168.20.200 for seqno 102 (Critical
Request) -- See https://wiki.asterisk.org/wiki/display/AST/SIP+Retransmissions
Packet timed out after 32000ms with no response
```

Obr. 4.17: Výpis z konzole Asterisk 11

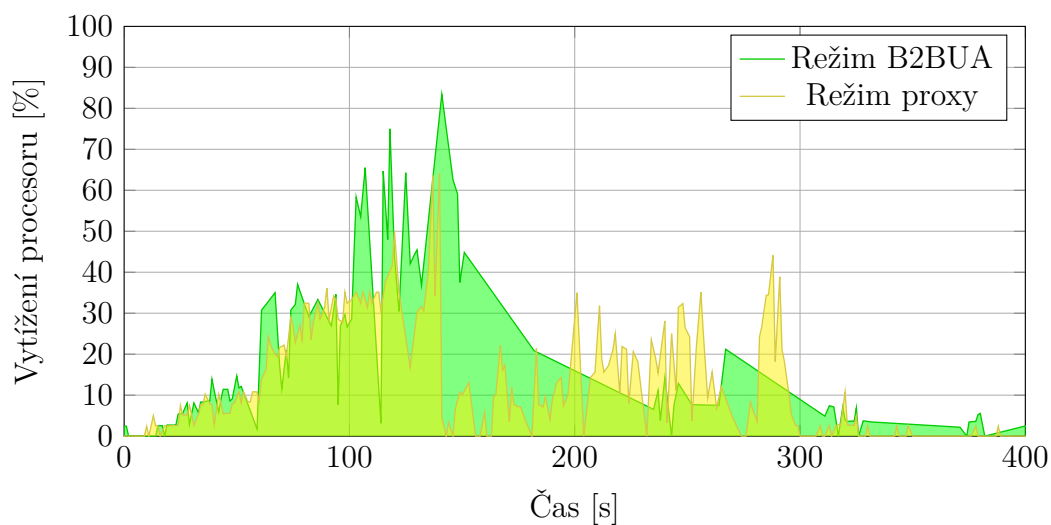
## Proxy režim

Oproti B2BUA, v režimu proxy se podařilo naměřit při 1536 MB RAM až 358 transakcí za sekundu, což je znatelný rozdíl. Jak je také vidět z obrázku 4.18, jakmile generátor začal snižovat zatížení, ústředna opět byla schopná vyřizovat transakce oproti předchozímu grafu na obrázku 4.16. To je dáno hlavně proxy režimem, protože RTP toky již neprocházejí skrz ústřednu, a také operační paměť, jelikož procesor a jeho vytížení se pohybovalo mezi 7-13 %.



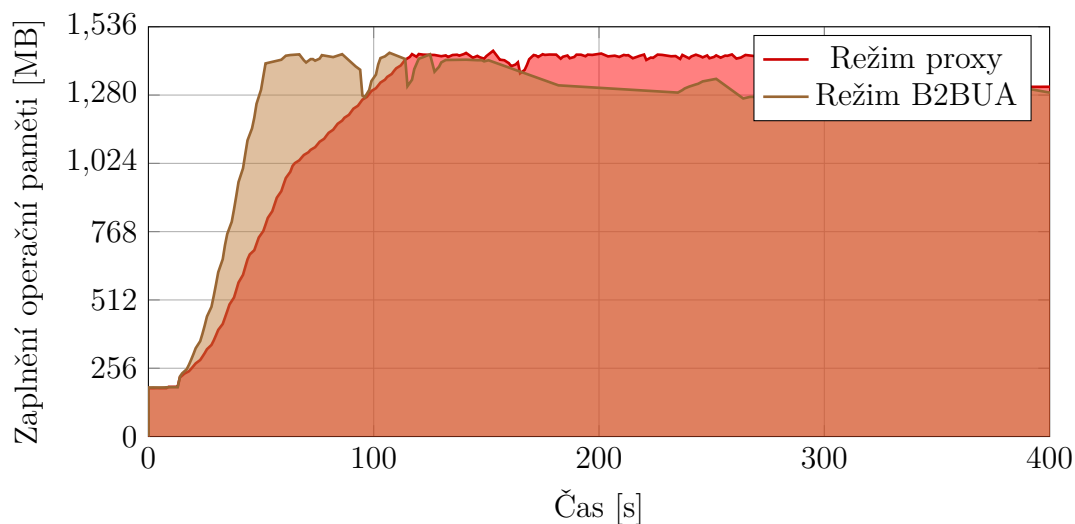
Obr. 4.18: Změřené transakce ústředny Asterisk 11 v režimu proxy s 1536 MB RAM

Jak již bylo zmíněno výše, z grafu na obrázku 4.19 je vidět, že v proxy režimu je ústředna daleko více stabilnější a využívá méně výkonu na procesoru.



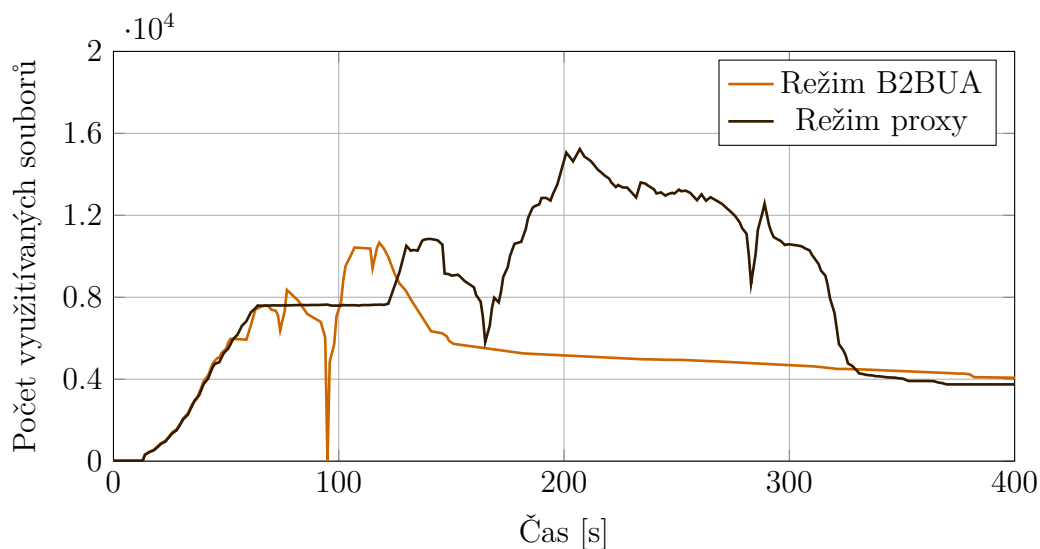
Obr. 4.19: Vytížení procesoru při simulaci počtu transakcí u ústředny Asterisk 11, v režimu B2BUA a proxy s 1536 MB RAM

Srovnání dvou režimů a jejich zaplnění operační paměti (obr. 4.20) je téměř identické. V obou případech se vyšplhala obsazená kapacita na maximum.



Obr. 4.20: Zaplnění operační paměti při simulaci počtu transakcí u ústředny Asterisk 11, v režimu B2BUA a proxy s 1536 MB RAM

Vzhledem k tomu, že v režimu proxy bylo dle předchozího grafu (obr. 4.18) naměřeno téměř dvojnásobný počet transakcí, logicky také vzrostl počet otevřených souborů (obr. 4.21)



Obr. 4.21: Využití souborů při simulaci počtu transakcí u ústředny Asterisk 11, v režimu B2BUA a proxy s 1536 MB RAM

## 4.4.2 Měření transakcí ústředny Asterisk 13

### B2BUA režim

Jak již bylo vysvětleno v podkapitole 4.3.2, na ústředně bylo možné pouze měření v režimu B2BUA. Výsledky PJSIP stacku nebyly příliš uspokojující. S nižší nastavenou operační pamětí dosáhl 69 transakcí za sekundu, což je nejméně ze všech měřených ústředěn. Avšak ovladatelný byl stále velmi dobře. Při 1536 MB RAM bylo naměřeno viz obrázek A.11, 134 transakcí za sekundu, což opět není oproti ostatním ústřednám mnoho.

## 4.4.3 Měření transakcí ústředny FreeSWITCH

### B2BUA režim

Stejně, jak při měření relací v kapitole 4.3.3, byl FreeSWITCH i při měření transakcí velmi nestabilní. Zejména u testu s menší operační pamětí, kde bylo sice dosaženo 89 transakcí, ale zhruba v polovině měření již ústředna nebyla schopná zvládat další požadavky. K podobnému výsledku došlo i s 1536 MB RAM, 86 transakcí, avšak v tomto případě ústředna vykonávala během celého měření práci, což je vidět na obrázku A.12. Po měření FreeSWITCH stále vyřizoval požadavky a do normálního stavu se vrátil zhruba za 15 minut.

### Proxy režim

Při proxy režimu bylo dosaženo s vyšší operační pamětí (viz obr. A.13) 123 transakcí, s menší pamětí tomu bylo 102 transakcí. Což se oproti klasickému B2BUA režimu příliš neliší. Vytížení ústředny bylo i zde pro oba typy konfigurací paměti velmi vysoké. Zajímavé však je, že ústředna měla otevřených velmi málo souborů (cca 78), operační paměť však měla zcela zaplněnou.

## 4.4.4 Měření transakcí ústředny OpenSER

### Proxy režim

V počtu transakcí se OpenSER oproti ostatním ústřednám velmi odlišil. U nastavení s 768 MB RAM bylo naměřeno 514 transakcí za sekundu, s 1536 MB RAM

to pak bylo 694 transakcí (obr. A.14). Po výkonnostní stránce byl oproti popisu v podkapitole 4.3.4 neměnný.

#### **4.4.5 Měření transakcí ústředny Kamilio**

##### **Proxy režim**

Ústředna Kamilio dosáhla nad očekávání v počtu transakcí daleko lepších výsledků než její předchůdce OpenSER. S vyšší operační pamětí bylo naměřeno 1085 transakcí za sekundu viz obrázek A.15, což je nejlepší výsledek z ústředen při měření transakcí. S dvojnásobně menší operační pamětí dosáhla maximálních 934 transakcí za sekundu.

## 4.5 Měření s jinými kodeky

Podobně jako měření maximálních relací probíhalo i měření s jinými kodeky. S tím rozdílem, že jsem si pouze pozměnil kodeky na generátoru jak na straně *server*, tak i na straně *klient*. Měřilo se na ústředně Asterisk 11, Asterisk 13 a FreeSWITCH. Kamailio a OpenSER fungují jako proxy server, tudíž by zde změna kodeků neměla smysl. Bylo tedy měřeno v režimu B2BUA.

Samozřejmě nastavit kodek bylo třeba i na ústředně. Na Asterisku 11 a 13 stačilo pozměnit hodnotu `allow` v souborech `sip.conf` a `pjsip.conf`. Na ústředně FreeSWITCH bylo třeba editovat soubor `vars.xml`. Volaný a volající tedy spolu s ústřednou měli nastavený stejný kodek, z čehož plyne, že ústředna neprováděla transkódování a hlasová data přes ní protékaly v pass-through režimu. To znamená, že nejsou využívány prostředky ústředny pro překódování audia v reálném čase, což by zcela jistě znamenalo rapidní pokles na výkonu a výsledný počet relací na ústředně by se podstatně snížil.

První kodek `alaw` (G.711a) byl již otestován v kapitole 4.3, dále byly vybrány kodeky, které byly dostupné jak v testeru, tak i u ústředen a to G.723 a G.729. Kodek G.729, jak bylo popsáno v podkapitole 4.2.1, byl pro ústřednu Asterisk 13 potřeba doinstalovat.

### 4.5.1 Chování ústředny Asterisk 11 s odlišnými kodeky

S menší operační pamětí měla ústředna horší výsledky oproti standardnímu kodeku `alaw` (obr. A.16) avšak s trochu nižším procesorovým vytížením (tab. A.5). Tyto horší hodnoty lze vysvětlit přeplněnou pamětí. Jakmile ale byla paměť navýšena na 1536 MB RAM, ústředny s oběma kodeky G.723 a G.729 dosáhly o 50 hovorů více, než tomu bylo u `alaw` kodeku.

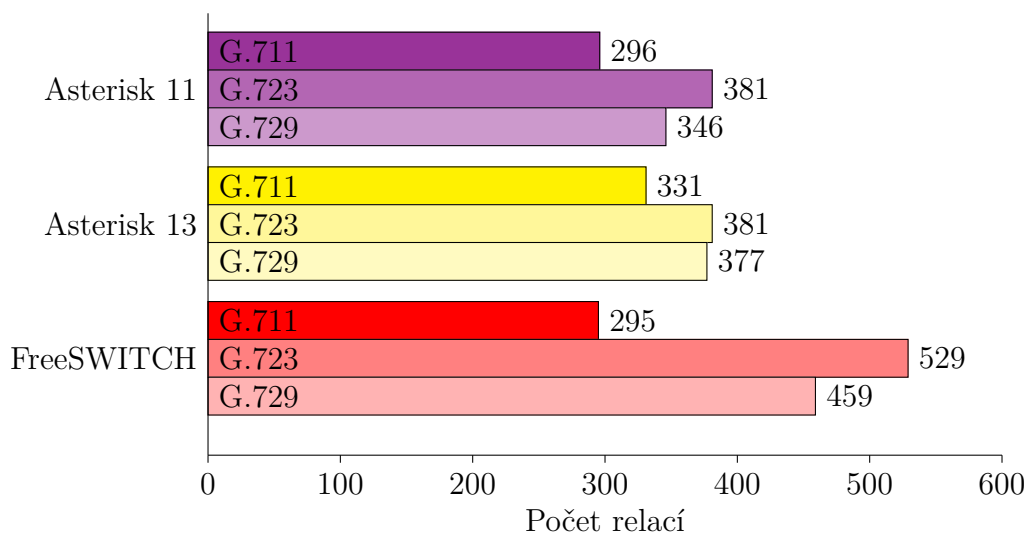
### 4.5.2 Chování ústředny Asterisk 13 s odlišnými kodeky

U ústředny Asterisk 13 tentokrát byl výsledek s kodeky G.723 a G.729 lepší v obou případech nastavení operační paměti. Nicméně ústředna vždy po zhruba dvou minutách provozu přes tyto kodeky havarovala.



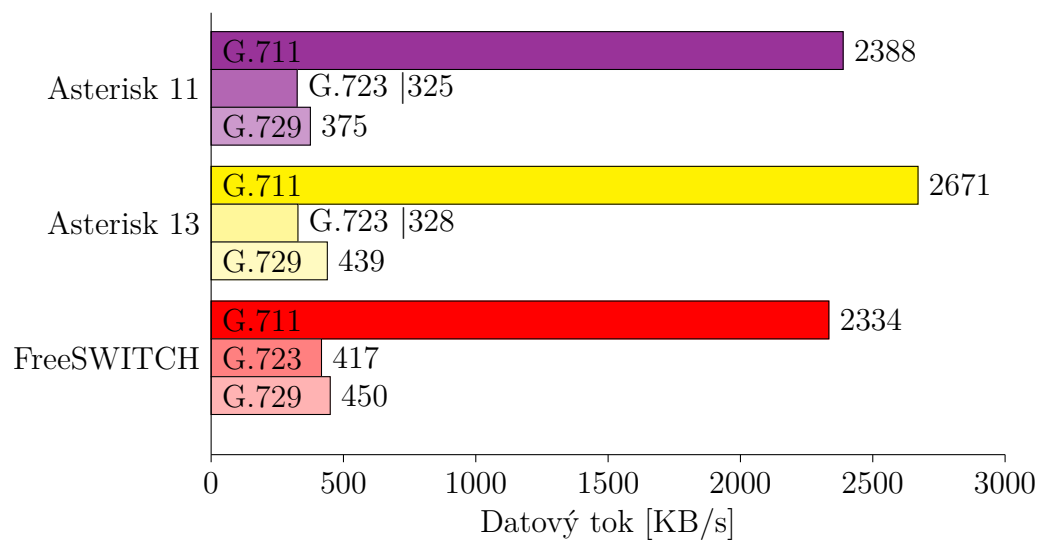
### 4.5.3 Chování ústředny FreeSWITCH s odlišnými kodeky

Zde se u FreeSWITCH ústředny oproti minulým měřením relací a transakcí situace obrátila a při měření s jinými kodeky vykazoval výborné výsledky. Nejlepšího výsledku dosáhl s kodekem G.723 a to až přes 500 aktivních relací (obr. 4.22). Byl stále velmi dobře ovládaný a stabilní během celé doby hovoru.



Obr. 4.22: Maximální dosažené relace v porovnání s různými kodeky s 1536MB operační pamětí

Pro zjištění důvodu, proč ústředny pracují s komplexnějšími kodeky stabilněji se stačilo podívat na intenzitu datového toku, který jimi tekl viz obrázek 4.23. Tento graf znázorňuje aktuální rychlost toku dat při maximálním dosaženém počtu hovorů na ústředně, která byla nakonfigurovaná na 1536 MB RAM. Jak je tedy vidět, kodeky G.723 a G.729 pracují s mnohonásobně vyšší kompresí zvuku a tím pádem ústřednami teče mnohem méně dat, což je méně zatěžuje a tím jsou i více stabilnější. Tato situace byla právě znatelná nejvíce u ústředny FreeSWITCH.



Obr. 4.23: Maximální toky dat přes ústředny s 1536MB operační pamětí a různými kodeky

Detailnější charakteristiky pro kompletní přehled výkonu ústředen je možné najít v elektronické příloze. Podobně jak v předchozích kapitolách pro větší přehlednost byly zbývající grafy přiloženy na konec práce do kapitoly A.3.3.

## 4.6 Měření s modifikovanými zprávami

Do ústředen byly pomocí generátoru Spirent zasílány upravené zprávy, které měly ověřit jejich stabilitu a bezpečnost. Veškeré podrobné výsledky z měření útoků na ústředny jsou také dostupné v tabulce A.6, kde jsou vypsány veškeré odpovědi, které ústředny učinily jako jejich reakci na přijatou zprávu.

Vybraných patnáct níže vypsáných útoků bylo prováděno dle doporučení RFC 4475 pod názvem *Torture Test Messages*, které se zaměřuje na speciální typy zpráv, které mohou způsobit neočekávaný chod ústředny nebo dokonce i její pád. Tímto si ověříme stabilitu každého stacku v zadaných a nainstalovaných ústřednách. Veškeré bezpečnostní testy tedy nebyly vytvářeny náhodně, přestože tester umožňuje jejich editaci [24, 6, 7].

Před začátkem testů jsem si nastavil všechny ústředny a jejich SIP stacky do debugovacího režimu. Tyto data jsem pak následně shromažďoval a analyzoval. Analýzu lze také provést pomocí programu Wireshark.

Spuštění debugovacího módu pro ústřednu Asterisk 11:

```
|| sip set debug on
```

Spuštění debugovacího módu pro ústřednu Asterisk 13:

```
|| pjsip set logger on
```

Spuštění debugovacího módu pro ústřednu FreeSWITCH:

```
|| sofia global siptrace on
```

V případě potřeby debugování v ústředně OpenSER přímo do konzole je nutné zadat příkaz `openser -E -dddd`, pro Kamailio pak `kamailio -E -dddd`.

### 4.6.1 Starší INVITE žádost

Princip spočívá v zaslání staré metody INVITE, která je definována dle RFC 2543. Aktuálně se však již používá nových žádostí dle novější verze SIP 2.0 a RFC 3261. Tímto se tedy otestuje zpětná kompatibilita ústředen. V případě, že ústředna není zpětně kompatibilní, může reakce na takovou zprávu znamenat až její pád.

### Struktura modifikované hlavičky

```
INVITE sip:UserB@example.com SIP/2.0
Via: SIP/2.0/UDP iftgw.example.com
From: <sip:+13035551111@iftg.client.example.net;user=phone>
Record-Route: <sip:UserB@example.com;maddr=ss1.example.com>
To: sip:+16505552222@ss1.example.net;user=phone
Call-ID: inv2543.1717@iftg.client.example.com
CSeq: 56 INVITE
Content-Type: application/sdp
```

Na zprávu reagovaly všechny ústředny kromě ústředny Asterisk 11, ta zprávu zahodila. Kamailio a OpenSER zaslaly odpověď 100 Trying a dále již ne zprávu nereagovaly.

## 4.6.2 Více hodnot v položce Content-Length

Na ústřednu je zaslána zpráva typu OPTIONS, která obsahuje dvě položky typu Content-Length. Tato položka udává velikost těla zprávy v bytech. Test ukazuje, jak se bude ústředna chovat v případě, že budou ve zprávě tyto položky dvě s odlišnými čísly.

### Struktura modifikované hlavičky

```
OPTIONS sip:user@example.com SIP/2.0
Via: SIP/2.0/UDP host5.example.net;branch=z9hG4bK293423
To: sip:user@example.com
From: sip:other@example.net;tag=3923942
Call-ID: mcl01.fhn2323orihawfdoa3o4r52o3irsdf
CSeq: 15932 OPTIONS
Content-Length: 13
Max-Forwards: 60
Content-Length: 5
Content-Type: text/plain
```

OpenSER zprávu zahodila, ostatní ústředny odpověděly chybou. Kamailio a Fre-eSWITCH dokonce reagovaly, že položka Content-Length je chybná.

## 4.6.3 Odpověď neznámého typu

Na požadavek je zaslána odpověď s číslem větším, než 699. To znamená, že by měla být zpráva zahozena.

### Struktura modifikované hlavičky

```
SIP/2.0 4294967301 better not break the receiver
Via: SIP/2.0/UDP 192.0.2.105;branch=z9hG4bK2398ndaoe
Call-ID: bigcode.asdof3uj203asdnf3429uasdhfas3ehjasdfas9i
CSeq: 353494 INVITE
From: <sip:user@example.com>;tag=39ansfi3
To: <sip:user@example.edu>;tag=902jndnke3
Content-Length: 0
Contact: <sip:user@host105.example.com>
```

Tuto zprávu všechny ústředny zahodily. Například reakce stacku PJSIP vypadala následovně:

```
Error processing packet from 192.168.20.241:1069: Invalid/
unexpected SIP status code (PJSIP_EINVALIDSTATUS)
```

### 4.6.4 Mezery v poli příjemce

Ve zprávě typu OPTIONS do pole To jsou úmyslně zaneseny mezery. Ústředna by tuto zprávu měla odmítnout a zaslat odpověď 400 Bad Request response.

### Struktura modifikované hlavičky

```
OPTIONS sip:user@example.org SIP/2.0
Via: SIP/2.0/UDP host4.example.com:5060;branch=z9hG4bKkdju43234
Max-Forwards: 70
From: "Bell, Alexander" <sip:a.g.bell@example.com>;tag=433423
To: "Watson, Thomas" < sip:t.watson@example.org >
Call-ID: badaspec.sdf0234n2nds0a099u23h3hnnw009cdkne3
Accept: application/sdp
CSeq: 3923239 OPTIONS
```

Asterisk 11 odpověděl chybou 404, Asterisk 13 chybou 401. To znamená, že zprávu nepřijaly. Stejně tak jí nepřijaly ani OpenSER a Kamailio, které jí rovnou zahodily. Avšak FreeSWITCH odpověděl jako 200 OK, což je dle doporučení RFC chyba.

### 4.6.5 Neznámá metoda

Je vyslána nová metoda s názvem NEWMETHOD. Stejně tak nesouhlasí ani značka CSeq. Jelikož je toto neznámá metoda, odpověď na ní by měla být 501 Not Implemented a to zvláště u proxy serverů.

### Struktura modifikované hlavičky

```
NEWMETHOD sip:user@example.com SIP/2.0
To: sip:j.user@example.com
From: sip:caller@example.net;tag=34525
Max-Forwards: 6
Call-ID: mismatch02.dj0234sxdf13
CSeq: 8 INVITE
Contact: <sip:caller@host.example.net>
Via: SIP/2.0/UDP host.example.net;branch=z9hG4bKkdjuw
Content-Type: application/sdp
```

Nejlépe dle doporučení reagovala Asterisk 11, která odpověděla s chybou 501 Method Not Implemented. Kamailio reagovala podobně, sice s chybou 400, avšak se zprávou CSeq method does not match request method. OpenSER zprávu zahodil, což není vhodné řešení. Ostatní ústředny oznámily chybu.

### 4.6.6 Různé typy transportu zprávy

V položce Via u metody OPTIONS je uvedeno několik typů transportů (UDP, TCP, SCTP, TLS...). Tato zpráva by měla být zpracována a měla by na ní přijít odpověď.

### Struktura modifikované hlavičky

```
OPTIONS sip:user@example.com SIP/2.0
To: sip:user@example.com
From: <sip:caller@example.com>;tag=323
Max-Forwards: 70
Call-ID: transports.kijh4akdnaqjkwendsasfdj
Accept: application/sdp
CSeq: 60 OPTIONS
Via: SIP/2.0/UDP t1.example.com;branch=z9hG4bKkdjuw
Via: SIP/2.0/SCTP t2.example.com;branch=z9hG4bKklasjdhf
Via: SIP/2.0/TLS t3.example.com;branch=z9hG4bK2980unddj
Via: SIP/2.0/UNKNOWN t4.example.com;branch=z9hG4bKasd0f3en
Via: SIP/2.0/TCP t5.example.com;branch=z9hG4bK0a9idfnee
```

Kamailio a OpenSER zde zprávu zahodily. Ústředny Asterisk obě oznámily chybu. FreeSWITCH opět potvrdila jako 200 OK, kterou zprávu akceptovala.

### 4.6.7 Nulová hodnota položky Max-Forwards

Hodnota v této položce se sníží pokaždé o jedna, jakmile zpráva projde přes daný prvek. Ústředna by měla reagovat formou odpovědi 483 Too Many Hops.

#### Struktura modifikované hlavičky

```
OPTIONS sip:user@example.com SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=3ghsd41
Call-ID: zeromf.jfasdlfnm2o2l43r5u0asdfas
CSeq: 39234321 OPTIONS
Via: SIP/2.0/UDP host1.example.com;branch=z9hG4bKkdjuw2349i
Max-Forwards: 0
Content-Length: 0
```

Správnou odpovědí 483 Too Many Hops reagovaly pouze OpenSER a Kamailio. FreeSWITCH jí přijal jako 200 OK a ústředny Asterisk značily chybu 404 a 401.

### 4.6.8 Špatně interpretovaný parametr v poli Contact

Metoda REGISTER obsahuje pole Contact, kde byl přidán zanesen neznámý parametr unknownparam.

#### Struktura modifikované hlavičky

```
REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/UDP saturn.example.com:5060;branch=z9hG4bKkdjuw
Max-Forwards: 70
From: sip:watson@example.com;tag=DkfVgjkrtMwaerKKpe
To: sip:watson@example.com
Call-ID: cparam01.70710@saturn.example.com
CSeq: 2 REGISTER
Contact: sip:+19725552222@gw1.example.net;unknownparam
```

Na tuto zprávu ústředny Asterisk a FreeSWITCH reagovaly chybou. OpenSER a Kamailio jí naopak zahodily.

### 4.6.9 Chybějící identifikátor transakce

V poli Via se nachází povinná položka branch. Tato položka udává unikátní identifikátor pro transakci. Identifikátor vždy začíná hodnotou z9hG4bK a za ní následuje dalších sedm znaků. Pro tuto zprávu však těchto sedm znaků bylo odstraněno.

Ústředna by přesto měla na tuto zprávu odpovědět. Styl této identifikace byl definován pro SIP 2.0.

### Struktura modifikované hlavičky

```
OPTIONS sip:user@example.com SIP/2.0
To: sip:user@example.com
From: sip:caller@example.org;tag=33242
Max-Forwards: 3
Via: SIP/2.0/UDP 192.0.2.1;branch=z9hG4bK
Accept: application/sdp
Call-ID: badbranch.sadonfo23i420jv0as0derf3j3n
CSeq: 8 OPTIONS
l: 0
```

I přes chybějící identifikátor zaslal FreeSWITCH odpověď 200 OK, na rozdíl od OpenSER a Kamailio, které zprávu zahodily a obě ústředny Asterisk odpověděly chybou.

## 4.6.10 Žádost se špatnou adresou

Tato registrační žádost obsahuje kontaktní adresu, do které byl zanesen speciální znak. I přes toto by měla ústředna registraci akceptovat.

### Struktura modifikované hlavičky

```
REGISTER sip:example.com SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=8
Max-Forwards: 70
Call-ID: regescrt.k345asrl3fdbv@192.0.2.1
CSeq: 14398234 REGISTER
Via: SIP/2.0/UDP host5.example.com;branch=z9hG4bKkdjuw
M: <sip:user@example.com?Route=%3Csip:sip.example.com%3E>
L:0
```

Stejně jak v případě chybného pole **Contact** i zde OpenSER a Kamailio zprávu zahodily a ostatní ústředny oznámily chybu. Registrace se tedy nevydařila na žádné z ústředen.



#### 4.6.11 Nesprávná hodnota velikosti těla zprávy

Velikost těla zprávy `Content-Length` je označena jako 9999. Ve skutečnosti je označena chybně, jelikož tak velké tělo neexistuje. Očekávaná odpověď je 400 `Bad Request` error v případě protokolu UDP.

##### Struktura modifikované hlavičky

```
INVITE sip:user@example.com SIP/2.0
Max-Forwards: 80
To: sip:j.user@example.com
From: sip:caller@example.net;tag=93942939o2
Contact: <sip:caller@hungry.example.net>
Call-ID: clerr.0ha0isndaksdjweiafasdk3
CSeq: 8 INVITE
Via: SIP/2.0/UDP host5.example.com;branch=z9hG4bK-39234-23523
Content-Type: application/sdp
Content-Length: 9999
```

Všechny ústředny na zprávu reagovaly. OpenSER jako jediný poslal odpověď typu 100 `Trying`. Zbývající ústředny odpověděly chybou. Kamailio dokonce specifikovala, že se jedná o položku `Content-Length`.

#### 4.6.12 Odstraněná mezera mezi jménem a adresou

Metoda `OPTIONS` obsahuje pole `From`, kde je definováno jméno a adresa odesílatele. Mezi těmito dvěma položkami bývá mezera. V modifikované zprávě však byla mezera odstraněna. Obecně by toto mělo být hodnoceno, jako validní zpráva. Naopak mezera, která se obvykle mezi tyto položky přidává je chybou ve specifikaci RFC 3261, která by měla být v dalších revizích odstraněna.

##### Struktura modifikované hlavičky

```
OPTIONS sip:user@example.com SIP/2.0
To: sip:user@example.com
From: "caller"<sip:caller@example.com>;tag=323
Max-Forwards: 70
Call-ID: lwsdisp.1234abcd@funky.example.com
CSeq: 60 OPTIONS
Via: SIP/2.0/UDP funky.example.com;branch=z9hG4bKkdjuw
l: 0
```

Správně se zachoval FreeSWITCH a zprávu akceptoval formou odpovědi 200 `OK`. Ústředny Asterisk zprávu odmítly s chybou a Kamailio a OpenSER jí dokonce

zahodily.

### 4.6.13 Parametry oddělené středníkem v poli Request-URI

V poli Request-URI je záměrně zanesen středník a speciální znak. Ústředna by takovou zprávu měla přijmout a tuto hodnotu zpracovat jako `user;par=u@example.net`.

#### Struktura modifikované hlavičky

```
OPTIONS sip:user;par=u%40example.net@example.com SIP/2.0
To: sip:j_user@example.com
From: sip:caller@example.org;tag=33242
Max-Forwards: 3
Call-ID: semiuri.0ha0isndaksdj
CSeq: 8 OPTIONS
Accept: application/sdp, application/pkcs7-mime, multipart/mixed
      , multipart/signed, message/sip, message/sipfrag
Via: SIP/2.0/UDP 192.0.2.1;branch=z9hG4bKkdjuw
1: 0
```

Zprávu akceptovala pouze FreeSWITCH. Kamailio a OpenSER jí zahodily, jak to dělají s většinou chybných zpráv, které nejsou definované dle RFC doporučení. Asterisk ústředny jí zase odmítly s chybou.

### 4.6.14 Neuzavřený parametr

Pole To obsahuje jméno adresáta a jeho adresu. Jméno adresáta by mělo být správně uzavřeno mezi uvozovkami. V této zprávě je však poslední uvozovka vynechána. Očekávaná odpověď by měla být 400 Bad Request error.

#### Struktura modifikované hlavičky

```
INVITE sip:user@example.com SIP/2.0
To: "Mr. J. User <sip:j.user@example.com>
From: sip:caller@example.net;tag=93334
Max-Forwards: 10
Call-ID: quotbal.aksdj
Contact: <sip:caller@host59.example.net>
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.59:5050;branch=z9hG4bKkdjuw39234
Content-Type: application/sdp
Content-Length: 152
```

Asterisk 13, OpenSER a Kamailio zprávu zahodily. Správně byla vyhodnocena ústřednou FreeSWITCH, jako výše definovaná chyba 400. Asterisk 11 oznámila chybu 404 Not Found. Následovně vypadala například chybová zpráva v logu stacku PJSIP:

```
Error processing 485 bytes packet from UDP 192.168.20.241:1076 :
    PJSIP syntax error exception when parsing '' header on line
    2 col 5
```

#### 4.6.15 Chybné pole Accept

Metoda INVITE s tímto polem označuje jakého typu má být tělo odpovědi. Standardně zde bývá hodnota `application/sdp`, avšak tentokrát byla zpráva upravena na hodnotu `text/nobodyKnowsThis`. Odpověď by měla být typu 406 Not Acceptable, popřípadě je vhodná i odpověď 400 Bad Request error.

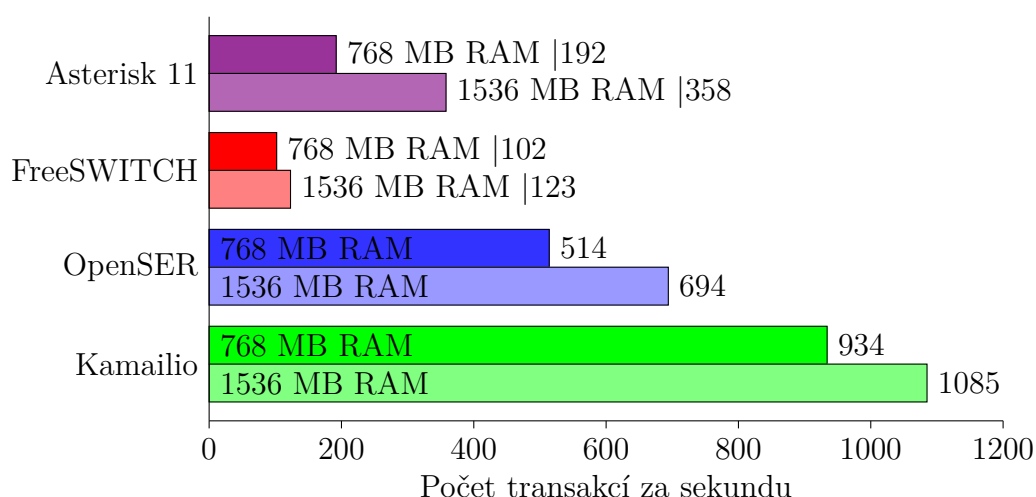
##### Struktura modifikované hlavičky

```
INVITE sip:user@example.com SIP/2.0
To: sip:j_user@example.com
Contact: <sip:caller@host15.example.net>
From: sip:caller@example.net;tag=234
Max-Forwards: 5
Call-ID: sdp01.ndaksdj9342dasdd
Accept: text/nobodyKnowsThis
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.15;branch=z9hG4bKkdjuw
Content-Length: 150
Content-Type: application/sdp
```

Nejvhodnější odpověď typu 406 Not Acceptable zaslal opět FreeSWITCH. Asterisk 11 reagoval s chybou 404 Not Found a Asterisk 13 401 Unauthorized. Kamailio a OpenSER pak zasílaly odpověď 100 Trying.

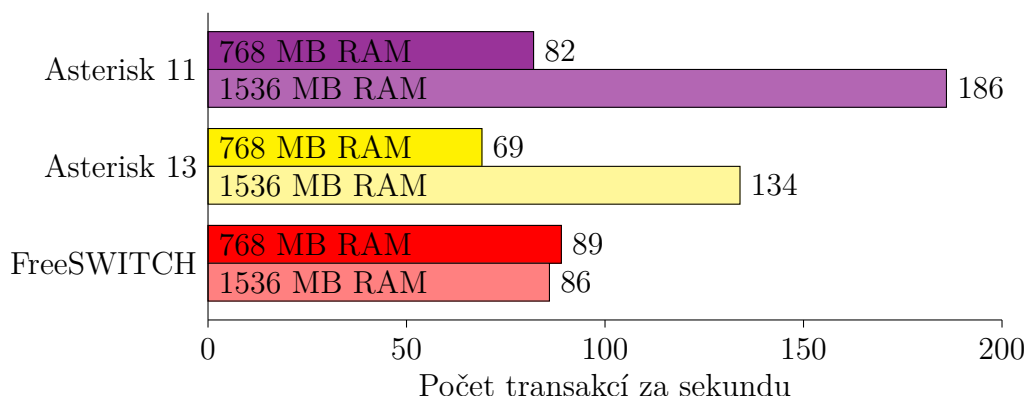
## 4.7 Celkové srovnání výsledků

Pro lepší přehlednost v naměřených hodnotách je vhodné si kompletní výsledky souhrnně srovnat. U vybraných ústředen a výkonu jejich SIP stacků lze konstatovat, že nejvýkonnější ústředna pro maximální přenos transakcí je Kamailio, jak lze shlédnout na sloupcovém grafu (obr. 4.24). Ihned za ní následuje její předchůdce OpenSER. Nutno však dodat, že z jejich charakteristik na obrázku A.14 a A.15 se jednalo o špičkový výkon, který se ustálil zhruba v jeho polovině. S největší pravděpodobností by ale bylo možné dosáhnout dalších výsledků postupným a detailnějším laděním ústředny při její kompilaci. Nejhuře si v proxy režimu při přenosu transakcí vedl FreeSWITCH, který byl rovněž velmi těžko ovladatelný. Ve všech těchto případech byl výkon znatelně lepší s větší operační pamětí.



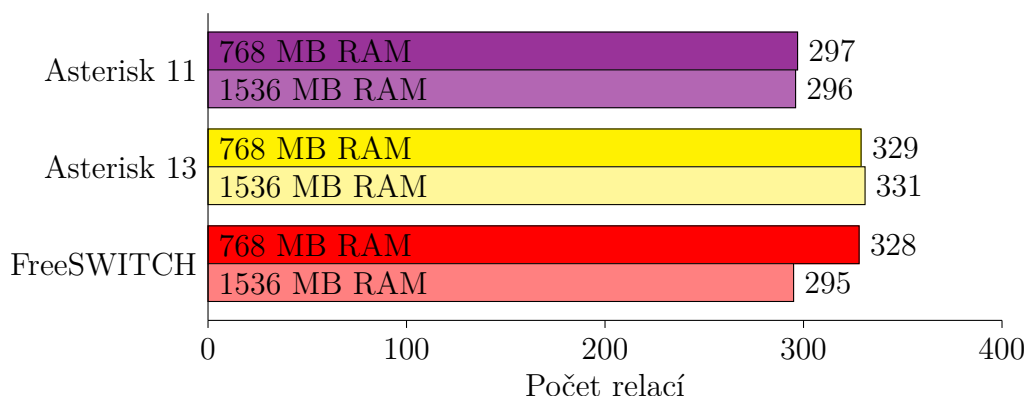
Obr. 4.24: Maximální dosažené transakce ústředen v režimu proxy

S konfigurací ve formě vyšší operační paměti, přenos transakcí jako B2BUA zvládal nejlépe Asterisk 11 a jeho *chan\_sip* stack, který byl v tomto případě výkonnější, jak PJSIP na obrázku 4.25.



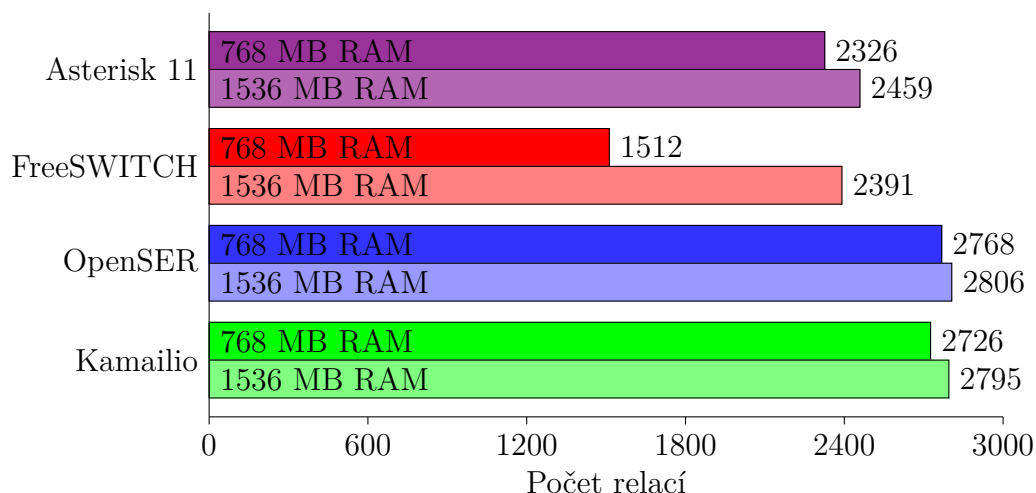
Obr. 4.25: Maximální dosažené transakce ústředen v režimu B2BUA

Pokud se zaměříme na nejlepší výsledky v oblasti maximálního počtu relací, všechny ústředny zvládaly hovory spojovat poměrně uspokojivým způsobem, jak v režimu proxy (obr. 4.27), tak i jako B2BUA (obr. 4.26).



Obr. 4.26: Maximální dosažené relace ústředen v režimu B2BUA

Pro proxy režim dosáhly maximum Kamailio a OpenSER s téměř totožnými výsledky. Menší problémy však nastaly u FreeSWITCHe, kde z důvodů jeho velkého zatížení, zejména u 768 MB RAM jeho výkon velmi kolísal.



Obr. 4.27: Maximální dosažené relace ústředn v režimu proxy

Kompletní naměřené výsledky lze shlédnout v tabulkách A.2-A.4 v příloze na konci práce. Charakteristiky, které zde nebyly uveřejněny se nachází v elektronické podobě, jako příloha práce a obsahují všechny grafy včetně nárůstů RAM paměti, výkonu CPU a počtu otevřených souborů pro každé měření.

Při měření odlišných kodeků bylo zjištěno, že v případě, když ústředna využívá kodek s menším datovým tokem, její zátěž je mnohonásobně nižší a dokáže v takovém případě zvládat i více relací naráz. To bylo prověřeno dle obrázku A.17 a 4.23. Bylo také zpozorováno, že během menšího toku dat skrz ústřednu s kodeky vyšší komprese nevyužila maximum své operační paměti, jak tomu bylo u kodeku G.711, kde ve všech případech byla dosažena maximální kapacita paměti vyjímaje ústřednu Kamailio a OpenSER.

Útokům na bezpečnost bylo u všech testech úspěšně odoláno. Stacky ve většině případu zprávy odmítaly nebo úplně zahazovaly. Nejčastěji se RFC doporučení držela ústředna FreeSWITCH, přesto ne úplně ve všech měřeních.

## 5 ZÁVĚR

Cílem této diplomové práce bylo měření výkonosti a stability několika vybraných knihoven implementujících protokol SIP, jejich testování vůči útokům a celkové srovnání využívaných prostředků systému při zatížení pobočkové ústředny. Tyto knihovny neboli stacky byly součástí předem zadaných ústředn, které se při testování porovnávaly vyjímaje PJSIP stacku, který bylo nejprve nutné do ústředny nainstalovat jako zásuvný modul. Měřilo se množství úspěšně vykonaných relací a transakcí a to ve dvou typech konfigurace operační paměti – 768 MB a 1536 MB RAM, dále rozdílový výkon při změně kodeků na ústřednách pro režim pass-thru a nakonec se zjišťovalo zabezpečení, jak dané stacky reagují na modifikované SIP zprávy. Rychlý přehled všech typů měření lze nalézt na začátku kapitoly 4. Testy se prováděly přes generátor zátěže Spirent TestCenter C1.

Jako nejvhodnější systém pro testování byla z důvodů stability vybrána linuxová distribuce CentOS. Je také nutné vzít v potaz, že výsledky se odvíjejí z jedné konkrétní hardwarové konfigurace a že se jednalo o virtualizované prostředí. V případě, že by byl nasazen server s vyšším výpočetním výkonem a nejednalo by se pouze o virtualizaci, počet relací a transakcí by se tímto zvýšil.

Do operačního systému byly zvlášť nainstalovány a konfigurovány ústředny Asterisk 11, Asterisk 13 s přidaným modulem PJSIP, FreeSWITCH, OpenSER a Kamailio. Každá pracovala na odlišném SIP stacku. Vytknout snad lze jen příliš malou rozdílnost mezi ústřednou OpenSER a Kamailio, jelikož Kamailio je přejmenovaná ústředna OpenSER a je její nástupce. Tato malá odlišnost byla ověřena, jak lze shlédnout například z obrázku 4.26. Pro testy stability a výkonu prováděné na testeru byly nadefinovány účty dvou stran – klient a server, tyto dvě strany pak za pomoci ústředny a jejího stacku uskutečňovaly společné relace. Počet těchto relací konstantně rostl podle předem nadefinované charakteristiky. Charakteristika byla úmyslně nadefinována tak, aby zde ústředna nebyla schopná dosáhnout požadovaného zatížení a abychom si tak mohli zjistit její maximální výkon. Příslušným softwarem pak byly vygenerovány veškeré potřebné statistiky. K přesnějším výsledkům byl použit také krátký vlastní skript, který sbíral data z ústředny (obr. 4.1.2).

Délka jednoho hovoru byla nastavena přesně na dobu celého prováděného testu. Tímto bylo zajištěno, že klient nebo server během půlky testu nezavěsí a ústředna se plně vytíží. V případě, že by se provádělo krátkých za sebou jdoucích 5sekundových hovorů, bylo by složitější a méně přesné určit maximální vytížení ústředny.

Z testů výkonu, jak již bylo zmíněno si vedly nejlépe ústředny OpenSER a Kamai-

lio, kde úspěšně zvládly spojit téměř 3000 hovorů za sekundu. Také vynikaly v počtu vykonaných transakcí, zvláště pak ústředna Kamilio (srovnání viz obr. 4.24). Změna kodeků v případě těchto ústředen neměla význam, jelikož slouží pouze jako proxy servery. Při bezpečnostních testech se ústředny chovaly spíše konzervativněji a spoustu modifikovaných zpráv jednoduše ignorovaly. Což i to může být aspekt v jejich rychlosti.

Ústředna FreeSWITCH naopak vyšla z porovnávacího testu u klasického kodeku G.711 nejhůře. Při maximálním vytížení se jevila velice nestabilní a byla neovladatelná. Opakovaně se stalo, že přestala zavěšovat hovory a poškodila vlastní databázi. Z těchto důvodů byly před každým testem databáze ústředen vyčištěny. Požději z testů rozdílných kodeků však ale bylo zjištěno, že ústředně dělal problém vysoký tok dat, kterým byl kodek G.711 na vině. Tímto se ústředně velmi rychle zaplnila operační paměť a vzrostl výkon procesoru. V případě, kdy byl přenos zvuku učiněn přes kodek G.723, tok dat skrz ústřednu několikanásobně klesl, tím i vzrostla její výkonnost a v testu rozdílných kodeků vyšla nejlépe (obr. 4.22). V některých případech měla i poměrně přesné reakce a odpovědi na bezpečnostní útoky (obr. A.6).

Asterisk 11 a Asterisk 13 s doinstalovaným modulem PJSIP se z výsledků měření jeví jako průměrné ústředny. V testech sice nedosahují nadlimitních výsledků, nicméně jsou poměrně stabilní a nevyužívají tolik výkonu při práci, jako je tomu například u ústředny FreeSWITCH. Z této zkušenosti si tedy i trůfám tvrdit, že by se hodily do infrastruktury větších společností ve spojení s ústřednou Kamilio sloužící jako proxy server. Na většinu útoků odpovídaly stacky z obou ústředen Asterisk chybou. Rozdíl u PJSIP stacku byl takový, že jeho nejčastější odpovědi byly 401 Unauthorized, zato u chan\_sip stacku 404 Not Found.

Během měření nebylo na žádném ze stacků nalezeno závažnější chyby a i přesto, že ne všechny ústředny se držely podle RFC doporučení, tak útoky modifikovaných zpráv neovlivnily jejich chod ani stabilitu.



# LITERATURA

- [1] *Asterisk Project Wiki* [online]. 2012 [cit. 2. 12. 2015]. Dostupné z URL: <<https://wiki.asterisk.org/wiki/display/AST/Home>>.
- [2] *Asterisk vs. FreeSWITCH* [online]. 1995-2015, poslední aktualizace 28. 5. 2008 [cit. 25. 11. 2015]. Dostupné z URL: <<http://anders.com/cms/266>>.
- [3] BAZALA, David. *Telekomunikace & VoIP telefonie I*. 1. vyd. Praha: BEN - technická literatura, 2006, 222 s. ISBN 80-7300-201-9.
- [4] BRYANT, Russell. *Asterisk: the definitive guide*. Fourth edition. Sebastopol: O'Reilly, 2013, xxxvi, 806 p. ISBN 9781449332426.
- [5] COLLINS, Daniel. *Carrier grade voice over IP*. 2nd ed. New York: McGraw-Hill, c2003, xviii, 522 p. ISBN 0071406344.
- [6] DWIVEDI, Himanshu. *Hacking VoIP: protocols, attacks, and countermeasures*. San Francisco: No Starch Press, c2009, xii, 211 p. ISBN 1593271638.
- [7] ENDLER, David a Mark COLLIER. *Hacking exposed VoIP: voice over IP security secrets & solutions*. New York: McGraw-Hill, c2007, xxvii, 539 s. ISBN 978-0-07-226364-0.
- [8] *FreeSWITCH Wiki* [online]. 2010, poslední aktualizace 22. 6. 2014 [cit. 20. 11. 2015]. Dostupné z URL: <[https://wiki.freeswitch.org/wiki/Main\\_Page](https://wiki.freeswitch.org/wiki/Main_Page)>.
- [9] HARTPENCE, Bruce. *Packet guide to Voice over IP*. First edition. Sebastopol, CA: O'Reilly Media, Inc., 2013, xiv, 224 p. ISBN 1449339670.
- [10] HAZLETT, Paul and Simon MILES and Greger V. TEIGRE. *SER - Getting Started*. ONsip.org, 2012, 85 p.
- [11] JOHNSTON, Alan B. *SIP: understanding the Session Initiation Protocol*. Boston: Artech House, 2001, xxi, 201 p. ISBN 1580531687.
- [12] *Kamailio SIP Server Documentation Wiki*[online]. 2011, poslední aktualizace 6. 12. 2015 [cit. 6. 12. 2015]. Dostupné z URL: <<http://www.kamailio.org/wiki/>>.
- [13] *libosip Documentation*[online]. 2005, poslední aktualizace 22. 2. 2005 [cit. 25. 10. 2015]. Dostupné z URL: <<http://www.gnu.org/software/osip/doc/html/>>.

- [14] MINESSALE, Anthony. *FreeSWITCH Cookbook: over 40 recipes to help you get the most out of your FreeSWITCH server : [quick answers to common problems]*. Olton Birmingham [England]: Packt Pub., 2012, ii, 134 p. ISBN 978-1-84951-540-5.
- [15] *OpalVoipWiki* [online]. 2007, poslední aktualizace 27. 11. 2012 [cit. 25. 10. 2015]. Dostupné z URL: <<http://wiki.opalvoip.org/>>.
- [16] *PJSIP - Open Source SIP Stack* [online]. 2006-2008, poslední aktualizace 2. 10. 2015 [cit. 24. 10. 2015]. Dostupné z URL: <[https://www.resiprocate.org/Main\\_Page](https://www.resiprocate.org/Main_Page)>.
- [17] *reSIProcate* [online]. 2007-2015 [cit. 1. 11. 2015]. PJSIP Reference. Dostupné z URL: <<http://www.pjsip.org/pjsip/docs/html/>>.
- [18] ŠILHAVÝ, P. *Telekomunikační a informační systémy*. 1. Brno: Vysoké učení technické v Brně, 2014. s. 1-140. ISBN: 978-80-214-5027-1.
- [19] SISALEM, Dorgham. *SIP security*. Chichester, U.K.: Wiley, 2009, xiv, 336 p. ISBN 0470516364.
- [20] *Sofia-SIP Library* [online]. 2007-2011, poslední aktualizace 11. 3. 2011 [cit. 27. 10. 2015]. About Sofia-SIP. Dostupné z URL: <<http://www.voip-info.org/wiki/view/VOCAL>>.
- [21] *Spirent Avalanche* [online]. 2013, poslední aktualizace duben 2013 [cit. 16. 11. 2015]. Security testing and application performance. Dostupné z URL: <[http://www.spirent.com/~media/Datasheets/Broadband/PAB/Avalanche/Avalanche\\_Security\\_Testing\\_and\\_Application\\_Performance\\_Datasheet.pdf](http://www.spirent.com/~media/Datasheets/Broadband/PAB/Avalanche/Avalanche_Security_Testing_and_Application_Performance_Datasheet.pdf)>.
- [22] *The Internet Engineering Task Force (IETF)* [online]. 1999 [cit. 4. 11. 2015]. RFC 2543 – SIP: Session Initiation Protocol. Dostupné z URL: <<http://tools.ietf.org/html/rfc2543>>.
- [23] *The Internet Engineering Task Force (IETF)* [online]. 2002 [cit. 4. 11. 2015]. RFC 3261 – SIP: Session Initiation Protocol. Dostupné z URL: <<http://tools.ietf.org/html/rfc3261>>.
- [24] *The Internet Engineering Task Force (IETF)* [online]. 2006 [cit. 20. 4. 2016]. RFC 4475 – Session Initiation Protocol (SIP) Torture Test Messages. Dostupné z URL: <<http://tools.ietf.org/html/rfc4475>>.

- [25] VAN BOSSE, John G a Fabrizio U DEVETAK. *Signaling in telecommunication networks*. 2nd ed. Hoboken, N.J.: Wiley-Interscience, c2007, xiv, 810 p. ISBN 9780471662884.
- [26] *VOCAL* – *voip-info.org* [online]. 2003-2015, poslední aktualizace 19.8.2005 [cit. 23.10.2015]. VOCAL - Vovida Open Communications Applications Library. Dostupné z URL: <<http://www.voip-info.org/wiki/view/VOCAL>>.

## SEZNAM ZKRATEK

|       |  |
|-------|--|
| B2BUA | agent, který plní funkci proxy serverů a zároveň slouží jako koncový bod pro signalizaci – Back-to-Back User Agent |
| CLI   | příkazový řádek – Command-line Interface   |
| CPU   | procesor, elektronická jednotka v PC, která umí vykonávat strojové instrukce – Central Processing Unit             |
| DNS   | protokol pro přidělování doménových jmen – Domain Name System  |
| DTLS  | protokol, umožňující zabezpečit datagramové protokoly – Datagram Transport Layer Security                          |
| FTP   | protokol pro přenos souborů mezi dvěma PC – File Transfer Protocol   |
| GNU   | svobodný operační systém projektu GNU – GNU's Not Unix   |
| GPLv2 | druhá verze licence pro svobodný software – General Public License   |
| H.323 | jeden z protokolů sloužící k přenosu hlasu a videa   |
| HTTP  | internetový protokol sloužící k přenosu obsahu ve formátu HTML – Hypertext Transfer Protocol                       |
| HTTPS | nástavba protokolu HTTP pro zabezpečení spojení – Hypertext Transfer Protocol Secure                               |
| ICE   | rozšíření pro TURN protokol – Interactive Connectivity Establishment   |
| IETF  | komise pro technickou stránku internetu – Internet Engineering Task Force  |
| IM    | služba, pomocí které mohou uživatelé spolu chatovat v reálném čase – Instant Messaging                             |
| IMAP4 | protokol sloužící ke vzdálenému přístupu k emailové schránce – Internet Message Access Protocol                    |
| IP    | nejdůležitější protokol pracující na síťové vrstvě – Internet Protocol   |
| IPv4  | internetový protokol verze 4 – Internet Protocol version 4   |
| IPv6  | internetový protokol verze 6 – Internet Protocol version 6   |

|       |   |
|-------|---|
| LGPL  | několik dalších oprávnění přidanych do GPLv3 – Lesser General Public License  |
| LTS   | stabilní verze s dlouhodobou podporou – Long Term Support   |
| MGCP  | VoIP protokol sloužící pro řízení komunikace <i>media gateways</i> v prostředí IP – Media Gateway Control Protocol  |
| MPL   | licence pro svobodný software vydávaný nejen pod Mozilla Corporation – Mozilla Public License                       |
| OPAL  | open source stack obsahující také podporu SIP protokolu – Open Phone Abstraction Library                            |
| OS    | operační systém – Operating System  |
| PBX   | pobočková telefonní ústředna – Private Branch Exchange  |
| PC    | označení stolního počítače – Personal Computer  |
| PJSIP | stack protokolu SIP, sloužící například jako modul pro ústřednu Asterisk  |
| POP3  | protokol verze 3 pro stahování emailových zpráv – Post Office Protocol  |
| PID   | číslo procesu – Process Identifier  |
| RAM   | operační paměť s přímým přístupem umožňující zápis i čtení – Random Access Memory                                   |
| RFC   | řada doporučení a dokumentů popisující internetové protokoly – Request for Comments                                 |
| RTP   | protokol sloužící k doručování zvukových a obrazových dat – Real-time Transport Protocol                            |
| RTMP  | protokol vyvinutý firmou Macromedia pro přenášení videa, zvukových a datových proudů – Real Time Messaging Protocol |
| RTSP  | protokol pro jednosměrné doručování obsahu ve formě datového proudu – Real Time Streaming Protocol                  |
| SDP   | protokol, který popisuje vlastnosti relace pro multimediální přenos dat – Session Description Protocol              |
| SSL   | protokol pro zabezpečení komunikace – Secure Sockets Layer  |

|        |  |
|--------|--|
| SIP    | protokol pro inicializaci relací – Session Initiation Protocol   |
| SMTP   | protokol pro odesílání emailových zpráv – Simple Mail Transfer Protocol  |
| STUN   | protokol umožňující komunikaci skrz NAT využívanou typicky u VoIP služeb – Session Traversal Utilities for NAT   |
| TCP    | spolehlivý, spojově orientovaný protokol sloužící k přenosu dat na transportní vrstvě – Transmission Control Protocol  |
| TLS    | kryptografický protokol využívající zabezpečené komunikace, následovník SSL – Transport Layer Security   |
| TURN   | protokol, který napomáhá při průchodu přes NAT či firewall u multimediálních aplikací – Traversal Using Relays around NAT                                    |
| UAC    | koncové zařízení v síti VoIP, které se stará o iniciaci hovorů – User Agent Client   |
| UAS    | koncové zařízení v síti VoIP, které reaguje na příchozí požadavky – User Agent Server  |
| UDP    | nespolehlivý, spojově neorientovaný protokol – User Datagram Protocol  |
| VOCAL  | další z open source stacků financovaný skupinou Vovida – Vovida Open Communication Application Library   |
| VoIP   | technologie umožňující přenos hlasu a videa pomocí protokolu UDP a TCP – Voice over Internet Protocol  |
| WebRTC | rozhraní pro spouštění telefonních hovorů, videochatu a P2P sdílení souborů ve webovém prohlížeči bez využití zasuvných modulů – Web Real-Time Communication |

# SEZNAM PŘÍLOH

|   |           |
|---|-----------|
| <b>A Přílohy</b>  | <b>79</b> |
| A.1 Konfigurační soubory ústředn . . . . .                | 79        |
| A.1.1 Konfigurační soubory ústředny Asterisk 11 . . . . . | 79        |
| A.1.2 Konfigurační soubory ústředny Asterisk 13 . . . . . | 79        |
| A.1.3 Konfigurační soubory ústředny Freeswitch . . . . .  | 81        |
| A.1.4 Konfigurační soubory ústředny OpenSER . . . . .     | 82        |
| A.2 Skript pro měření výkonu . . . . .                    | 83        |
| A.3 Grafy . . . . .                                       | 84        |
| A.3.1 Měření relací . . . . .                             | 84        |
| A.3.2 Měření transakcí . . . . .                          | 89        |
| A.3.3 Měření s odlišnými kodeky . . . . .                 | 92        |
| A.4 Tabulky . . . . .                                     | 93        |
| <b>B Obsah přiloženého CD</b>                             | <b>99</b> |

## A PŘÍLOHY

### A.1 Konfigurační soubory ústředn

V následujících podkapitolách se nachází informace a konfigurační soubory, dle kterých byly ústředny plně funkční.

#### A.1.1 Konfigurační soubory ústředny Asterisk 11

Níže zmíněné vypsané soubory se nacházejí ve složce `/etc/asterisk/`.

Soubor `extensions.conf`

```
[from-internal]
exten=>1000,1,Dial(SIP/1000,20)
exten=>1001,1,Dial(SIP/1001,20)
```

Soubor `sip.conf`

```
[general]
transport=udp

[friends_internal](!)
type=friend
host=dynamic
context=from-internal
disallow=all
allow=ulaw

[1000](friends_internal)
secret=1234

[1001](friends_internal)
secret=1234
```

#### A.1.2 Konfigurační soubory ústředny Asterisk 13

Níže zmíněné vypsané soubory se nacházejí ve složce `/etc/asterisk/`.



## Soubor `extensions.conf`

Struktura souboru je velice podobná s jediným rozdílem, že SIP v závorkách nahradíme za nově použitý stack PJSIP.

```
[from-internal]
exten=>1000,1,Dial(PJSIP/1000,20)
exten=>1001,1,Dial(PJSIP/1001,20)
```

## Soubor `pjsip.conf`

V tomto případě je definice klapků pro PJSIP stack mírně složitější oproti původnímu stacku.

```
[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0

[endpoint_internal](!)
type=endpoint
context=from-internal
disallow=all
allow=ulaw

[auth_userpass](!)
type=auth
auth_type=userpass

[aor_dynamic](!)
type=aor
max_contacts=9999
remove_existing=yes

[1000](endpoint_internal)
auth=1000
aors=1000
[1000](auth_userpass)
password=1234
username=1000
[1000](aor_dynamic)

[1001](endpoint_internal)
auth=1001
aors=1001
```

```
[1001](auth_userpass)
password=1234
username=1001
[1001](aor_dynamic)
```

### A.1.3 Konfigurační soubory ústředny Freeswitch

#### Soubor switch.conf.xml

Soubor nalezneme pod cestou `/etc/freeswitch/autoload_configs/`. Níže je vy-  
psaná pouze jeho část pro povolení více relací. Zbytek kódu zůstal nezměněn.

```
...
<!-- Max number of sessions to allow at any given time. -->
  <param name="max-sessions" value="10000"/>
<!--Most channels to create per second -->
  <param name="sessions-per-second" value="10000"/>
...
```

#### Soubor 1000.xml

Soubor 1000.xml nacházející se v `/etc/freeswitch/directory/default/` obsa-  
huje, jak již název napovídá definici klapky 1000. Nastavení pro klapku 1001 je  
obdobné.

```
<include>
  <user id="1000">
    <params>
      <param name="password" value="${default_password}"/>
      <param name="vm-password" value="1000"/>
    </params>
    <variables>
      <variable name="toll_allow" value="domestic,international,
local"/>
      <variable name="accountcode" value="1000"/>
      <variable name="user_context" value="default"/>
      <variable name="effective_caller_id_name" value="Extension
1000"/>
      <variable name="effective_caller_id_number" value="1000"/>
      <variable name="outbound_caller_id_name" value="${
outbound_caller_name}"/>
      <variable name="outbound_caller_id_number" value="${
outbound_caller_id}"/>
```

```
<variable name="callgroup" value="techsupport"/>
</variables>
</user>
</include>
```

## A.1.4 Konfigurační soubory ústředny OpenSER

### Soubor openserctlrcl

Jelikož je obsah souboru poměrně rozsáhlý, níže jsou zmíněny pouze řádky, které byly odkomentovány a upraveny.

```
## your SIP domain
SIP_DOMAIN=192.168.20.200

## database type: MYSQL or PGSQL, by default none is loaded
DBENGINE=MYSQL

## database host
DBHOST=localhost

## database name
DBNAME=openser

## database read/write user
DBRWUSER=openser

## database read/write password
DBRWPW=openserrw

## database read only user
DBROUSER=openserro

## password for database read only user
DBROPW=openserro

## database super user
DBROOTUSER="root"

## type of aliases used: DB - database aliases; UL - usrloc
aliases
## - default: none
ALIASES_TYPE="DB"
```

## Soubor `opernser.cfg`

V souboru byly odkomentovány a editovány následující řádky.

```
loadmodule "/usr/lib/openser/modules/mysql.so"
modparam("usrloc", "db_mode", 2)
if (!www_authorize("192.168.20.200", "subscriber")) {
www_challenge("192.168.20.200", "0");
break;
};
```

## A.2 Skript pro měření výkonu

Skript přiložený níže byl využit pro měření výkonu procesoru, využití operační paměti a počtu otevřených souborů procesem ústředny.

```
#!/bin/bash
# Performance measurement

# Type of PBX
PBX=$(pidof freeswitch)

# Show info
printf "CPU [%%]: "
cat <(grep 'cpu ' /proc/stat) <(sleep 0.1 && grep 'cpu ' /proc/
stat) | awk -v RS="" '{print ($13-$2+$15-$4)*100/($13-$2+$15-
$4+$16-$5)}'

printf "Memory [MB]: "
free | grep Mem | awk '{print ($2-$4)/1024}'

printf "Open files: "
# PID='/proc/2285/fd'
PID='/proc/'$PBX'/fd'
ls -l $PID | wc -l

#####
# Memory measurement
date +%X' | tr '\n' ' ' >> mem.txt
free | grep Mem | awk '{print ($2-$4)/1024}' >> mem.txt

# CPU measurement
date +%X' | tr '\n' ' ' >> cpu.txt
```

```

cat <(grep 'cpu ' /proc/stat) <(sleep 0.1 && grep 'cpu ' /proc/
stat) | awk -v RS="" '{print ($13-$2+$15-$4)*100/($13-$2+$15-
$4+$16-$5)}' >> cpu.txt

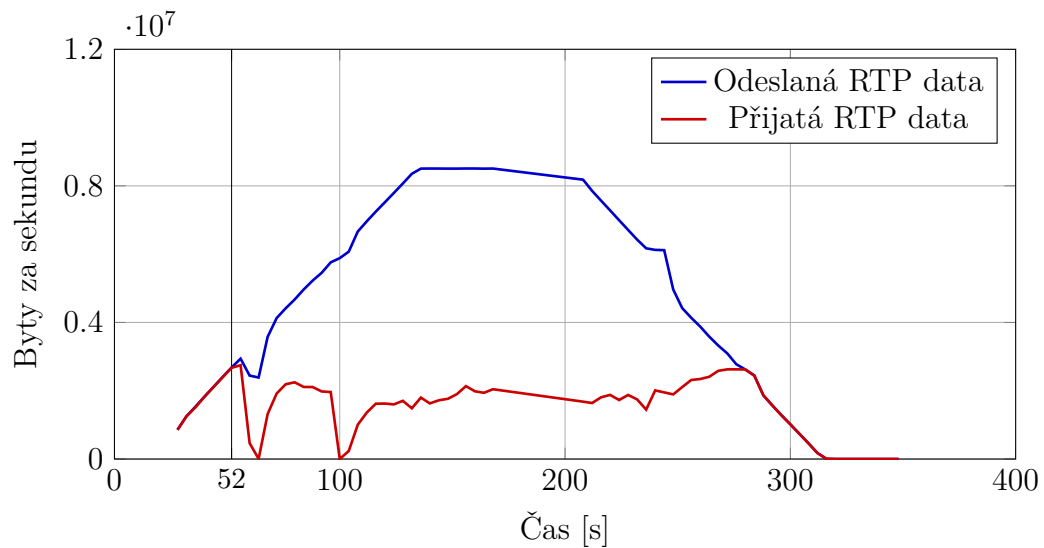
# Open files measurement
date '+%X' | tr '\n' ' ' >> files.txt
ls -l $PID | wc -l >> files.txt

```

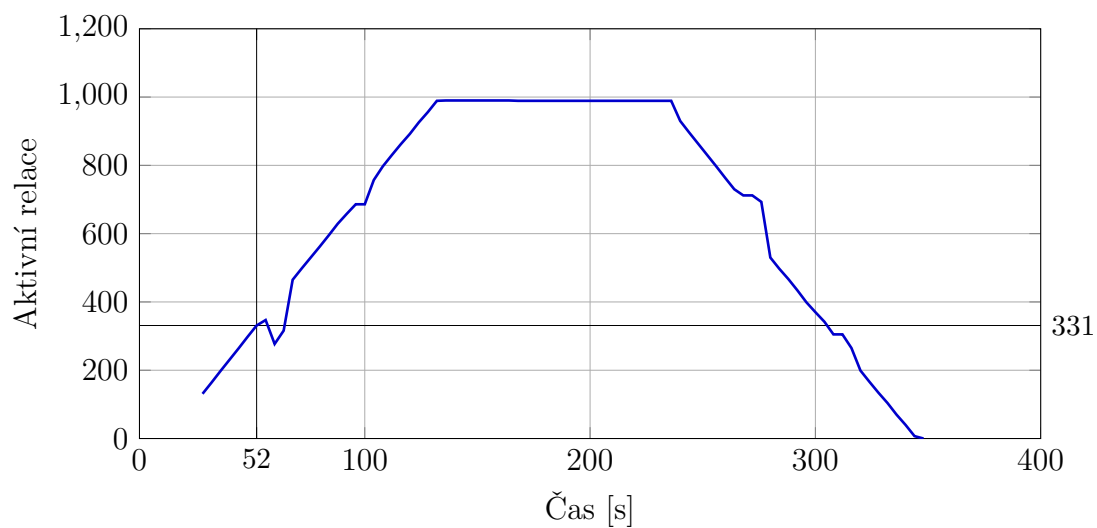
## A.3 Grafy

### A.3.1 Měření relací

#### Asterisk 13

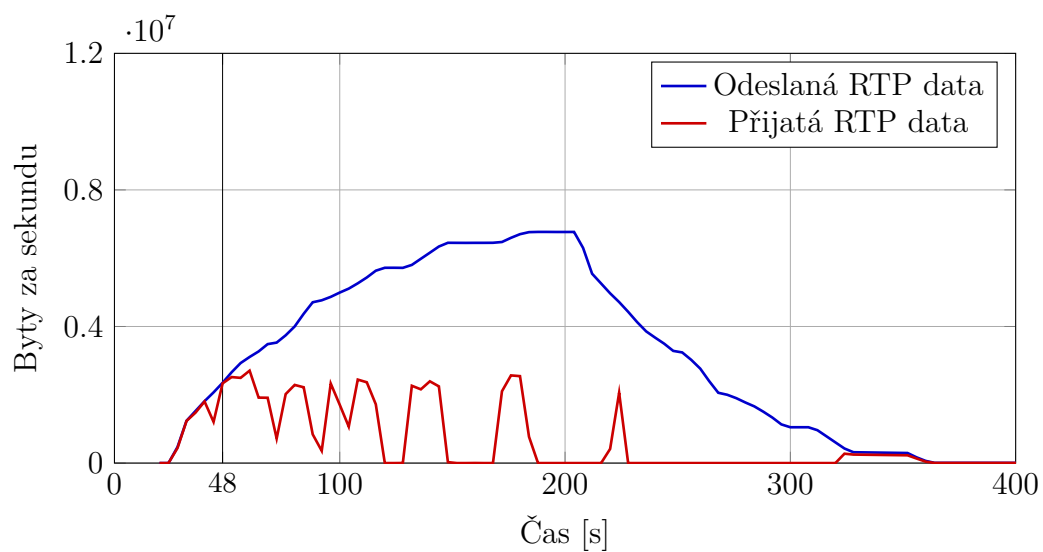


Obr. A.1: Změřené RTP toky ústředny Asterisk 13 v režimu B2BUA s 1536 MB RAM

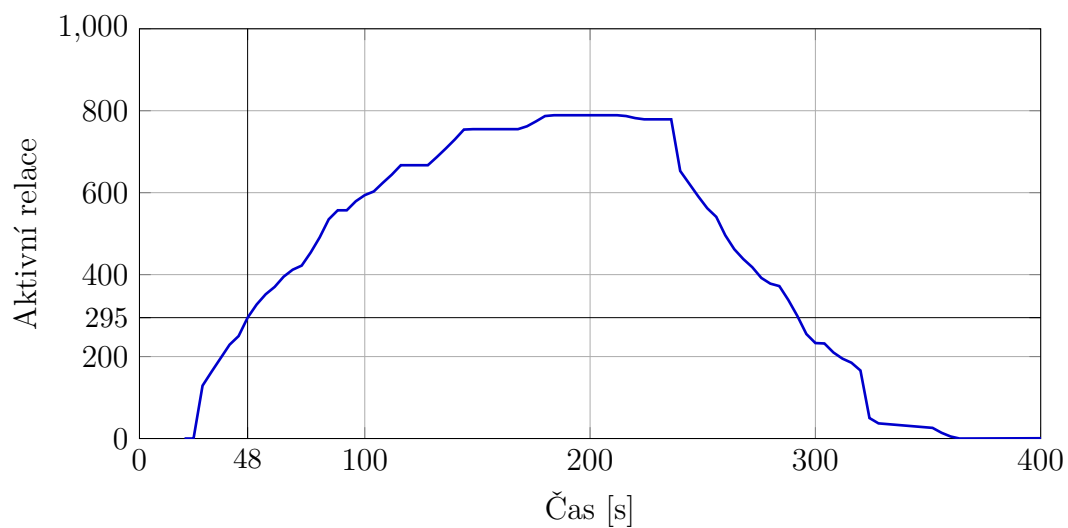


Obr. A.2: Změřený počet relací ústředny Asterisk 13 v režimu B2BUA s 1536 MB RAM

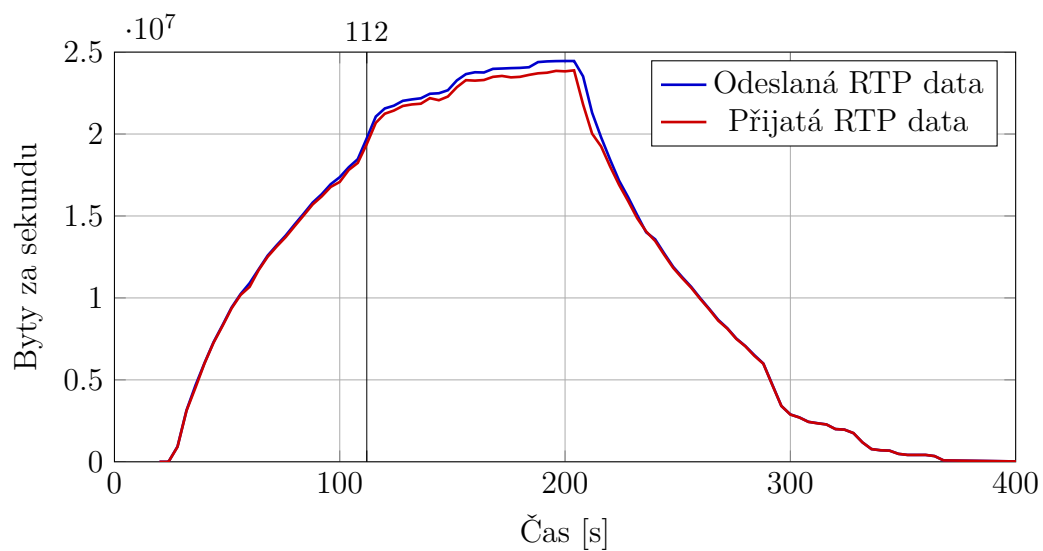
### FreeSWITCH



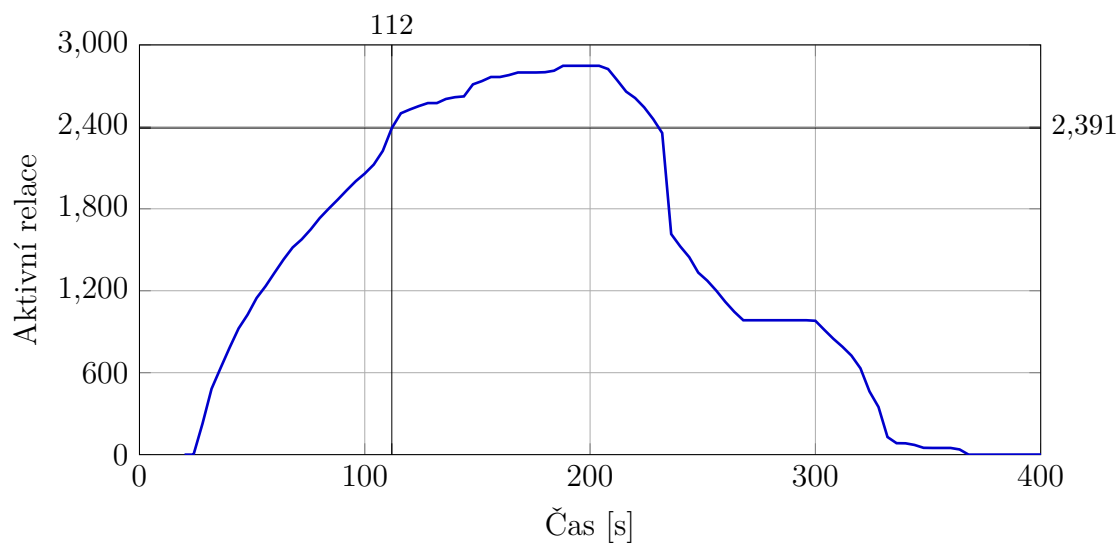
Obr. A.3: Změřené RTP toky ústředny FreeSWITCH v režimu B2BUA s 1536 MB RAM



Obr. A.4: Změřený počet relací ústředny FreeSWITCH v režimu B2BUA s 1536 MB RAM

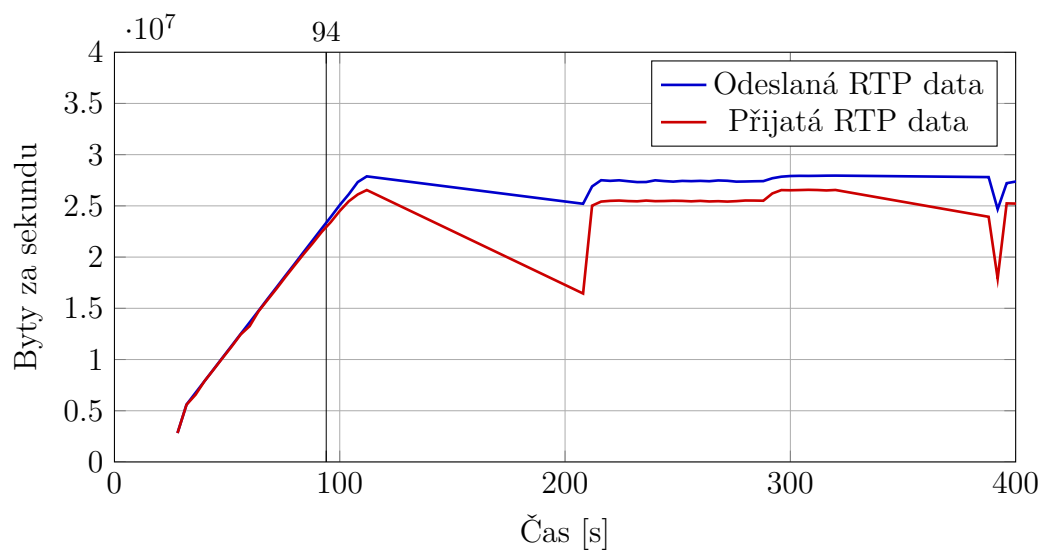


Obr. A.5: Změřené RTP toky ústředny FreeSWITCH v režimu proxy s 1536 MB RAM



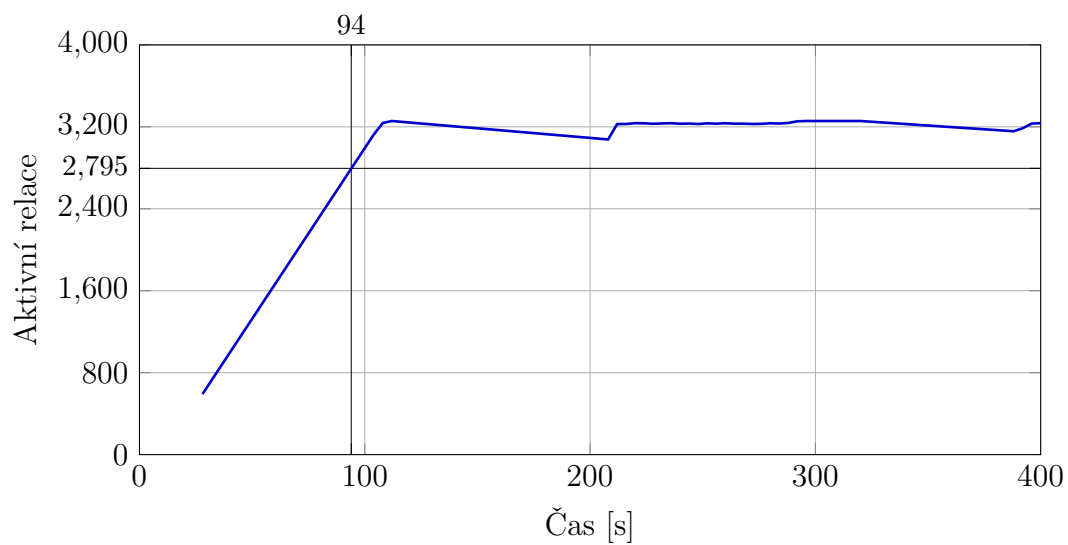
Obr. A.6: Změřený počet relací ústředny FreeSWITCH v režimu proxy s 1536 MB RAM

## OpenSER



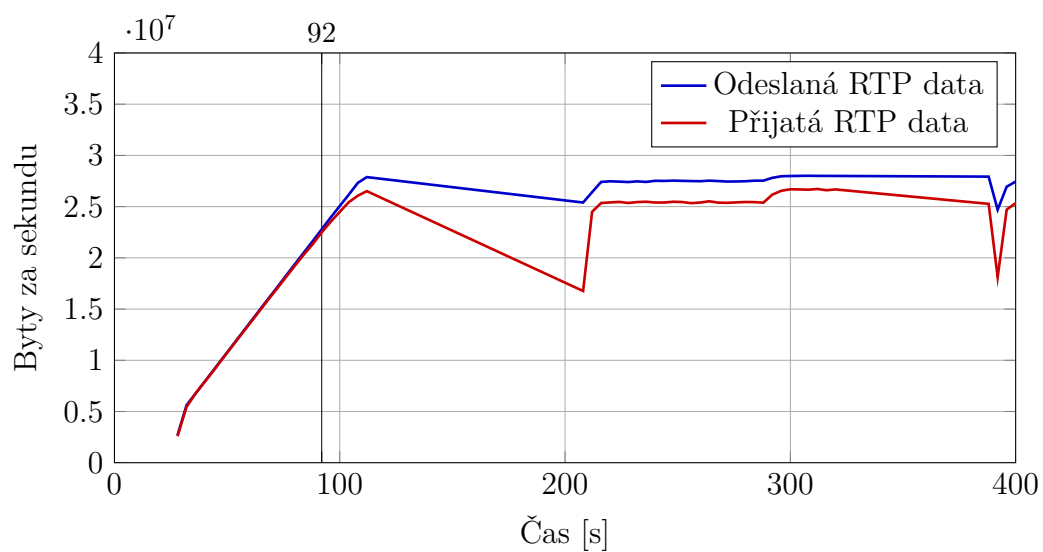
Obr. A.7: Změřené RTP toky ústředny OpenSER v režimu proxy s 1536 MB RAM



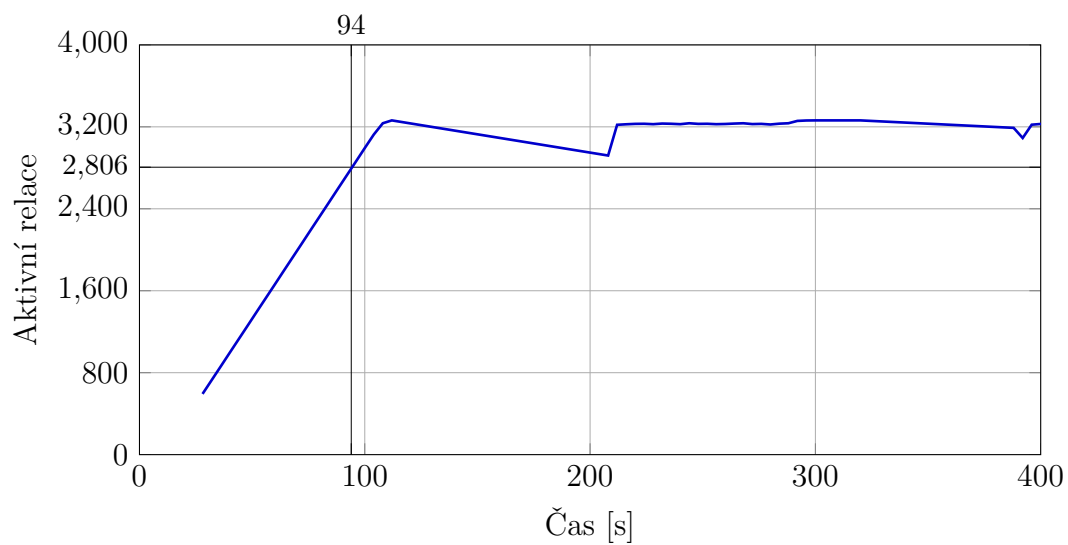


Obr. A.8: Změřený počet relací ústředny OpenSER v režimu proxy s 1536 MB RAM

## Kamailio



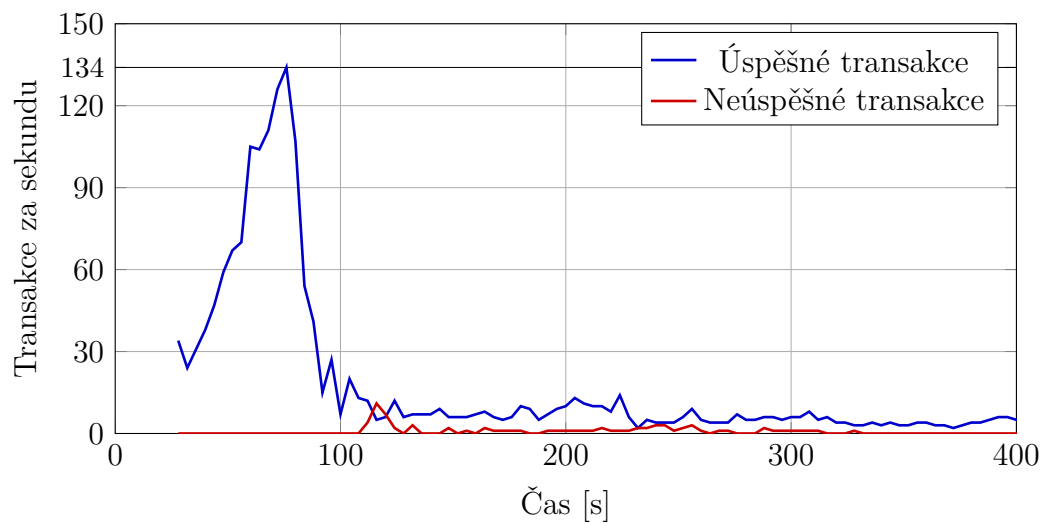
Obr. A.9: Změřené RTP toky ústředny Kamailio v režimu proxy s 1536 MB RAM



Obr. A.10: Změřený počet relací ústředny Kamilio v režimu proxy s 1536 MB RAM

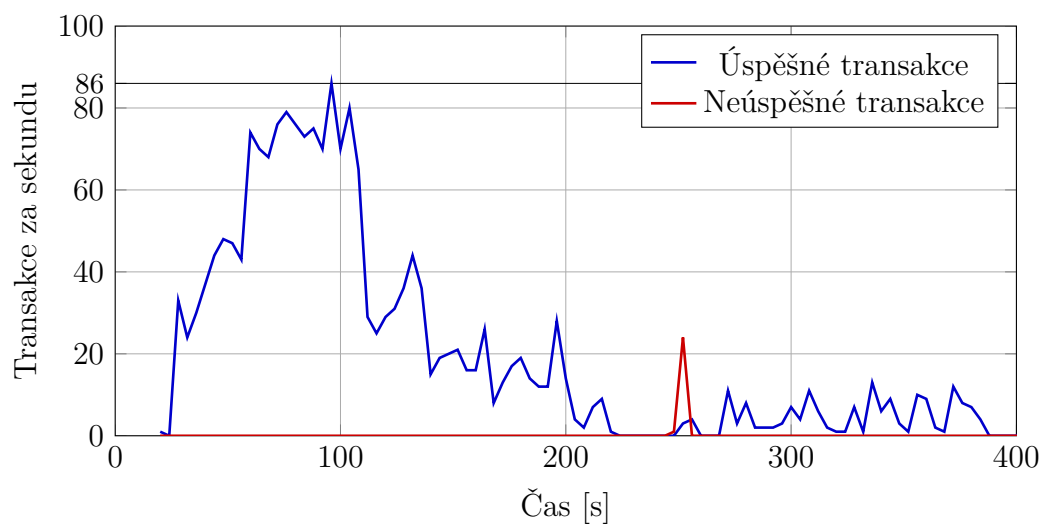
### A.3.2 Měření transakcí

#### Asterisk 13

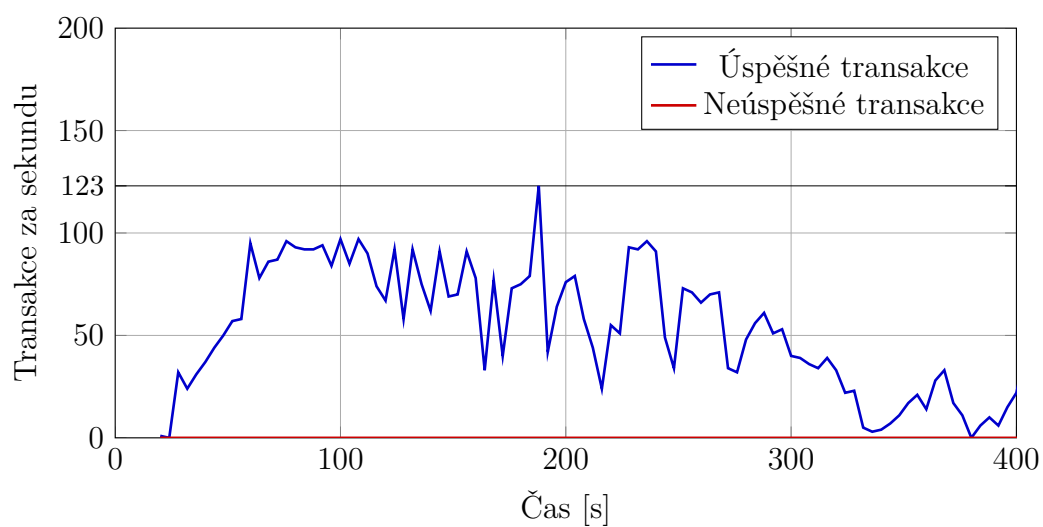


Obr. A.11: Změřené transakce ústředny Asterisk 13 v režimu B2BUA s 1536 MB RAM

## FreeSWITCH

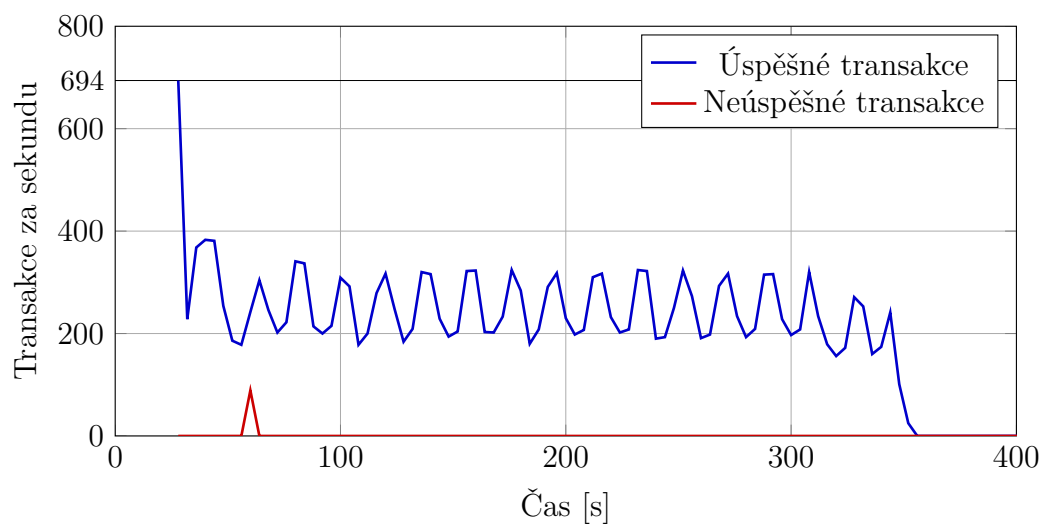


Obr. A.12: Změřené transakce ústředny FreeSWITCH v režimu B2BUA s 1536 MB RAM



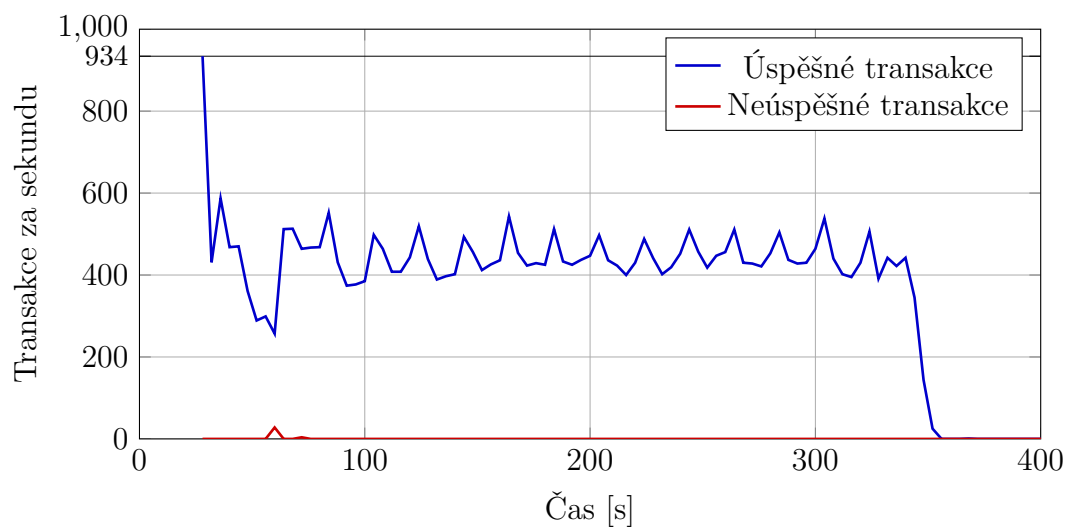
Obr. A.13: Změřené transakce ústředny FreeSWITCH v režimu proxy s 1536 MB RAM

## OpenSER



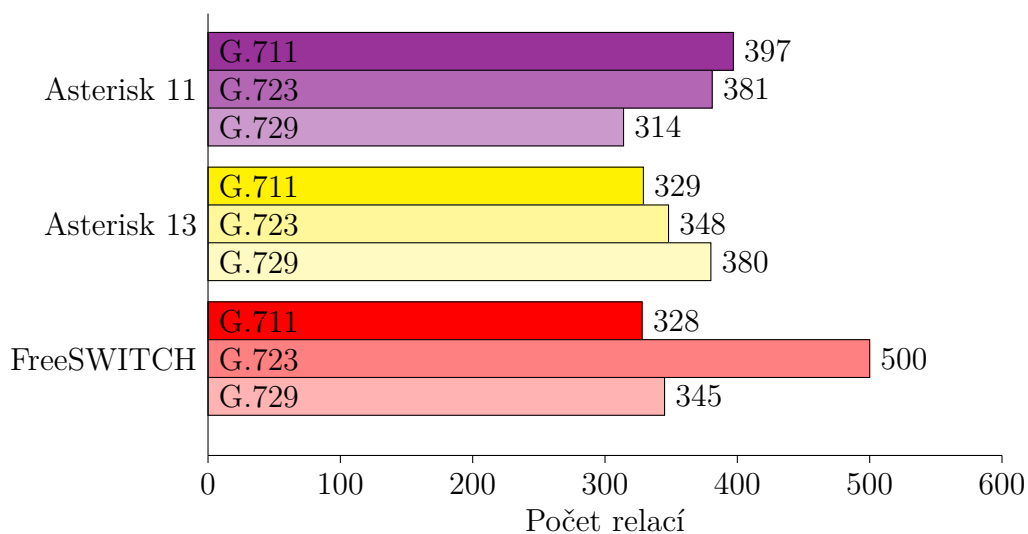
Obr. A.14: Změřené transakce ústředny OpenSER v režimu proxy s 1536 MB RAM

## Kamailio

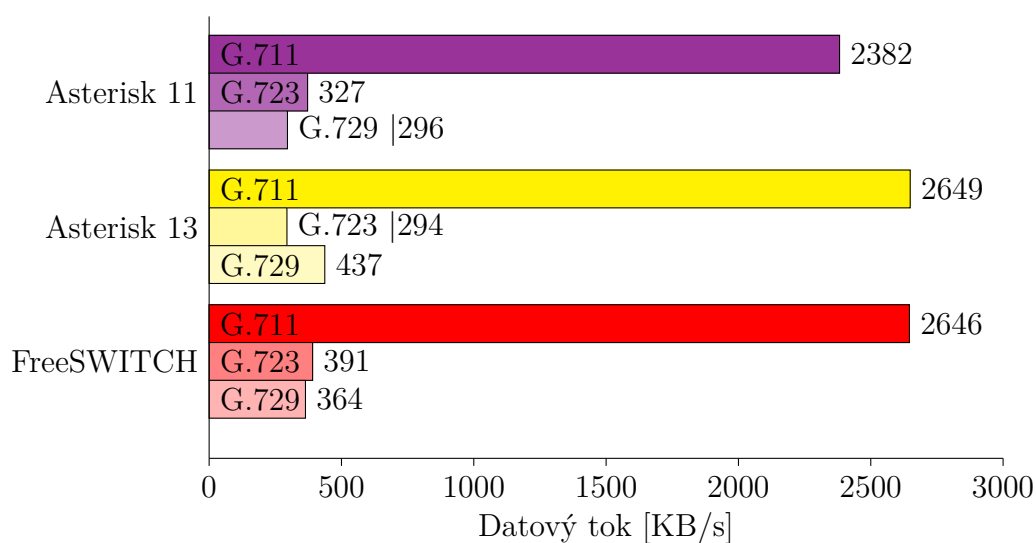


Obr. A.15: Změřené transakce ústředny Kamailio v režimu proxy s 768 MB RAM

### A.3.3 Měření s odlišnými kodeky



Obr. A.16: Maximální dosažené relace v porovnání s různými kodeky s 768MB operační pamětí



Obr. A.17: Maximální toky dat přes ústředny s 768MB operační pamětí a různými kodeky

## A.4 Tabulky

Tab. A.1: Srovnání nejpodstatnějších vlastností stacků SIP

|                                     | <b>PJSIP</b> | <b>Sofia-sip</b> | <b>reSIPprocate</b> | <b>oSIP</b> | <b>OPAL</b> | <b>VOCAL</b>   |
|-------------------------------------|--------------|------------------|---------------------|-------------|-------------|----------------|
| Licence                             | GPLv2        | LGPL             | Vovida              | LGPL        | MPL         | Vovida         |
| Napsáno v jazyce                    | C            | C                | C++                 | C           | C++         | C++            |
| Dokumentace                         | Výborná      | Dobrá            | Výborná             | Chvalitebná | Výborná     | Dobrá          |
| Podpora SSL/TLS                     | Ano          | Ano              | Ano                 | ?           | Ne          | Ano            |
| Otisk v paměti                      | <150 KB      | <600 KB          | <1 MB               | 400 KB      | Velká*      | Velká*         |
| Multiplatformní                     | Ano          | Ano              | Ano                 | Ano         | Ano         | Pouze Linux    |
| Schopnost pracovat na více vláknech | Ano          | Ano              | Ano                 | Ano         | Ano         | Ano            |
| Podpora RFC 3261                    | Ano          | Ano              | Ano                 | Ano         | Ano         | Pouze částečná |

*\*jde o několik knihoven, velikost stacku záleží na výběru a použití těchto knihoven, je však v řádu MB*

Tab. A.2: Shrnutí měření dosažených relací a transakcí

|                    | RAM<br>[MB] | Režim B2BUA     |                    |                                | Režim proxy     |                    |                                |
|--------------------|-------------|-----------------|--------------------|--------------------------------|-----------------|--------------------|--------------------------------|
|                    |             | Počet<br>relací | Počet<br>transakcí | Doba sestavení<br>hovoru [ms]* | Počet<br>relací | Počet<br>transakcí | Doba sestavení<br>hovoru [ms]* |
| <b>Asterisk 11</b> | 768         | 297             | 82                 | 241                            | 2326            | 192                | 552                            |
|                    | 1536        | 296             | 186                | 690                            | 2459            | 358                | 871                            |
| <b>Asterisk 13</b> | 768         | 329             | 69                 | 547                            | -               | -                  | -                              |
|                    | 1536        | 331             | 134                | 162                            | -               | -                  | -                              |
| <b>FreeSWITCH</b>  | 768         | 328             | 89                 | 385                            | 1512            | 102                | 2808                           |
|                    | 1536        | 295             | 86                 | 145                            | 2391            | 123                | 36                             |
| <b>OpenSER</b>     | 768         | -               | -                  | -                              | 2726            | 514                | 110                            |
|                    | 1536        | -               | -                  | -                              | 2795            | 694                | 110                            |
| <b>Kamailio</b>    | 768         | -               | -                  | -                              | 2768            | 934                | 141                            |
|                    | 1536        | -               | -                  | -                              | 2806            | 1085               | 124                            |

\*měřeno pouze u maximálního počtu relací

Tab. A.3: Shrnutí naměřeného výkonu na ústřednách při měření relací

|             | RAM<br>[MB] | Režim B2BUA   |         |                      |                       | Režim proxy   |         |                      |                       |
|-------------|-------------|---------------|---------|----------------------|-----------------------|---------------|---------|----------------------|-----------------------|
|             |             | Výkon CPU [%] |         | Dosažená<br>RAM [MB] | Otevřených<br>souborů | Výkon CPU [%] |         | Dosažená<br>RAM [MB] | Otevřených<br>souborů |
|             |             | Průměr        | Maximum |                      |                       | Průměr        | Maximum |                      |                       |
| Asterisk 11 | 768         | 26            | 89      | 701                  | 3071                  | 7             | 33      | 698                  | 20024                 |
|             | 1536        | 33            | 81      | 1418                 | 4607                  | 13            | 63      | 1442                 | 20024                 |
| Asterisk 13 | 768         | 24            | 94      | 681                  | 5204                  | -             | -       | -                    | -                     |
|             | 1536        | 44            | 93      | 1434                 | 4981                  | -             | -       | -                    | -                     |
| FreeSWITCH  | 768         | 38            | 96      | 691                  | 3549                  | 24            | 71      | 694                  | 528                   |
|             | 1536        | 44            | 69      | 1385                 | 4489                  | 27            | 79      | 1437                 | 77                    |
| OpenSER     | 768         | -             | -       | -                    | -                     | 3             | 41      | 326                  | 30                    |
|             | 1536        | -             | -       | -                    | -                     | 3             | 15      | 336                  | 30                    |
| Kamailio    | 768         | -             | -       | -                    | -                     | 3             | 14      | 339                  | 33                    |
|             | 1536        | -             | -       | -                    | -                     | 15            | 45      | 454                  | 33                    |



Tab. A.4: Shrnutí naměřeného výkonu na ústřednách při měření transakcí

|                    | RAM<br>[MB] | Režim B2BUA   |         |                      |                       | Režim proxy   |         |                      |                       |
|--------------------|-------------|---------------|---------|----------------------|-----------------------|---------------|---------|----------------------|-----------------------|
|                    |             | Výkon CPU [%] |         | Dosažená<br>RAM [MB] | Otevřených<br>souborů | Výkon CPU [%] |         | Dosažená<br>RAM [MB] | Otevřených<br>souborů |
|                    |             | Průměr        | Maximum |                      |                       | Průměr        | Maximum |                      |                       |
| <b>Asterisk 11</b> | 768         | 11            | 84      | 701                  | 3071                  | 8             | 42      | 700                  | 12137                 |
|                    | 1536        | 17            | 83      | 1439                 | 10662                 | 17            | 64      | 1446                 | 15233                 |
| <b>Asterisk 13</b> | 768         | 7             | 67      | 701                  | 3364                  | -             | -       | -                    | -                     |
|                    | 1536        | 24            | 67      | 1447                 | 6761                  | -             | -       | -                    | -                     |
| <b>FreeSWITCH</b>  | 768         | 37            | 93      | 701                  | 1113                  | 37            | 71      | 696                  | 78                    |
|                    | 1536        | 48            | 97      | 1446                 | 1541                  | 39            | 77      | 1446                 | 79                    |
| <b>OpenSER</b>     | 768         | -             | -       | -                    | -                     | 4             | 14      | 338                  | 30                    |
|                    | 1536        | -             | -       | -                    | -                     | 4             | 11      | 343                  | 30                    |
| <b>Kamailio</b>    | 768         | -             | -       | -                    | -                     | 16            | 53      | 403                  | 33                    |
|                    | 1536        | -             | -       | -                    | -                     | 15            | 45      | 454                  | 33                    |

Tab. A.5: Shrnutí naměřeného výkonu na ústřednách při měření kodeků

|             | RAM<br>[MB] | Kodek G.723   |         |                      |                       | Kodek G.729   |         |                      |                       |
|-------------|-------------|---------------|---------|----------------------|-----------------------|---------------|---------|----------------------|-----------------------|
|             |             | Výkon CPU [%] |         | Dosažená<br>RAM [MB] | Otevřených<br>souborů | Výkon CPU [%] |         | Dosažená<br>RAM [MB] | Otevřených<br>souborů |
|             |             | Průměr        | Maximum |                      |                       | Průměr        | Maximum |                      |                       |
| Asterisk 11 | 768         | 19            | 48      | 641                  | 3196                  | 22            | 67      | 689                  | 4002                  |
|             | 1536        | 19            | 48      | 783                  | 3220                  | 44            | 82      | 1271                 | 4444                  |
| Asterisk 13 | 768         | 33            | 70      | 633                  | 4762                  | 32            | 73      | 768                  | 4676                  |
|             | 1536        | 33            | 62      | 834                  | 4918                  | 41            | 83      | 633                  | 4724                  |
| FreeSWITCH  | 768         | 58            | 97      | 698                  | 4268                  | 59            | 96      | 689                  | 3700                  |
|             | 1536        | 43            | 71      | 1380                 | 4808                  | 56            | 78      | 1034                 | 3940                  |

Tab. A.6: Seznam všech odpovědí na modifikované zprávy

| Modifikovaná zpráva (kapitola)                   | Asterisk 11                | Asterisk 13      | FreeSWITCH                    | OpenSER           | Kamailio                                      |
|--|----------------------------|------------------|-------------------------------|-------------------|---|
| Starší INVITE žádost                             | Zpráva zahozena            | 401 Unauthorized | 400 Missing Contact Header    | 100 Giving a try  | 100 Trying                                    |
| Více hodnot v položce Content Length             | 404 Not Found              | 401 Unauthorized | 400 Bad Content-Length Header | Zpráva zahozena   | 400 Content-Length mis-match                  |
| Odpověď neznámého typu                           | Zpráva zahozena            | Zpráva zahozena  | Zpráva zahozena               | Zpráva zahozena   | Zpráva zahozena                               |
| Mezery v poli příjemce                           | 404 Not Found              | 401 Unauthorized | 200 OK                        | Zpráva zahozena   | Zpráva zahozena                               |
| Neznámá metoda                                   | 501 Method Not Implemented | 401 Unauthorized | 400 Bad Request               | Zpráva zahozena   | 400 CSeq method does not match request method |
| Různé typy transportu zprávy                     | 404 Not Found              | 401 Unauthorized | 200 OK                        | Zpráva zahozena   | Zpráva zahozena                               |
| Nulová hodnota položky Max-Forwards              | 404 Not Found              | 401 Unauthorized | 200 OK                        | 483 Too Many Hops | 483 Too Many Hops                             |
| Špatně interpretovaný parametr v poli Contact    | 401 Unauthorized           | 401 Unauthorized | 401 Unauthorized              | Zpráva zahozena   | Zpráva zahozena                               |
| Chybějící identifikátor transakce                | 404 Not Found              | 401 Unauthorized | 200 OK                        | Zpráva zahozena   | Zpráva zahozena                               |
| Žádost se špatnou adresou                        | 401 Unauthorized           | 401 Unauthorized | 401 Unauthorized              | Zpráva zahozena   | Zpráva zahozena                               |
| Nesprávná hodnota velikosti těla zprávy          | 404 Not Found              | 401 Unauthorized | 400 Bad Request               | 100 Giving a try  | 400 Content-Length mis-match                  |
| Odstraněná mezera mezi jménem a adresou          | 404 Not Found              | 401 Unauthorized | 200 OK                        | Zpráva zahozena   | Zpráva zahozena                               |
| Parametry oddělené středníkem v poli Request-URI | 404 Not Found              | 401 Unauthorized | 200 OK                        | Zpráva zahozena   | Zpráva zahozena                               |
| Neuzavřený parametr                              | 404 Not Found              | Zpráva zahozena  | 400 Bad To Header             | Zpráva zahozena   | Zpráva zahozena                               |
| Chybné pole Accept                               | 404 Not Found              | 401 Unauthorized | 406 Not Acceptable            | 100 Giving a try  | 100 Trying                                    |

## B OBSAH PŘÍLOŽENÉHO CD

Z důvodů velmi velkého množství souborů na CD, zejména logy z měření, zde není vypsána stromovitá struktura kořenového adresáře, ale obsah pouze slovně okomentován.

- Tato práce v elektronické podobě ve formě PDF dokumentu
- Vlastní skript `perf.sh` pro měření některých prostředků systému
- Vygenerované grafy v PDF dokumentu pro veškerá zátěžová měření
- Konfigurační soubory ústředny
- Měření dosažených relací
  - Exportované logy RTP toků a aktivních relací z testeru
  - Naměřené hodnoty CPU, RAM a otevřených souborů ústředny
- Měření dosažených transakcí
  - Exportované logy počtu transakcí z testeru
  - Naměřené hodnoty CPU, RAM a otevřených souborů ústředny
- Měření s více kodeky
  - Exportované logy RTP toků a aktivních relací z testeru
  - Naměřené hodnoty CPU, RAM a otevřených souborů ústředny
- Realizované útoky na ústřednu
  - Exportované logy probíhající komunikace v souboru PCAP
  - Logy některých ústředny