

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

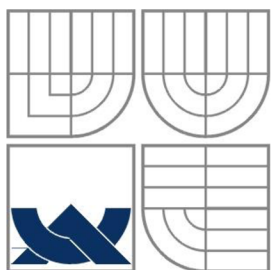
TESTOVÁNÍ VÝKONNOSTI SÍTĚ SIP VOIP POMOCÍ
SPIRENT TEST CENTER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

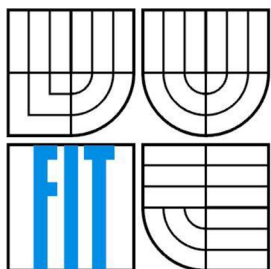
AUTOR PRÁCE
AUTHOR

Miroslav Slivka

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

TESTOVÁNÍ VÝKONNOSTI SÍTĚ SIP VOIP POMOCÍ SPIRENT TEST CENTER

TESTING PERFORMANCE PARAMETERS OF SIP VOIP USING SPIRENT TEST CENTER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Miroslav Slivka

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Matoušek, Ph.D.

BRNO 2013

Abstrakt

Táto práca sa zaoberá testovaním siete SIP VoIP pomocou zariadenia Spirent TestCenter a jeho aplikačnej nadstavby Avalanche. Výsledky tejto práce ukážu spôsob využitia Spirent TestCenter Avalanche na testovanie sietí SIP. Riešenie bude spočívať vo vytvorení metodológie testovania, obsahujúcej konkrétne testy v závislosti na licenčnom obmedzení zariadenia Spirent TestCenter spolu s predvedením výstupov týchto testov. Okrem vytvorenia testov pre Spirent TestCenter budú ukázané spôsoby testovania VoIP sietí inými technológiami.

Abstract

This work deals with benchmarking of SIP VoIP networks by Spirent TestCenter device and its software application Avalanche. Results of this work show the way of using Spirent TestCenter Avalanche for SIP VoIP testing. Solution of this work involves creation of testing methodology, containing test scenarios depending on license limitations of Spirent TestCenter along with a presentation of outputs of the tests. Besides creating of tests for Spirent TestCenter there will also be presented another technologies for testing VoIP networks.

Kľúčové slová

SIP, VoIP, testovanie, monitorovanie, prenos hlasu

Keywords

SIP, VoIP, benchmarking, monitoring, voice transmission

Citácia

Slivka Miroslav: Testovanie výkonnosti siete SIP VoIP pomocou Spirent Test Center, bakalárska práca, Brno, FIT VUT v Brně, 2013

Testovanie výkonnosti siete SIP VoIP pomocou Spirent Test Center

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením

Ing. Petra Matouška Ph.D. .

Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Miroslav Slivka

10.5.2013

Pod'akovanie

Týmto by som rád poďakoval môjmu vedúcemu bakalárskej práce Ing. Petrovi Matouškovi, Ph.D. za odbornú pomoc.

© Miroslav Slivka, 2013

Tato práca vznikla ako školské dielo na Vysokom učení technickom v Brne, Fakulte informačných technológií. Práca je chránená autorským zákonom a jej použitie bez udelenia oprávnenia autorom je nezákonné, s výnimkou zákonom definovaných prípadov.

Obsah

Obsah	1
1 Úvod.....	2
2 Protokol SIP	4
3 Spirent TestCenter	6
3.1 Popis rozhrania aplikácie Avalanche	7
4 Asterisk	9
4.1 Konfigurácia – SIP účty a volací plán	10
5 Metriky testovania SIP VoIP	12
5.1 Štandardné metriky merania výkonnosti SIP.....	12
5.2 Metriky kvality hovoru podľa RTCP.....	14
5.3 Analýza výsledkov testov pomocou STC	17
6 Metodológia a výsledky testovania.....	18
6.1 Metodológia a výsledky testov STC	18
6.2 Metodológia a výsledky kvalitatívnych testov	29
6.2.1 Výsledky testov BLUELIGHT BL400A	30
6.2.2 Výsledky testov Solarwinds VNQM	33
7 Záver	37
Literatúra	38

1 Úvod

S nástupom paketových sietí a Internetu sa ľudia snažili poskytnúť v týchto sieťach aj hlasové služby, dovtedy implementované pomocou sietí s prepínaním okruhov. Hlasové služby v IP sieti sa nazývajú VoIP (Voice over IP). Pri prenose hlasových služieb do IP siete bolo potrebné implementovať protokoly zabezpečujúce signalizáciu (vyzváňanie, informácia o položenom/zodvihnutom slúchadle a pod.) a protokoly zabezpečujúce samotný prenos hlasu. Príkladom takýchto protokolov je SIP, signalizačný protokol zabezpečujúci vytvorenie relácii medzi komunikujúcimi stranami a protokol RTP/RTCP zabezpečujúci prenos multimedialných dát.

Používanie VoIP má mnoho výhod. Jednou z nich, pre mnoho ľudí najdôležitejšou, je cena. Vďaka tomu, že si vystačíte s pripojením k internetu, respektíve IP sieťou, je cena omnoho menšia. Medzi počítačmi či terminálmi pripojenými do siete môžete volať zadarmo. Medzi ďalšie výhody VoIP patrí registrácia užívateľov, vďaka ktorej si každý užívateľ berie číslo vždy so sebou, nech sa nachádza kdekoľvek.

Vo VoIP hrá veľkú úlohu oneskorenie paketov. Pri signalizácii nám nemusí vadieť, že telefón na druhej strane začne zvoniť o sekundu neskôr (<2,5s), ale pri konverzácii by už tak veľké oneskorenie bolo neprijemné. Podľa štandardu ITU-T G.114 [1] by sa oneskorenie malo pohybovať najviac do 150 ms. Oneskorenie väčšie ako 400 ms je už nepoužiteľné. Ďalším problémom vo VoIP je rozptyl oneskorenia hlasových paketov – jitter. Kvôli takýmto výkyvom kvality je potrebné, aby sme vedeli ako monitorovať a testovať našu VoIP sieť.

Táto práca sa zaoberá testovaním siete SIP VoIP pomocou zariadenia Spirent TestCenter a jeho aplikačnej nadstavby Avalanche. Cieľom práce je ukázať spôsob využitia Spirent TestCenter Avalanche na testovanie sietí SIP. Výstupom práce bude metodológia testovania SIP siete pomocou tohto zariadenia a porovnanie možností zariadenia Spirent TestCenter a aplikácie Avalanche s inými možnosťami testovania VoIP sietí.

V práci sa postupne oboznámime stručne s protokolom SIP, povieme si niečo o samotnom zariadení Spirent TestCenter a jeho rozhraní Avalanche, oboznámime sa so základnou konfiguráciou a používaním ústredne Asterisk a predvedieme výsledky vytvorenej metodológie testovania. V krátkej kapitole Metriky testovania SIP VoIP sa oboznámime s metrikami používanými pre testovanie protokolu SIP a metrikami, ktoré sa prenášajú v správach protokolu RTCP, ktoré nás informujú o kvalite prebiehajúceho hovoru. Nasledovať bude kapitola obsahujúca metodológiu testovania, postup a výsledky testov. Na zariadení Spirent TestCenter boli vytvorené tri typy testov zamerané na signalizáciu. V týchto troch testoch je zobrazené a popísané všetko, čo môžeme sledovať pri testovaní signalizácie SIP pomocou zariadenia Spirent TestCenter.

Ďalšie dve podkapitoly sa zaoberajú inými technológiami na testovanie či monitorovanie sietí VoIP. Prvá z nich je venovaná zariadeniu BLUELIGHT BL400A od firmy BlueScope. Pre toto zariadenie bol vytvorený jeden test, ktorý pokrýva väčšinu toho, čo môžeme sledovať na zariadení pri monitorovaní VoIP sietí. Druhou testovanou technológiou je *VoIP & Network Quality Manager* od firmy *Solarwinds*, čo je nástroj určený k dlhodobému monitorovaniu VoIP sietí postavených primárne na technológiách Cisco Call Manager.

2 Protokol SIP

Vzhľadom na to, že témou tejto práce je primárne testovať SIP siete, v tejto kapitole čitateľa stručne oboznámim so základnými princípmi protokolu SIP [2].

Protokol SIP je signalizačný protokol, ktorý slúži na vytvorenie, modifikáciu a ukončenie tzv. relácií. Protokol SIP je založený na transakčnom modeli žiadosť/odpoveď, podobnom ako môžeme vidieť u HTTP. Spolu so SIP sa používajú aj iné protokoly, ktoré zaisťujú prenos real-time dát (RTP) či popis relácie (SDP).

Základné metódy (žiadosti) protokolu SIP:

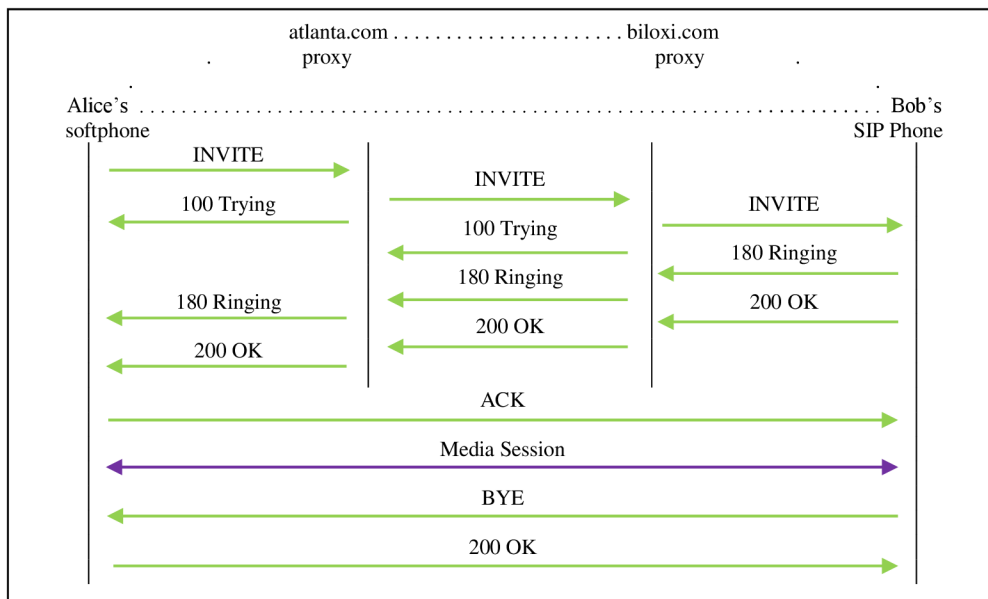
- REGISTER - registrácie
- INVITE, ACK, CANCEL – vytváranie relácie
- BYE – ukončenie relácie
- OPTIONS – možnosti prenosu

Architektúra SIP obsahuje tieto prvky:

- UAC
- UAS
- SIP Proxy

UAC je skratka pre *User Agent Client* a základnou úlohou tohto prvku je generovanie správ, napr. žiadosť o vytvorenie spojenia (INVITE). UAS je skratka pre *User Agent Server* a jeho úlohou je odpovedať na žiadosti vygenerované UAC. Prvok typu UAC môže byť napríklad každé koncové zariadenie, no prvok typu UAS môže byť ako koncové zariadenie tak aj SIP Proxy. SIP Proxy je prvok, ktorý smeruje žiadosti o spojenia k UAS a odpovede na tieto žiadosti k UAC. SIP Proxy môže odpovedať priamo na žiadosti, napríklad ak potrebuje prihlasovacie údaje od klienta. Vtedy sa SIP Proxy správa ako UAS. SIP Proxy môže byť stavový (udržiava stav dialógov/transakcií, rozpoznáva cykly) alebo bezstavový (iba smeruje správy). SIP Proxy hrá rolu lokalizačného a registračného serveru.

Na pochopenie základného princípu komunikácie protokolu SIP bude použitý obrázok z dokumentu RFC 3261 [2], kde sa snažia spolu spojiť dve strany, Alica a Bob, cez dva SIP servery.



Obrázok 1: Vytvorenie SIP relácie

Pre lepšie pochopenie toho, čo sa na obrázku deje si popíšeme vzniknutú situáciu. Alica a Bob si chcú zatelefonovať. Obaja majú registrované SIP účty a zapnutých klientov. Alica má svoj účet registrovaný u poskytovateľa atlanta.com a Bob u biloxi.com. Alica vytočí Bobovo číslo.

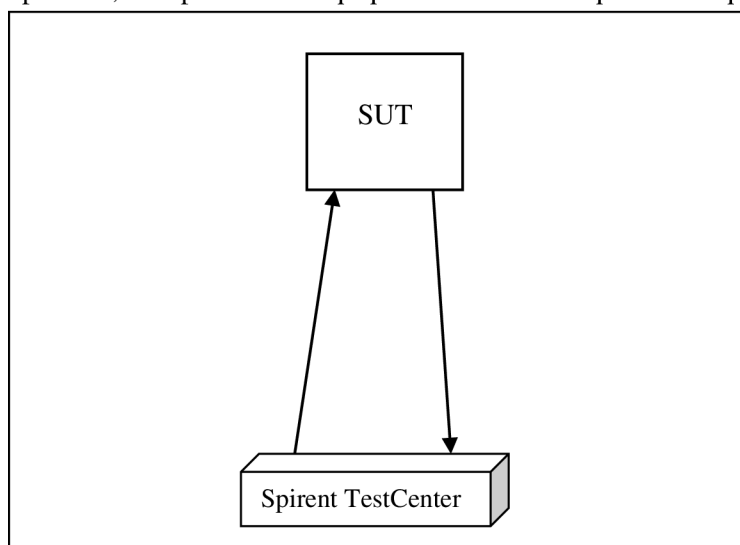
Od tejto chvíle už signalizácia prebieha ako je popísané na obrázku. Alicin klient vygeneruje správu INVITE, ktorá je poslaná na SIP server atlanta.com. Tento server zistí, kde má hľadať Boba. Zistí, že cesta k Bobovi vedie cez server biloxi.com. Takto sa preposiela správa INVITE, až kým sa nedostane k Bobovi. Po tom ako Bobov klient prijme správu INVITE, začne u Boba zvonit' telefón a naspäť sa posiela správa Ringing, čo na druhej strane u Alice vyvolá vyzváňací tón. Po prijatí hovoru sa od Boba pošle správa OK oznamujúca, že hovor bol prijatý. Po tejto správe už môže komunikácia prebiehať priamo, bez pomoci proxy serverov. Vytvorí sa relácia s prenosom hlasových dát. Hovor je ukončený správou BYE.

V tejto práci využijeme tieto znalosti pri navrhovaní metodológie testovania a navrhovaní testovacích topológií.

3 Spirent TestCenter

V tejto kapitole sa čitateľ oboznámi so zariadením Spirent TestCenter¹ (STC) a jeho aplikačnou nadstavbou Avalanche, t.j. možnosťami tejto aplikácie. Obsah kapitoly sa zameriava výlučne na testovanie sietí SIP VoIP.

Spirent TestCenter je zariadenie určené na testovanie sietí a zariadení v sieti. Dokáže simulovať klientov, servery prípadne celé pod/siete a emulovať sieťové prvky. Testy spočívajú v generovaní prevádzky rôznych vrstiev modelu OSI z jedného, alebo viacerých portov a prijímaní dát na inom porte alebo portoch, ich spracovanie a prípadné odoslanie odpovede naspäť (viď obrázok 2).



Obrázok 2: Schéma zapojenia zariadenia Spirent TestCenter

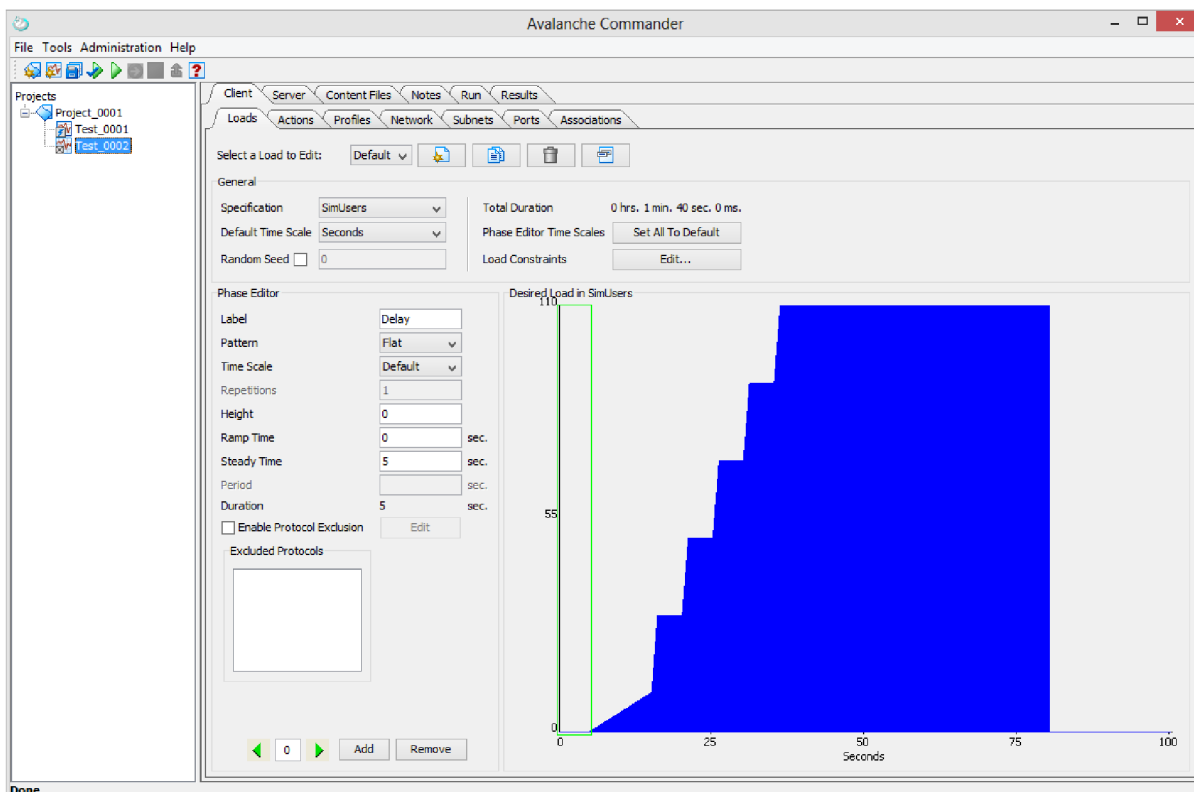
Na generovanie prevádzky existujú dve aplikácie. Rovnomenná aplikácia Spirent TestCenter je určená na generovanie prevádzky vrstiev L1 – L3. Mimo to slúži tiež na obsluhu firmware v zariadení². V tejto práci budeme používať aplikáciu nazvanú Avalanche, určenú na generovanie prevádzky vrstiev L4 – L7. Ku každej z týchto aplikácii patrí aj samostatná aplikácia na analyzovanie výsledkov testov.

¹ <http://www.spirent.com/>

² Na prácu s aplikáciou Avalanche bude potrebné upraviť firmware na portoch zariadenia.

3.1 Popis rozhrania aplikácie Avalanche

Aplikácia Avalanche je rozdelená na dve časti. V ľavej časti (vid' obrázok 3) máme zoznam projektov. Každý projekt sa môže skladať z jedného a viacerých testov.



Obrázok 3: Ukážka prostredia aplikácie Avalanche

V aplikácii máme na výber z dvoch kategórií testov – Application a Device (Aplikácia a Zariadenie). Prvý spomínaný simuluje iba klientov, druhý simuluje taktiež správanie serverov. Na základe toho, čo budeme testovať rozlišujeme pojem System Under Test (SUT³) a Device Under Test (DUT⁴).

Aplikácia nám podľa náročnosti konfigurácie ponúkne tri typy testov – Quick (Rýchly), Advanced (Pokročilý) a EZ Test. Najmenej možností nastavenia nám ponúkne EZ Test, najviac Advanced Test. Pri prvom oboznamovaní sa s aplikáciou si vystačíme s Quick Testom, v ktorom celkom rýchlo nakonfigurujeme funkčný test a vďaka malému množstvu nastavení na nič nezabudneme. V ďalšom pokračovaní tejto práce ale bude nutné vytvárať pokročilé testy s rôznou záťažou a rôznym správaním simulovaných klientov a na to už s Quick testom nevystačíme. Advanced test nám vďaka Asociáciám dovoľuje nakonfigurovať odlišné správanie pre skupiny simulovaných klientov a rôzne ovplyvňovať generovanú záťaž (vid' Príloha 1).

³ Typicky sa jedná o celé topológie sietí.

⁴ Testuje sa jedno zariadenie, všetky ostatné časti siete sú simulované a emulované.

Počas testu nám Avalanche dovoľuje sledovať aktuálne štatistiky. Avšak čo sa týka protokolu SIP, možnosti sú celkom obmedzené.

4 Asterisk

V tejto kapitole oboznámim čitateľa s aplikáciou Asterisk⁵. Táto aplikácia nám počas testov bude poskytovať funkčnosť SIP proxy. Z toho dôvodu sa v tejto kapitole budem zaoberať základnou konfiguráciou Asterisku⁶. Znalosti z tejto kapitoly budú využiteľné pri replikovaní testovacích scenárov.

Na rozdiel od klasického SIP proxy sa Asterisk správa ako koncový bod. Pri spracovaní hovoru je dialóg v smere od zdroju k Asterisku iným dialógom ako dialóg v smere od Asterisku k cieľu. Hovoríme, že Asterisk je „back-to-back user agent“ (B2BUA). Toto správanie sa ale dá ovplyvniť v konfigurácii SIP Asterisku.

Balíček programu Asterisk je dostupný priamo z repozitárov väčšiny distribúcií Linuxu. Na domovských stránkach môžeme nájsť aj obraz disku AsteriskNOW, čo je distribúcia CentOS s už nainštalovaným Asteriskom a prídavným software. Detailný postup inštalácie je možné nájsť na wiki stránkach.

Novú inštanciu Asterisku spustíme príkazom `asterisk`. Pridaním parametru `-c` zaistí spustenie na popredí a automatický prechod do konzole Asterisku. Pri štarte môžeme ovplyvniť aj úroveň toho, ako veľmi nás bude Asterisk informovať o tom čo sa deje, pomocou parametru `-v`. Čím viac „véčok“ tým viac ladiacich výpisov. Ak už inštancia Asterisku beží, pripojíme sa k nej parametrom `-r`. Pomocou parametra `-x` môžeme zadávať príkazy Asterisku bežiacemu na pozadí s výpisom do terminálu. CLI príkaz pre Asterisk musí nasledovať bezprostredne za parametrom v úvodzovkách.

Niekoľko nápomocných príkazov pre Asterisk:

- `core show help` – zobrazí zoznam príkazov pre Asterisk
- `core restart now` – reštartuje Asterisk
- `core stop now` – zastaví Asterisk
- `sip show peers` – zoznam všetkých nakonfigurovaných SIP zariadení
- `dialplan show` – zobrazí aktívny číselný plán
- `sip reload` – znovu načíta konfiguráciu v súbore `sip.conf`
- `dialplan reload` – znovu načíta konfiguráciu volacieho plánu

⁵ <http://www.asterisk.org/>

⁶ <https://wiki.asterisk.org/wiki/display/AST/Beginning+Asterisk>

4.1 Konfigurácia – SIP účty a volací plán

Konfiguračné súbory sa nachádzajú v `/etc/asterisk/`. Súbory sú rozdelené do kontextov. Názov kontextu je v hranatých zátvorkách (`[a]`). V názve kontextu sa nesmú nachádzať medzery a záleží na veľkosti písmen. Jednotlivé nastavenia sú vo formáte `názov=hodnota` prípadne `názov=>hodnota`.

SIP účty budeme pridávať v súbore `/etc/asterisk/sip.conf`. Tento súbor sa začína kontextom `[general]`, obsahuje predvolené nastavenia pre každú linku. Tieto nastavenia môžu byť prepísané v kontexte konkrétneho účtu. Za týmto kontextom môžeme pridávať jednotlivé účty. Účty sú identifikované názvom kontextu, odporúča sa však využívať aj hodnota `username`.

```
[bob]
type=friend
context=sip-phones
host=dynamic
username=bob
secret=password
```

Typ „friend“ znamená, že tento účet môže spracovávať prichádzajúce aj odchádzajúce hovory. Položka „context“ značí, do ktorého kontextu sa vstúpi v číselnom pláne, keď sa budú spracovávať hovory odchádzajúce z tohto účtu. Hodnota „dynamic“ v položke „host“ znamená, že užívateľ sa prihlasuje z rôznych adries.

Volací plán sa konfiguruje v súbore `/etc/asterisk/extensions.conf`. Volací plán sa skladá zo štyroch základných častí – kontextov (contexts), pravidiel (extensions), priorít (priorities) a príkazov (applications). Volací plán je rozdelený do kontextov, ktoré obsahujú pravidlá. Pravidlo v jednom kontexte je úplne oddelené od pravidla v inom kontexte. Na začiatku volacieho plánu sa nachádzajú dva špeciálne kontexty nazvané `[general]` a `[globals]`.

Každé pravidlo volacieho plánu sa začína kľúčovým slovom `exten` a má tvar `exten => názov,priorita,príkaz`. Dané pravidlo sa použije, keď volané číslo je zhodné s názvom. Najjednoduchšie je použiť priamo telefónne číslo, aj keď Asterisk podporuje v názve aj písmená. Musí existovať aspoň jedno pravidlo s daným názvom a prioritou 1. Priorita určuje poradie spracovania pravidiel počínajúc prioritou 1. Každé ďalšie pravidlo musí mať prioritu o 1 vyššiu ako predošlé alebo hodnotu n – „next“. Príkaz je názov funkcie, ktorá sa má vykonať.

Regulárne výrazy sa v číselnom pláne začínajú znakom podčiarkovník „_“. Regulárne výrazy využijeme na to, aby sme nemuseli písať vždy každé telefónne číslo v osobitných pravidlách. Používané zástupné znaky nájdete v tabuľke 1.

X	Čísla 0-9
Z	Čísla 1-9
N	Čísla 2-9
.	Hocijaký počet hocijakých čísel
!	Pravidlo s týmto znakom sa použije ak sa nenašla žiadna zhoda

Tabuľka 1: Niekoľko zástupných znakov používaných vo volacom pláne Asterisku

Neodporúča sa používať výraz `_.`, pretože zahŕňa aj špeciálne pravidlá *i*, *t* a *h*.

Niekoľko dôležitých príkazov:

- `Answer()` – otvorí kanál
- `Wait()` – pozastaví vykonávanie volacieho plánu na požadovaný počet sekúnd
- `Hangup()` – zavrie aktívny kanál
- `Dial()` – pokúsi sa vytočiť zariadenie zadané prvým parametrom funkcie (so špecifikovaným protokolom), druhým parametrom je čas v sekundách, ako dlho sa bude snažiť dovolať, kým prejde na nasledujúce pravidlo; `Dial(SIP/user, 20)`
 - Pre volanie na inú ústredňu je potrebné do prvého parametru pridať premennú `${EXTEN}`

5 Metriky testovania SIP VoIP

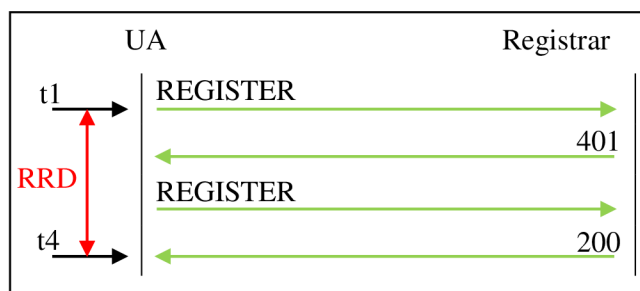
V tejto kapitole sa čitateľ oboznámi s metrikami, ktoré definuje dokument RFC 6076 [3] pre meranie výkonnosti SIP, oboznámi sa s metrikami používanými pre určovanie kvality hovoru vo VoIP a metriky výkonnosti SIP porovnáme s tým, čo môžeme sledovať v aplikácii Avalanche. Pri analýze výsledkov testov pomocou STC sa nebudem zaoberať metrikami kvality pretože školou vlastnené licencie pre STC nedovoľujú posielat' RTP dáta.

5.1 Štandardné metriky merania výkonnosti SIP

Registration Request Delay (RRD)

RRD je miera odozvy na žiadosť REGISTER. Jednotkou sú milisekundy. Zaznamenávať by sa mali iba úspešné registrácie, zatiaľ čo neefektívne pokusy o registráciu (IRAs) by mali byť hlásené ako zlyhania.

RRD = Čas odpovede - Čas žiadosti REGISTER



Obrázok 4: Registration Request Delay

Ineffective Registration Attempts (IRAs)

IRAs nám dovoľuje zistiť zlyhania pri pokusoch o registráciu. IRAs sa meria v percentách.

$IRAs \% = \text{Počet IRAs} / \text{Celkový počet žiadostí REGISTER}$

Neúspešný pokus o registráciu môže byť spôsobený buď jednou z odpovedí o zlyhaní⁷ od serveru alebo vypršaním časovača. Dovoľuje nám zistiť problémy pri signalizácii smerom od serveru ku klientovi, či zahltenie serveru a jeho nemožnosť odpovedať.

⁷ Odpovede typu 4xx (okrem 401, 402 a 407), 5xx alebo 6xx

Session Request Delay (SRD)

SRD nám dovoľuje zistiť problémy spôsobujúce oneskorenia odpovedí na žiadosť klienta o vytvorenie relácie. SRD sa meria ako pre odpovede indikujúce úspešné vytvorenie relácie tak aj pre odpovede indikujúce neúspech, avšak nesmú sa kombinovať.

SRD = Čas odpovede indikujúcej stav relácie - Čas INVITE

Session Duration Time (SDT)

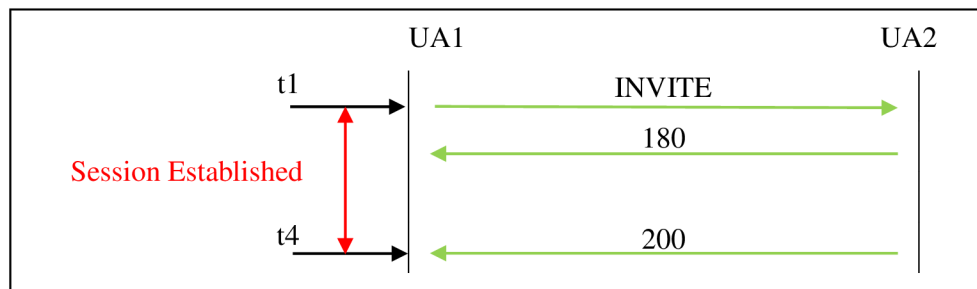
Táto metrika je určená na detekciu problémov spôsobujúcich krátke trvanie relácie. Meria sa v rádoch sekúnd na jednom konci SIP dialógu.

SDT = Čas príchodu správy BYE alebo vypršanie časovača - Čas príchodu správy 200 OK ako odpoveď na INVITE

Session Establishment Ratio (SER)

Táto metrika je používaná na detekciu schopnosti klienta alebo proxy serveru úspešne nadviazať reláciu s novou žiadosťou INVITE. Meria sa v percentách pomocou tejto rovnice:

$$SER = \frac{\text{Počet žiadostí INVITE, na ktoré prišla odpoveď 200 OK}}{\text{(Všetky žiadosti INVITE) - (Žiadosti INVITE, na ktoré prišla odpoveď 3xx)}}$$



Obrázok 5: Nadviazanie relácie

Session Establishment Effectiveness Ratio (SEER)

Táto metrika je doplnková k metrike SER. Oproti metrike SER však berie v úvahu potenciálnu činnosť užívateľa.

$$SEER = \frac{\text{Počet žiadostí INVITE spojené s 200, 480, 486, 600 alebo 603}}{\text{(Všetky žiadosti INVITE - Počet žiadostí INVITE s odpoveďou 3XX)}} \times 100$$

Ineffective Session Attempts (ISAs)

Neúčinný pokus o vytvorenie relácie nastáva, keď server alebo klient vypustí žiadosť o vytvorenie relácie kvôli zlyhaniu alebo preťaženiu. Meria sa v percentách.

$$ISA \% = \frac{\text{Počet ISA}}{\text{Celkový počet žiadostí o vytvorenie relácie}} \times 100$$

Session Completion Ratio (SCR)

Dokončenie relácie je definované ako SIP dialóg, ktorý skončí bez chýb spôsobených nedostatkom odpovedí od serveru alebo klienta. Meria sa v percentách.

$SCR \% = \frac{\text{Počet úspešne dokončených relácií}}{\text{Počet žiadostí o vytvorenie relácie}} \times 100$

5.2 Metriky kvality hovoru podľa RTCP

Prenos hlasových dát je vo VoIP zabezpečený protokolom RTP popísaným v RFC 3550 [4]. Tento protokol pracuje spolu s protokolom RTCP, ktorý zabezpečuje spätnú väzbu na kvalitu prenášaných dát. RTCP správy *Sender report (SR)* a *Receiver report (RR)* obsahujú informácie o kvalite prebiehajúcej komunikácie.

Analýzou správ SR/RR zistíme tieto informácie:

- Round-trip delay
- Medzipríchodový jitter
- Strata paketov

Okrem správ SR a RR bola pre RTCP navrhnutá správa *Extended report (XR)* popísaná v RFC 3611 [5], ktorá obsahuje zvláštny blok s metrikami VoIP:

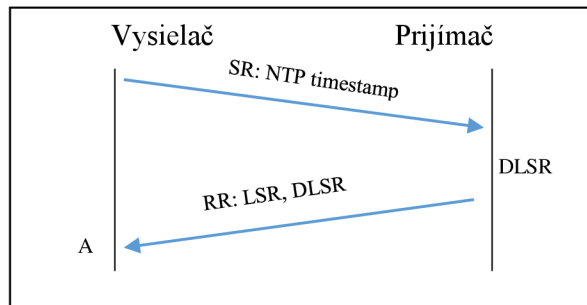
- Metriky straty a zahadzovania paketov (*Packet Loss and Discard Metrics*)
- *Burst metrics*
- *Delay metrics*
- Signálové metriky (*Signal Related Metrics*)
- Metriky kvality hovoru alebo kvality prenosu (*Call Quality or Transmission Quality Metrics*)

Round-trip delay

Round trip delay je čas od odoslania správy po prijatie odpovede od prijímača. K výpočtu Round-trip delay medzi vysielačom RTP dát a prijímačom RTP dát sú dôležité správy RR. Tieto správy obsahujú polia LSR (last SR timestamp – časová značka poslednej prijatej správy SR) a DLSR (delay since last SR – doba od poslednej prijatej správy SR), z ktorých môže vysielač priamo vypočítať Round-trip delay. Podľa vzorca

$$D = A - LSR - DLSR,$$

kde A je čas, kedy bola prijatá správa RR. Táto metrika sa dá využiť k približnému určeniu vzdialenosti vysielača a prijímača.



Obrázok 6: Round-trip delay [7]

Medzipríchodový jitter

Medzipríchodový jitter je priebežne pozorovaná zmena v príchodoch RTP paketov. Najprv sa počíta rozdiel D vzdialenosti dvoch RTP paketov v čase na strane prijímača v porovnaní s vysielateľom podľa vzorca

$$D = (R_j - R_i) - (S_j - S_i),$$

kde R je čas príchodu a S je RTP časová značka daného paketu. Táto zmena sa počíta s každým prichádzajúcim RTP paketom. Pre vyhnutie sa dočasnému kolísaniu je finálna hodnota vyhladená podľa rovnice

$$J_i = J_{i-1} + (|D| - J_{i-1})/16 = (15/16)J_{i-1} + (1/16)|D|$$

Zmena v tejto hodnote by mala indikovať zahltenie pred tým ako sa začnú zahadzovať pakety.

Strata paketov

Správa *receiver report* obsahuje informáciu o stratených paketoch. Toto číslo je definované ako zlomok počtu stratených paketov a počtu očakávaných paketov. Tento počet očakávaných paketov je vypočítaný na základe prijatých RTP paketov a ich najvyššieho sekvenčného čísla. Táto metrika môže byť použitá ako indikátor zahltenia a informovať tak vysielateľ aby zredukoval rýchlosť odosielania dát.

Metriky straty a zahadzovania paketov

V RTCP XR máme dve počítadlá straty paketov. Jedno počíta pakety stratené v sieti, druhé pakety zahodené v jitter bufferi. Toto oddelenie počítadiel nám umožňuje zistiť zdroj degradácie hovoru.

Bursts metrics

V tejto správe je čas zhľuku (burst time), v ktorom je vysoký pomer stratených či zahodených paketov. Zhľuk je definovaný ako najdlhšia sekvencia, ktorá

- začína stratou alebo zahodením paketu,
- neobsahuje výskyt viacerých za sebou nezahodených paketov,
- končí strateným alebo zahodeným paketom.

Medzera (gap) je čas slabého strácania paketov. Formálne je to jedno z

- a) čas od začiatku RTP relácie do času prijatia posledného paketu pred začatím zhuku,
- b) čas od konca posledného zhuku do správy *report* alebo do konca RTP relácie,
- c) čas medzi dvoma zhukmi.

V tejto medzere musí každý stratený/zahodený paket predchádzať a nasledovať minimálne Gmin paketov, ktoré sú prijaté a nezahodené. Pole hodnoty Gmin sa nachádza v bloku VoIP metrík v správe RTCP XR. Odporúčaná hodnota je 16, ktorá spôsobí v medzere charakteristiku korešpondujúcu dobrej kvalite. V samotnej správe sa vzťahujú k metrikám zhukov tieto polia: *loss rate*, *discard rate*, *burst density*, *gap density*, *burst duration*, *gap duration*.

Delay metrics

Okrem *round-trip delay*, čo je časová vzdialenosť medzi dvoma RTP rozhraniami, sa prenáša aj tzv. *end system delay*, čo je časové oneskorenie nabrané medzi hlasovou aplikáciou RTP rozhraním. Vo vysielacom smere je toto oneskorenie definované ako súčet oneskorenia pri zbieraní vzoriek a oneskorenia pri kódovaní. V prijímacom smere je to súčet oneskorenia v pamäti jitter buffer, oneskorenia pri dekódovaní a oneskorenia v pamäti playout buffer.

Signálové metriky

Tieto metriky by mali poskytnúť informácie o elementoch nesúvisiacich s paketmi v systémoch VoIP na identifikáciu problémov postihujúcich kvalitu hovoru. Prenášajú sa tieto informácie:

- hladina signálu (*signal level*)
- hladina šumu (*noise level*)
- zostatkový útlm echa (*residual echo return loss*)

Metriky kvality hovoru alebo kvality prenosu

Patria sem priame metriky kvality hovoru či prenosu ovplyvňované typom kodeku, stratou/zahadzovaním paketov, oneskorením a pod. V správe sa prenáša R Faktor [6], celočíselná hodnota od 0 do 100, kde všetko menšie ako 50 označuje nepoužiteľnú kvalitu. Ďalej sa v správe prenáša MOS-LQ (zrozumiteľnosť) a MOS-CQ (kvalita konverzácie), celočíselné hodnoty od 10 do 50 odpovedajúce $MOS \times 10$, kde MOS 1 je neakceptovateľná kvalita a MOS 5 je perfektná kvalita. Pri MOS-CQ sa na rozdiel od MOS-LQ uvažuje aj efekt oneskorenia a ostatné efekty, ktoré môžu ovplyvniť kvalitu konverzácie. Vzťah medzi hodnotami MOS a R Faktor môžete vidieť v tabuľke 2.

Spokojnosť	R Faktor	MOS CQ
Veľmi spokojní	90 – 100	4,34 – 5,0
Spokojní	80 – 90	4,03 – 4,34
Niektorí užívatelia spokojní	70 – 80	3,60 – 4,03
Mnoho užívateľov nespokojných	60 – 70	3,10 – 3,60
Takmer všetci nespokojní	50 – 60	2,58 - 3,10
Nepoužiteľné	0 – 50	1,0 – 2,58

Tabuľka 2: Vzťah medzi MOS CQ a R Faktor z ITU-T G.107

5.3 Analýza výsledkov testov pomocou STC

Spirent TestCenter Layer 4-7 Application nám dovoľuje sledovať štatistiky už za behu testu. V tomto bode môžeme v prehľadných grafoch sledovať aktívne relácie, pokusy o vytvorenie relácie, úspešné vytvorenie relácie, neúspešné vytvorenie relácie a zrušené relácie.

V prostredí STC bohužiaľ nevidíme konkrétne žiadnu z metrik spomínaných v kapitole 5.2. Avšak vďaka tomu, že výsledky sú zobrazované v grafoch, môžeme vidieť závislosť medzi množstvom pokusov o nadviazanie spojenia a úspešných, či neúspešných relácií, či odozvu protokolu.

Po skončení testu máme možnosť si zobrazit' výsledky testu. Vďaka tomu, že sa výsledky ukladajú vo formáte CSV s nimi môžeme ďalej pracovať. Priamo výsledkový analyzátor nám ponúka možnosť vyexportovať výsledky do PDF súboru alebo ako HTML stránku.

6 Metodológia a výsledky testovania

Táto práca si kladie za cieľ vytvoriť metodológiu testovania siete SIP VoIP pomocou zariadenia Spirent TestCenter. Okrem testovania výkonnosti siete SIP pomocou STC, budeme inými nástrojmi taktiež sledovať kvalitatívne metriky VoIP. Pri testovaní SIP sa zameriame na odozvy protokolu v rôznych situáciách, pri vytváraní relácií, registrácií užívateľov na server prípadne na známe útoky, napríklad (D)DoS. Primárnym cieľom týchto testov bude ukázať možnosti zariadenia Spirent TestCenter pri testovaní SIP sietí. Sekundárnym cieľom testovania bude zamerať sa na sledovanie kvalitatívnych metrik v prostredí VoIP. Pre sledovanie kvalitatívnych metrik boli vybrané dve technológie. Jednou z nich je HW zariadenie značky BlueScope⁸. Druhou technológiou je monitorovací nástroj *VoIP & Network Quality Manager (VNQM)*⁹ od firmy Solarwinds.

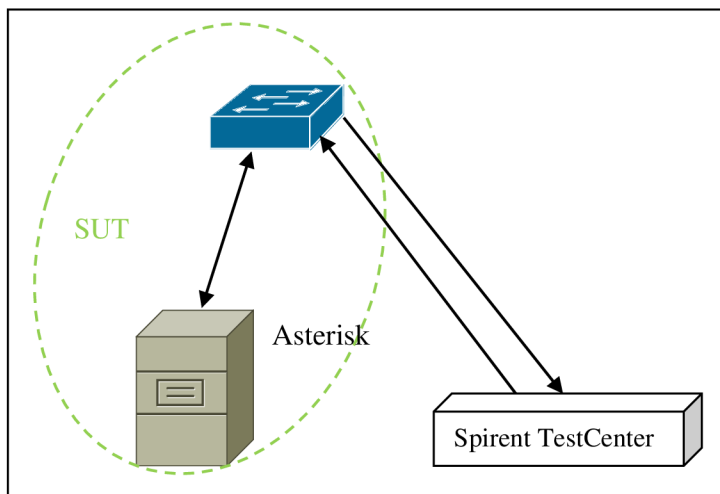
6.1 Metodológia a výsledky testov STC

Názov: *Úspešnosť vytvárania spojenia*

Cieľ: Cieľom tohto testu je ukázať ako si vie ústredňa Asterisk poradiť s veľkým množstvom žiadostí o vytvorenie spojenia

Nastavenia:

- V tomto teste použijeme zariadenie STC v zapojení s ústredňou Asterisk cez jeden prepínač (viď obrázok 7).



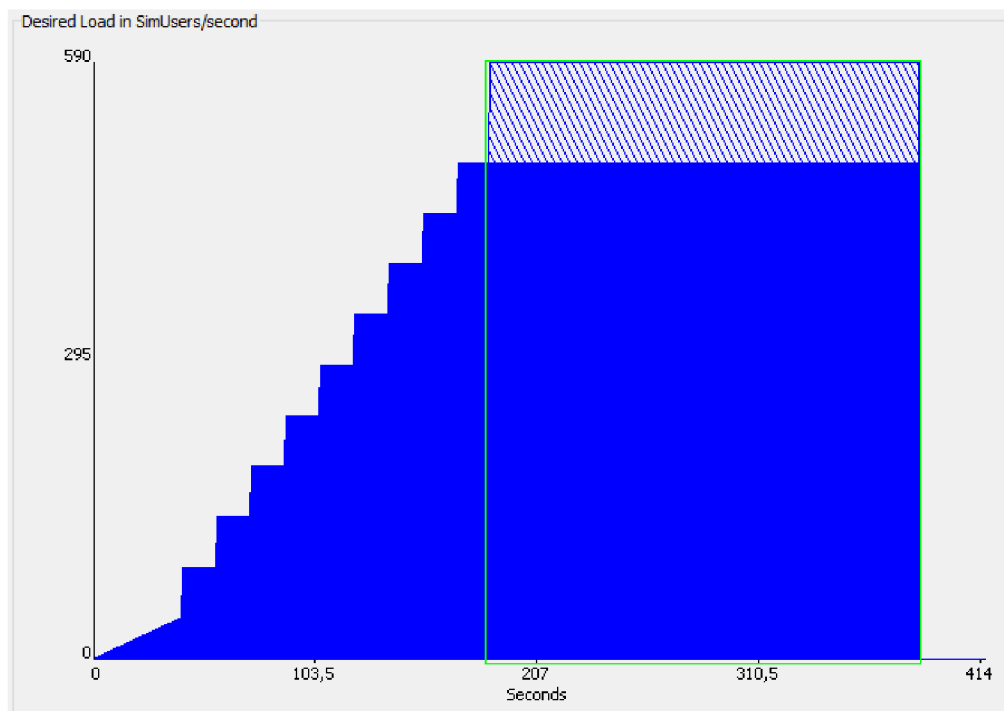
Obrázok 7: Topológia zapojenia s ústredňou Asterisk

⁸ <http://www.bluelighttec.com/bluelight/product/product02.htm>

⁹ <http://www.solarwinds.com/voip-network-quality-manager.aspx>

Konfigurácia:

- Asterisk
 - SIP účty
 - V sekcii [global] nastavíme hodnotu `host` na `dynamic` a nastavíme kontext pre pravidlá volacieho plánu
 - Vytvoríme sekcie užívateľov s položkou `type=friend`
 - Volací plán
 - Vytvoríme rovnaký počet pravidiel s akciou `Answer()` ako je počet užívateľov. Táto akcia bude reprezentovať zdvihnutie slúchadla užívateľa na druhej strane.
- Spirent TestCenter
 - Vytvoríme novú *Advanced test* z kategórie *Application*
 - Tvar záťaže nastavíme podobne ako na obrázku 8. V teste môžeme postupne vyskúšať typ záťaže `SimUsers/s`¹⁰ či `SimUsers`¹¹. Tvar postupného schodovitého zvyšovania počtu užívateľov za sekundu s dĺžkou schodu okolo 15 sekúnd. Vo finálnom stave som nastavil náhodné rozloženie záťaže s výškou 100. Najvyšší počet užívateľov za sekundu nastavte podľa uváženia.



Obrázok 8: Požadovaný tvar záťaže

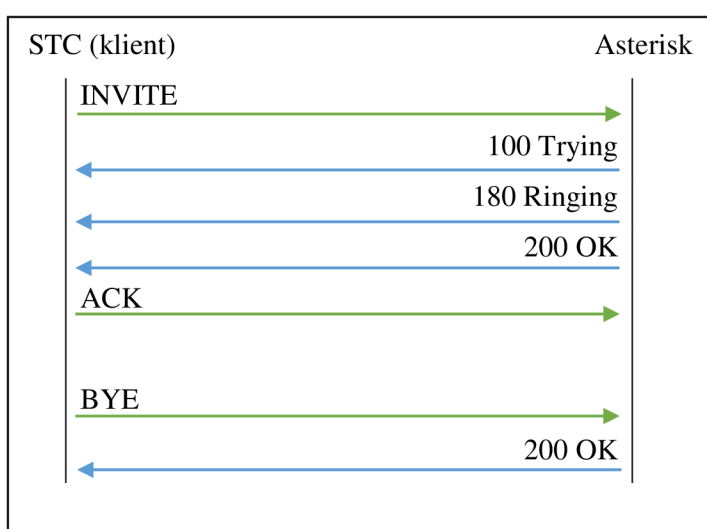
¹⁰ Kontroluje tempo generovanej záťaže. Nemá kontrolu nad počtom aktívnych užívateľov (otvorených spojení)

¹¹ Kontroluje počet aktívnych užívateľov (otvorených spojení). Vytvára zhluky pri snahe udržať počet aktívnych užívateľov (otvorených spojení)

- V *action liste* necháme len riadok začínajúci *sipng://*
- Vytvoríme telefónny zoznam pre *sipng*, do stĺpca *IP address* dáme IP adresu serveru, *Caller* bude tvorený prihlasovacím menom volajúceho užívateľa a IP adresou serveru. *Callee* bude tvorený volacím číslom a IP adresou serveru (ja som vytvoril do zoznamu sto rôznych užívateľov volajúcich na rôzne čísla).
- Vytvoríme podsieť so správnymi adresami.
- Vytvoríme asociáciu

Priebeh testu:

- Priebeh testu s očakávanými generovanými dátami je na obrázku 9



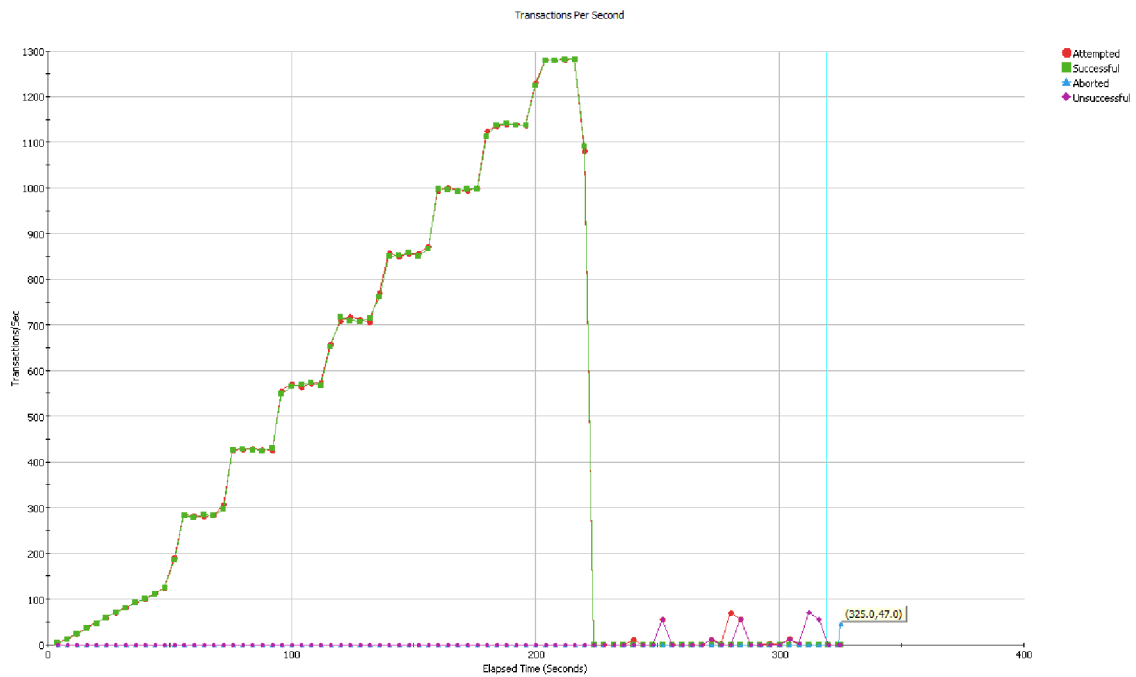
Obrázok 9: Priebeh testu

Očakávané výsledky:

- Zvyšovanie sa odozvy ústredne v závislosti na počte žiadostí o nadviazanie relácie

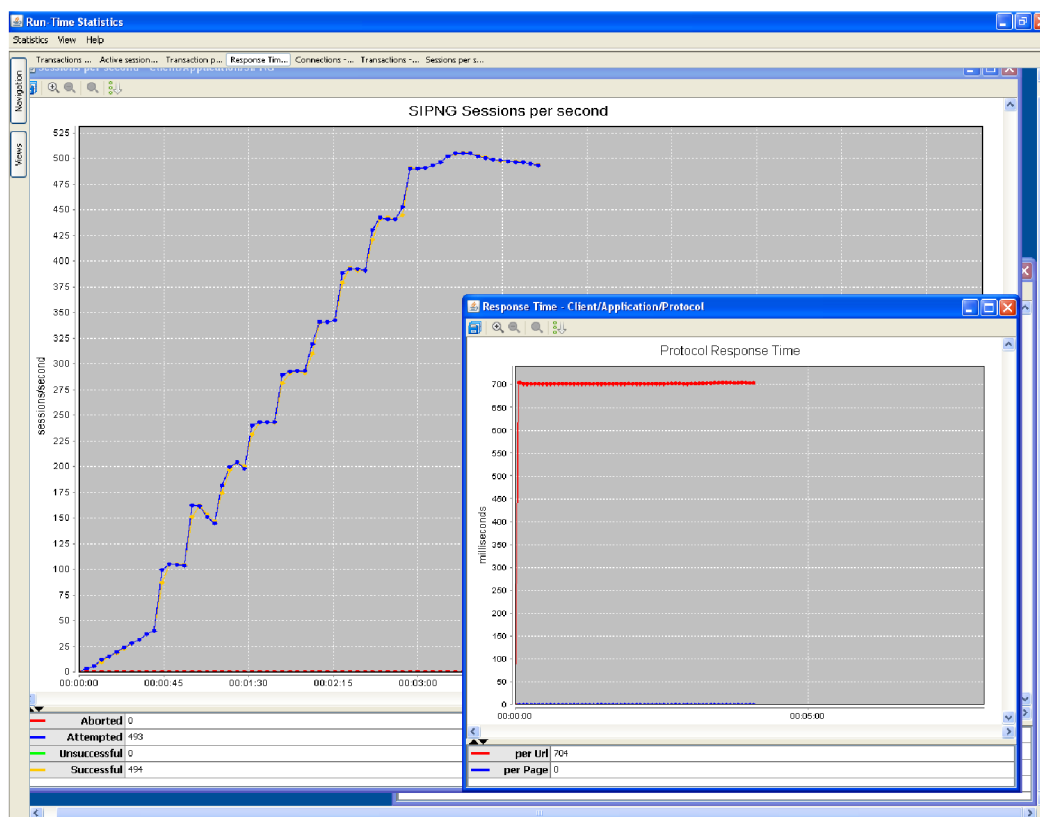
Výsledky:

Prvé spustenia testu ukázali limity serveru. Po dosiahnutí istého počtu relácií Asterisk ukončil činnosť s chybou segmentácie. Dosiahnutý limit sa týka počtu otvorených súborov v systéme danou aplikáciou.



Obrázok 10: Výsledný priebeh testu 1 s dosiahnutím limitu systému a zrušteniu serveru.

Spomínaný limit bol dosiahnutý pri približne 650 reláciách/s čo odpovedalo približne 1300 transakciám/s (vid' obrázok 10).



Obrázok 11: Ukážka run-time štatistík. Priebeh vytvárania relácií a odozvu protokolu.

Po upravení krivky zátáže tak, aby jej vrchol bol približne na hranici tejto hodnoty, sa podarilo test úspešne ukončiť. Odozva protokolu sa držala konštantne pod jednou sekundou, čo môžeme vidieť na obrázku 11.

Vyhodnotenie:

Tento test nám ukázal limity serveru, ktoré sa týkali množstva otvorených súborov aplikáciou. Preto sa výsledky môžu odlišovať v závislosti na nastavení systému na serveri. Okrem tejto skutočnosti nám test ukázal, že Asterisk zvládal vybavovať veľké množstvo prichádzajúcich požiadaviek na vytvorenie relácie takmer bezprostredne. Treba ale brať v úvahu, že Asterisk sa nemusel starať o žiaden iný prebiehajúci hovor, na samotnom serveri nebežala iná serverová aplikácia a v sieti neboli iné dáta.

V porovnaní so štandardnými metrikami môžeme zobrazené relácie/s či transakcie/s prirovnať k SRD.

Názov: Úspešnosť vytvárania spojenia, ktoré predchádzala registrácia

Cieľ: Tento test má ukázať ako ovplyvní výkon serveru vytváranie relácii spolu s registráciou a aký to má vplyv na odozvu protokolu.

Nastavenia:

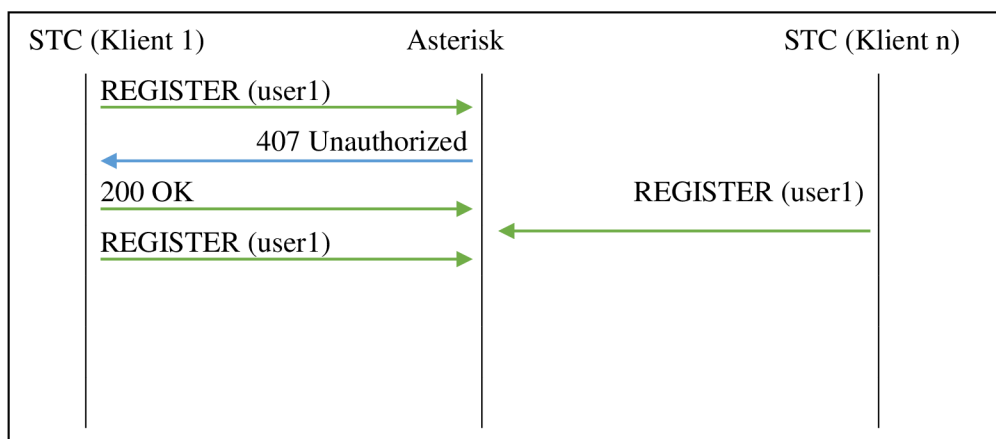
- Topológia bude rovnaká ako v predchádzajúcom teste (viď obrázok 7).

Konfigurácia:

- Asterisk
 - Riešením problému s registráciami z rôznych IP adries je nekonfigurovať na ústredni heslo pre SIP účty, čím dovolíme vytvoriť spojenie, aj keď registrácia prišla z inej adresy. Preto bude nastavenie Asterisku rovnaké ako pri predchádzajúcom teste.
- Spirent TestCenter
 - Vytvoríme si nový *advanced test*, kategória *application*.
 - Krivku záťaže ponecháme rovnakú ako v predchádzajúcom teste (viď obrázok 8).
 - Vytvoríme nový *action list*.
 - Vytvoríme kópiu telefónneho zoznamu z predchádzajúceho testu a v tejto kópii vyplníme položky potrebné na registráciu.
 - Pridáme podsieť a vytvoríme asociácie.

Priebeh testu:

- Očakávaný priebeh testu s generovanými správami je na obrázku 12



Obrázok 12: Priebeh testu pri využití telefónneho zoznamu v STC

- Zobrazený priebeh testu je ovplyvnený konfiguráciou jediného telefónneho zoznamu klientov
 - Pri takejto konfigurácii testu, sa stane to, že sa na Asterisk odošle niekoľko žiadostí o registráciu jedného účtu z rôznych klientov (IP adries)
 - Inou možnosťou konfigurácie je pre každého simulovaného klienta nakonfigurovať vlastný telefónny zoznam, vlastný *Action List*, vlastnú podsieť s jedinou jeho IP

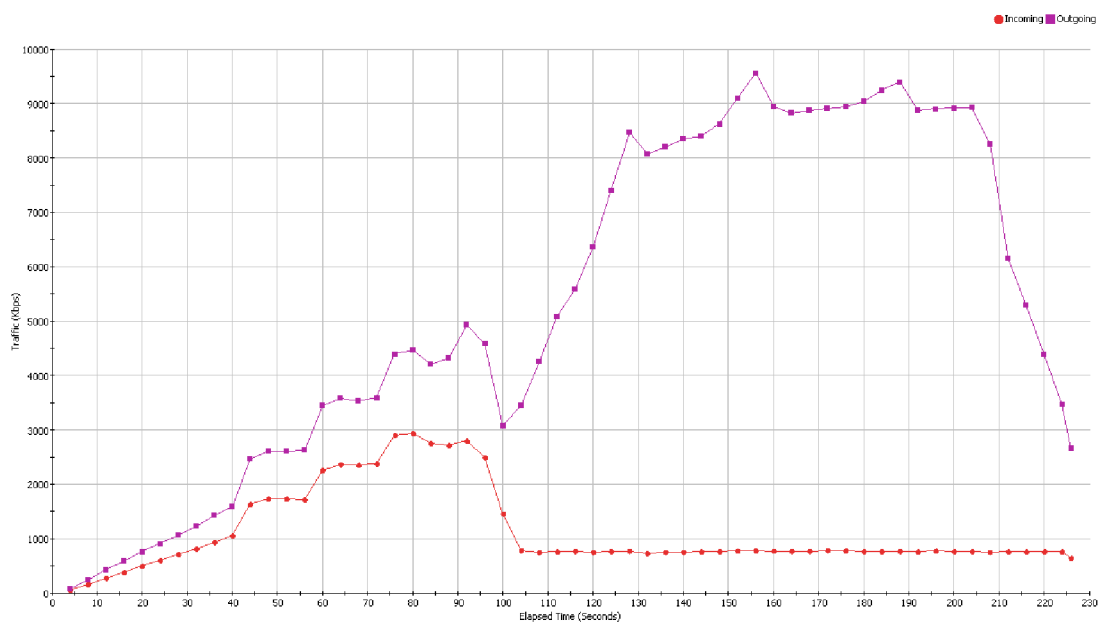
adresou a pre každého klienta vytvoriť vlastnú asociáciu. S krivkou záťaže stúpajúcou k 600 simulovaných užívateľov by sme potrebovali rovnaký počet osobitne nakonfigurovaných klientov/asociácii a telefónnych zoznamov.

Očakávané výsledky:

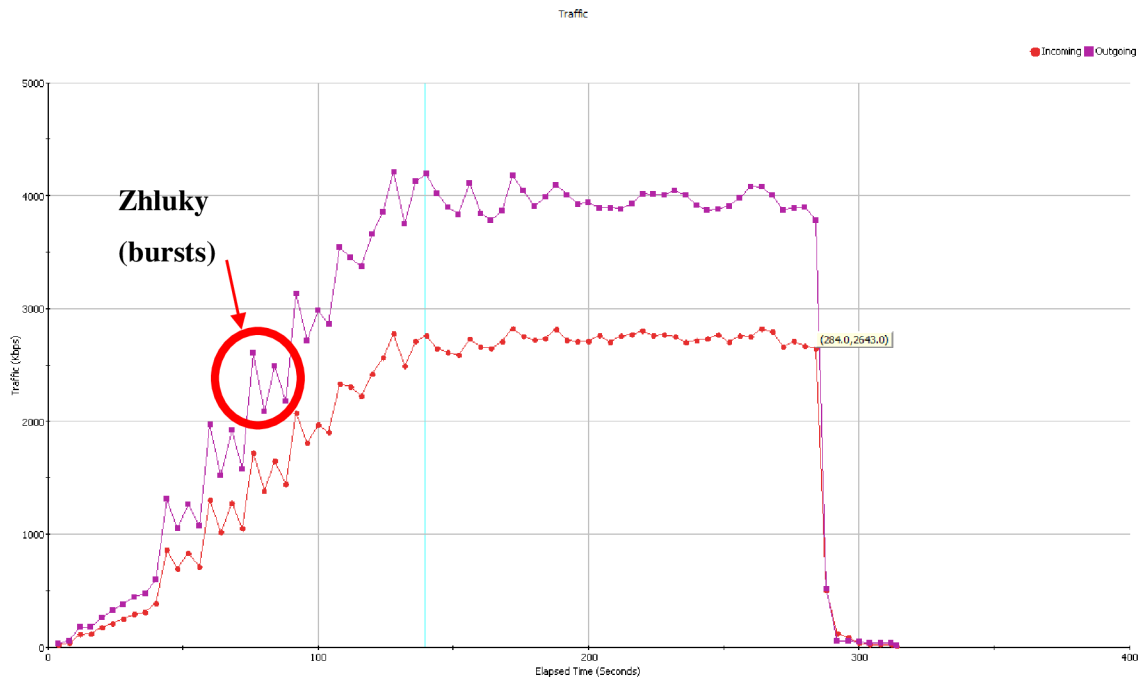
- Vzhľadom na to, že registrácie budú prichádzať naraz z rôznych strán očakávame veľký nárast odozvy

Výsledky:

Spustenie testov s pôvodnou krivkou záťaže v SimUsers/s skončilo vždy zrušením serveru. Aj po upravení výšky krivky sa opakoval rovnaký scenár (viď obrázok 13).

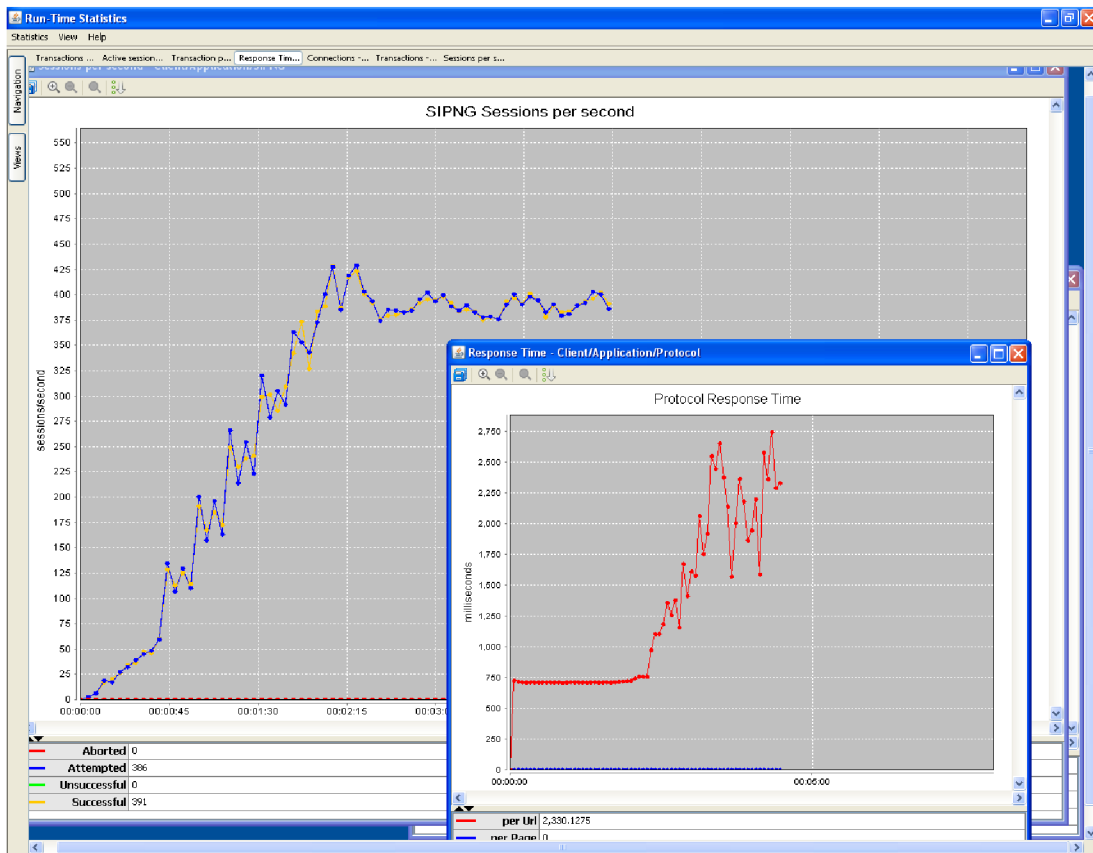


Obrázok 13: Graf prevádzky pri type záťaže SimUsers/s. V tomto teste sa server zrútil v čase okolo 80s.



Obrázok 14: Graf prevádzky pri type záťaže v SimUsers.

Po zmene typu krivky na SimUsers test prebehol úspešne do konca. Za povšimnutie stojí približne o polovicu menšia prevádzka ako v predchádzajúcich pokusoch (viď obrázok 14).

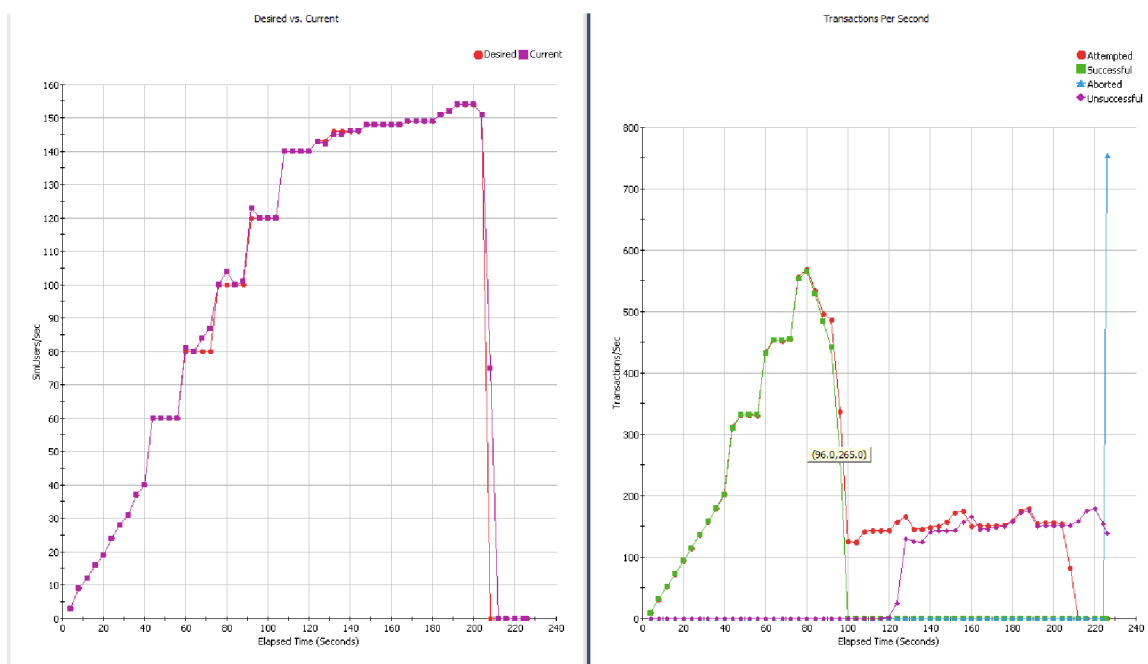


Obrázok 15: Run-time štatistiky ukázali nárast odozvy protokolu pri zvyšovaní počtu klientov.

Pri sledovaní štatistik počas testu sme si všimli nárast odozvy protokolu až na 2,75s (viď obrázok 15).

Vyhodnotenie:

Tento test nám ukázal hlavne nedostatky pri konfigurácii testu, kedy sa viacerí simulovaní klienti (rôzne IP adresy) snažia registrovať pod rovnakým menom. Ďalej sme zistili, že pri pokusoch o registráciu pred vytvorením relácií sa generuje dvojnásobná prevádzka pri type záťaže SimUsers/s v porovnaní s typom záťaže v SimUsers, ktorá dokáže zahltiť server aj s okolo 100 SimUsers/s (viď obrázok 16). Taktiež sme v tomto teste videli nárast odozvy protokolu so zvyšujúcim sa počtom požiadaviek.



Obrázok 16: Graf znázorňujúci pád serveru pri počte simulovaných užívateľov.

Názov: Úspešnosť vytvárania spojenia v sieti s HTTP serverom

Cieľ: Cieľom tohto testu je predviesť vytváranie testu, ktorý obsahuje ako simulovaných klientov tak aj simulovaný server. Takéto typy testov sa môžu hodiť napríklad pri testovaní QoS. Ak by sme chceli vidieť vplyv iného serveru na našu ústredňu, vytvorili by sme server priamo na ústredni.

Nastavenia:

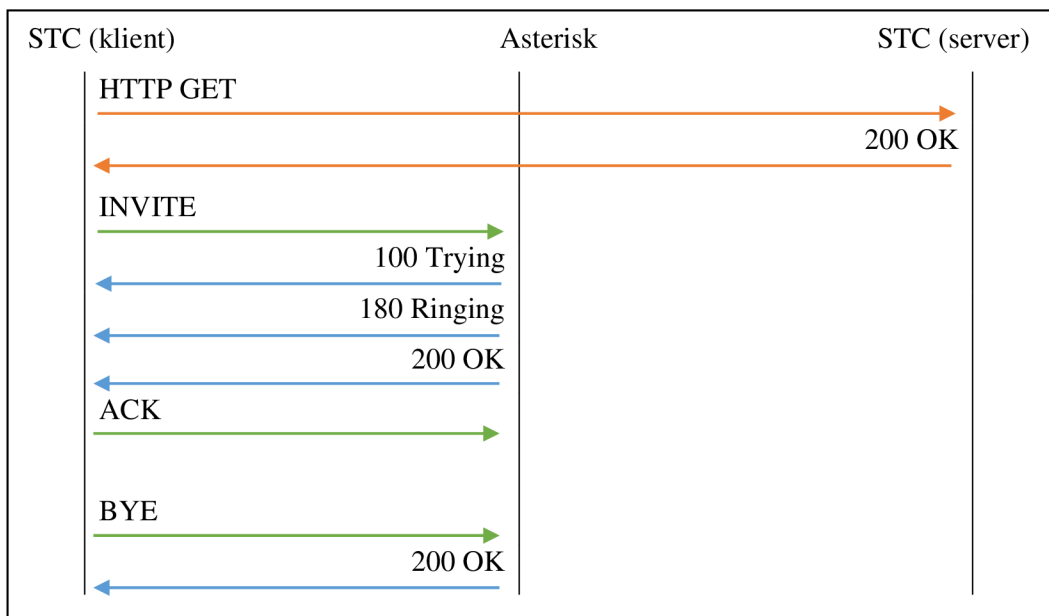
- Topológia ostáva rovnaká ako na obrázku 7.

Konfigurácia:

- Asterisk
 - V tomto teste sa konfigurácia ústredne oproti predchádzajúcim testom nijako nemení
- Spirent TestCenter
 - Vytvoríme si nový *advanced test*, kategória device
 - Krivku záťaže ponecháme ako v teste 1 (viď obrázok 8)
 - Vytvoríme nový *action list*, kde okrem riadku so SIPNG ponecháme aj prvé dva riadky
 - Vytvoríme podsieť a asociáciu
 - Na karte serveru vyberieme typ serveru HTTP a protokol HTTP 1.1
 - Vytvoríme podsieť a v asociácii do IP rozsahu dáme iba jednu adresu serveru

Priebeh testu:

- Očakávaný priebeh testu je zobrazený na obrázku 17.



Obrázok 17: Očakávaný priebeh testu

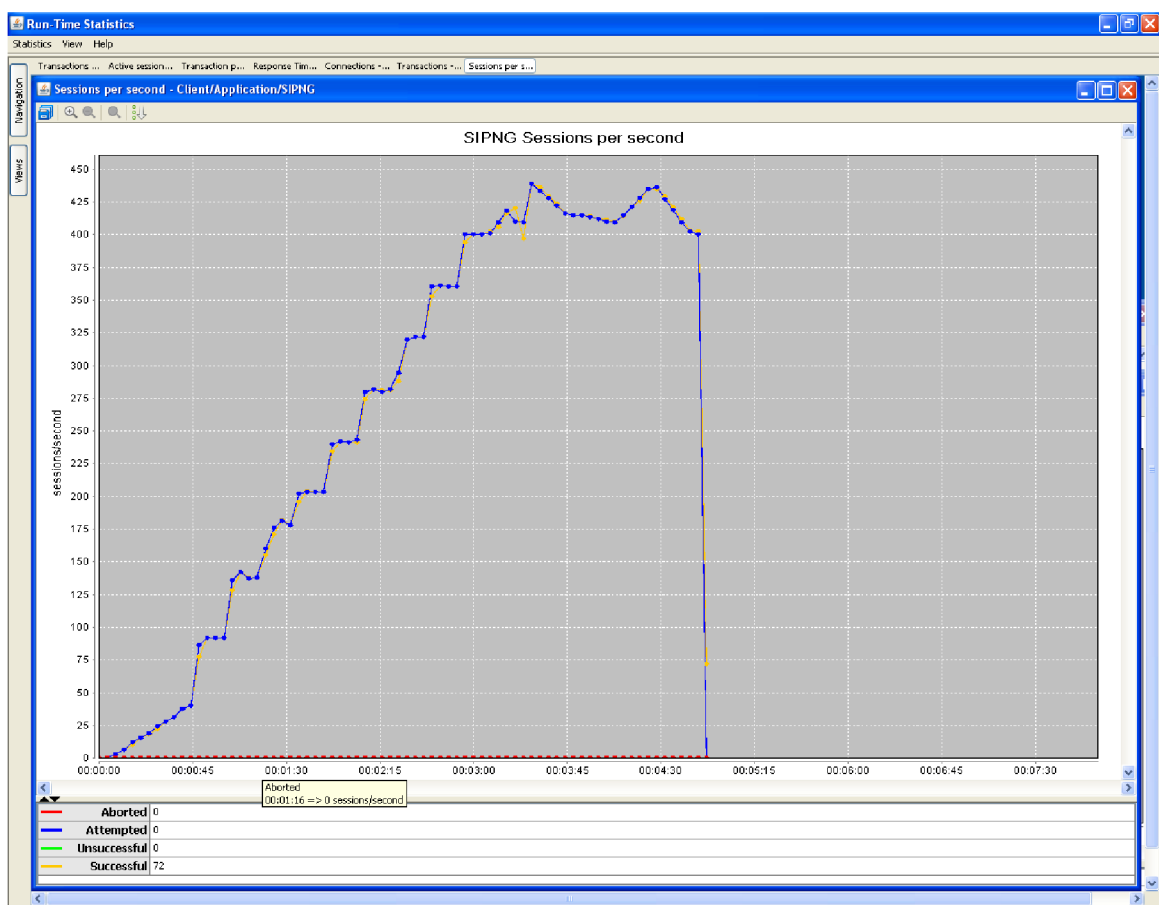
- Poradie správ HTTP a SIP je ovplyvnené poradím akcií v *Action Liste* v Avalanche

Očakávané výsledky:

Vzhľadom na to, že spomínaný HTTP server nebeží na rovnakom stroji ako Asterisk, v tomto teste budeme očakávať podobné výsledky ako v prvom teste. Odozvu ústredne bude ovplyvňovať len množstvo prichádzajúcich požiadavkov.

Výsledky:

V tomto teste vidíme, že prítomnosť iného serveru v topológii nemá vplyv na odozvy ústredne, graf počtu pokusov o nadviazanie spojenia takmer dokonale opisuje graf úspešne nadviazaných spojení (viď obrázok 18). Tento test som vyskúšal taktiež so zapnutou funkciou fragmentácie na IP vrstve a poprehadzovaním fragmentov, avšak to tiež neukázalo žiadne zmeny v odozve.



Obrázok 18: Priebeh testu 3 s povolenou fragmentáciou nepreukázal žiadne výrazné zmeny v odozve.

Vyhodnotenie:

Tento test nám ukázal, že aj pri zapojení dvoch serverov v našej testovacej topológii všetko prebiehalo podľa očakávaní a nezaznamenali sme žiadne výrazné zmeny v odozve. Tento test nám taktiež ukázal ako nakonfigurovať test so simulovaným serverom. Ďalšou obmenou tohto testu by bolo spustiť serverovú aplikáciu na ústredni a zistiť ako vplýva takáto kombinácia na výkon ústredne.

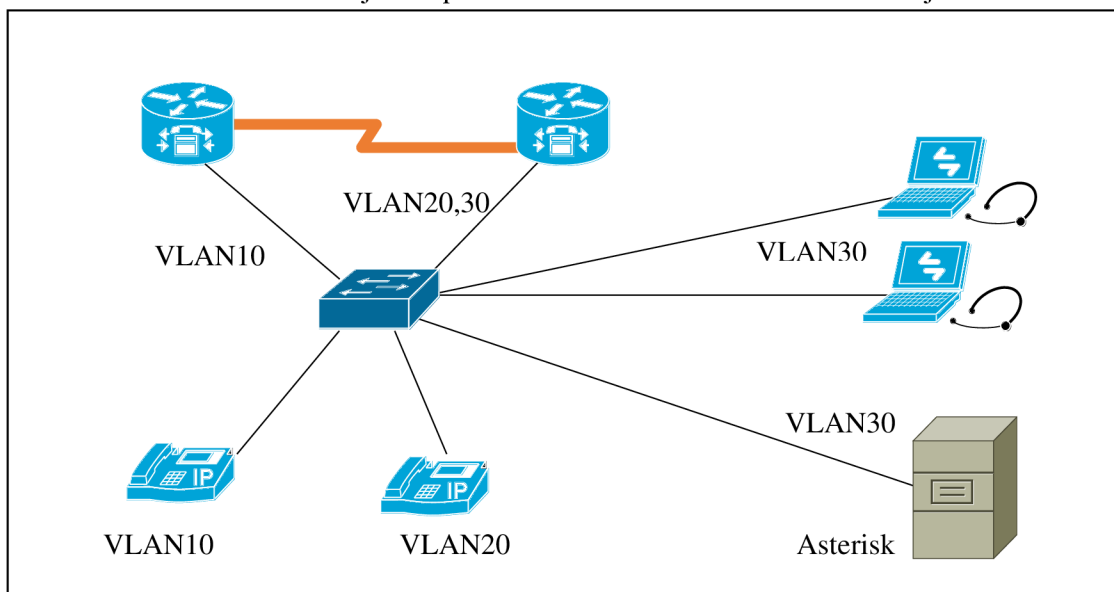
6.2 Metodológia a výsledky kvalitatívnych testov

Názov: *Kvalita hovoru v prostredí s rôznymi ústredňami*

Cieľ: Cieľom tohto testu je monitorovanie kvality hovorov.

Nastavenia:

- Pre tento test použijeme topológiu na obrázku 19
- Ako monitorovací nástroj bude použité zariadenie BLUELIGHT a nástroj SolarWinds VNQM



Obrázok 19: Topológia pri testovaní kvality hovorov

Konfigurácia:

- Nakonfigurujeme Call Manager na dvoch zariadeniach (viď Príloha 2)
- Na prepínači nakonfigurujeme potrebné VLAN (trunkly a access porty)
- Nakonfigurujeme Asterisk tak, aby sa bolo možné spojiť s ostatnými ústredňami (viď Príloha 2)
 - Pravidlo s akciou `Dial(SIP/<IP_adresa_CCM_VLAN30>/${EXTEN})`
- Pripojíme softwarové telefóny k ústredni Asterisk

Príbeh testu:

- Pri testovaní zariadenia BLUELIGHT bude test spočívať vo vytváraní reálnych hovorov takých, ktoré bude tento nástroj schopný zachytiť
- Pri testovaní SolarWinds VNQM bude test spočívať vo vytvorení SLA operácií týmto nástrojom na Cisco ústredniach a preskúmaním možností tohto nástroju pre zobrazovanie kvalitatívnych metrik

Očakávané výsledky:

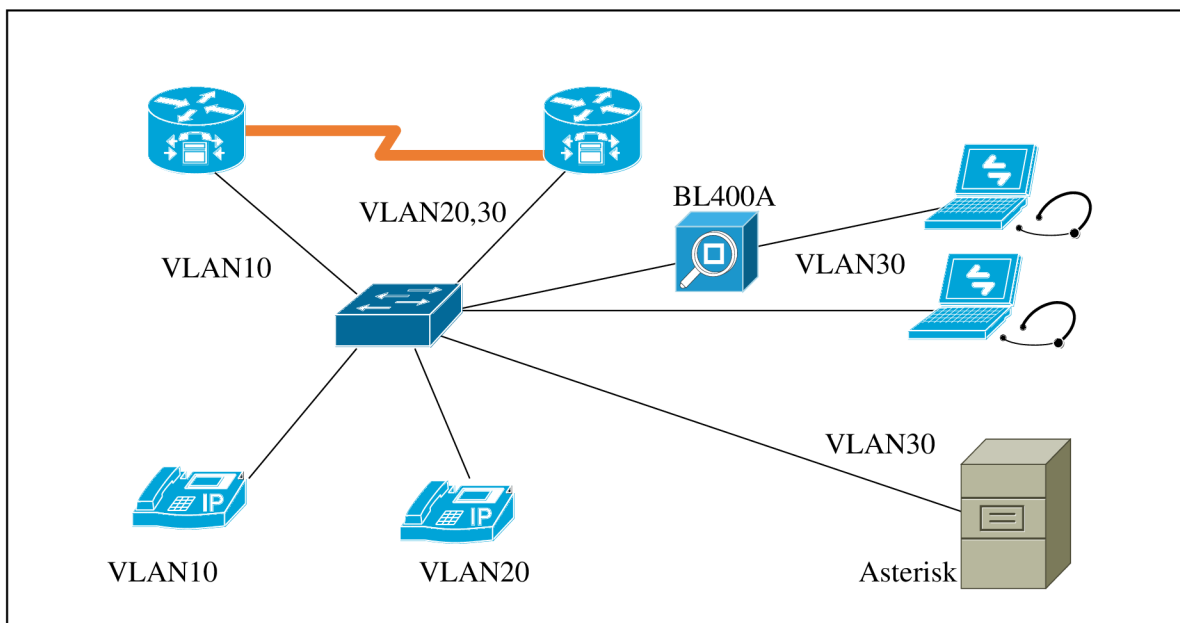
- Rôzna nameraná kvalita hovorov v jednej logickej sieti a medzi týmito sieťami

6.2.1 Výsledky testov BLUELIGHT BL400A

Zariadenie od BLUELIGHT je prenosný tablet používaný na testovanie sietí a sieťových prvkov. Toto zariadenie na rozdiel od Spirent TestCenter neposkytuje záťažové testy pre VoIP siete, ale testy kvality hovorov a meranie základných metrik kvality hovoru ako sú oneskorenie, jitter, R-Faktor a MOS. Tieto informácie zariadenie získava z RTCP paketov aktuálne prebiehajúcich hovorov.

Zariadenie poskytuje dva typy testov, *Monitor (Terminated)* a *Monitor (PassThrough)*. Pri druhom spomínanom zariadenie analyzuje pakety, ktoré mu prídu na jeden port a nezmenené ich pošle von druhým portom. Pri teste *Monitor (Terminated)* by sa zariadenie, ako názov naznačuje, malo správať ako koncový bod. Toto správanie sa mi však nepodarilo potvrdiť ani po viacerých pokusoch.

Na obrázku 20 je zobrazené konkrétne zapojenie zariadenie BLUELIGHT BL400A v testovacej topológii



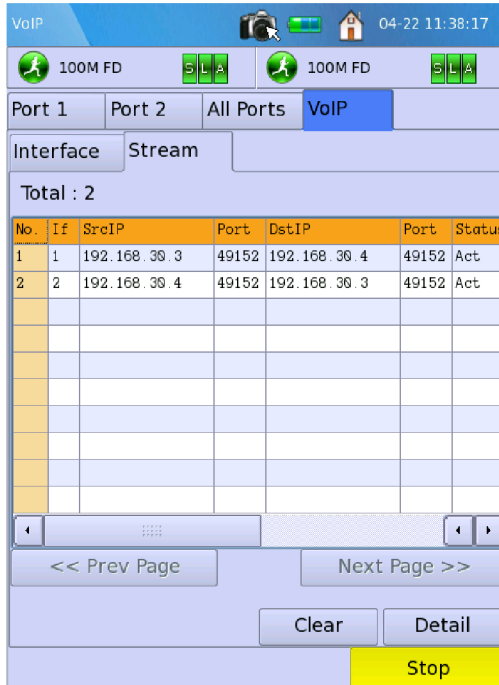
Obrázok 20: Topológia pri testovaní BLUELIGHT BL400A

Postup merania:

- Na zariadení som vybral VoIP test, typ *Monitor(PassThrough)*
- Pripojil som zariadenie medzi prepínač a SIPPhone
- Porty je potrebné nastaviť tak, aby pracovali na rovnakej rýchlosti a rovnakom duplexnom režime
- Pod kartou *All Ports* aktivujeme porty a na karte *VoIP* spustíme test
- Postupne voláme na telefón, na ktorom je pripojené zariadenie

Výsledky:

Prvý zachytený hovor sa uskutočnil medzi dvoma SJPhonami. Na zariadení sa nám objavili dva streamy. V ich detailnom náhľade vidíme metriky získané z RTCP (viď obrázok 21)



VoIP 04-22 11:38:17

100M FD SLA 100M FD SLA

Port 1 Port 2 All Ports VoIP

Interface Stream

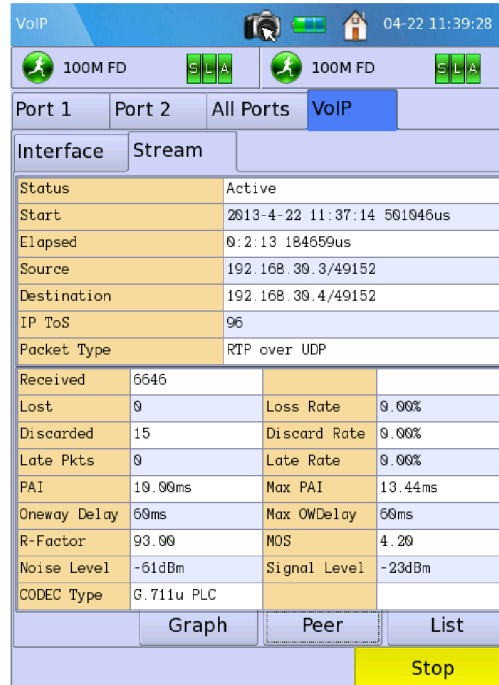
Total : 2

No.	If	SrcIP	Port	DstIP	Port	Status
1	1	192.168.30.3	49152	192.168.30.4	49152	Act
2	2	192.168.30.4	49152	192.168.30.3	49152	Act

<< Prev Page Next Page >>

Clear Detail

Stop



VoIP 04-22 11:39:28

100M FD SLA 100M FD SLA

Port 1 Port 2 All Ports VoIP

Interface Stream

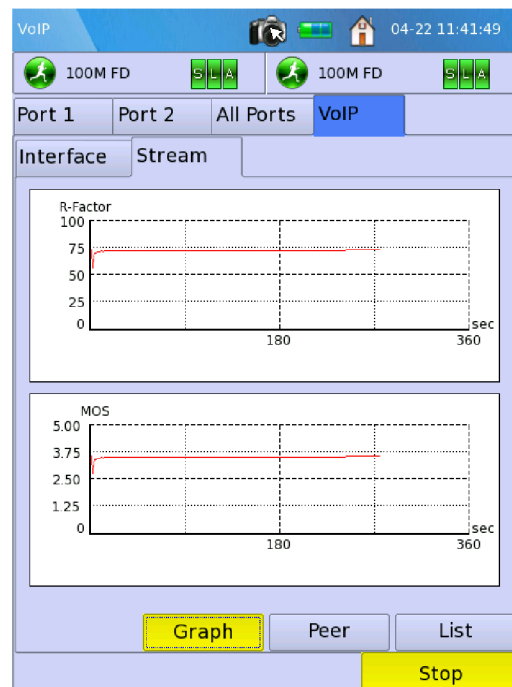
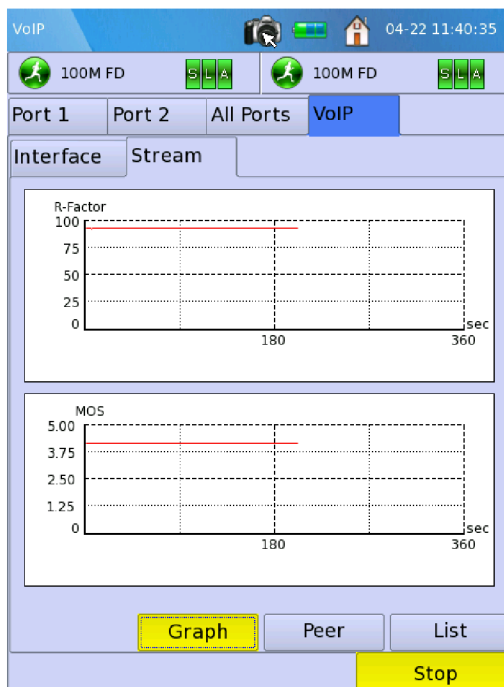
Status	Active
Start	2013-4-22 11:37:14 501046us
Elapsed	0:2:13 184659us
Source	192.168.30.3/49152
Destination	192.168.30.4/49152
IP ToS	96
Packet Type	RTP over UDP

Received	6646		
Lost	0	Loss Rate	0.00%
Discarded	15	Discard Rate	0.00%
Late Pkts	0	Late Rate	0.00%
PAI	10.00ms	Max PAI	13.44ms
Oneway Delay	60ms	Max OWDelay	60ms
R-Factor	93.00	MOS	4.20
Noise Level	-61dBm	Signal Level	-23dBm
CODEC Type	G.711u PLC		

Graph Peer List

Stop

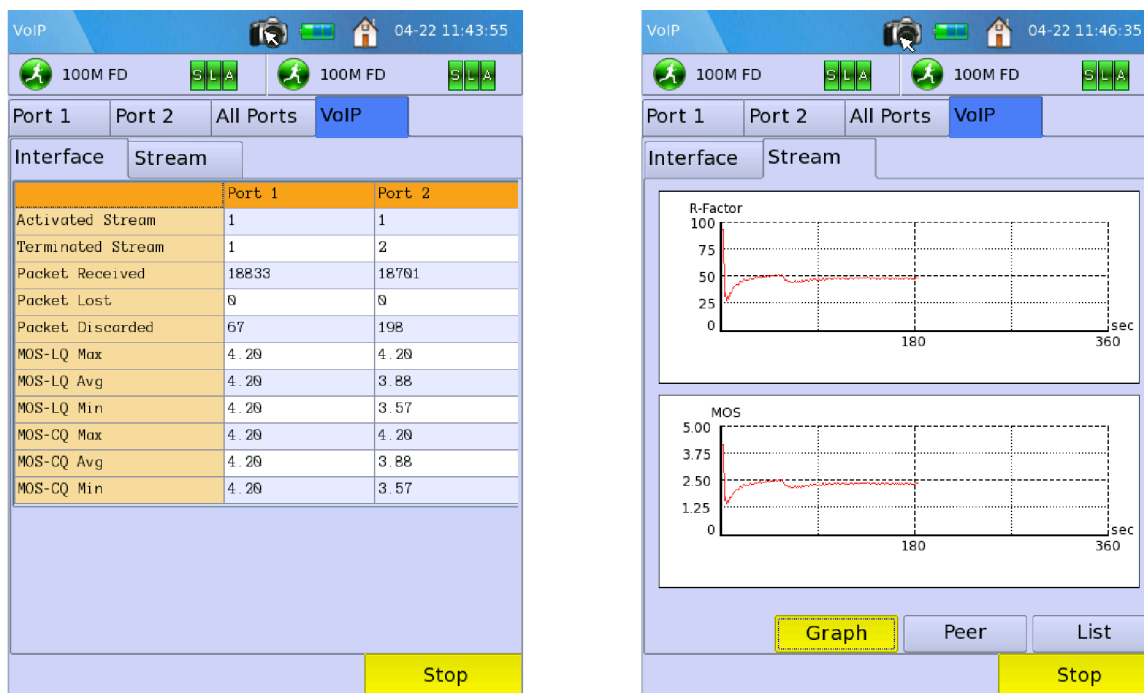
Obrázok 21: Zobrazenie RTP streamov prebiehajúcich pri jednom hovore a detailný náhľad streamu



Obrázok 22: Grafy priebehu merania hodnoty MOS a R-Faktor. Vľavo stream detailu z obrázku 18

Pri preštudovaní grafov jednotlivých streamov si môžeme všimnúť mierného rozdielu v kvalite (vid' obrázok 22).

Ďalší hovor bol iniciovaný z IP Telefónu na Vlan 10 (t.j. vzdialenejší Cisco Call Manager) s tým, že pôvodný hovor nebol zrušený, ale podržaný. Na zariadení sa podržané hovory zobrazujú ako ukončené streamy. Zvláštnosť je, že počet ukončených streamov bol na jednotlivých portoch zariadenia odlišný (obrázok 23 vľavo).



Obrázok 23: Štatistiky na jednotlivých rozhraniach a grafy kvality hovoru streamu od Cisco Call manageru

Na obrázku 23 vpravo vidíme, že kvalita hovoru je tesne pod hranicou prijateľnosti. Zaujímavé je, že v druhom smere bola kvalita ešte vyhovujúca. To naznačuje možné problémy pri prenose zvukových dát na sériovom spoji medzi ústredňami Cisco Call Manager.

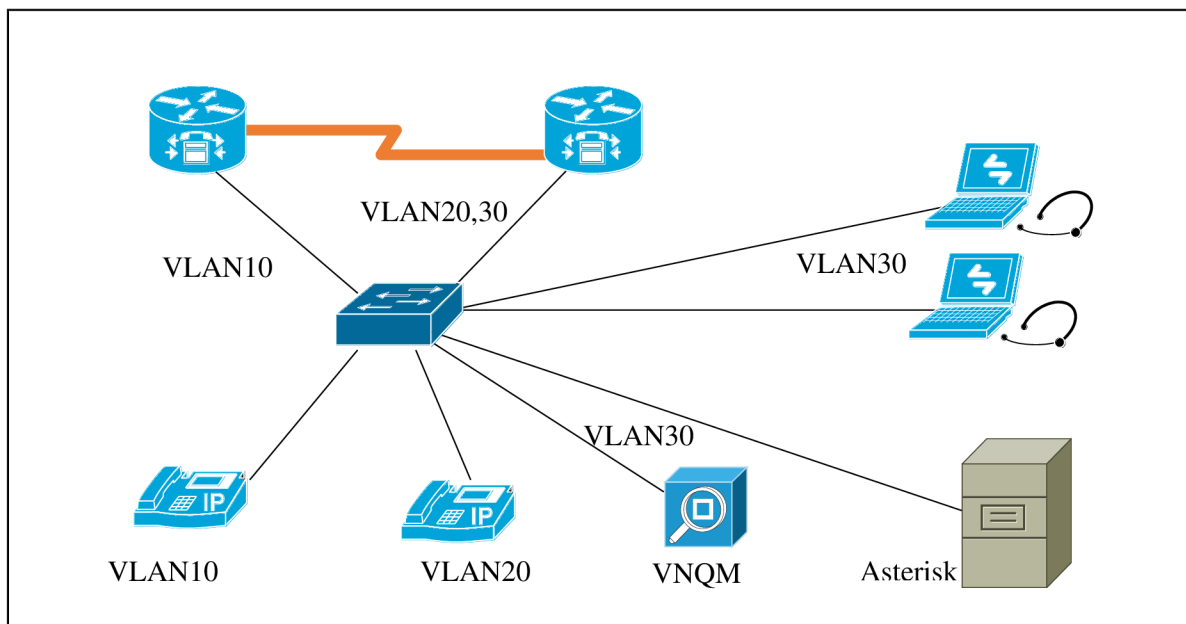
Vyhodnotenie:

Zariadenie BLUELIGHT BL400A je dobre použiteľné pri meraní kvality hovorov prebiehajúcich aj na veľké vzdialenosti. Tým, že pripojíme zariadenie medzi koncové zariadenie a prepínač, získame prehľad o všetkých hovoroch smerujúcich z a na daný terminál. Zariadenie získa z RTCP správ hodnoty a čitateľne ich zobrazí v tabuľkách a grafoch.

6.2.2 Výsledky testov Solarwinds VNQM

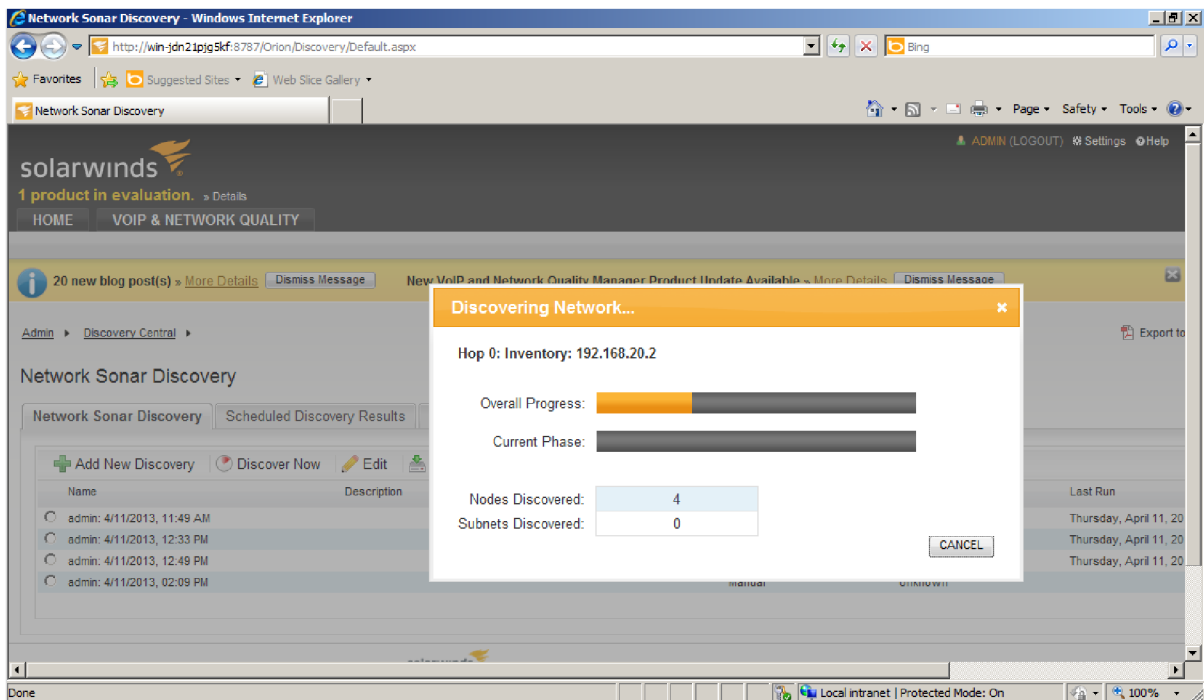
Nástroj od firmy Solarwinds slúži na monitorovanie VoIP sietí. VNQM poskytuje monitorovacie, výstražné a záznamové funkcie. Monitor využíva Cisco SLA na generovanie simulovanej VoIP prevádzky. Vďaka simulovanej prevádzke môžeme vedieť, aký je stav siete kedykoľvek, aj keď reálne neprebíha žiaden hovor. Na rozdiel od zariadenia Spirent TestCenter neposkytuje možnosť vytvárania záťažových testov.

Tento nástroj pracuje so zariadeniami Cisco, konkrétne *Call manager* prípadne *Call manager express*. Z toho dôvodu som nástroj testoval na rovnakej topológii ako zariadenie BLUELIGHT (viď obrázok 24).



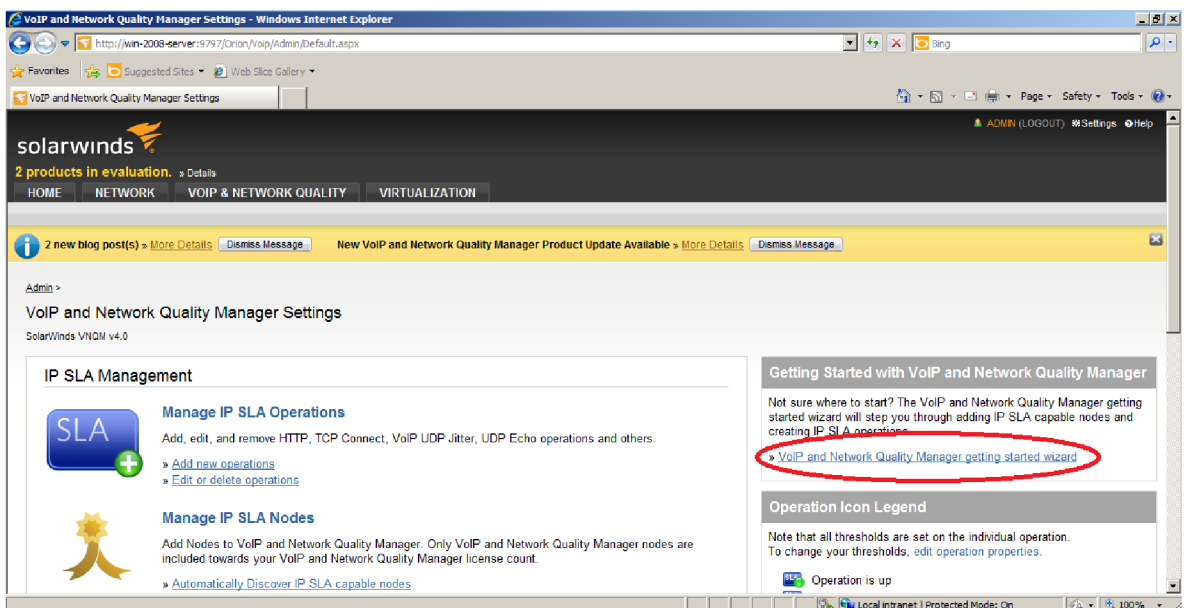
Obrázok 24: Topológia pri testovaní VNQM

Na rozdiel od zariadenia BLUELIGHT však tento nástroj nemusí byť zapojený priamo medzi volajúcimi stranami. K správne fungovaniu potrebujeme mať nainštalované *Internet Information Services (IIS)*, ktoré povolíme v ovládacích paneloch Windows. Po nainštalovaní môžeme prístupit k webovému rozhraniu (*Web Console*). Pomocou voľby *Network Discovery* preskúmame našu sieť a získame znalosti o každom zariadení (obrázok 25).



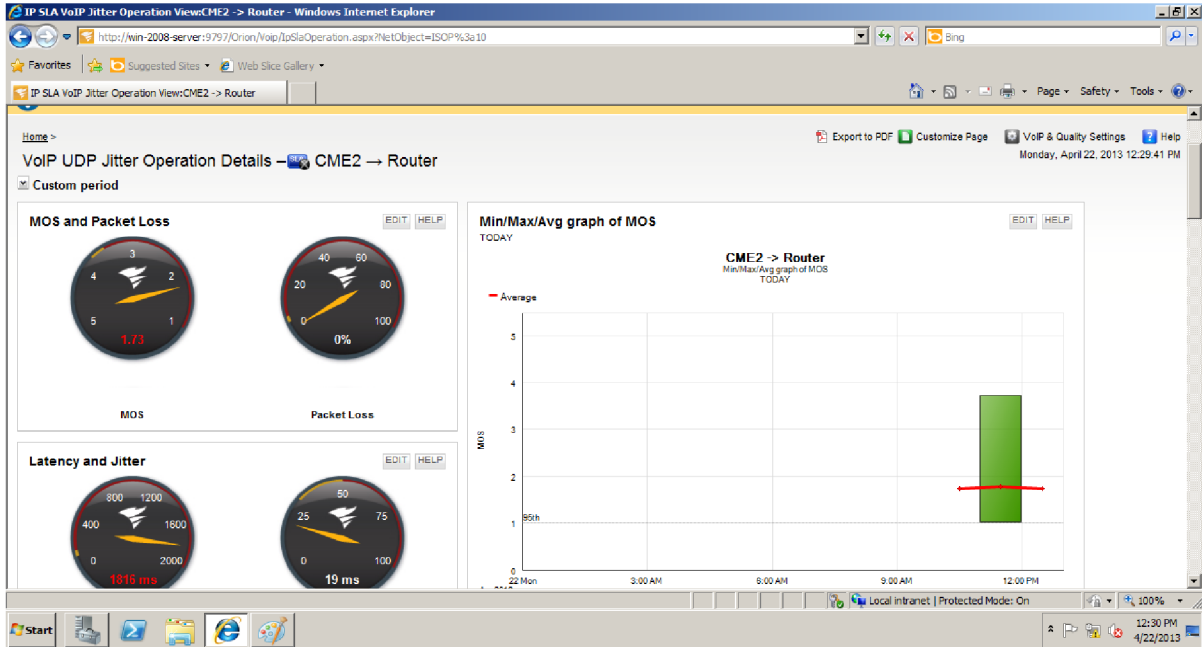
Obrázok 25: Solarwinds priebeh procesu Network Discovery

Na zariadeniach Cisco je potrebné mať zapnuté SNMP a nakonfigurovanú SNMP komunitu typu *rw*. To nám však k tomu, aby sme v tejto aplikácii videli zaujímavé dáta nestačí. V aplikácii potrebujeme nakonfigurovať tzv. SLA operácie. SLA operácie na Cisco zariadeniach fungujú ako aktívny monitor/sonda, tzn. že generuje vlastnú prevádzku za účelom merania výkonnosti. Pridať novú SLA operáciu môžeme napríklad cez *Getting Started Wizard* v nastaveniach (viď obrázok 26). VNQM potom pomocou SNMP správy *SetRequest* nastaví potrebné SLA operácie na smerovači/ústredni.



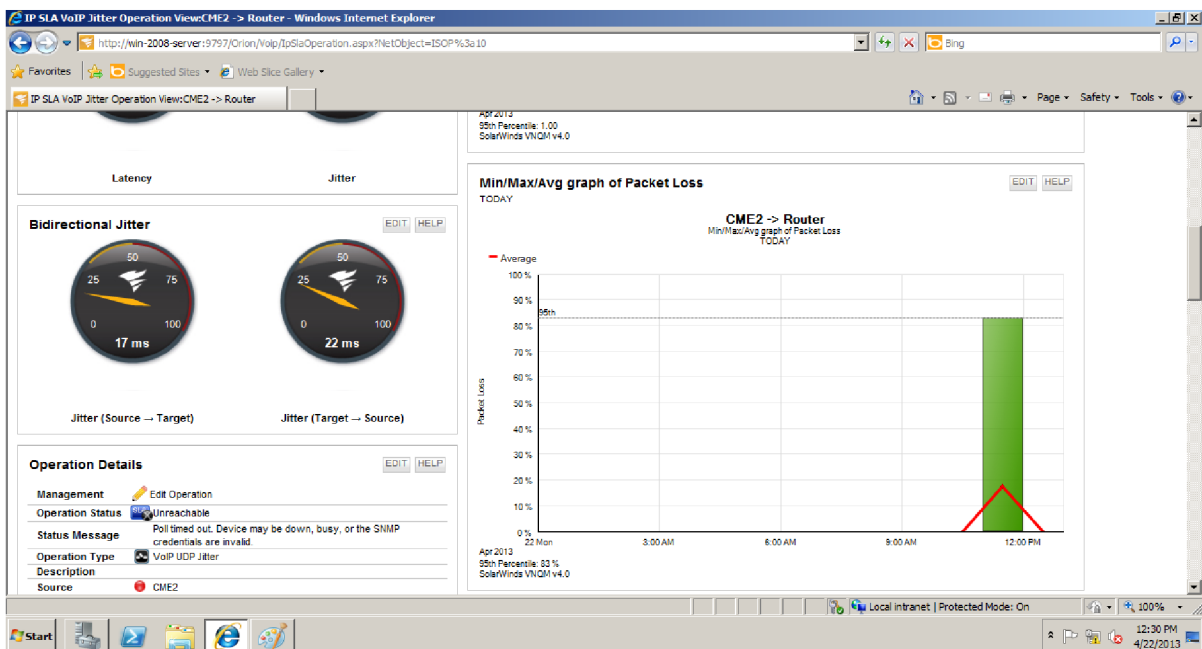
Obrázok 26: Umiestnenie sprievodcu v časti týkajúcej sa nastavení VNQM

Po pridaní SLA operácií už v nástroji môžeme vidieť informácie napríklad o MOS na jednotlivých linkách, ktoré sme pridalí v predchádzajúcom kroku. V detailoch operácie vidíme aj históriu kvality. VNQM tieto dáta získava pomocou SNMP správ GetRequest, GetNextRequest a GetBulkRequest.



Obrázok 27: Detail operácie Jitter

V detailnom náhľade operácie *Jitter* medzi dvoma Call managermi vidíme históriu vývoja kvality v metrike MOS, taktiež vidíme aktuálnu hodnotu *MOS*, *Packet Loss*, *Latency* a *Jitter* na budíkoch vľavo (obrázok 27 a obrázok 28).



Obrázok 28: Detail operácie Jitter (iný pohľad)

Vyhodnotenie:

Solarwinds VNQM je výborný nástroj pre dlhodobé monitorovanie VoIP siete, primárne postavenej na technológii Cisco. Podľa užívateľského manuálu¹², VNQM ponúka aj prídanie nie Cisco ústrední a to vytvorením vlastných MIB dotazov (*Management Information Base*).

¹² <http://www.solarwinds.com/documentation/IPSLA/docs/VNQMAdministratorGuide.pdf>

7 Záver

Siete VoIP sa postupne stávajú bežnou súčasťou života moderného človeka. Vo VoIP sieťach však sledujeme isté výkyvy kvality, ktoré v klasických telefónnych sieťach nie sú obvyklé. Preto je veľmi dôležité poznať spôsoby a prostriedky monitorovania a testovania VoIP sietí.

V tejto práci som sa zaoberal vytváraním testov siete SIP VoIP v prostredí Sirent TestCenter a porovnaním možností tohto zariadenia s inými technológiami na testovanie sietí VoIP. Táto práca si vyžadovala naštudovanie základov protokolu SIP, konfigurácie niektorej SIP telefónnej ústredne a naštudovanie rozhrania Spirent TestCenter Avalanche.

Po naštudovaní týchto princípov som začal vytvárať testovacie scenáre. V tejto časti som sa inšpiroval hlavne dokumentom RFC 6076, v ktorom sú popísané štandardné metriky pri testovaní výkonu SIP. Jednotlivé testy sú vytvorené v rámci licenčných obmedzení a v rámci obmedzení samotného softwaru.

Samotné testy nám ukázali spôsob testovania určitých vlastností siete SIP. Testy nám taktiež ukázali, že zariadenie Spirent TestCenter je určené na výkonnostné testovanie zariadení, rôznych implementácií serverov alebo sieťových topológií či sieťových prvkov. Pri testovaní protokolu SIP môžeme sledovať odozvu protokolu či pomer úspešne nadviazaných relácií a neúspešných či odmietnutých a zrušených relácií. Kvôli licenčným obmedzeniam som nemohol posielat' RTP dáta, takže sledovanie metrik typu jitter či MOS nebolo možné. Po preskúmaní aplikácie je otáznе či sledovanie niektorých z týchto metrik Avalanche poskytuje.

Kvôli spomínaným licenčným obmedzeniam a nemožnosti sledovať základné metriky VoIP sme sa rozhodli do práce zakomponovať taktiež porovnanie s inými technológiami na testovanie či monitorovanie VoIP sietí.

Prvou takouto technológiou sa stal softwarový nástroj od spoločnosti SolarWinds nazvaný *VoIP & Network Quality Manager*, ktorý slúži na monitorovanie VoIP siete postavenej na zariadeniach Cisco a samotné monitorovanie prebieha na simulovanej prevádzke vytváranej medzi týmito zariadeniami pomocou SLA operácií.

Ďalšou technológiou bol prenosný počítač od spoločnosti BlueScope, *BLUELIGHT BL400A*. Podľa užívateľského manuálu tento nástroj dokáže vytvárať výkonnostné testy na vrstvách L1 – L4 modelu OSI a okrem toho poskytuje aj možnosti monitorovania siete VoIP. Monitorovanie prebieha na reálnej sieti a oproti produktu SolarWinds obsahuje aj metriku kvality hovorov R-Faktor. Výsledky monitoru získava zo správ RTCP prechádzajúcich zariadením.

Literatúra

1. ITU-T. *One-way transmission time*. G.114, Máj 2003.
2. ROSENBERG, J. et al. *SIP: Session Initiation Protocol*. RFC 3261, Jún 2002.
3. MALAS, D. a A. MORTON. *Basic Telephony SIP End-to-End Performance Metrics*. RFC 6076, Január 2011.
4. SCHULZRINNE, H. et al. *RTP: A Transport Protocol for Real-Time Applications*. RFC 3550, Júl 2003.
5. FRIEDMAN, T. R. CACERES a A. CLARK. *RTP Control Protocol Extended Reports (RTCP XR)*. RFC 3611, November 2003.
6. ITU-T. *The E-model: a computational model for use in transmission planning*. G.107, December 2011.
7. KOISTINEN, T. *Protocol overview: RTP and RTCP*. [cit. 2013-Apríl]. Dostupné z: <http://www.netlab.tkk.fi/opetus/s38130/k99/presentations/4.pdf>

Zoznam príloh

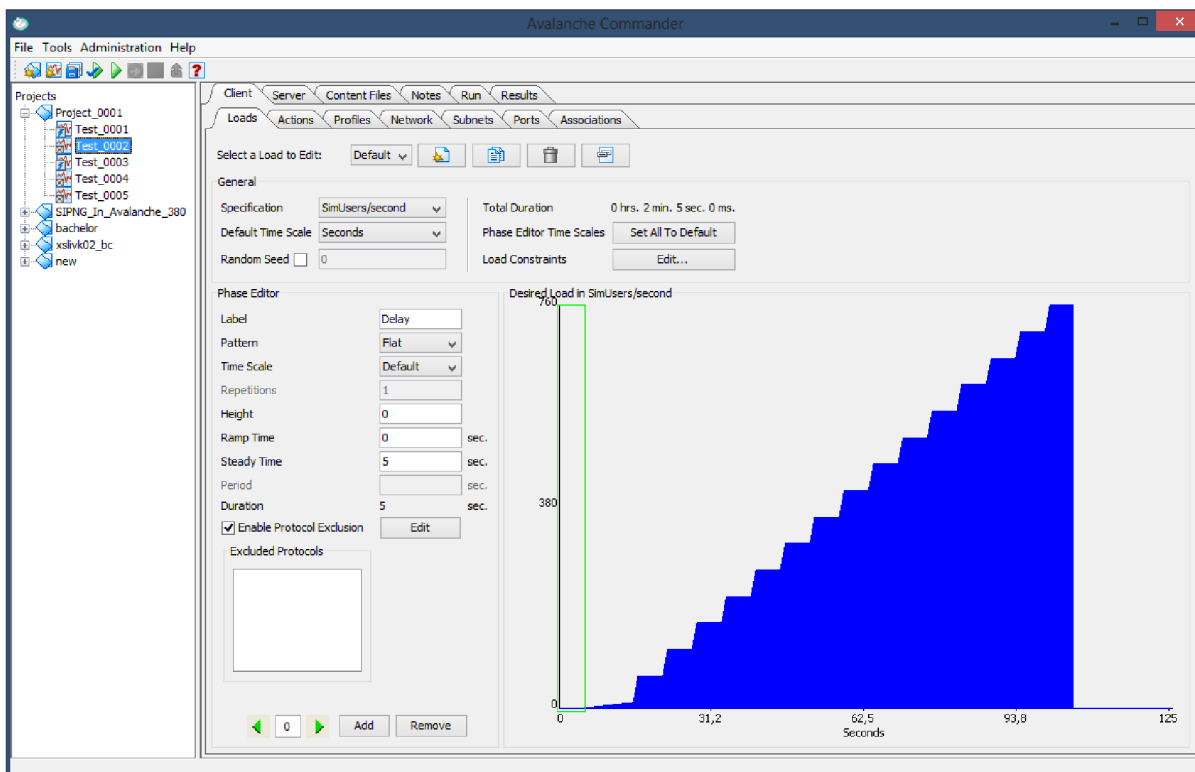
Príloha 1 Rýchly manuál k Spirent TestCenter Layer 4-7 Application (Avalanche).

Príloha 2 Konfiguračné súbory Cisco CME a Cisco Switch plus konfiguračné súbory Asterisk.

Príloha 3 CD s archívom testov a ich výsledkami pre Spirent TestCenter.

Príloha 1: Rýchly manuál k Spirent TestCenter Layer 4-7 Application (Avalanche)

Okno aplikácie Avalanche je rozdelené na dve časti. Vľavo je časť kde môžeme spravovať projekty, vpravo časť kde konfiguruje jednotlivé testy. Testy sa zoskupujú do projektov a v rámci týchto projektov zdieľajú niektoré nastavenia ako napr. zoznamy akcií.

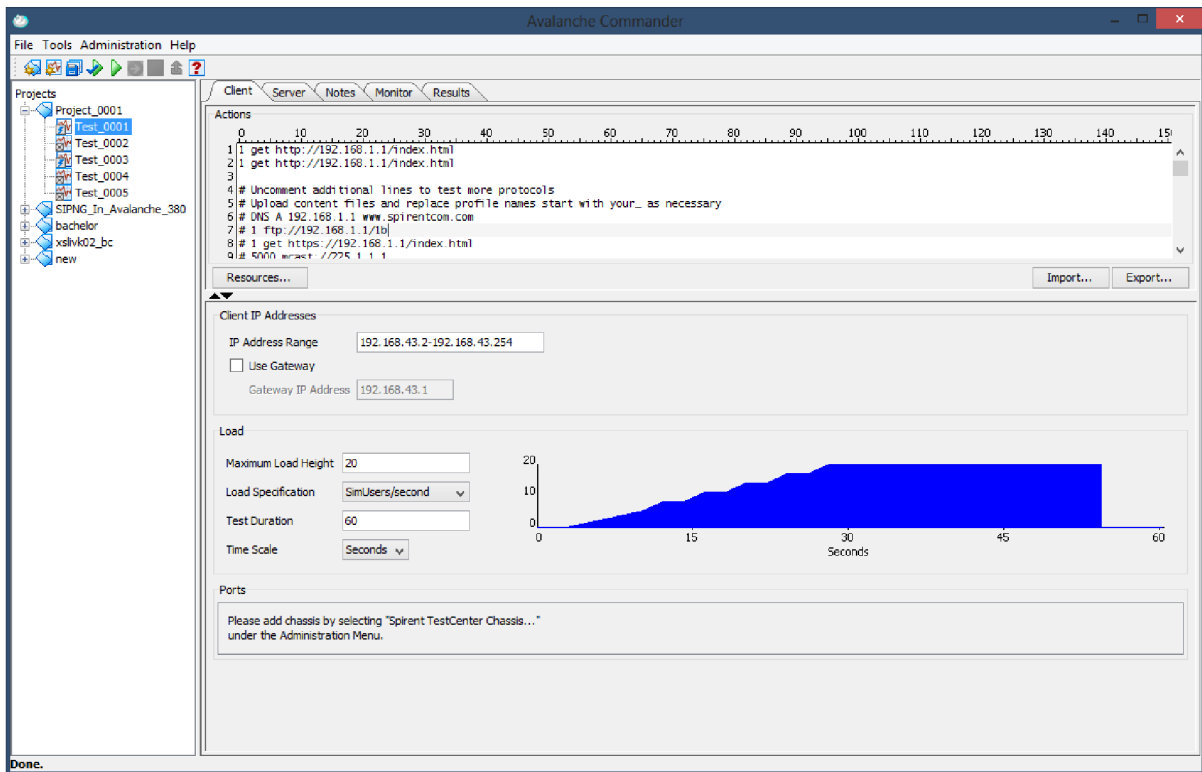


Obrázok 29: Rozhranie aplikácie Avalanche

Pri vytvorení testu máme na výber z dvoch kategórií testov a to test aplikácie a test zariadenia. V teste aplikácie bude Avalanche simulovať iba správanie klientov, no v teste zariadenia bude simulovať obe strany, ako časť klientov tak aj server.

V ďalšom kroku sa testy delia do troch typov, podľa toho, koľko voľnosti poskytujú pri ich konfigurácii. Najmenej možností nám poskytuje takzvaný EZ test. Pri prvých krokoch s prostredím odporúčam začať *Quick* testom. My budeme najčastejšie používať *Advanced* test, pretože poskytuje lepšiu manipuláciu s krivkou záťaže.

Quick test sa skladá z niekoľkých kariet. Na karte klienta vidíme zoznam akcií, ktoré každý klient má vykonať. Ďalej si na tejto karte môžeme nastaviť rozsah IP adries klientov a môžeme zľahka ovplyvniť tvar krivky záťaže. Telefónne zoznamy, ktoré využívame aj v tejto práci, vytvoríme stlačením tlačidla

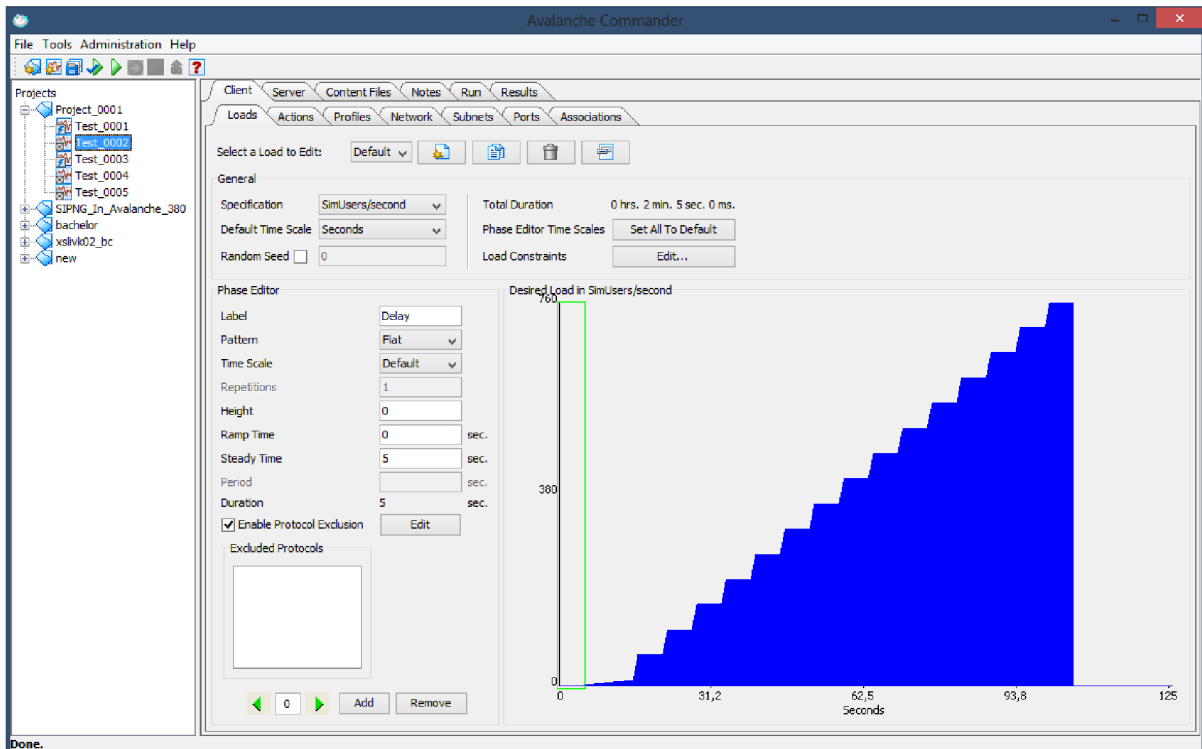


Obrázok 30: Rozhranie aplikácie Avalanche s vytvoreným rýchlym testom

Resources pod kartou Phonebook.

Ak sme si vybrali test zariadenia ponúkne sa nám aj nastavenie chovania serveru. V *Quick* teste máme možnosť nastaviť podporované protokoly a rozsah IP adries. Spustenie testu nájdeme na karte *Monitor*. Tu môžeme sledovať aj štatistiky počas testu. Po skončení testu sa nám výsledky objavia v karte *Results*.

Advanced test je test s najväčšou škálou možností konfigurácie testu a hlavnou výhodou oproti *Quick* testu je, že môžeme upravovať každú fázu krivky záťaže. Krivku záťaže vidíme hneď na prvej karte. Pohyb medzi jednotlivými krokmi je možný pomocou zelených šípok dole. Vľavo od grafu môžeme nastavovať charakteristiku jednotlivých fáz. Na karte *Subnets* môžeme nastaviť podsieť klientov. Dôležitá je karta *Associations* kde všetky nastavenia spojíme dohromady. Máme možnosť vytvoriť viac asociácií a tým pádom aj viac správaní klientov.



Obrázok 31: Advanced test

V nastavení serverov sú posledné 4 karty rovnakého významu ako v nastavení klientov. V prvých troch kartách nájdeme nastavenie typu serverov či nastavenie transakcií, čo sa týka toho, ako budú servery odpovedať.

Príloha 2 Konfiguračné súbory Cisco CME a Cisco Switch plus konfiguračné súbory Asterisk

CME1-config

```
!  
version 12.4  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname CME1  
!  
boot-start-marker  
boot-end-marker  
!  
!  
no aaa new-model  
memory-size iomem 15  
!  
!  
ip cef  
no ip dhcp use vrf connected  
!  
ip dhcp pool Voice  
    network 192.168.10.0 255.255.255.0  
    default-router 192.168.10.1  
    option 150 ip 192.168.10.1  
!  
ip dhcp pool Data  
    network 192.168.15.0 255.255.255.0  
    default-router 192.168.15.1  
!  
!  
ip auth-proxy max-nodata-conns 3  
ip admission max-nodata-conns 3  
!  
!  
voice-card 0  
    no dspfarm  
!  
!  
interface FastEthernet0/0  
    no ip address  
    duplex auto  
    speed auto  
!  
interface FastEthernet0/0.10  
    encapsulation dot1Q 10  
    ip address 192.168.10.1 255.255.255.0
```

```

!
interface FastEthernet0/0.15
 encapsulation dot1Q 15
 ip address 192.168.15.1 255.255.255.0
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0/0
 ip address 192.168.1.100 255.255.255.0
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!
ip forward-protocol nd
ip route 192.168.20.0 255.255.255.0 192.168.1.200
ip route 192.168.25.0 255.255.255.0 192.168.1.200
ip route 192.168.30.0 255.255.255.0 192.168.1.200
!
!
no ip http server
no ip http secure-server
!
!
control-plane
!
!
dial-peer voice 1 voip
 destination-pattern 2...
 session target ipv4:192.168.20.1
!
dial-peer voice 2 voip
 destination-pattern 3...
 session protocol sipv2
 session target ipv4:192.168.30.2
 codec g711alaw
!
!
telephony-service
 max-ephones 10
 max-dn 10
 ip source-address 192.168.10.1 port 2000
 auto assign 1 to 10
 system message Hello2
 create cnf-files version-stamp Jan 01 2002 00:00:00
 keepalive 10

```

```
max-conferences 8 gain -6
!
!
ephone-dn 1
  number 1001
  name user2
!
!
ephone 1
  mac-address 0030.94C3.6464
  type 7960
  button 1:1
!
!
line con 0
line aux 0
line vty 0 4
  login
!
scheduler allocate 20000 1000
!
end
```

CME2-config

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CME2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 10
!
!
ip cef
no ip dhcp use vrf connected
!
ip dhcp pool Voice
  network 192.168.20.0 255.255.255.0
  default-router 192.168.20.1
  option 150 ip 192.168.20.1
!
ip dhcp pool Data
  network 192.168.25.0 255.255.255.0
```

```

default-router 192.168.25.1
!
ip dhcp pool Asterisk
  network 192.168.30.0 255.255.255.0
  default-router 192.168.30.1
!
!
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
!
voice-card 0
  no dspfarm
!
!
interface FastEthernet0/0
  no ip address
  duplex auto
  speed auto
!
interface FastEthernet0/0.20
  encapsulation dot1Q 20
  ip address 192.168.20.1 255.255.255.0
!
interface FastEthernet0/0.25
  encapsulation dot1Q 25
  ip address 192.168.25.1 255.255.255.0
!
interface FastEthernet0/1
  no ip address
  duplex auto
  speed auto
!
interface FastEthernet0/1.30
  encapsulation dot1Q 30
  ip address 192.168.30.1 255.255.255.0
!
interface Serial0/0/0
  ip address 192.168.1.200 255.255.255.0
  clock rate 64000
!
interface Serial0/0/1
  no ip address
  shutdown
  clock rate 2000000
!
ip forward-protocol nd
ip route 192.168.10.0 255.255.255.0 192.168.1.100
ip route 192.168.15.0 255.255.255.0 192.168.1.100
!
!

```

```

no ip http server
no ip http secure-server
!
!
control-plane
!
!
dial-peer voice 1 voip
destination-pattern 1...
session target ipv4:192.168.10.1
!
dial-peer voice 2 voip
destination-pattern 3...
session protocol sipv2
session target ipv4:192.168.30.2
codec g711alaw
!
!
telephony-service
max-ephones 10
max-dn 10
ip source-address 192.168.20.1 port 2000
auto assign 1 to 10
system message Hello
create cnf-files version-stamp Jan 01 2002 00:00:00
keepalive 10
max-conferences 8 gain -6
!
!
ephone-dn 1
number 2001
name user1
!
!
ephone 1
mac-address 0013.806C.2F0C
type 7960
button 1:1
!
!
line con 0
line aux 0
line vty 0 4
login
!
scheduler allocate 20000 1000
!
end

```

Switch-config

```
!  
! Last configuration change at 00:33:10 UTC Mon Mar 1 1993  
!  
version 12.2  
no service pad  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname Switch  
!  
boot-start-marker  
boot-end-marker  
!  
!  
no aaa new-model  
system mtu routing 1500  
!  
!  
spanning-tree mode pvst  
spanning-tree extend system-id  
!  
vlan internal allocation policy ascending  
!  
!  
interface FastEthernet0/1  
  switchport trunk encapsulation dot1q  
  switchport trunk allowed vlan 30  
  switchport mode trunk  
!  
interface FastEthernet0/2  
  switchport trunk encapsulation dot1q  
  switchport trunk allowed vlan 10,15  
  switchport mode trunk  
!  
interface FastEthernet0/3  
  switchport access vlan 30  
  switchport mode access  
!  
interface FastEthernet0/4  
  switchport access vlan 10  
  switchport mode access  
!  
interface FastEthernet0/5  
  switchport access vlan 30  
  switchport mode access  
!  
interface FastEthernet0/6  
  switchport access vlan 15
```



```
switchport mode access
switchport voice vlan 10
spanning-tree portfast
!
interface FastEthernet0/7
switchport access vlan 30
switchport mode access
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
interface FastEthernet0/12
!
interface FastEthernet0/13
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 20,25
switchport mode trunk
!
interface FastEthernet0/14
!
interface FastEthernet0/15
switchport access vlan 20
switchport mode access
!
interface FastEthernet0/16
!
interface FastEthernet0/17
switchport access vlan 25
switchport mode access
switchport voice vlan 20
spanning-tree portfast
!
interface FastEthernet0/18
!
interface FastEthernet0/19
!
interface FastEthernet0/20
!
interface FastEthernet0/21
!
interface FastEthernet0/22
!
interface FastEthernet0/23
!
interface FastEthernet0/24
!
```

```
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
no ip address
shutdown
!
interface Vlan30
ip address 192.168.30.100 255.255.255.0
!
ip http server
ip http secure-server
!
!
logging esm config
!
!
line con 0
line vty 5 15
!
end
```

Asterisk- SIP.conf

```
[general]
domain=test.dom
context=incoming
```

```
[user0]
username=user0
type=friend
context=users
host=dynamic
secret=heslo
```

```
[user1]
username=user1
type=friend
context=users
host=dynamic
secret=heslo
```

Asterisk- extensions.conf

```
[global]
```

```
[general]
```

```
[users]
exten=>3001,1,Dial(SIP/user0,30)
exten=>3001,n,Hangup()
exten=>3002,1,Dial(SIP/user1,30)
exten=>3002,n,Hangup()
exten=>_1xxx,1,Dial(SIP/192.168.10.1/${EXTEN},30)
exten=>_1xxx,n,Hangup()
exten=>_2xxx,1,Dial(SIP/192.168.20.1/${EXTEN},30)
exten=>_2xxx,n,Hangup()
```

```
[incoming]
include=>users
```

Príloha 3 CD s archívom testov a ich výsledkami pre Spirent TestCenter

CD

```
|—STC.results
|   STC-SIP.spf  -archív testov a ich výsledkov vytvorené v STC Avalanche v4.10
|
|—configuration.files
|   |—STC      -konfiguračné súbory použité pri výkonnostnom testovaní SIP
|   |   |—Asterisk -5000 užívateľských účtov a 5000 čísel pre volací plán
|   |       5000.users.for.Asterisk-SIP.txt
|   |       5000.extensions.for.Asterisk-dialplan.txt
|   |
|   |—bl400a-solarwinds -súbory týkajúce sa testov monitorovania kvality hovorov
|   |   |—cisco      -konfiguračné súbory Cisco zariadení
|   |       cme1-config
|   |       cme2-config
|   |       switch-config
|   |
|   |—Asterisk -konfiguračné súbory obsahujúce prepojenie s ústredňami CME
|   |   extensions.conf
|   |   sip.conf
|
|—BP      -elektronická podoba BP
|   BP-SIP_STC.pdf
|   BP-SIP_STC.docx
```