VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

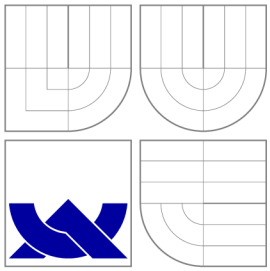
PROSTŘEDÍ PRO MONITOROVÁNÍ A SPRÁVU VOIP S VYUŽITÍM TECHNOLOGIE ONEPK

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

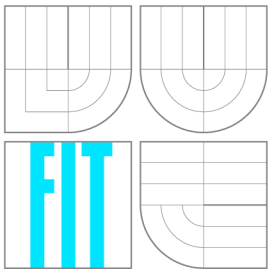
AUTOR PRÁCE
AUTHOR

Bc. DÁVID ANTOLÍK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PROSTŘEDÍ PRO MONITOROVÁNÍ A SPRÁVU VOIP S VYUŽITÍM TECHNOLOGIE ONEPK

VOIP TRAFFIC MONITORING AND MANAGEMENT IN ONEPK ENABLED NETWORKS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. DÁVID ANTOLÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ONDŘEJ RYŠAVÝ, Ph.D.

BRNO 2015

Abstrakt

Cílem této diplomové práce je obeznámení s principy softwarově definovaných sítí na platformě Cisco One Platform Kit a postupy monitorování v těchto sítích se zaměřením na monitorování kvality Voice over IP komunikace. Součástí této práce je návrh a implementace rozšiřitelného monitorovacího prostředí OneMon na platformě Cisco One Platform Kit prostřednictvím analyzátorů pro monitorování různých typů síťového provozu. V diplomové práci byl implementován analyzátor VoIP provozu pracující nad protokoly SIP a RTP. Tento analyzátor poskytuje informace o telefonních spojeních v monitorované síti a jejich kvalitě.

Abstract

The main goal of this master's thesis is to apprise of principles of Cisco One Platform Kit based on software defined networks and with monitoring techniques in that type of networks. The focus is concentrated on monitoring the quality of Voice over IP communication. Next part of this thesis is a proposal and implementation of the extensible monitoring environment OneMon on the Cisco One Platform Kit. It is possible to extend OneMon environment using specific analyzers to monitor various types of network traffic. The part of this master's thesis is also implementation of VoIP traffic analyzer for SIP and RTP protocols. This analyzer provides information about phone calls and their quality in a monitored segment of a computer network.

Klíčová slova

softwarově definované sítě, cisco onepk, voip, monitorování kvality

Keywords

software defined networks, cisco onepk, voip, quality measurement

Citace

Dávid Antolík: Prostředí pro monitorování a správu VoIP s využitím technologie OnePK, diplomová práce, Brno, FIT VUT v Brně, 2015

Prostředí pro monitorování a správu VoIP s využitím technologie OnePK

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Ondřeja Ryšavého, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Dávid Antolík
25. května 2015

Poděkování

Ďakujem vedúcemu mojej diplomovej práce pánovi Ing. Ondřejovi Ryšavému, Ph.D., za jeho čas, hodnotné pripomienky a podnety k jej riešeniu.

Obsah

Kapitola 1

Úvod

Cieľom mojej diplomovej práce je vytvorenie monitorovacieho prostredia s využitím architektúry One Platform Kit pre softvérovo definované siete od spoločnosti Cisco. Softvérovo definované siete, o ktorých pojednáva kapitola 2, zažívajú v poslednej dobe veľký rozmach. Je to spôsobené najmä rozvojom nových technológií, protokolov a služieb, pre ktoré už tradičné sieťové architektúry nie sú dostatočné flexibilné. Prínos softvérovo definovaných sietí je predovšetkým v možnostiach algoritmizácie riadenia sietí a sieťovej prevádzky, čo odbúrava mnohé problémy konvenčných sieťových architektúr.

S rozvojom počítačových sietí súvisia aj zvýšené nároky na zaistenie spoľahlivosti a bezpečnosti týchto sietí a jej užívateľov. K tomu je potrebné sledovanie prevádzky na sieti a jej analýza [4]. Tú je možné vykonávať priamo za behu, teda real-time, alebo dáta zbierať a ukladať pre neskoršie spracovanie. V tejto diplomovej práci bude predstavený návrh aj implementácia monitorovacieho prostredia OneMon nad technológiou Cisco OnePK. Technológii Cisco OnePK sa bližšie venuje kapitola 3. Prostredie OneMon poskytuje možnosti a prostriedky pre real-time analýzu. Toto prostredie uplatňuje princípy modulárnosti a berie do úvahy možnosti rozširovania, ktoré reflektujú neustále pribúdajúce protokoly a služby v počítačových sieťach.

Pri vývoji monitorovacej platformy v tejto diplomovej práci, ktorej popis je obsahom kapitoly 5, bolo vychádzané z požiadaviek na monitorovanie Voice over IP prevádzky (kapitola 4), teda k monitorovaniu prenosu hlasových telefónnych hovorov cez počítačovú sieť. Pri VoIP má význam predovšetkým real-time monitorovanie. Analyzátor vyvinutý pre monitorovanie VoIP prevádzky v kapitole 6 tejto diplomovej práce sleduje správy protokolov SIP a RTP, ktoré sú používané k prenosu telefónnych hovorov cez počítačovú sieť. Tento analyzátor umožňuje sledovanie telefónnych hovorov ustanovených signalizačným protokolom SIP. Zo správ signalizačného protokolu SIP analyzátor získa potrebné informácie pre sledovanie multimediálnej komunikácie telefónnych hovorov, ktorá je realizovaná protokolom RTP. Protokol RTP prenáša hlasové dáta medzi komunikujúcimi účastníkmi. Z pohľadu zaistenia kvality telefónnych služieb v počítačových sieťach je preto dôležité práve monitorovanie RTP paketov. Problémy pri prenose RTP paketov cez počítačovú sieť majú za následok ich stratovosť, veľké oneskorenie pri doručení alebo kolísanie oneskorenia pri doručení paketov (nazývaného jitter), ktoré môžu byť spôsobené napríklad asymetrickým smerovaním. Všetky tieto faktory je potrebné neustále monitorovať a vyhodnocovať. Sledovaním kvality VoIP prevádzky sa dá predísť problémom pri komunikácii a nespokojnosti užívateľov s IP telefóniou.

Monitorovacia platforma OneMon poskytuje okrem monitorovania a analýzy tiež prostriedky pre zásah do konfigurácie siete prostredníctvom funkcií technológie Cisco OnePK.

Na základe výsledkov monitorovania a analýzy môže správca algoritmicky definovať reakcie na vzniknuté situácie. Môže to byť napríklad reakcia na zvýšenú stratovosť RTP paketov a tým zhoršenie kvality VoIP telefónnych hovorov. Definovanie automatických akcií, napríklad zmena v nastavení QoS[1] alebo smerovania, môže pomôcť zachovať funkčnosť týchto služieb pri výskyte takýchto situácií a minimalizovať riziko výpadku alebo nedostupnosti služby.

Posledné dve kapitoly 7 a 8 sú venované testovaniu platformy OneMon, jej spoľahlivosti, zhodnoteniu vytvoreného monitorovacieho prostredia a prezentácii nápadov a podnetov, ktoré môžu prispieť k budúcemu zdokonaleniu platformy OneMon. Je to hlavne grafická nadstavba WebAPI, ktorá zjednoduší konfiguráciu monitorovania a prístupu k jeho výsledkom.

Táto diplomová práca nadväzuje na môj semestrálny projekt, z ktorého vychádzajú kapitoly o princípoch VoIP komunikácie a hodnotení jej kvality. Na základe semestrálneho projektu som tiež postupoval pri tvorbe architektúry monitorovacej platformy OneMon. Táto architektúra bola prispôsobená novým požiadavkam a možnostiam, ktoré som objavil pri práci s technológiou Cisco OnePK.

[1] Quality of Service (QoS) zahŕňa služby pre rezerváciu prenosovej kapacity v počítačových sieťach, využíva sa predovšetkým pri dátach, ktoré vyžadujú bezodkladné doručenie a prioritizáciu pred ostatnými dátami, napríklad pri VoIP internetovej telefónii.

Kapitola 2

Softvérovo definované siete

Softvérovo definované siete (SDN) je pojem označujúci novú architektúru počítačových sietí. Tá umožňuje administrátorom siete lepšiu a prehľadnejšiu správu sieťových služieb a dátových tokov. Architektúra SDN zariadení v paketových sieťach oddeľuje data-plain časť, teda vysokorýchlostný subsystém pre prepínanie paketov od control-plain časti. Control-plain predstavuje logiku sieťových elementov. V konvenčných počítačových sieťach je control-plain časť zabudovaná priamo do sieťového zariadenia. Administrátorovi ponúka iba limitovanú množinu možností konfigurácie. O interpretáciu konfiguračných príkazov sa pritom stará control-plain. Administrátor nemôže v plnom rozsahu zasahovať do postupu riadenia, ktorým sieťový prvok dosahuje zamýšľané chovanie počítačovej siete.

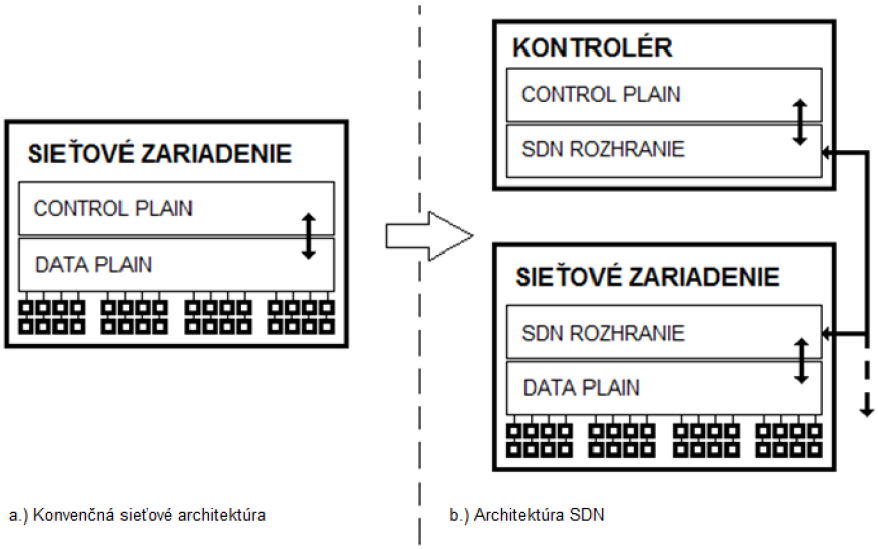
2.1 Koncept SDN

V softvérovo definovaných sieťach je control-plain úplne oddelený od sieťového prvku a nezávisle umiestnený v rámci sieťovej topológie ako nový prvok. Často sa tento nový komponent označuje pojmom kontrolér (obrázok 2.1). Najčastejšie je to server, na ktorom v podobe počítačového programu beží softvérový control-plain. Ten je spojený pomocou vhodného a k tomu určeného komunikačného protokolu so sieťovým prvkom. Sieťový prvok je v tomto prípade možné chápať len ako hardware bez rozhodujúcej logiky (alebo len so základnými funkciami), ktorý nie je schopný plnej funkcionality. Celé chovanie siete riadi kontrolér, ktorý konfiguruje sieťové prvky v súlade so zamýšlanou sieťovou konfiguráciou. Táto sieťová konfigurácia je implementovaná algoritmicky v podobe kontroléra, ktorý spravuje jednotlivé sieťové prvky v súlade s týmto algoritmom.

Sieťové prvky v koncepte SDN sietí často pracujú s jednotkami, označovanými ako sieťový tok. Sieťový tok je možné definovať ako množinu po sebe idúcich paketov v určitom časovom intervale, ktoré majú zhodné jeden alebo viacero spoločných parametrov. Tými môžu byť:

- Zdrojová a cieľová IP adresa
- Zdrojový a cieľový port
- Protokolové číslo v rámci IP paketu
- Protokol aplikačnej vrstvy

Tok môže byť určený tiež zdrojovou a cieľovou MAC adresou a ďalšími špecifickými parametrami, ktoré umožňujú kategorizovať pakety do skupín so spoločnými vlastnosťami.



Obrázek 2.1: Ilustrácia oddelenia data-plain časti a control-plain časti v softvérovo definovaných sieťach.

2.2 Architektúra a prvky SDN

Ako už bolo uvedené, architektúra SDN sietí sa odlišuje od klasického konceptu počítačových sietí oddelením control-plain časti zo sieťových zariadení a jeho centralizáciou. K tomu do siete pribudnú nové prvky, predovšetkým kontrolér, ktorý slúži ako hostiteľ pre SDN aplikácie. Súčasťou SDN architektúr je linka umožňujúca prenos dátovej prevádzky medzi sieťovými prvkami a kontrolérom, nazývaná DataPath.

Kontrolér

Kontrolér je logicky centralizovaný prvok v SDN architektúre, ktorého primárnou funkciou je umožniť beh SDN aplikáciám a zabezpečiť rozhranie so sieťovými prvkami v SDN topológii. Obvykle je to server s dostatočnou výpočetnou kapacitou pre beh riadiacich algoritmov. SDN kontrolér poskytuje sieti logiku, a preto musí byť opatrený voči výpadku zavedením vhodnej redundancie.

Aplikácia pre SDN

Aplikácia v softvérovo definovanej sieti je program, ktorý implementuje logiku nad počítačovou sieťou, teda jej správanie a reakcie na vzniknuté situácie.

Pracuje na základe algoritmu, ktorý pri riadení siete realizuje požiadavky administrátora. Pritom je schopná reagovať na aktuálny stav siete.

SDN DataPath

Je logické sieťové zariadenie, ktoré predstavuje kanál pre prenos dátových paketov v SDN sieti medzi sieťovými prvkami a kontrolérom. Je to dedikovaná spoľahlivá linka, ktorá umožňuje obojsmerný prenos dát. Implementačné detaily a spôsob použitia sú závislé na konkrétnej platforme.

2.3 Najznámejšie SDN architektúry

Softvérovo definované siete sú obecný pojem, ktorý zahŕňa viacero implementačne a architektonicky odlišných protokolov. Tieto protokoly môžu byť korporátne a zamerané na množinu zariadení jedného výrobcu, alebo obecne dostupné a otvorené pre všetkých výrobcov sieťových prvkov a administrátorov počítačových sietí. Aj napriek odlišnostiam medzi nimi zostávajú zachované hlavné princípy softvérovo definovaných sietí. Medzi najznámejšie architektúry SDN patria OpenFlow a technológia Cisco One Platform Kit.

2.3.1 OpenFlow

OpenFlow je otvorený štandard, na začiatku bol určený prevažne výskumníkom. Umožňuje vytvárať a testovať experimentálne protokoly a nové princípy v počítačových sieťach. OpenFlow sa už ale vyskytuje aj v komerčných produktoch, predovšetkým prepínačoch (vyrábaných napríklad spoločnosťou Hewlett Packard) a osvojuje si ho stále viac výrobcov sieťových zariadení.

OpenFlow naplno uplatňuje architektúru softvérovo definovaných sietí, teda rozdelenie data-plain a control-plain častí. Control-plain je umiestnený na server a nazýva sa kontrolér. Kontrolér so sieťovým prvkom komunikuje pomocou protokolu OpenFlow.

Princíp OpenFlow sietí je postavený na tabuľke nazývanej Flow-Table. Pre každý prepínač v OpenFlow sieťovej topológii existuje samostatná tabuľka Flow-Table. Flow-Table obsahuje údaje, ktoré sú potrebné pre správne fungovanie data-plain časti prepínača. Je zložená zo záznamov, pričom každý záznam obsahuje minimálne:

Popis sieťového toku
: Štruktúra popisujúca spoločné vlastnosti paketov, patriacich do tohto sieťového toku.

Akciu
: Akciu, ktorá sa aplikuje na paket, patriaci do tohto sieťového toku.

Platnosť
: Časový údaj platnosti záznamu.

Tabuľka Flow-Table principiálne nahrádza CAM-tabuľku, používanú na bežných prepínačoch, pracujúcich nad linkovou vrstvou. Pri príchode paketu na rozhranie prepínača sa ten pokúsi najskôr paket klasifikovať a nájsť zodpovedajúci záznam v tabuľke Flow-Table. Ak takýto záznam nájde, uplatní na tento paket akciu, definovanú týmto záznamom. Akciou môže byť napríklad odoslanie paketu na stanovené sieťové rozhranie, zachytenie paketu a jeho odoslanie prostredníctvom Datapath na kontrolér alebo zahodenie paketu. V prípade, že paket nezodpovedá žiadnemu sieťovému toku vo Flow-Table, posiela prepínač paket prostredníctvom Datapath na analýzu kontroléru. Kontrolér obdrží celý obsah paketu a analyzuje ho. Potom algoritmus bežiaci na kontroléri rozhodne, akú akciu je s paketom potrebné vykonať. Väčšinou sa jedná o prvý paket toku, za ktorým budú nasledovať ďalšie pakety. Kontrolér preto môže nastaviť nové pravidlo do Flow-Table príslušného sieťového zariadenia s definovanou akciou. Toto pravidlo bude platné pre ďalšie pakety tohto sieťového toku.

Úlohou kontroléra je napĺňať a udržiavať tabuľky Flow-Table v prepínačoch, ktoré sú súčasťou ním spravovanej sieťovej topológie v súlade so zamýšľanou sieťovou konfiguráciou. K tomu si kontrolér udržiava vlastný vnútorny model týchto tabuliek pre každý prepínač v jeho topológii.

Pre správu Flow-Table v prepínačoch používa kontrolér protokol OpenFlow. Tento protokol definuje niekoľko typov správ. Najdôležitejšie z nich sú:

Packet received
: Správa od sieťového prvku, ktorá informuje kontrolér o novom prijatom pakete. Požaduje od kontroléra odpoveď, čo ma s týmto paketom urobiť. Súčasťou správy je aj payload prijatého paketu.

Send packet out
: Správa od kontroléra pre sieťový prvok. Može byť reakciou na správu "Packet received", kedy sieťovému prvku hovorí, na ktoré sieťové rozhranie má paket ďalej poslať, prípadne mu prikáže paket zahodiť. Pomocou tejto správy môže byť taktiež do siete vložený nový paket a odoslaný cez určené rozhranie.

Modify flow table
: Správa pre sieťový prvok, ktorej súčasťou je špecifikácia sieťového toku a akcia, ktorá sa má aplikovať na všetky pakety patriace do tohto sieťového toku. Takýto záznam sa pridá do tabuľky Flow-Table sieťového prvku. Po dobu platnosti záznamu v tabuľke bude sieťový prvok vykonávať nad paketmi tohto sieťového toku nastavenú akciu.

Get statistics
: Vyžiadanie štatistických údajov od sieťového prvku, ako napríklad počet prenesených paketov, bajtov, zaplnenosť tabuľky Flow-table apod.

2.3.2 Cisco One Platform Kit

Cisco One Platform Kit (skrátene Cisco OnePK) je rovnako ako OpenFlow softvérovo definovaná architektúra počítačových sietí, ktorá zahŕňa framework pre vývoj SDN kontroléra a abstraktnú vrstvu, ktorá je súčasťou sieťových zariadení Cisco s podporou OnePK. Aj keď Cisco OnePK tiež uplatňuje myšlienku sietí SDN, od architektúry OpenFlow sa výrazne odlišuje. Cisco OnePK by bolo možné nazvať aj konfiguračnou nádstavbou nad sieťovými zariadeniami Cisco, než ucelenou SDN architektúrou. OnePK poskytuje prostriedky, ktorými je možné zasahovať do konfigurácie sieťových zariadení Cisco z bežiacej aplikácie - kontroléra. Tým je možné dynamicky meniť bežiacu konfiguráciu na týchto sieťových zariadeniach.

Veľmi prínosnou sa javí možnosť využívať technológiu Datapath, ktorá pracuje s podobnými princípmi ako v architektúre OpenFlow. Umožňuje zachytávať dáta prechádzajúce sieťovými zariadeniami a odovzdávať ich na spracovanie kontroléru. Tiež umožňuje vkladanie nových dátových paketov priamo do sieťovej prevádzky. Na technológii Datapath je postavená významná časť tejto diplomovej práce, ktorá sa zaoberá tvorbou rozšíriteľného monitorovacieho prostredia. Toto monitorovacie prostredie bude využívať Datapath k zberu dát (zaujímavých z pohľadu monitorovania) a ich doručovaniu do kontroléra k real-time analýze.

Architektúre Cisco One Platform Kit je venovaná celá nasledujúca kapitola 3.

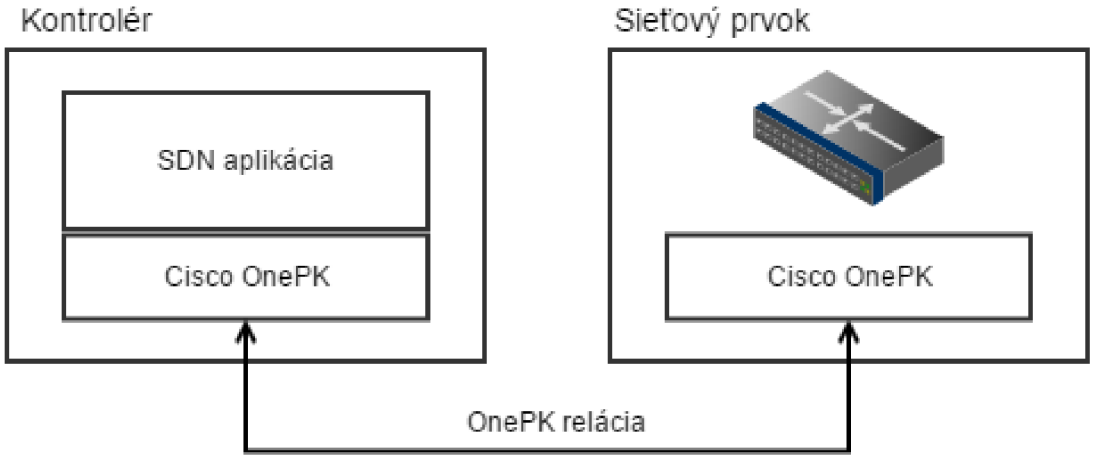
Kapitola 3

Cisco One Platform Kit

Cisco OnePK (Cisco One Platform Kit) je funkcionalita operačného systému Cisco IOS. Zároveň toto označenie nesie framework - súbor knižníc umožňujúci vývoj aplikácií, ktoré úzko spolupracujú s podporovanými Cisco zariadeniami. Pre vývoj je k dispozícii API rozhranie v troch programovacích jazykoch - C, Java, Python a REST API. Pre vývoj monitorovacieho prostredia, ktoré je predmetom tejto diplomovej práce bude použité API v jazyku C, pretože pre Datapath Service Set (kapitola 3.2.1) je kvôli požadovanej rýchlosti spracovania podporovaný iba vývoj v jazyku C.

3.1 Základné pojmy

Základná schéma architektúry Cisco OnePK je znázornená na obrázku 3.1. Nachádza sa v nej kontrolér, na ktorom beží aplikácia naprogramovaná s využitím Cisco OnePK API. Cez OnePK rozhranie vytvorí aplikácia reláciu so sieťovými prvkami.



Obrázek 3.1: Schéma architektúry Cisco OnePK.

3.1.1 Aplikácia

Aplikácia pre Cisco OnePK je spustiteľný program, napísaný s využitím API rozhrania Cisco OnePK v niektorom z troch dostupných programovacích jazykov, prípadne s využitím REST API. Aplikácie pre platformu OnePK môže vyvíjať ktokoľvek so znalosťami API, ku ktorému je dostupná dokumentácia [1] na stránkach spoločnosti Cisco. Aplikácia zvyčajne beží na samostatnom serveri, avšak môže byť aj priamo súčasťou sieťového prvku, ako je popísané v kapitole 3.1.4.

3.1.2 Sieťový prvok

Zariadenie, ktoré podporuje platformu Cisco OnePK. Vo všeobecnosti sa jedná o Cisco smerovače s operačným systémom Cisco IOS a platformy Cisco IOS-XE, Cisco IOS-XR a Cisco NX-OS Software.

3.1.3 Relácia

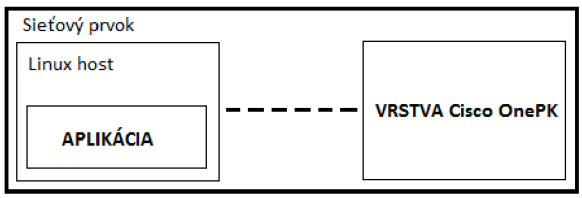
Medzi OnePK aplikáciou a sieťovým prvkom, s ktorým táto aplikácia kooperuje, vznikne relácia (v ang. literatúre označovaná session). Relácia je vytvorená aplikáciou po jej spustení, keď sa aplikácia pripojí k sieťovému prvku. Súčasťou vytvárania relácie je autentizácia aplikácie pomocou užívateľského mena a hesla alebo certifikátu.

3.1.4 Modely OnePK

Cisco OnePK je možné s istým nadhľadom označiť ako architektúru client-server. Klientom sa v tomto prípade rozumie sieťový prvok, serverom je aplikácia, ktorá sieťový prvok spravuje. OnePK podporuje 3 modely - process hosting, end-node hosting a blade hosting.

3.1.5 Model process hosting

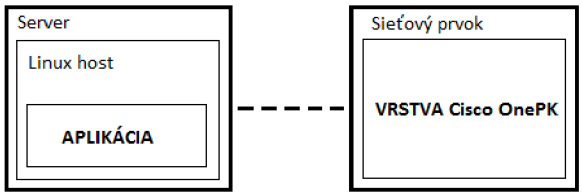
V tomto modeli (znázornenom na obrázku 3.2) siete beží aplikácia v linuxovom operačnom systéme, ktorý beží priamo v sieťovom prvku. Operačný systém je izolovaný od systému, ktorý riadi sieťový prvok. Väčšinou sa jedná o virtualizovaný operačný systém pomocou KVM/QEMU.



Obrázek 3.2: Grafické znázornenie modelu process hosting.

3.1.6 Modely blade hosting a end-node hosting

Vzniknú vyčlenením linuxového prostredia zo sieťového prvku a presunutím na samostatný server. Server pritom môže byť umiestnený celkom nezávisle vzhľadom k sieťovému prvku. Tento model sa nazýva end-node hosting, zobrazený na obrázku 3.3. Model blade hosting sa odlišuje od end-node hosting modelu iba tým, že server, ktorý hostí aplikáciu, je vonkajšou súčasťou sieťového prvku. Znamená to, že môžu zdielať fyzickú krabicu ako jedno zariadenie.



Obrázek 3.3: Grafické znázornenie modelu end-node hosting.

3.2 Service sety

Service set je súhrn elementárnych funkcií a procesov, ktoré na seba navzájom naväzujú a ako celok plnia špecifickú komplexnú funkciu. Service sety predstavujú základ pre fungovanie OnePK aplikácií. Samotné prostredie Cisco OnePK ponúka niekoľko základných service setov:

3.2.1 DataPath service set (DPSS)

DataPath je špecifická funkcionalita softwarovo definovaných sietí. Je to prostriedok, pomocou ktorého sa vo všeobecnosti zachytávajú a prenášajú dátové pakety zo sieťovej komunikácie prechádzajúcej sieťovým prvkom k centrálnemu riadiacemu arbitrovi siete (ktorý v našom prípade označujeme kontrolér). V prostredí OnePK je možné používať tento service set v troch režimoch:

Divert V tomto režime je paket odklonený do aplikácie. To znamená, že nepokračuje ďalej v ceste, pokiaľ ho aplikácia nespracuje a nepošle pomocou DataPath späť na sieťový prvok. Aplikácia môže paket rozbaliť na príslušnú vrstvu a modifikovať (zmena QoS, smerovania, údajov na aplikačnej vrstve apod.). Následne aplikácia paket pošle späť sieťovému prvku, ktorý ho zaradí na miesto pôvodného paketu. Aplikácia môže taktiež nariadiť zahodenie paketu.

Copy Skopíruje paket a odošle ho prostredníctvom DataPath do aplikácie. Na rozdiel od režimu Divert paket pokračuje obvyklým spôsobom ďalej v ceste, aplikácia dostane iba jeho presnú kópiu.

Inject Je špeciálny režim, ktorý umožňuje aplikácii vygenerovať vlastný paket a prostredníctvom DataPath ho distribuovať na sieťový prvok. Ten s takýmto paketom naloží tak, akoby sa jednalo o akýkoľvek iný paket v sieťovej komunikácii.

DataPath je zovšeobecním pre linku, ktorou sa pakety doručujú zo sieťového prvku do aplikácie. V prostredí Cisco OnePK je k tomuto účelu pri zapnutí aplikácie vytvorený medzi sieťovým prvkom a aplikáciou GRE tunel [21]. GRE tunnel existuje po celú dobu behu aplikácie, ktorá pracuje s DPSS.

3.2.2 Policy service set

Predstavujú rozhranie pre konfiguráciu Access control listov, ACEs, QoS politík. Prostredníctvom nich sa taktiež konfiguruje záujmový traffic, ktorý spracováva DataPath Service set.

3.2.3 Routing service set

Umožňuje aplikácii pristupovať ku konfigurácii smerovania (Routing information base - RIB). Podporuje vkladanie, modifikáciu a mazanie smerovacích informácií.

3.2.4 Element service set

Predstavuje systémový model zariadení, spravuje prihlasovanie a organizuje zariadenia v OnePk topológii.

3.2.5 Discovery service set

Umožňuje prístup k topológii OnePK siete, teda mapuje OnePK zariadenia pripojené k aplikácii.

3.2.6 Configuration management service set

Všetky konfiguračné aktivity, ktoré vývojár prostredníctvom OnePK vytvorí, je nutné správne zahrnúť do aktuálnej konfigurácie routra (running-config), spravuje NVRAM a je schopný po dokončení behu aplikácie znova vrátiť konfiguráciu routra do pôvodného stavu aký bol pred spustením aplikácie (transakcia rollback).

3.2.7 Event service set

Spracováva výnimočné aj neočakávané udalosti, ku ktorým dôjde počas behu OnePK aplikácie. Spracováva tieto udalosti a vytvára logovacie záznamy so špecifikáciou stavu, pri ktorom došlo k výnimke pre vývojárov, aby bolo možné výnimku korektne ošetriť.

3.2.8 Developer service set

Tento service set poskytuje vývojárom prístup k logom aplikácie a ladiacim výpisom, slúži k ľahšiemu debugovaniu pri vývoji OnePK aplikácií.

3.3 Network-Based Application Recognition - NBAR

Cisco Network-Based Application Recognition je funkcionalita dostupná na sieťových prvkoch Cisco s operačným systémom IOS. Jej hlavným cieľom je poskytnúť efektívnu možnosť klasifikácie sieťových tokov na základe aplikačného protokolu, ktorému sieťový tok prináleží.

NBAR podporuje širokú škálu aplikačných protokolov architektúry klient-server, ktoré dynamicky a nezávisle komunikujú prostredníctvom služieb TCP alebo UDP na ktoromkoľvek vhodnom porte. Potom, čo je aplikácia rozpoznaná, môže s tokom sieťový prvok začať odlišne zaobchádzať, napríklad modifikovať jeho nastavenia QoS alebo vykonávať monitoring tohto sieťového toku prostredníctvom platformy OnePK.

Funkcia NBAR na zariadeniach Cisco disponuje štandardnou sadou približne sto najpoužívanejších sieťových protokolov. Podporuje zároveň aj užívateľsky definované špeciálne protokolové sady.

Funkciu NBAR je nutné pred použitím na smerovačoch Cisco aktivovať. Pre použitie s monitorovacou platformou OneMon, ktorej návrh a implementácia budú predstavené v kapitole 5, postačuje aktivovať funkciu NBAR na všetkých sieťových rozhraniach, z ktorých bude komunikácia zachytávaná.

Postup aktivácie NBAR na sieťovom rozhraní *GigabitEthernet 0/0* smerovača je uvedený na nasledujúcom príklade:

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet 0/0
Router(config-if)# ip nbar protocol-discovery
```

Kapitola 4

Princípy Voice over IP

Voice over Internet Protocol (v skratke VoIP) je technológia umožňujúca prenos digitalizovaného hlasu zabaleného najčastejšie v paketoch UDP cez počítačovú sieť vybudovanú nad IP protokolom. Najčastejšie sa využíva k telefonovaniu cez internet. Jednotlivé UDP pakety prenášajú vo svojom tele úseky telefónneho hovoru v dĺžke niekoľkých desiatok milisekúnd. Sú zabalené v protokole RTP (bližšie predstavený v kapitole 4.4) a zakódované vhodným kodekom (viac o kodekoch v kapitole 4.6). Okrem prenosu hlasových dát pomocou protokolu RTP sú pri VoIP telefónii prenášané aj riadiace a signalizačné informácie. Signalizácia je dôležitá pre uskutočnenie telefónneho hovoru, nastavovanie parametrov komunikácie (napríklad voľba kodeku) a ukončenie telefónneho hovoru. Protokolom RTCP (kapitola 4.5) môžu byť prenášané aj údaje o kvalite spojenia.

4.1 Session Initiation Protocol

Session Initiation Protocol (v skratke SIP) [23] je protokol určený pre vytváranie multimediálnych spojení v IP paketových sieťach. Vo VoIP telefónii je používaný ako signalizačný protokol. Implementáciu súčasnej verzie tohto protokolu detailne popisuje RFC 3261 [7] od IETF [1].

SIP je textový protokol typu Request-Response, ktorý sa svojou štruktúrou približuje protokolom HTTP a SMTP. Pracuje na siedmej vrstve referenčného modelu ISO/OSI [2]. Vznikol ako reakcia na príchod protokolu H.323, ktorý vyvinula organizácia ITU-T [10]. Oproti H.323 je SIP značne jednoduchší a založený na osvedčených princípoch. Vďaka týmto charakteristikám je protokol SIP vhodný pre integráciu do internetových služieb, čím významne rozširuje ich multimediálne možnosti, najmä prenos telefónnych hovorov cez internet. Protokol SIP slúži výhradne pre prenos riadiacich informácií a neprenáša žiadne užívateľské multimediálne dáta.

Hlavnou úlohou signalizačného protokolu SIP je vytvorenie multimediálnej relácie medzi účastníkmi konferencie. K tomu sú potrebné viaceré funkcie, predovšetkým:

[1] Internet Engineering Task Force (skratka IETF) je organizácia, ktorá sa zaoberá vývojom a udržiavaním štandardov používaných na Internete.

[2] Referenčný model ISO/OSI vypracovala štandardizačná organizácia ISO, má sedem vrstiev a jeho cieľom je poskytnúť základňu pre vypracovávanie noriem pre prepojovanie systémov a ich súčastí.

Lokalizácia účastníkov
Vyhľadanie polohy koncových staníc a cesty k nim.

Kontrola stavu účastníkov
Zistenie okolností, či je s účastníkom možné nadviazať spojenie.

Dojednanie parametrov spojenia
Podľa možností koncových staníc sa dohodnú parametre spojenia protokolom SDP (typ kodeku, prenosová rýchlosť, typ posielaných dát, atď.).

Nadviazanie spojenia
Zahájanie prenosu RTP paketov, ktoré sú nositeľmi multimediálnych dát medzi účastníkmi.

Riadenie multimediálnej relácie
Počas hovoru môže dôjsť k zmenám, v dôsledku čoho je potrebné prejednať parametre komunikácie (zmeniť kodek apod.). Postará sa taktiež o ukončenie multimediálnej relácie.

4.1.1 Architektúra SIP

Zaistenie týchto funkcií majú na starosti sieťové komponenty v SIP architektúre. Tieto komponenty SIP vytvárajú vlastnú sieť nad aplikačnou vrstvou.

Užívateľský agent

Užívateľský agent je súčasťou klientských zariadení, ktoré vyžívajú protokol SIP. Môže pracovať v režime klienta alebo servera. Tieto dva režimy sa v praxi často prelínajú a zariadenia pracujú súčasne v oboch režimoch súčasne. Ich rozdelenie má význam najmä pre pochopenie fungovania protokolu SIP a jeho správnu implementáciu. Jedná sa o architektúru klient-server so spojením typu point-to-point.

Užívateľský agent v režime klient (UAC)

V tomto režime užívateľský agent iniciuje spojenia a dotazuje protistranu, na ktorej beží užívateľský agent v režime server (UAS). Dotazy, ktoré môže UAC odosielať, sú INVITE, ACK, OPTIONS, CANCEL, BYE, REGISTER a ďalšie. Význam tých najpodstatnejších je uvedený v tabuľke 4.1.

Príkaz	**Použitie**
INVITE	Užívateľský agent žiada o iniciovanie telefónneho hovoru
ACK	Potvrdenie doručenia poslednej odpovede na žiadosť INVITE
CANCEL	Žiadosť o zrušenie nadväzovaného spojenia
BYE	Žiadosť o ukončenie prebiehajúceho spojenia
OPTIONS	Zistenie podporovaných funkcií UAS
REGISTER	Žiadosť o registráciu u Registrar serveru

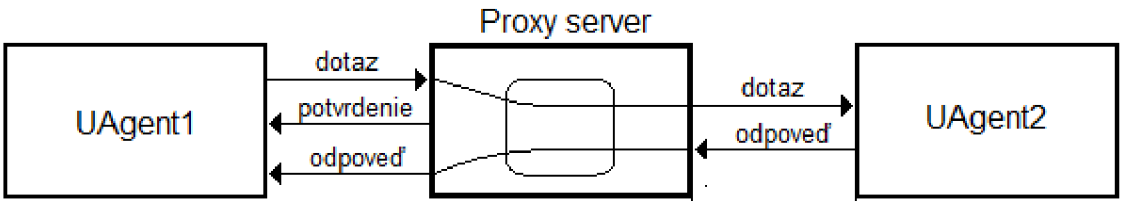
Tabulka 4.1: Prehľad typov žiadostí protokolu SIP definovaných v RFC 3261.

Užívateľský agent v režime server (UAS)

Užívateľský agent v režime servera prijíma dotazy od UAC a zasiela odpovede výhradne na tieto dotazy. Na jeden dotaz môže poslať aj niekoľko rôznych typov odpovedí. Sám v tomto režime ale neiniciuje žiadnu komunikáciu. Jedná sa teda o konkurentný server, ktorý dokáže bezodkladne obslúžiť viacero spojení s UAC.

Proxy server

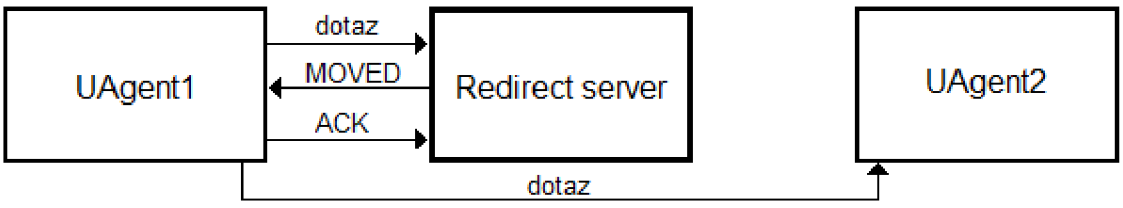
Proxy server plní funkciu prostredníka v architektúre SIP a smeruje žiadosti o spojenia medzi užívateľskými agentmi. Ak chce komunikovať užívateľský agent *UAgent1* s užívateľským agentom *UAgent2*, *UAgent1* zašle svoju požiadavku proxy serveru, ktorý ju prepošle cieľovému *UAgent2*. Proxy server môže byť stavový, ktorý monitoruje všetky správy a udržiava reláciu medzi komunikujúcimi stranami. Stavový proxy server umožňuje napríklad účtovanie. Bezstavový proxy server neudržuje medzi komunikujúcimi stranami reláciu ani nesleduje väzby medzi správami. Slúži iba na zabezpečenie smerovania v rámci SIP domény.



Obrázek 4.1: Funkcia proxy servera v architektúre SIP.

Redirect server

Redirect server je jednoduchý prostriedok, ktorý na žiadosť užívateľského agenta o komunikáciu s iným agentom (cieľovou stanicou) odpovie jeho aktuálnou adresou. Redirect server využíva pri lokalizácii cieľovej stanice databázu Registrar servera v kapitole 4.1.1.

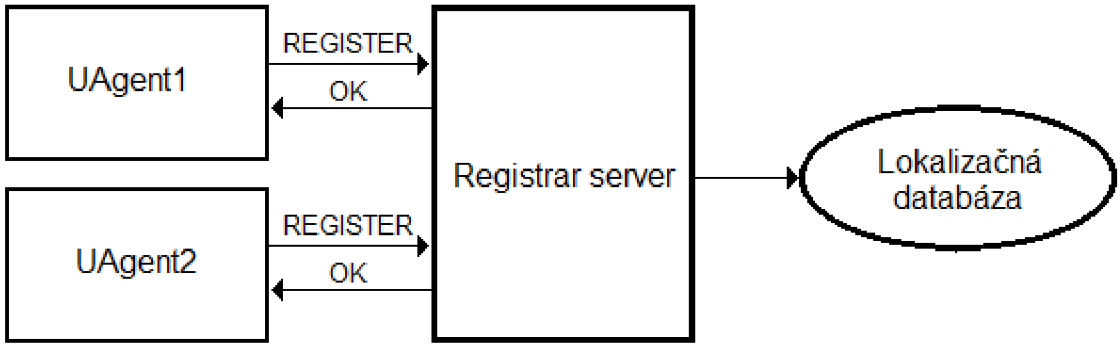


Obrázek 4.2: Redirect server v architektúre SIP.

Registrar server

Zabezpečuje registráciu užívateľských agentov a udržiava aktuálnu lokalizačnú databázu účastníkov. Jedná sa teda o zabezpečenie mapovania adries účastníkov na adresy fyzických

zariadení, na ktorých sa účastník práve vyskytuje.



Obrázek 4.3: Registračný server v architektúre SIP.

Registrácia je proces, pri ktorom účastnícka stanica, ktorá požaduje registráciu, zašle Registrar serveru požiadavku *REGISTER*, ten potvrdí prijatie a aktualizuje lokalizačnú databázu.

Lokalizačná databáza

Udržiava informácie o aktuálnej polohe registrovaných účastníkov, teda mapuje účastnícke mená na IP adresy a príslušný port, na ktorom načúva užívateľský agent na prichádzajúce spojenia.

4.1.2 Správy protokolu SIP

Komunikácia v protokole SIP je založená na výmene správ medzi užívateľskými agentmi - klientom a serverom. Rozlišujú sa pritom dva typy správ - žiadosti a odpovede. Žiadosti zasiela vždy klient na server, odpovede zase posiela server smerom ku klientovi.

Žiadosti v protokole SIP

Žiadosti zasiela klient serveru. Štruktúru žiadosti je možné vidieť na nasledujúcom príklade. V tejto žiadosti dotazuje klientska časť užívateľského agenta *helena@192.168.0.200* užívateľského agenta s URI *david@192.168.0.100* a žiada od neho zaslať späť akcie, ktoré tento užívateľský agent podporuje.

```
OPTIONS sip:david@192.168.0.100 SIP/2.0
Via: SIP/2.0/UDP 192.168.0.200:5060;rport;branch=z9hG4bK1556578081
From: <sip:helena@192.168.0.200>;tag=1847037790
To: <sip:david@192.168.0.100>
Call-ID: 1313112572
CSeq: 20 OPTIONS
Max-Forwards: 70
Content-Length: 0
```

V prvom riadku sa nachádza typ žiadosti, URI adresa užívateľského agenta, ktorého dotazujeme a verzia protokolu SIP, ktorú klient používa. Správa so žiadosťou obsahuje

identifikáciu odosielateľa v poli *From* a cieľového užívateľského agenta v poli *To*. Pole *Via* v žiadosti zabezpečuje správne smerovanie odpovedí pri prechode cez SIP Proxy servery.

Dôležitým polom je *Call-ID*, ktoré nesie identifikátor dialógu. Dialóg je možné chápať ako reláciu medzi užívateľskými agentmi (klientom a serverom), ktorá má nejaký cieľ, napríklad získanie podporovaných akcií protistrany alebo ustanovanie telefónneho hovoru. Pole *Max-Forwards* špecifikuje maximálny počet proxy serverov, cez ktoré môže byť správa preposlaná, pričom každý proxy server, cez ktorý správa prejde túto hodnotu dekrementuje o jedničku. Do správy môžu byť pridané aj iné voliteľné polia, ktoré protokol SIP podporuje, napríklad *Contact* a *Route*.

Najdôležitejšie a najčastejšie používané typy žiadostí v SIP protokole sú stručne charakterizované v tabuľke 4.1. Tieto žiadosti definuje štandard RFC 3261 [3]. Okrem nich existujú ďalšie typy žiadostí, ktoré zaisťujú doplnkové služby protokolu SIP. Takou žiadosťou je napríklad *MESSAGE*, definovaná v štandarde RFC 3428 [4], popisujúcom výmenu rýchlych textových správ (Instant messaging) pomocou protokolu SIP.

Odpoveď na žiadosť v SIP protokole

Odpoveď zasiela server klientovi ako reakciu na zaslanú žiadosť. V prvom riadku odpovede od SIP servera sa nachádza verzia SIP protokolu, ktorým protistrany komunikujú. Nasleduje kód odpovede a jeho textový popis. Kódy odpovedí sa delia do kategórií, ktoré sú uvedené v tabuľke 4.2. Príklad odpovede na dotaz, ktorým bol ukázaný formát zasielaných žiadostí protokolu SIP v podkapitole 4.1.2 je možné vidieť na nasledujúcej ukážke.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.0.100:5060;rport;branch=z9hG4bK1556578081
From: <sip:david@192.168.0.100>;tag=1847037790
To: <sip:helena@192.168.0.200>;tag=199057352
Call-ID: 1313112572
Allow: INVITE, ACK, BYE, CANCEL, OPTIONS, MESSAGE, SUBSCRIBE, NOTIFY, INFO
CSeq: 20 OPTIONS
Content-Length: 0
```

Kód	Kategória	Popis
1xx	Provisional	reakcia na spracovanie žiadosti
2xx	Success	kladná reakcia na zaslanú žiadosť
3xx	Redirection	potreba vykonať presmerovanie
4xx	Client error	chyba v žiadosti na strane klienta
5xx	Server error	server nemôže spracovať zaslanú žiadosť
6xx	General error	žiadosť nemôže byť vybavená

Tabulka 4.2: Kategórie kódov používaných v odpovediach protokolu SIP.

4.2 Session Description Protocol

Session Description Protocol (SDP) je protokol určený k prenosu informácií o vlastnostiach multimediálnej relácie. SDP býva obvykle súčasťou správy protokolu SIP. Jeho primárnou

[3] http://tools.ietf.org/html/rfc3261
[4] http://tools.ietf.org/html/rfc3428

úlohou je vyjednanie parametrov medzi účastníkmi, ktorí nadväzujú telefónny hovor prostredníctvom protokolu SIP. Vyjednávané parametre môžu byť napríklad typ média, ktoré sa bude prenášať (audio, video), transportný protokol, použitý pre prenos multimediálnych dát, použitý kodek a vzorkovacia frekvencia, prenosová rýchlosť a podobne. Protokol SDP je popísaný v RFC 4566 [9].

4.3 H.323

Protokol H.323 [22] je signalizačný protokol pre multimediálne relácie. Je definovaný ako štandard organizáciou ITU-T. Používaný je vo viacerých aplikáciách pre real-time komunikáciu po sieti, napríklad Ekiga. Jeho použitie nie je obmedzené iba pri IP telefónii, uplatňuje sa aj pri videokonferenciách. V tejto práci však podpora pre protokol H.323 nie je zahrnutá, a preto zmienka o tomto protokole je len informatívna.

Architektúra systému využívajúceho protokol H.323 je založená na sieťových prvkoch, ktoré kooperujú na dosiahnutí multimediálnej relácie. Medzi kľúčové prvky patria:

Brány Sú zariadenia, ktoré sprístupňujú komunikáciu medzi H.323 sieťou a vonkajším prostredím, napríklad PSTN alebo inou sieťou, ktorá nevyužíva protokol H.323.

Gatekeepery Sú voliteľné prvky H.323 sietí, ktoré poskytujú prvkom v H.323 sieťach radu dodatočných služieb. Medzi tieto služby patrí registrácia koncových zariadení, adresovanie, ovládanie hovorov, autentizácia užívateľov a podobne. Najdôležitejšou funkciou gatekeeperov je adresovanie umožňujúce prepojenie dvoch koncových bodov, ktoré spolu majú záujem komunikovať.

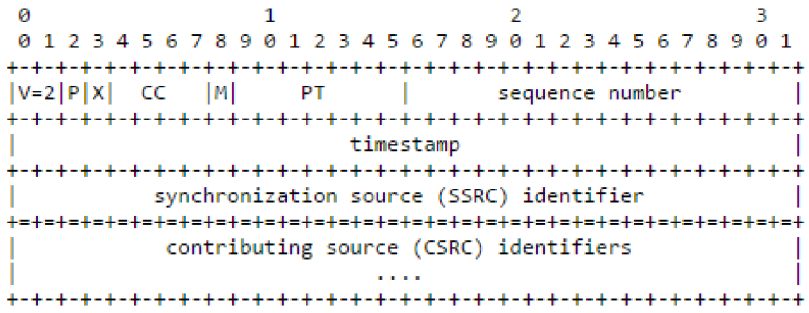
Terminály Sú elementárne prvky každej H.323 siete. Sú to koncové zariadenia, napríklad IP telefónne prístroje alebo videokonferenčné zariadenia. Taktiež to môže byť softwarový klient vo funkcii IP telefónnej stanice. Terminály disponujú sadou protokolov kompatibilných so štandardom H.323, ktoré umožňujú vytvorenie multimediálneho spojenia s inými terminálmi.

Kontrolné jednotky MCU Majú na starosti konferencie s viacerými komunikujúcimi stranami. Slúžia ako zmiešavače audio a video dát, aby užívatelia mali možnosť súčasne počuť aj vidieť všetkých účastníkov konferencie.

4.4 Real-time Transport Protocol

Real-Time Transport Protocol (RTP) je protokol určený k prenosu multimediálnych dát (obrazových a hlasových) cez internet. Pôvodne bol špecifikovaný už v roku 1996 v RFC 1889 [6]. V tomto štandarde je definovaný spoločne s protokolom RTCP (kapitola 4.5), ktorý slúži ako kontrolný protokol k monitorovaniu parametrov a riadeniu RTP komunikácie [13]. Tento štandard bol neskôr nahradený RFC 3550 [8].

Protokol RTP sa najčastejšie používa pre streamovanie audio a video dát v prostredí IP telefónie (VoIP) a pri videokonferenciách. Prenáša však iba dátové toky a neslúži pre dohodnutie parametrov spojenia ani negarantuje kvalitu služieb. K tomu slúžia signalizačné protokoly (napríklad SIP a H.323) a kontrolný protokol RTCP. RTP dáta sú skoro vždy prenášané pomocou nespojovanej služby UDP. Pre komunikáciu RTP protokolom sa využívajú párne čísla portov. O jedno väčšie číslo portu potom slúži pre zodpovedajúci RTCP prenos riadiacich informácii k danému RTP toku.



Obrázek 4.4: Štruktúra RTP paketu (prevzaté z [8]).

Na obrázku 4.4 je zobrazená štruktúra RTP paketu. Obsahuje informácie o verzii protokolu RTP (možno špecifikovať hodnotu 1 pre starú verziu RTP). Ďalej obsahuje položky:

PT - Payload type pole identifikuje formát obsahu RTP paketu a určuje spôsob jeho interpretácie aplikáciou.

Sequence number sekvenčné číslo RTP paketu, ktoré je inkrementované o jedničku s každým vyslaným RTP paketom. Príjemcovi umožňuje detekovať stratu paketov a zostaviť pakety do správnej postupnosti v akej boli vyslané. Počiatočná hodnota sekvenčného čísla môže byť náhodná a jej voľba súvisí s bezpečnosťou. Dĺžka sekvenčného čísla je 16 bitov.

Timestamp je časové razítko, ktoré reflektuje okamih počiatku vzorkovania prvého oktetu multimediálneho obsahu, ktorý je RTP paketom prenášaný. Umožňuje korekciu v časovaní prenášaných dát. Počiatočná hodnota sa volí náhodne, rovnako ako počiatočná hodnota sekvenčného čísla.

Synchronization source - SSRC Hodnota s dĺžkou 32 bitov je špecifická pre každý tok RTP paketov. Pakety s rovnakou hodnotou SSRC majú teda zhodné časovanie a sekvenčné čísla patriace do rovnakého rozsahu. Hodnota SSRC je zvolená pri ustanovení spojenia a volí sa náhodne. Musí sa však zaručiť jedinečnosť tejto hodnoty vrámci RTP relácie.

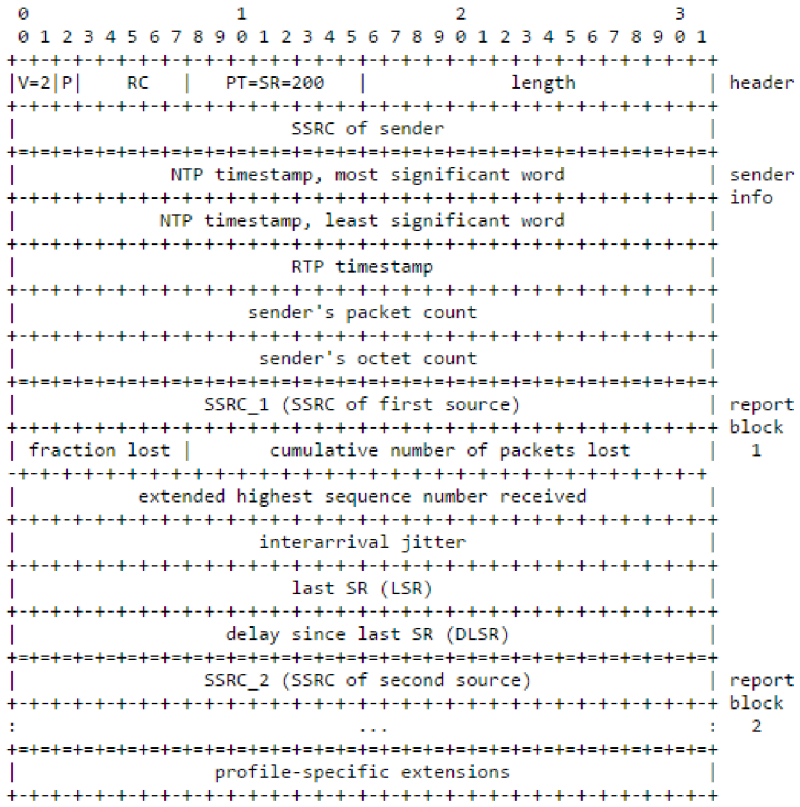
Contributing source - CSRC Je taktiež hodnota s dĺžkou 32 bitov a špecifikuje zdroj RTP paketov, ak sú tieto vkladané do iného RTP toku, napríklad komponentou nazývanou RTP mixer pri konferenčnom hovore. V tomto hovore potom pakety obsahujú identifikáciu odosielateľa v položke CSRC. Hodnota SSRC identifikuje RTP mixer komponentu.

4.5 Real-Time Control Protocol

Je riadiaci protokol pracujúci na aplikačnej vrstve, ktorý dopĺňa protokol RTP pri real-time prenose zvuku a videa cez počítačovú sieť. Jeho štruktúra paketu je definovaná v RFC 3550 [8] spoločne s protokolom RTP. RTCP prenáša riadiace informácie a štatistické údaje o toku RTP dát. Obvykle tvorí približne 5% veľkosti prenášaných RTP dát. Perióda vysielania RTCP informácií je zvyčajne niekoľko sekúnd. Sám však RTCP žiadne multimediálne dáta neprenáša. Hlavnou úlohou RTCP paketov je poskytovať spätnú väzbu o kvalite služby,

na základe ktorej je možné ovplyvniť parametre prenosu RTP dát, napríklad zmena smerovania, úprava QoS na danej sieti, zmena kodekov použitých pre kódovanie prenášaných dát a podobne. RTCP umožňuje zlúčiť viac RTCP paketov do jedného, pričom jednotlivé časti sú spracované nezávisle.

Vysielanie RTCP paketov však nie je podmienkou pri RTP prenose a jedná sa iba o doplnkovú službu. Pri hodnotení kvality sa na prítomnosť RTCP správ nedá spoľahnúť, preto je vhodné vedieť určovať kvalitu multimediálneho spojenia priamo z RTP dátového toku.



Obrázek 4.5: Štruktúra RTCP paketu typu Sender Report (prevzaté z [8]).

RTCP pakety obsahujú informácie o RTP prenose, a to predovšetkým:

- Počet prenesených bajtov
- Počet prenesených paketov
- Počet stratených paketov
- Jitter - kolísanie oneskorenia doručenia paketov do cieľa, môže byť spôsobené napríklad rozdielnou cestou jednotlivých paketov
- NTP časová značka - časové razítko odoslania RTCP kontrolného paketu
- Round trip delay - oneskorenie vzniknuté pri ceste tam a späť

Ďalšou úlohou, ktorú zabezpečuje Real-Time Control Protocol, je distribúcia kanonického mena, ktoré slúži ako trvalý identifikátor RTP zdroja. Tento identifikátor je vo formáte

username@hostname. Ak je to jednopoužívateľský stroj, môže byť týmto identifikátorom iba hostname. Do RTCP paketov je možné pridať špecifické voliteľné parametre, napríklad informácie o sedení, ktoré sú bežne distribuované prostredníctvom iných k tomu určených protokolov.

4.5.1 Typy RTCP paketov

RTCP používa rovnako ako RTP k prenosu UDP protokol. Jednotlivé pakety tohto protokolu sa líšia typom správy, ktorú RTCP paket zapúzdruje. Rozlišujú sa nasledujúce typy RTCP správ [25]:

Sender Report - SR

Správy typu Sender Report odosielajú účastníci RTP komunikácie a sú určené pre aktívnych účastníkov. Tieto správy obsahujú informácie o prebiehajúcej komunikácii a prijímacie štatistiky pre všetky RTP pakety. Tieto informácie majú príjemcovi pomáhať odhadovať kvalitu a spoľahlivosť prenosu. Ten na základe nich môže prispôsobovať parametre komunikácie. Správy sú synchronizované na základe časového razítka v formáte UNIX-TIMESTAMP.

Receiver Report - RR

Správy Receiver Report slúžia pre pasívnych účastníkov komunikácie, teda pre účastníkov, ktorí nie sú momentálne aktívnymi odosielateľmi. Správy svojim obsahom informujú účastníkov o kvalite služieb a prenosu, o problémoch prijímateľov a obsahujú počty stratených paketov a informácie o jitter-i na strane príjemcu. Vo výsledku môže aplikácia na základe takejto správy znížiť alebo zvýšiť kvalitu multimediálneho obsahu.

Source Description Message - SDES

Správy tohto typu sú vysielané pravidelne a nesú informácie o vysielateľovi, ktoré takto sprístupňuje ostatným účastníkom. SDES správy definuje RFC 1889 [6]:

CNAME vysiela kanonické meno odosielateľa

NAME reálne meno a priezvisko užívateľa alebo jednoznačný identikátor zdroja

EMAIL e-mailová adresa užívateľa

PHONE telefónne číslo používateľa

LOC geografická poloha užívateľa

TOOL názov aplikácie a informácie o jej verzii, pomocou ktorej užívateľ komunikuje

ďalšie typy napríklad aplikačne špecifické správy pre rôzne prípady použitia RTP spojenia

End of participation - BYE

Vysielač informuje o ukončení zasielania RTP dát a opustení konferencie ostatných účastníkov.

Application-Specific Message - APP

Slúži pre zasielanie špecifických správ, ktoré nie sú priamo definované štandardom. Tieto môžu byť použité napríklad pre experimentálne účely.

4.6 Kodeky pre kódovanie hlasu

Kodek je algoritmický prostriedok pre kódovanie hlasovej informácie (analógovej veličiny) do digitálnej podoby, v ktorej je možné preniesť túto informáciu počítačovou sieťou. Je nevyhnutnou súčasťou všetkých koncových zariadení používaných v IP telefónii. Typ použitého kodeku je špecifikovaný v hlavičke RTP paketu v poli Payload Type.

Kodek obsahuje prostriedky, ktoré kódujú analógová signál na digitálny. Táto úprava zmenšuje kvalitu prenášanej hlasovej informácie. Kodek môže definovať rôznu úroveň kompresie hlasovej informácie. Kompresia však musí spĺňať kritériá, ktoré zabezpečia vhodnú kvalitu kódovanej hlasovej informácie s rozumnými nárokmi na výpočtové prostriedky. Zároveň však prenášaná informácia v IP telefónii musí zaberať čo najmenšie prenosové pásmo a teda čo najmenej zaťažovať prenosovú linku.

Jednotlivé kodeky sa v týchto parametroch odlišujú. Každý kodek definuje vlastný kompresný algoritmus a požadované prenosové pásmo. Kodeky sa odlišujú taktiež v hardwarových nárokoch. Vhodná voľba kodeku je mimoriadne dôležitá pre kvalitný prenos hlasu cez internet a rozumné využitie dostupného prenosového pásma. V nasledujúcej časti sú stručne charakterizované kodeky G.711 a G.729 najčastejšie používané v IP telefónii a telekomunikáciách [5] a kodek Speex.

4.6.1 G.711

Kodek G.711 je najpoužívanejší kodek používaný v telekomunikáciách. Organizáciou ITU-T bol schválený už v roku 1988. Jeho jednoduchosť je však nevýhodou kvôli potrebe väčšieho prenosového pásma. Používa pulznú kódovú moduláciu a pracuje na vzorkovacej frekvencii 8kHz s veľkosťou jednej vzorky 8 bitov. To udáva požadovanú prenosovú rýchlosť 64kbit/s. Pri kodeku G.711 existujú 2 algoritmy kódovania. Najrozšírenejší je algoritmus A-law, v Japonsku a Severnej Amerike sa používa algoritmus μ-law .

4.6.2 G.729

G.729 je veľmi úsporný kodek, ktorý vyžaduje prenosové pásmo iba 8kbit/s a poskytuje mimoriadne dobrú kvalitu prenášaného hlasu. Je teda vhodný pre použitie v IP telefónii, pretože nezaťažuje nadmerne prenosové pásmo a umožňuje pracovať aj pri nízkych prenosových rýchlostiach. Kodek využíva algoritmus CS-ACELP. Existujú modifikácie tohto kodeku s požiadavkami na prenosové pásmo iba 6.4kbit/s s horšou a 11.8kbit/s s lepšou výslednou kvalitou kódovaných dát. Jeden paket obsahuje 10 milisekúnd prenášanej audio informácie.

4.6.3 Speex

Speex je otvorený a slobodný kodek primárne navrhnutý pre kódovanie hlasu vo VoIP. Je navrhnutý pre použitie so vzorkovacími frekvenciami 8kHz, 16kHz a 32kHz. Umožňuje nastaviť kvalitu kódovania hlasových vzorkov s konštantnou alebo variabilnou bitovou šírkou.

Kódovanie s variabilnou bitovou šírkou umožňuje kodeku dynamickú adaptáciu na základe zložitosti kódovaného vzorku. Kodek má zabudovanú podporu technológie *Voice Activity Detection*, ktorá má význam najmä pri internetovej telefónii. Umožňuje rozoznať, či účastník práve hovorí, alebo má práve prestávku v reči. Vtedy produkuje iba malé množstvo bitov, ktoré postačia na generovanie šumu. V prípade dlhšej neaktivity účastníka úplne pozastaví prenos dát.

4.7 Monitorovanie kvality VoIP hovorov

Metódy hodnotenia kvality vo VoIP, teda kvality prenosu hlasu cez paketové siete, sa delia na objektívne a subjektívne [3]. Subjektívne metódy využívajú pri hodnotení vnemové schopnosti poslucháčov, ktorí na základe vypočutého udelia hodnotenie kvality prenosu. Aritmetickým priemerom získaných hodnotení od jednotlivých poslucháčov sa vypočíta hodnota MoS (Mean Opinion Score).

Objektívne metódy spočívajú v nahradení človeka v roli poslucháča a hodnotiaceho člena vhodným algoritmom. Algoritmus počíta hodnotu MoS zo vstupného vzorku. Dôležitým kritériom kladeným na objektívne metódy hodnotenia kvality VoIP komunikácie je dosiahnutie čo najpresnejších výsledkov MoS v porovnaní so subjektívnymi metódami hodnotenia.

Nie je možné odporučiť, ktoré metódy sú pre hodnotenie kvality hovorov prenášaných cez IP siete vhodnejšie. Subjektívne aj objektívny metódy majú svoje kladné aj záporné stránky. Výhodou subjektívnych metód merania je priame hodnotenie dojmov z telefonickej komunikácie užívateľmi, čo nie je možné dosiahnuť žiadnym algoritmom. Algoritmy, používané objektívnymi metódami, dokážu hodnotu MoS vypočítať len približne na základe parametrov prenášaných dát. Výhodou algoritmického výpočtu hodnoty MoS však je možnosť automatizovaného určovania kvality VoIP a možnosť rýchleho zásahu v prípade, že je detekované náhle zníženie kvality hovorov. Subjektívne metódy naviac môžu byť ovplyvnené náladou užívateľov, ktorá môže súvisieť s počasím, zdravotnými aspektami a ďalšími okolnosťami, ktoré môžu kladne aj záporne vplývať na hodnotu MoS. Subjektívne metódy hodnotenia sú využívané prevažne na kalibráciu algoritmov pre objektívne hodnotenie.

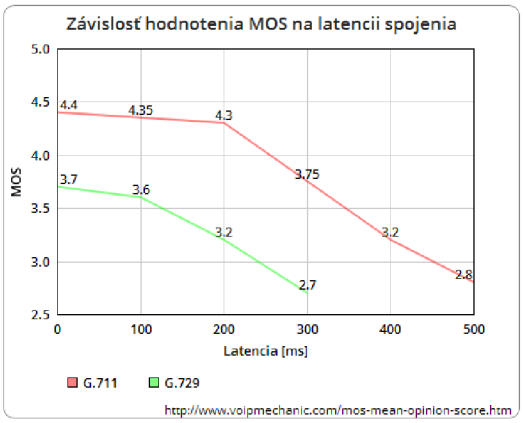
4.7.1 Subjektívne metódy hodnotenia kvality

Sú založené na subjektívnom vnímaní kvality hovoru užívateľom. Každá osoba interpretuje kvalitu hlasového prenosu odlišným spôsobom, pričom na hodnotení sa odráža aj jej nálada. Hodnoty kvality prenosu sú obvykle vyjadrené metrikami, ktoré sú funkciami kvality sieťového prenosu vplývajúcich na kvalitu danej služby. Stanovisko spokojnosti užívateľa s kvalitou služby sa vyjadruje v hodnotách MoS (Mean Opinion Score). Tabuľka 4.3 zhruba vyjadruje spokojnosť užívateľov v závislosti na hodnote MoS.

hodnota MOS	**subjektívna kvalita**	**snaha o porozumenie**
5	vynikajúca	porozumenie bez výhrad
4	dobrá	porozumenie dobré aj bez vynaloženia úsilia
3	primeraná	porozumenie možné s menším úsilím
2	zlá	porozumenie možné s veľkým úsilím
1	nedostatočná	nie je možné porozumieť

Tabulka 4.3: Spokojnosť užívateľov v závislosti na hodnote MoS.

Na grafe 4.6 je pre ilustráciu zobrazené hodnotenie užívateľov MoS v závislosti na latencii linky pre bežne používané kodeky G.711 a G.729.

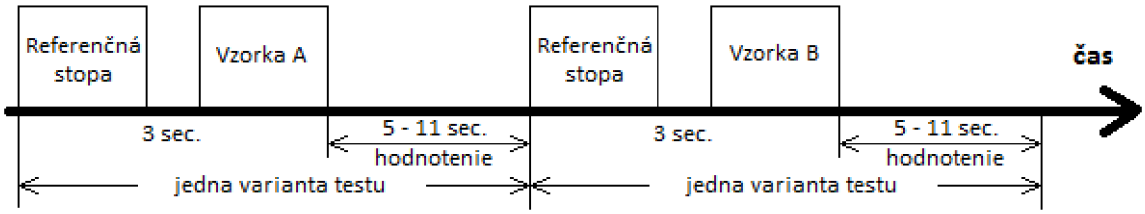


Obrázek 4.6: Závislosť hodnotenia MOS na latencii spojenia (dáta prevzaté z http://www.voipmechanic.com/mos-mean-opinion-score.htm).

Degradating Category Rating

Metódou používanou pri subjektívnom hodnotení kvality je Degradating Category Rating, ktorá udeľuje ako známku strednú hodnotu degradácie, teda aritmetický priemer všetkých vykonaných meraní. Je odporúčaná štandardom ITU-T P.910 [12]. Niekedy sa táto metóda nazýva tiež Double Stimulus Impairment Scale (DSIS). Výhodou je väčšia miera citlivosti k výsledku aj pri malých rozdieloch v kvalite.

Pri tejto metóde užívateľ postupne ohodnocuje vzorky s rôznym stupňom degradácie. Priebeh hodnotenia je znázornený na obrázku 4.7.



Obrázek 4.7: Priebeh hodnotenia subjektívnou metódou DCR.

Užívateľ postupne prechádza variantami testu, pričom v každej variente si vypočuje najskôr originálnu, neskreslenú referenčnú vzorku. Jej dĺžka je približne 10 sekúnd, môže byť však aj iná. Príliš krátka vzorka neobsiahne dostatočnú škálu hodnotiacich faktorov, dlhá vzorka však môže byť užívateľovi nepríjemná, čo sa môže prejaviť na jeho hodnotení. Po krátkej prestávke nasleduje testovacia vzorka, ktorá je istým spôsobom skreslená. Po jej vypočutí má užívateľ primeraný čas na hodnotenie stupňa degradácie skreslenej vzorky voči referenčnej vzorke. Tento spôsob testovania sa využíva aj v iných aplikáciách, napríklad pri hodnotení kvality videa. Naledujúca tabuľka 4.4 orientačne vyjadruje skóre degradácie vzhľadom k pocitom užívateľa.

Skóre DCR	Stupeň degradácie
5	Nepostrehnuteľný
4	Postrehnuteľný, ale bez výhrad
3	Postrehnuteľný, mierne znepokojujúci
2	Prítomný, znepokojujúci
1	Veľmi znepokojujúci

Tabulka 4.4: Skóre hodnotenia pre metódu DCR.

Absolute Category Rating

Výstup metódy Absolute Category Rating - ACR udeľuje známku, ktorá reprezentuje spokojnosť užívateľa. Testovanie prebieha obdobným spôsobom ako pri metóde DCR (4.7.1). Rozdiel je, že užívateľ nemá k dispozícii pri každej variante referenčnú vzorku, ale hodnotí svoju spokojnosť s vypočutým testovacím vzorkom, ktorý je nejakým spôsobom skreslený. Výsledné skóre hodnotenia je aritmetickým priemerom hodnotenia všetkých variant testu. Hodnoty skóre pre rôzne varianty sú uvedené v nasledujúcej tabuľke 4.5.

Skóre ACR	Kvalita
5	Vynikajúca
4	Bez výhrad
3	Malé výhrady
2	Prijateľná s výhradami
1	Neprijateľná, veľmi zlá

Tabulka 4.5: Skóre hodnotenia pre metódu ACR.

V niektorých modifikáciách sa používajú aj varianty s 10 stupňami hodnotenia. Pri tvorbe rôznych variant testu je možné vymazávať postupne 1%, 2%, 5% až do 20% z reči náhodne alebo zhlukovo z celej vzorky. Takto je simulovaná strata paketov v sieťovej prevádzke.

4.7.2 Objektívne metódy hodnotenia kvality

Objektívne metódy hodnotenia kvality hovorov nie sú založené na posluchu vzoriek človekom, ale využívajú porovnávacie metódy. Delia sa na intruzívne a neintruzívne. Intruzívne objektívne metódy hodnotenia kvality (napríklad PESQ, PEAQ, PAMS) sú nazývané aj aktívne, pretože k ohodnoteniu kvality prenesenej hlasovej vzorky vyžadujú dostupnosť referenčnej neskreslenej vzorky. Výsledné hodnotenie potom spočíva v porovnaní prenesenej vzorky so skreslením a zodpovedajúcej originálnej vzorky bez skreslenia.

Neintruzívne metódy hodnotenia kvality hovorov sú pasívne a nevyžadujú referenčnú vzorku k ohodnoteniu kvality. Sú založené na analýze zachytených paketov s hlasovými dátami, alebo na analýzach skreslenej vzorky bez použitia referenčnej vzorky. Z pohľadu monitorovania paketov prenášajúcich multimediálnu prevádzku je vhodným práve použitie neintruzívnych metód hodnotenia kvality VoIP. Takouto metódou hodnotenia kvality je napríklad E-Model (sekcia 4.7.3).

Perceptual Evaluation of Speech Quality - PESQ

PESQ [24] je metóda štandardizovaná ITU-T ako P.862. Je to objektívna intruzívna metóda porovnania hlasových signálov. Porovnáva pôvodný signál X(t) s degradovaným signálom Y(t). Degradácia signálu je dosiahnutá prenosom signálu cez komunikačné linky zo zdroja k príjemcovi. Výsledkom PESQ je hodnotenie kvality vypočutej hlasovej vzorky na strane príjemcu, ktoré je porovnateľné so subjektívnym hodnotením poslucháča. Metóda PESQ porovnáva hlasový signál prenesený cez testovanú komunikačnú linku s originálnym neskresleným vzorkom. Pri vysokých paketových stratách nad 20% sú však výsledky tejto metódy nepresné. Výsledné hodnotenie udáva hodnotu MoS.

Perceptual Evaluation of Audio Quality - PEAQ

V aktívnej intruzívnej metóde PEAQ je použitý psychoakustický model, ktorý vytvára premenné, založené na porovnaniach medzi referenčným signálom a tým istým signálom spracovaným konkrétnym kodekom. Tieto premenné sa použijú k odhadnutiu subjektívnej kvality, ktorá zodpovedá spracovanému signálu. Objektívne meranie muselo byť predtým kalibrované výsledkami z množiny uskutočnených posluchových testov.

Základná verzia tejto metódy je dostatočne rýchla aj pre real-time monitorovanie. Psycho-akustický model je založený na rýchlej Fourierovej transformácii, pomocou ktorej mení signál do časovo-frekvenčnej reprezentácie.

Perceptual Analysis Measurement System - PAMS

PAMS (Perceptual Analysis Measurement System) je intruzívna metóda objektívneho hodnotenia kvality, ktorej výsledkom je hodnotenie zodpovedajúce celkovému subjektívnemu hodnoteniu kvality počujúceho. Je založená na vytvorení modelu s časovo synchronizovanými vzorkami referenčného signálu a degradovaného signálu. Model je schopný identifikovať oneskorenia medzi úsekmi vzoriek. Tie môžu vzniknúť napríklad ako dôsledok asymetrického smerovania.

Obidva signály sa vyrovnajú na štandardnú posluchovú úroveň 79dB. Vykonajú sa sluchové transformácie, čo zahŕňa prechod cez filter modelujúci ušnú dutinu a zvukovod, lineárne filtre pre transformáciu do rôznych frekvenčných pasiem. Nasleduje výpočet výkonu pre každé frekvenčné pásmo pre rámce o dĺžke 4 milisekundy.

Následnou parametrizáciou chyby sa detekuje miera odlišnosti jednotlivých tried skreslenia, ktoré sú spriemerované v čase a namapované do výsledku cez nelineárnu funkciu, ktorá zachováva monotónny vzťah medzi každým parametrom a výsledkom kvality.

4.7.3 Hodnotenie kvality VoIP založené na E-modeli

E-Model je výpočetný model, slúžiaci pre vyjadrenie R-faktoru ako prostriedku pre ohodnotenie kvality telefónneho hovoru. Pri výpočte R-faktoru sa berú do úvahy účinky, ktoré pozitívne alebo negatívne pôsobia na kvalitu hovoru. Komponenty z ktorých sa výpočet R-faktoru skladá popisuje štandard G.107 [11]. Výpočet kombinuje všetky prenosové parametre, ktoré vplývajú na spojenie. R-faktor sa skladá z:

$$R = R_0 - I_S - I_D - I_{E-EFF} + A$$

, kde:

R_0 - je koeficient vyjadrujúci pomer signálu a šumu (Signal to noise ratio - SNR)

I_S - súčet všetkých znehodnotení, ktoré môžu nastať v súvislosti s prenosom hlasu

I_D - faktor znehodnotenia, reflektujúci všetky druhy oneskorenia hlasového signálu

I_{E-EFF} - fakt zhoršenia vplyvom paketovej straty, vychádza z hodnoty I_E, ktorá je subjektívne zmeraná pre konkrétny použitý kodek, ktorým je hlasová informácia zakódovaná

A - faktor zvýhodnenia, ktorý kompenzuje horšiu kvalitu v náročnejších podmienkach (napríklad mobilné telefóny, satelitné stanice apod.)

Výslednú hodnota R-faktoru je potom možné previesť na hodnotu MoS. V tabuľke 4.6 je vidieť približnú závislosť hodnoty MoS na vypočítanej hodnote R-faktoru.

Užívateľské hodnotenie	**R-factor**	**MOS**
Maximum, dosiahnuteľné s kodekom G.711	*93 %*	*4,4*
Veľmi spokojní	90 - 100 %	4,3 - 5,0
Spokojní	80 - 90 %	4,0 - 4,3
Väčšina užívateľov spokojná	70 - 80 %	3,6 - 4,0
Niektorí užívatelia nespokojní	60 - 70 %	3,1 - 3,6
Skoro všetci užívatelia nespokojní	50 - 60 %	2,6 - 3,1
Absolútne neuspokojivé	0 - 50%	1,0 - 2,6

Tabulka 4.6: Tabuľka približne vyjadrujúca hodnoty MoS v závislosti na hodnotách R-faktoru, zdroj: http://www.tamos.com/htmlhelp/voip-analysis/mosandr_factor.htm.

4.7.4 Charakteristiky prenosu vplývajúce na kvalitu VoIP

V predchádzajúcich kapitolách boli prezentované metódy určovania kvality VoIP komunikácie. V tejto diplomovej práci sa zameriavam na určenie kvality VoIP telefónnych hovorov na základe charakteristík, ktoré je možné zmerať z dát, zachytených medzi účastníkmi telefónneho hovoru. Takýmito charakteristikami sú najmä oneskorenie, jitter a stratovosť paketov [15].

Oneskorenie

Parameter oneskorenie v telekomunikáciách a prenose hlasu vyjadruje dobu, ktorá uplynie medzi okamžikom, keď volajúci prehovorí a okamžikom, keď si volaný túto správu vypočuje. V IP telefónii sa hodnoty oneskorenia pohybujú v intervale 50 - 150ms. Takéto oneskorenie účastníci ani neregistrujú. Oneskorenie nad 400ms je už možné postrehnúť, avšak nespôsobuje nezrozumiteľnosť [19]. Tento faktor je dosť dobre akceptovateľný aj pri vyšších hodnotách. Výrazné zvýšenie hodnoty oneskorenia však spôsobuje, že účastníci telefónneho hovoru sa prerušujú (neúmyselne si skáču do reči).

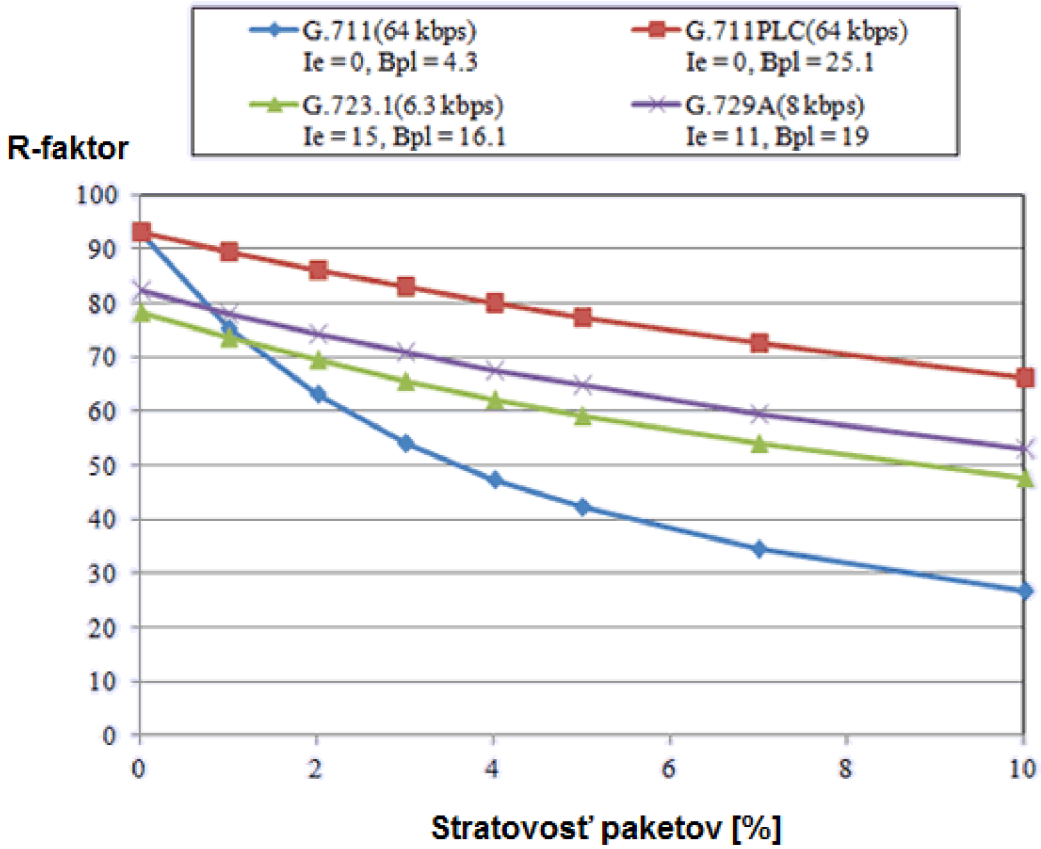
Jitter

Očakávaný čas príchodu ďalšieho paketu v multimediálnom toku je možné predpovedať na základe znalosti frekvencie vysielania týchto paketov. Jitter je časový interval medzi očakávaným a skutočným príchodom paketu. Na jitter majú zvyčajne vplyv rozdielne cesty, ktorými sú pakety doručované od vysielača k príjemcovi. Môže sa tak stať, že RTP paket s vyšším sekvenčným číslom (bol vygenerovaný a odoslaný neskôr) dorazí k príjemcovi

skôr ako RTP paket s nižšou hodnotou sekvenčného čísla. Ku kompenzácii jitteru sa používa vyrovnávacia pamäť (jitter buffer), do ktorej sa pakety zaznamenávajú a zoraďujú do správneho poradia predtým, než sú znova spracované kodekom a prehrané užívateľovi.

Stratovosť paketov

Stratovosť paketov je faktor, ktorý priamo vplýva na kvalitu hovoru. Je to pomer medzi počtom stratených paketov a počtom všetkých paketov, ktoré boli odoslané k príjemcovi. Počet stratených paketov je možné tiež vyjadriť ako rozdiel hodnôt počtu odoslaných paketov a počet doručených paketov. Pri prenose dát v IP telefónii protokolom RTP sa najčastejšie používa nespojovaná služba UDP, kde sa stratovosť pripúšťa. Kodeky používané vo VoIP sú schopné kompenzovať malý počet stratených RTP paketov bez toho, aby to užívateľ postrehol ako zhoršenie kvality telefónneho hovoru. Približný odhad závislosti R-faktoru a stratovosti paketov je uvedený v grafe na obrázku 4.8. Dôležité však je, aby straty neboli zhlukového charakteru, teda že stratených je viac po sebe idúcich paketov. Zhlukové straty nie je možné efektívne zahladiť kodekom a dochádza k prerušovaniu telefónneho hovoru, čo výrazne znižuje spokojnosť užívateľov s jeho kvalitou.



Obrázek 4.8: Hodnota R-faktoru v závislosti na stratovosti paketov pre rôzne typy kodekov. Zdroj: http://www.ntt.co.jp/qos/qoe/eng/technology/sound/05_2.html

Kapitola 5

Prostredie pre monitorovanie sieťovej prevádzky

Táto diplomová práca sa zaoberá návrhom a tvorbou univerzálneho a rozšíriteľného monitorovacieho prostredia s využitím technológie Cisco OnePK. Prostredie, ktorého tvorba je súčasťou tejto diplomovej práce, bolo nazvané OneMon. Tento názov vznikol spojením slov *One*, čo je skratka marketingového označenia *Open Network Environment* spoločnosti Cisco pre produkty, kam patrí aj technológia Cisco OnePK a slova *Mon* symbolizujúceho monitorovanie. Prostredie OneMon bolo vyvinuté pre potreby výskumnej skupiny Sec6Net[1]. Má slúžiť predovšetkým pre zachytávanie dát zo sieťových zariadení, zaujímavých z hľadiska real-time analýzy, napríklad pre monitorovanie dostupnosti a kvality služieb, alebo pre uloženie zachytených dát a ich neskoršiu off-line analýzu.

5.1 Požiadavky na monitorovanie

Na začiatku vývoja monitorovacieho nástroja OneMon bola požiadavka vyvinúť takéto prostredie na existujúcej komerčnej platforme Cisco. Zariadenia spoločnosti Cisco sú vo veľkom počte rozšírené na všetkých úrovniach počítačových sietí. Najnovšie verzie smerovačov od spoločnosti Cisco podporujú technológiu One Platform Kit (bola predstavená v kapitole 3). Súčasťou technológie OnePK je DataPath Service Set, prostredníctvom ktorého je možné zbierať dáta prechádzajúce sieťovým prvkom, obvykle smerovačom. Táto práca sa sústreďuje na zachytávanie dát súvisiacich s IP telefóniou, predovšetkým paketov signalizačného protokolu SIP a dátových paketov RTP, prenášajúcich multimediálne zvukové dáta. Real-time spracovanie týchto dát umožňuje sledovanie telefónnych hovorov uskutočnených nad týmito protokolmi, ktoré prechádzajú sledovaným segmentom siete.

Monitorovacie prostredie OneMon však nie je určené iba pre zachytávanie prevádzky, ale súčasne je navrhnuté tak, aby poskytovalo možnosť zasahovať do konfigurácie siete a ovplyvňovať prevádzku na sieti niekoľkými spôsobmi, čo korešponduje s hlavnou myšlienkou softwarovo definovaných sietí. Takto by mohlo napríklad vykonávať zmeny v paketoch, ktoré prechádzajú smerovačom. Môže zasahovať do nastavenia kvality služieb QoS a dynamicky ju meniť v závislosti na kvalite spojenia, prípadne upraviť a korigovať smerovanie.

[1] http://www.fit.vutbr.cz/research/view_project.php.cs?id=517&notitle=0&format=0&shortname=0

5.2 Návrh prostredia OneMon

Pri návrhu prostredia OneMon sa vychádzalo z uvedených požiadaviek a dôraz sa kládol na modularitu a rozšíriteľnosť. Viedlo to k robustnejšiemu a náročnejšiemu prostrediu. Jeho využitie však nie je limitované iba jednou aplikáciou, ale vďaka vytvoreniu OneMon API je možné prostredie prispôsobiť aj pre monitorovanie iných služieb ako VoIP.

Bolo navrhnuté a vyvinuté ako rozšíriteľné prostredie, ktorého jadro tvorí súbor funkcií na konfiguráciu zachytávania sieťovej prevádzky priamo zo sieťových zariadení s podporou Cisco OnePK. Prenos zachytených dát zo sieťových elementov do monitorovacej aplikácie OneMon, ktorá ich spracuje je vykonávaný prostredníctvom DataPath Service Set-u (predstavený v kapitole 3.2.1), ktorý je súčasťou platformy Cisco OnePK.

Komponentou jadra je ďalej systém pre zber zachytených dát od sieťových elementov, ktorý umožňuje ich real-time analýzu, alebo ukladanie pre off-line spracovanie v budúcnosti. Nad vrstvou jadra sa nachádza rozhranie pre správu monitorovania. Toto rozhranie umožňuje správcovi špecifikovať požadované sieťové toky, pričom vďaka funkcionalite NBAR (kapitola 3.3) je možné vykonávať rozpoznanie aplikačného protokolu (protokolu na siedmej vrstve ISO/OSI modelu) už v sieťovom zariadení, ktorého prevádzka je cieľom monitorovania. Týmto spôsobom je odbúrané množstvo paketov aplikačných protokolov, ktoré nie sú z pohľadu monitorovania zaujímavé a ich distribúcia do monitorovacej platformy OneMon by pre ňu predstavovala iba prebytočnú záťaž.

5.3 Architektúra

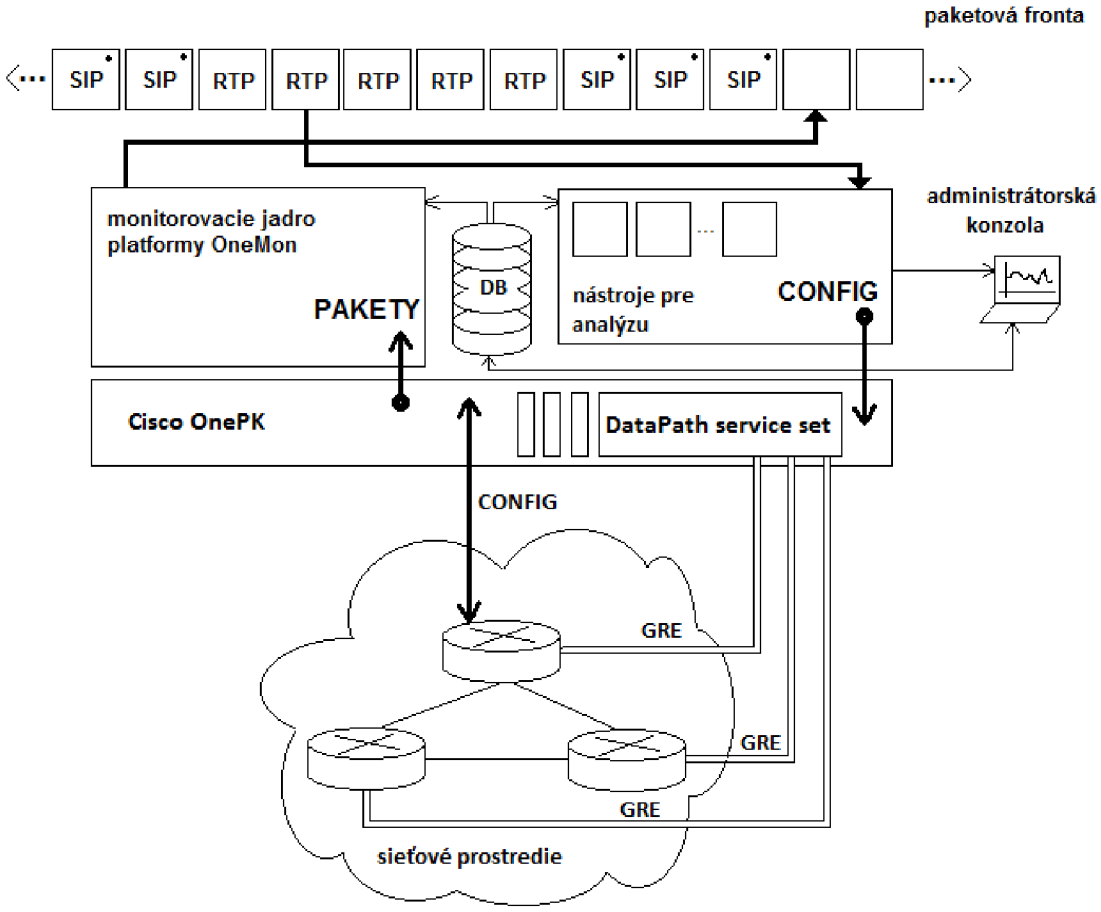
Architektúra je konceptuálne znázornená na obrázku 5.1. Spodná časť predstavuje OnePK sieť, nad ktorou pracujú nástroje pre monitorovanie a analýzu sieťovej prevádzky.

5.3.1 Počítačová sieť s podporou OnePK

Základom pre nasadenie monitorovacej platformy OneMon, ktorej vytvorenie je predmetom tejto diplomovej práce, je počítačová sieť pozostávajúca zo zariadení od spoločnosti Cisco, podporujúcich technológiu Cisco OnePK. Pomocou týchto zariadení je možné vykonávať zber dát zo sieťovej komunikácie, ktoré sú zaujímavé z hľadiska monitorovania s cieľom sledovať komunikáciu užívateľov, monitorovať stav počítačovej siete a služieb, ktoré sú nad ňou prevádzkované, prípadne vykonať zmenu konfigurácie počítačovej siete na základe výsledkov analýzy zachytených dát.

5.3.2 Cisco OnePK vrstva

Je vrstva, ktorá zabezpečuje prepojenie hornej časti prostredia (nástroje pre monitorovanie a analýzu) s OnePK počítačovou sieťou. Pre aplikácie v hornej časti prostredia poskytuje táto vrstva rozhranie API. Cez toto rozhranie prebieha obojstranná komunikácia medzi sieťovými prvkami (smerovačmi) v sieti a aplikáciami pre monitoring a správu siete, súhrnne označenými názvom OneMon. Dôležitou súčasťou tejto vrstvy je DataPath Service Set (kapitola 3.2.1), ktorý má na starosti prenos dát sieťovej prevádzky a ich doručenie do monitorovacej platformy OneMon.



Obrázek 5.1: Architektúra prostredia pre monitoring a správu OnePK počítačových sietí.

5.3.3 Monitorovacie jadro platformy OneMon

Jadro platformy OneMon tvorí subsystém pre zber dát sieťovej prevádzky, zachytených pomocou Cisco OnePK. Tieto dáta sú doručené prostredníctvom DataPath Service Set-u špeciálnemu procesu DPSS_MP, bežiacemu na aplikačnom serveri vedľa aplikácie OneMon. Nástroj DPSS_MP je popísaný v kapitole 5.6.1. Doručujú sa však iba tie dáta, ktoré spadajú do sieťových tokov označených správcom siete ako zaujímavé pre ciele monitorovania. Zachytené dáta sú ukladané do vyrovnávacej pamäte - paketovej fronty, odkiaľ sú potom odoberané špecifickými analyzátormi na spracovanie. Po spracovaní analyzátorom ich jadro monitorovacej platformy korektne uvoľní z pamäti.

5.3.4 Nástroje pre analýzu

Pracujú nad zachytenými dátami z monitorovacieho subsystému. Sú špecifické pre rôzne požiadavky monitorovania - sledovanie objemu prenesených dát, monitorovanie obsahu prenesených dát (hlavičiek protokolov) a podobne. Analyzátor na svoj vstup dostáva dáta z paketovej fronty, ktorej veľkosť je limitovaná veľkosťou operačnej pamäte servera, na ktorom beží prostredie OneMon. Analyzátor vykonáva nad dátami výpočty v rámci svojich výpočtových možností a vytvára svoj vlastný model analýzy, ktorú implementuje. Podľa

tohto modelu analyzátor poskytuje výstupné dáta, ktoré predstavujú správcom požadovaný výstup z monitorovania sieťovej prevádzky na základe zachytených dát.

5.3.5 Vyrovnávacia pamäť pre zachytené pakety

Dáta, zachytené monitorovacím subsystémom, sú uložené v paketovej fronte. Z nej sú odoberané analyzátormi prevádzky. Real-time analýzou sa z dát extrahujú požadované údaje, ktoré sú použité analyzátormi pri ďalších výpočtoch. Paketová fronta je priebežne naplňovaná nástrojom pre monitorovanie, ktorý do nej vkladá dáta zo zachytenej sieťovej prevádzky. Prostredníctvom špeciálneho analyzátora, ktorého zdrojový kód tvorí Prílohu B tejto diplomovej práce, môžu byť dáta tiež ukladané do PCAP súborov pre neskoršiu off-line analýzu prevádzky.

5.3.6 Administrátorská konzola

Predstavuje stanicu, ktorá slúži pre zobrazenie výsledkov monitorovania. Administrátor bude mať prostredníctvom tejto konzoly možnosť manuálne zasahovať do fungovania siete a ovplyvňovať podstatné faktory s cieľom zvýšenia spoľahlivosti a zlepšenia fungovania siete, prípadne k dosiahnutiu zamýšľaného chovania. Umožňuje tiež nastaviť automatizované reakcie siete na vzniknuté situácie prostredníctvom spúšťačov a udalostí. Príkladom administrátorskej konzoly je grafická nadstavba WebAPI, popísaná v sekcii 8.1. Administračná konzola sa pripája k databázovému rozhraniu monitorovacej platformy, ktoré slúži ako API. V databázovej schéme vykonáva úpravy a tým špecifikuje ciele monitorovania (napríklad filtrovanie zaujímavých protokolov RTP, RTCP, SIP). Slúži aj ako nástroj pre zobrazovanie výstupov z analyzátora. Výstupy z analyzátora sú dostupné už vo finálnej podobe, nie je nad nimi potrebné vykonávať žiadne ďalšie operácie (pokiaľ užívateľ nezamýšľa inak). Môže sa teda jednať o grafické znázornenie sieťových parametrov alebo parametrov služby, ktorej sieťová prevádzka prechádza monitorovanou sieťou (napríklad monitorovanie kvality VoIP hovorov, sledovanie prenosu súborov po sieti, atď.). Platforma umožňuje súčasné pripojenie jednej alebo viacerých administračných konzolí. Konfliktom pri konfigurácii siete a vzájomné vylúčenie je zabezpečené v réžii databázovej vrstvy a použitým databázovým systémom MySQL. Systém je možné prevádzkovať aj bez pripojenej administrátorskej konzoly, pretože všetky potrebné nastavenia k monitorovaniu a riadeniu sú prezistentne uložené v databáze.

5.4 Konfigurácia Cisco zariadení pre podporu OneMon

Prvým predpokladom pre nasadenie monitorovacieho prostredia OneMon do siete je použitie sieťových zariadení od spoločnosti Cisco, ktoré podporujú technológiu Cisco One Platform Kit. V dobe písania tejto diplomovej práce boli podporované smerovače z rady ISR druhej generácie. Na týchto zariadeniach musí bežať verzia operačného systému IOS 15.4(2)T alebo novšia. Spoločnosť Cisco rozširuje podporu technológie OnePK postupne aj na ďalšie svoje produkty. Pred nasadením prostredia OneMon je vhodné overiť, či na monitorovacích zariadeniach je nainštalovaná najnovšia verzia operačného systému Cisco IOS a či je v ich operačnom systéme dostupná podpora technológie One Platform Kit.

Pre účely vývoja aplikácie v tejto diplomovej práci boli používane virtuálne smerovače s operačným systémom IOS, ktoré spoločnosť Cisco uvoľnila pod názvom *IOS on Unix*

(v skratke IOU). IOU je emulátor operačného systému Cisco IOS založený na operačnom systéme Unix. Je k dispozícii ako obraz operačného systému pre virtualizačný nástroj QEMU. IOU umožňuje vytvoriť vlastnú virtuálnu topológiu, zloženú zo sieťových zariadení, na ktorých beží operačný systém IOS od spoločnosti Cisco a prepojiť túto virtuálnu topológiu s fyzickým sieťovým rozhraním, čim sa vytvorí most medzi virtuálnou a fyzickou sieťovou topológiou.

5.4.1 Postup konfigurácie smerovačov

Skôr, než prostredie OneMon bude schopné monitorovať dáta prechádzajúce podporovaným smerovačom, je potrebné vykonať iniciálnu konfiguráciu smerovača, ktorá pozostáva z nastavenia certifikátov pre komunikáciu protokolom Cisco OnePK medzi smerovačom a prostredím OneMon. Ďalej je potrebné aktivovať funkcionalitu OnePK v operačnom systéme smerovača a nastaviť jeho parametre. Keďže k monitorovaniu dát a ich prenosu zo smerovača je využívaý DataPath Service Set, je taktiež potrebné zapnúť jeho podporu v sieťovom zariadení a nakonfigurovať parametre prenosu medzi sieťovým elementom a aplikáciou.

Nastavenie certifikátov pre komunikáciu protokolom OnePK

Monitorovacia aplikácia OneMon využíva pre komunikáciu so sieťovým prvkom protokol Cisco OnePK. Ten vyžaduje vygenerovanie certifikátov, potrebných pre zabezpečenie komunikácie a ich import do sieťového elementu. Najbežnejšou cestou je vygenerovanie certifikátov na serveri pomocou skriptu *createNEp12.sh*, ktorého použitie je uvedené na nasledujúcom príklade. Skript vyžaduje špecifikovanie parametrov príkazového riadku, ich význam je uvedený pod príkladom. Skript *createNEp12.sh* tvorí prílohu tejto diplomovej práce a nachádza sa na priloženom optickom médiu. Príklad spustenia skriptu *createNEp12.sh*:

```
./createNEp12.sh -cn <hostname> -ip <ip> -out <cert_fn>.p12 -pass <password>
```

-cn špecifikuje HOSTNAME sieťového elementu, na ktorom bude certifikát použitý

-ip IP adresa sieťového elementu, monitorovacia aplikácia OneMon bude prostredníctvom tejto IP adresy s elementom nadväzovať spojenie

-out cesta a názov súboru, kam bude certifikát vygenerovaný

-pass heslo pre zabezpečenie certifikátu, rovnaké musí byť zadané pri importe certifikátu do sieťového zariadenia

Po vygenerovaní certifikátu, resp. balíka s certifikátmi vo formáte PKCS12 skriptom *createNEp12.sh* je potrebné sprístupniť tieto certifikáty pomocou TFTP servera. Je nevyhnutné importovať certifikáty priamo z TFTP servera príkazom *crypto pki import*. Pri mojich pokusoch síce import z FLASH pamäte na prvý pohľad fungoval, aplikácia OneMon však nedokázala so sieťovým prvkom korektne nadviazať spojenie. Nasledujúci príklad demonštruje importovanie certifikátu vo formáte PKCS12 z TFTP servera, ktorý beží na IP adrese *192.168.0.1* a súbor s certifikátom na tomto TFTP serveri sa nazýva *Router.p12*. Certifikát je v tomto príklade pomenovaný *onepTP* a pri jeho vygenerovaní bol zabezpečený heslom *cisco*.

```
Router> enable
Router# configure terminal
Router(config)# crypto pki import onepTP
                pkcs12 tftp://192.168.0.1/Router.p12 password cisco
```

Týmto príkazom bol na zariadenie importovaný certifikát *onepTP*. Teraz je potrebné zapnúť podporu technológie Cisco One Platform Kit. To zabezpečí príkaz *onep* zadaný z globálneho konfiguračného režimu Cisco zariadenia. Príkaz presunie užívateľa do režimu konfigurácie OnePK na danom zariadení. V tomto režime sa priradí certifikát *onepTP*, importovaný v predchádzajúcom príklade do tohto zariadenia k bežiacej inštancii OnePK na zariadení.

```
Router> enable
Router# configure terminal
Router(config)# onep
Router(config-onep)# transport type tls localcert onepTP
                     disable-remotecert-validation
```

V globálnom konfiguračnom móde musia byť nastavené prihlasovacie údaje pre aplikáciu OneMon k zariadeniu a nastavené užívateľské oprávnenia.

```
Router> enable
Router# configure terminal
Router(config)# username onemon password OnEMoN
Router(config)# username onemon privilege 15
```

Posledná časť konfigurácie, ktorú je potrebné vykonať na zariadení je aktivácia Data-Path Service Set-u a jeho konfigurácia. Je potrebné zvoliť jedinečný číselný identifikátor zariadenia *sender-id* a špecifikovať fyzické rozhranie smerovača, z ktorého bude vytvorený GRE tunel na server, kde beží aplikácia OneMon.

```
Router> enable
Router# configure terminal
Router(config)# onep
Router(config-onep)# datapath transport gre sender-id 2
                     interface GigabitEthernet0/1
```

Pre platformu OneMon je kľúčová podpora pre rozpoznávanie aplikačných protokolov (NBAR) už v smerovači, aby nedochádzalo k jej zahlcovaniu nežiadúcimi dátami. Je preto nevyhnutné zapnúť podporu rozpoznávania aplikačných protokolov na vstupných rozhraniach smerovača, z ktorých bude zachytávaná komunikácia a odovzdávaná do monitorovacej platformy OneMon. Postup pre zapnutie funkcie NBAR bol popísaný v podkapitole 3.3.

Po tejto konfigurácii je zariadenie pripravené na pripojenie monitorovacej aplikácie OneMon.

5.5 Implementácia jadra monitorovacieho systému

Jadro monitorovacieho systému OneMon, ako bolo popísané na začiatku tejto kapitoly, sa skladá z funkcií pre konfiguráciu zachytávania, teda špecifikovanie zaujímavých tokov,

ktoré budú zaradené do sledovania. Druhou časťou jadra je subsystém, ktorý zbiera dáta zachytené a preposlané sieťovým zariadením do monitorovacej aplikácie OneMon.

Po preložení zdrojových súborov monitorovacieho systému OneMon vznikne spustiteľná aplikácia so staticky linkovanými knižnicami platformy Cisco OnePK a ďalšími bežne dostupnými knižnicami, ktoré vyžadujú moduly rozhrania OneMon.

5.5.1 Rozhranie pre konfiguráciu monitorovania

Konfigurácia monitorovania je proces, pri ktorom sa transformujú požiadavky správcu na monitorovanie konkrétnych sieťových tokov na konfiguráciu. Táto konfigurácia sa aplikuje prostredníctvom platformy Cisco OnePK na jednotlivých sieťových zariadeniach, ktoré monitorovanie podporujú a sú korektne nakonfigurované pre monitorovanie nástrojom OneMon. Sú to predovšetkým smerovače, najmä kvôli použitej technológii Cisco OnePK. Aplikácia je však schopná vykonávať monitorovanie aj z iných typov zariadení, ktoré podporujú OnePK a DataPath Service Set 3.2.1. Aplikovanie konfigurácie na smerovače je vykonávané prostredníctvom funkcií jadra OneMon, ktoré je nádstavbou nad rozhraním Cisco OnePK.

Monitorované dátové toky

Zaujímavé dáta z pohľadu monitorovania sú špecifikované prostredníctvom sieťových tokov. Monitorovaný sieťový tok je v tomto prípade množina paketov, ktorá prechádza sieťovým zariadením v časovom intervale, keď je monitorovanie aktívne a pakety, ktoré do tohto sieťového toku patria majú nejaké spoločné vlastnosti. Špecifikovaním týchto vlastností správca definuje monitorované sieťové toky, výsledkom čoho sú metadáta popisujúce sieťový tok. Medzi metadáta, ktorých definovanie podporuje monitorovacie prostredie OneMon patria predovšetkým:

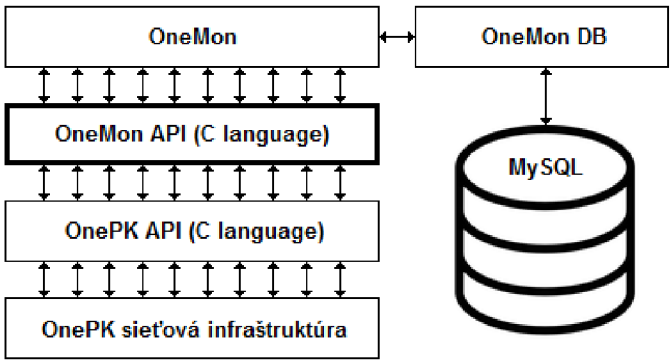
- Rozsah zdrojových a cieľových IP adries
- Zdrojové a cieľové číslo portu transportného protokolu (TCP/UDP)
- Protokoly linkovej a sieťovej vrstvy
- Protokol aplikačnej vrstvy

Takto vytvorené metadáta popisujúce sieťové toky, ktoré majú byť monitorované, je potrebné prostredníctvom platformy OnePK odovzdať smerovaču, ktorý bude zachytávanie realizovať. K tomu slúži vlastné API rozhranie platformy OneMon vyvinuté nad OnePK API. Nadstavbové OneMon API rozhranie bolo vytvorené kvôli zjednodušeniu a sprehľadneniu konfigurácie zariadení, pretože Cisco OnePK API svojou veľkou granularitou síce poskytuje mnoho možností, ale pre programátora je príliš neprehľadné a náchylné k zaneseniu chyby. Súčasťou nadstavbového OneMon API je tiež ošetrenie chybových stavov a neočakávaných parametrov, ktoré môžu byť spôsobené chybou na strane užívateľa - správcu, ktorý konfiguruje platformu OneMon, alebo chýb zanesených treťou stranou, komunikačným problémom a podobne.

OneMon API ako nadstavba nad Cisco OnePK SDK

Nadstavbové OneMon API je využívané jadrom monitorovacieho systému OneMon k nastavovaniu sieťových prvkov. Konfigurácia monitorovania, teda metadáta popísujúce monitorované sieťové toky, je uložená v databáze MySQL. Ku komunikácii medzi databázou

MySQL a platformou OneMon slúži databázová vrstva OneMon DB, ktorá odtieňuje špecifiká databázy MySQL pre prípad, že by bolo vhodné v budúcnosti zvoliť inú databázu alebo zdroj metadát o konfigurácii monitorovacieho systému OneMon. Táto hierarchia je pre úplnosť schematicky znázornená na obrázku 5.2.



Obrázek 5.2: Schéma nadstavbového rozhrania OneMon API nad rozhraním Cisco OnePK.

Konceptuálna architektúra systému OneMon

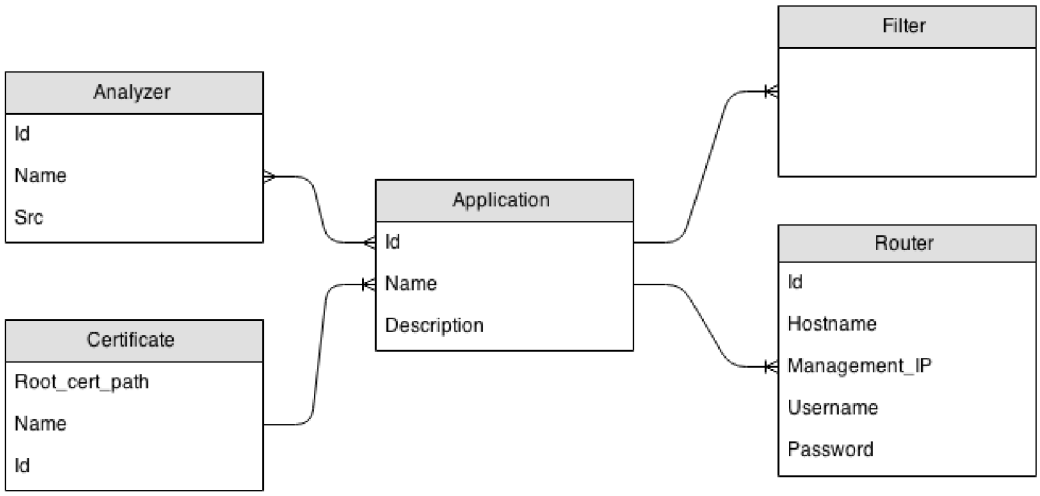
K metadátam o monitorovaných tokoch sa v databáze nachádza množstvo asociácií, ktoré z monitorovacej platformy OneMon robia komplexné prostredie schopné monitorovania viacerých aspektov v jednej alebo viacerých OnePK doménach.

Koncept monitorovacieho systému OneMon a jeho databázovej vrstvy prešiel od začiatku vývoja viacerými etapami. Schému databázy bolo potrebné niekoľkokrát transformovať v dôsledku neustáleho vývoja technológie Cisco OnePK, do ktorej pribúdala nová funkcionalita. Taktiež sa počas vývoja upresňovali prípady nasadenia tejto platformy, kedže na začiatku bolo ťažké odhadnúť jej schopnosti a výkonnosť. V niektorých prípadoch nebolo zrejmé ani to, či funkcie, ktoré sú zadokumentované v OnePK API, sú plne funkčné a bolo ich potrebné najskôr implementovať a dôkladne otestovať na funkčnej aplikácii. Táto etapa vývoja bola časovo najviac obtiažna, pretože k mnohým funkciám OnePK API nebolo okrem ich deklarácie a veľmi stručného popisu dostupné žiadne doplňujúce vysvetlenie ich funkčnosti ani príklady použitia.

Konečná podoba, ku ktorej prostredie OneMon dospelo s aktuálnou verziou rozhrania Cisco OnePK 1.3 bude predstavená v nasledujúcom texte. Dobrú predstavu o architektúre databázovej vrstvy OneMon, na ktorej je celé prostredie OneMon postavené, je možné urobiť si na základe ER diagramu na obrázkoch 5.3 a 5.4. Ten je zjednodušenou verziou reálnej databázovej schémy MySQL, nad ktorou OneMon pracuje.

Základným vstupným bodom do tohto ER diagramu je entita *Application*. Táto entita reprezentuje konkrétny prípad použitia, ktorý má jasne definovaný cieľ monitorovania a smerovače, na ktorých bude monitorovanie vykonávané. Príkladom je napríklad aplikácia pre analýzu SIP prevádzky, ktorá sleduje aktívne prebiehajúce hovory medzi účastníkmi.

Pri spustení monitorovacej platformy OneMon sa očakáva v iniciálnej konfigurácii nastavenie identifikátora aplikácie, ktorú má táto instancia platformy OneMon vykonávať (kapitola 5.6.2). K tejto aplikácii sú priradené smerovače, na ktorých správca požaduje monitorovanie prevádzky. Smerovače musia byť korektne nakonfigurované podľa kapitoly 5.4. U každého smerovača je potrebné poznať IP adresu, pod ktorou je dostupný pre aplikáciu



Obrázek 5.3: Zjednodušená podoba ER-diagramu databázovej vrstvy OneMon, výsek okolo entitnej množiny Application.

OneMon, užívateľské meno a heslo pre prihlásenie do privilegovaného režimu smerovača. K aplikácii súčasne musí existovať sada certifikátov, ktoré sú dôveryhodné pre ustanovenie zabezpečeného spojenia protokolu Cisco OnePK medzi platformou OneMon a smerovačmi.

K aplikácii platformy OneMon sú správcom priradené analyzátory prevádzky. Každý analyzátor je v skutočnosti modul naprogramovaný v jazyku C, vykonávajúci real-time spracovanie zachytených dát. Tento modul musí byť pred použitím správne preložený a prilinkovaný k platforme OneMon. O postupe pri vytváraní, registrácii a spôsobe použitia modulov s analyzátormi pojednáva sekcia 5.6.3.

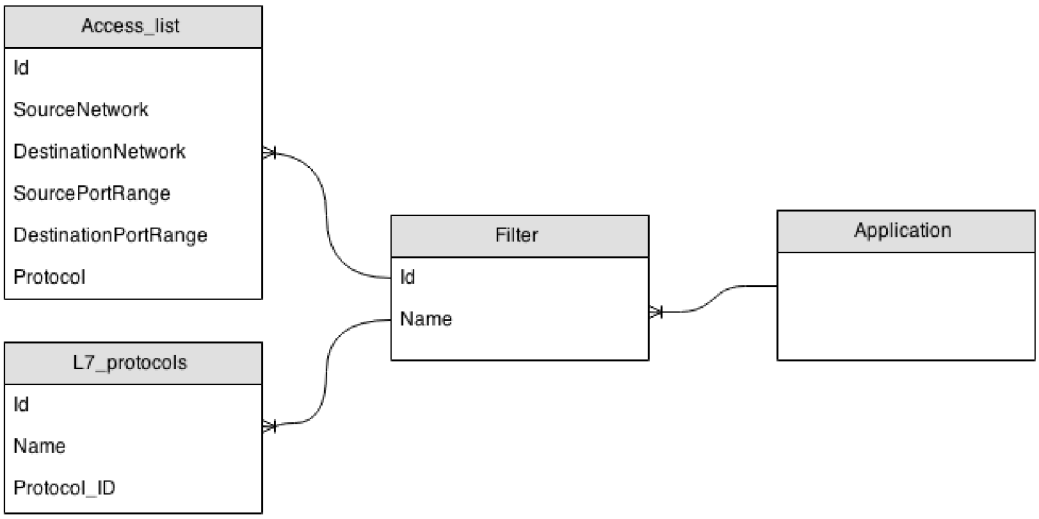
Na obrázku 5.4 je znázornený ER diagram okolo entitnej množiny *Filter*. Táto entitná množina predstavuje filtrovacie pravidlá, ktoré musia spĺňovať monitorované dáta. Ku každej aplikácii je možné vytvoriť nula až N filtrovacích pravidiel. Medzi filtrovacími pravidlami, priradenými k aplikácii pritom platí vzťah logické ALEBO (OR). Dáta sú teda zaujímavé z pohľadú monitorovania pre konkrétnu aplikáciu, ak úplne spĺňujú aspoň jedno filtrovacie pravidlo priradené k tejto aplikácii.

Filtrovacie pravidlá umožňujú definovať zdrojové a cieľové atribúty dát pomocou prístupových listov ACL. Filtrovanie je tiež možné vykonávať na základe protokolu aplikačnej vrstvy. Toto filtrovanie zabezpečuje funkcia NBAR na smerovačoch Cisco (kapitola 3.3).

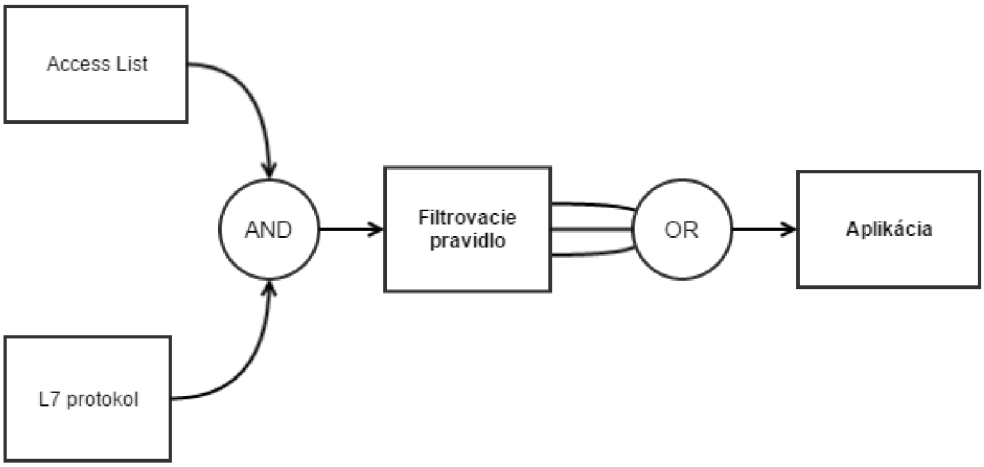
K jednému filtrovaciemu pravidlu je možné priradiť niekoľko prístupových listov a tiež špecifikovať niekoľko aplikačných protokolov. Dôležité je mať na pamäti, že všetky filtrovacie atribúty, priradené k jednému filtrovaciemu pravidlu sú vo vzťahu logické A ZÁROVEŇ (AND). Aby dáta splnili podmienky konkrétneho filtrovacieho pravidla, musia spĺňovať všetky atribúty tohto pravidla. Tieto vzťahy ilustruje obrázok 5.5.

5.5.2 Rozhranie pre zber zachytených dát

Monitorovacie prostredie OneMon po spustení nakonfiguruje sieťové zariadenia, ktoré zachytávajú prevádzku a odosielajú ju cez kanál Datapath do jej jadra. Jadro tieto prijaté pakety ukladá do paketovej fronty, ktorá je implementovaná ako zreťazený zoznam. Prostredníctvom OneMon API sú potom dáta z paketovej fronty dostupné pre analyzátory prevádzky, ktoré dáta z paketovej fronty čítajú. Jadro sa stará tiež o uvoľňovanie paketov



Obrázek 5.4: Zjednodušená podoba ER-diagramu databázovej vrstvy OneMon, výsek okolo entitnej množiny Filter.



Obrázek 5.5: Aby pakety vyhovovali pravidlám aplikácie, je nutné, aby existovalo aspoň jedno filtrovacie pravidlo, ktoré paket akceptuje. Aby paket vyhovoval filtrovaciemu pravidlu, je nutné splniť všetky atribúty tohto pravidla.

z paketovej fronty po ich spracovaní.

Princíp analyzátorov bude vysvetlený v podkapitole 5.6.3. Analyzátory spracovávajú pakety zhlukovo. Implementačne je spracovanie riešené cez volanie call-back funkcie analyzátora jadrom systému OneMon vždy, keď je v paketovej fronte dostupných N nespracovaných paketov. Parameter N je pritom možné nastaviť, postup pri voľbe tohto parametru bude vysvetlený tiež v podkapitole o analyzátoroch. Pri zavolaní call-back funkcie analyzátora jadrom OneMon, dostane analyzátor 2 parametre. Prvým je ukazateľ na paketovú frontu, resp. na prvý paket, ktorý má analyzátor spracovať, keďže paketová fronta je implementovaná ako lineárny zreťazený zoznam. Druhou položkou je záražka STOP. Analyzátor využíva pre získanie paketov z paketovej fronty funkciu OneMon API *GetNextItem()*, ktorej

predá ako parametre ukazateľ do paketovej fronty a zarážku STOP. Funkcia *GetNextItem()* vráti vždy ďalší nespracovaný paket z paketovej fronty, až pokiaľ nenarazí na zarážku STOP, kedy vráti hodnotu NULL. Ak analyzátor namiesto paketu obdrží hodnotu NULL, je to signál, že v paketovej fronte už nie sú ďalšie pakety na spracovanie pre aktuálnu instanciu volania call-back funkcie analyzátora. V paketovej fronte medzitým už môžu byť ďalšie pakety na spracovanie, ale tie už môže spracovávať iný analyzátor, alebo paralelne bežiaca ďalšia instancia tohto analyzátora, preto analyzátor nesmie pokračovať v spracovaní za zarážkou STOP.

5.6 Inštalácia, nastavenie a spustenie platformy OneMon

Platforma OneMon je rozsiahla aplikácia, ktorú je možné spustiť pod operačným systémom Linux. Odporúčaná distribúcia je Linux Mint 17 (Maya), na ktorej prebiehal aj vývoj platformy a jej testovanie. Platforma OneMon je distribuovaná vo forme zdrojových kódov v jazyku C. Pred inštaláciou a spustením platformy musia byť tieto zdrojové kódy preložené a korektne zlinkované. K tomu je potrebné mať nainštalované prostredie Cisco OnePK Software Development Kit vo verzii 1.3 alebo vyššej. Cisco OnePK SDK je možné voľne získať na stránkach spoločnosti Cisco určenej pre vývojárov venujúcich sa prostrediu Cisco OnePK [2].

5.6.1 Nástroj DPSS_MP

Súčasťou balíka Cisco OnePK SDK 1.3 je aj jednoduchý nástroj *DPSS_MP*, ktorý slúži ako prostredník medzi monitorovacou platformou OneMon a sieťovými zariadeniami Cisco k odovzdávaniu zachytených paketov. Táto aplikácia je do OnePK SDK zavedená pre podporu Service Setu Datapath, pre ktorý zatiaľ Cisco neimplementovalo podporu priamo do C - API.

Nástroj *DPSS_MP* je pred spustením potrebné nakonfigurovať vykonaním niekoľkých zmien v preddefinovanom konfiguračnom súbore, ktorý je s ním dodávaný. Detailne je postup konfigurácie objasnený na stránkach, venujúcich sa nástroju *DPSS_MP* [3].

Pre spustenie s monitorovacou platformou OneMon postačí, ak v konfiguračnom súbore nastavíme *LOCAL_IP* parameteru IP adresu rozhrania, na ktorom bude nástroj *DPSS_MP* načúvať a prijímať prichádzajúce pakety od smerovača cez OnePK Datapath Service Set. Druhým parametrom, ktorý je potrebné nastaviť je *SENDER_ID*. Tento parameter špecifikuje identifikátor zariadenia, ktorý bol nastavený na každom smerovači pri iniciálnej konfigurácii zariadení, ktorá bola popísaná v kapitole 5.4.1. Nástroj *DPSS_MP* by teda mal prijímať iba zachytené pakety od zariadenia so zhodným identifikátorom *SENDER_ID*. Pri testovaní sa však zistilo, že tento parameter nie je braný do úvahy a nástroj *DPSS_MP* zachytáva pakety od všetkých korektne nakonfigurovaných smerovačov v monitorovanej topológii bez ohľadu na nastavenú hodnotu parametra *SENDER_ID*. Pravdepodobne sa jedná o úmyselne spôsobenú chybu zo strany vývojárov platformy OnePK, ktorí zaviedli koncept *SENDER_ID*. Operačný systém však neumožní viacnásobné spustenie nástroja *DPSS_MP* pre rozličné hodnoty *SENDER_ID*, pretože ten sa asociuje k fyzickému sieťovému rozhraniu serveru. V tom prípade by však bolo možné zbierať dáta cez Datapath Service Set na jednom serveri iba z jedného smerovača, čo je obmedzujúce a výrazne by limitovalo široké možnosti použitia konceptu Datapath.

[2] https://developer.cisco.com/site/onepk/
[3] https://developer.cisco.com/media/onepk_c_tutorials/DatapathTutorial/DatapathConfiguration.html

5.6.2 Preklad a spustenie nástroja OneMon

Predpokladom pre úspešný preklad nástroja OneMon je nainštalované Cisco OnePK SDK 1.3 alebo vyššie. Zároveň je potrebné nastaviť cestu k rozhraniu OnePK API v jazyku C do premennej *ONEP_SDK* pre shell, v ktorom pracujeme a túto premennú urobiť viditeľnú pre všetky programy.

```
cisco@linux-mint17:~# export ONEP_SDK=/opt/cisco/onep/c64/sdk-c64-1.3.0.181/
```

Po nastavení korektnej cesty k OnePK API je možné aplikáciu OneMon preložiť, preklad trvá obvykle niekoľko desiatok sekúnd. Výsledkom je binárna spustiteľná aplikácia OneMon. Pred opätovným prekladom je vhodné vymazať všetky objektové súbory vytvorené predchádzajúcim prekladom, najmä pri zmenách v moduloch s analyzátormi.

```
cisco@linux-mint17:~# cd OneMon/src
cisco@linux-mint17:~# make clean && make
```

Iniciálna konfigurácia OneMon

Aplikácia OneMon potrebuje k svojmu spusteniu súbor s iniciálnou konfiguráciou, v ktorom sa nastavujú údaje pre pripojenie k databáze a identifikátor monitorovacej aplikácie, ktorú má správca v úmysle spustiť v novej instancii OneMon. Koncept aplikácií bol opísaný v predchádzajúcich kapitolách, ktoré sa venujú jadru prostredia OneMon.

Súbor s iniciálnou konfiguráciou OneMon je rozdelený na sekcie pre konfiguráciu databázy a aplikácie. V sekcii konfigurácie databázy sa nastaví typ databázy, nad ktorou bude pracovať databázová vrstva aplikácie. Tento parameter slúži k zvoleniu správneho databázového wrappera použitého pri komunikácii platformy OneMon s databázou. V súčasnosti je podporovaná iba databáza MySQL. Nasledujúce parametre slúžia pre pripojenie aplikácie OneMon k databáze. Parameter *schema* špecifikuje názov databázy, v ktorej je uložená schéma databázy využívanej systémom OneMon. Správna štruktúra konfiguračného súboru pre aplikáciu OneMon je nasledovná:

```
[DATABASE]
type=mysql
hostname=127.0.0.1
username=root
password=password
schema=onemon_db

[APPLICATION]
id=1
```

Výslednú preloženú aplikáciu OneMon je možné následne spustiť, vyžaduje jediný parameter príkazového riadku, ktorým je cesta k súboru s iniciálnou konfiguráciou platformy OneMon.

```
cisco@linux-mint17:~# ./OneMon config.dat
```

5.6.3 Analyzátory zachytenej prevádzky

V súlade s architektúrou systému OneMon je možné používať užívateľsky vytvorené analyzátory monitorovanej prevádzky. Tieto analyzátory vykonávajú nad zachytenými dátami výpočty a analýzy, ktorými sú plnené ciele monitorovania. Príkladom jednoduchého analyzátora je ukladanie všetkých zachytených dát do súboru PCAP. Takýto analyzátor je už súčasťou základnej inštalácie systému OneMon. Ďalším analyzátorom, ktorý je v systéme OneMon dostupný, je analyzátor SIP prevádzky. Tento analyzátor sleduje SIP komunikáciu medzi užívateľskými agentmi protokolu SIP a detekuje aktívne telefónne hovory. Potom sleduje pakety RTP, prenášajúce multimediálne dáta týchto telefónnych hovorov a určuje kvalitu telefónneho hovoru z pohľadu prenosu multimediálnych dát (jitter, stratovosť). Implementácii tohto analyzátora je venovaná nasledujúca kapitola 6.

Aj ked je analyzátor označovaný ako real-time, v skutočnosti spracováva dáta zhlukovo. Počet paketov, ktoré budú nárazovo spracované, je možné nastaviť v súbore so zdrojovým kódom `tahoe.c` v adresári projektu. Následne je potrebné opätovne preložiť celý projekt postupom popísaným v sekcii 5.6.2.

Správny počet paketov, spracovaných v jednom zhluku nie je možné vopred konštantne definovať. Ak sa tento parameter nastaví na hodnotu jedna, bude sa analyzátorom spracovávať samostatne každý jeden paket, zachytený jadrom monitorovacej platformy OneMon - pôjde teda o skutočne bezodkladné real-time spracovanie. To ale prinesie veľké spomalenie súvisiace s réžiou vloženia zarážky na koniec fronty paketov v jadre systému OneMon. Naopak, ak sa zvolí neúmerne veľká hodnota tohto parametru, môže to znamenať odklad spracovania niektorých dôležitých paketov. Voľba tejto hodnoty je individuálna a záleží od konkrétnych požiadaviek na monitorovanie a taktiež od množstva dát, ktoré monitorovanou sieťou prechádzajú. Ak sa parameter nastaví na veľkú hodnotu, pričom sieťou nejakú dobu nebude prechádzať dostatočný objem monitorovaných dát, môže dôjsť k javu označovanému ako vyhladovanie. Dáta, ktoré je potrebné spracovať analyzátorom, sú k dispozícii vo fronte, ale čaká sa za úplným naplnením fronty po zarážku.

Rozhranie pre tvorbu analyzátorov

V súlade s vyššie opísaným konceptom zhlukového spracovania paketov je analyzátor volaný vždy po naplnení fronty paketov po zarážku. Analyzátor dostane ako parameter ukazateľ na prvý paket vo fronte a zarážku. Volaním funkcie *GetNextItem* z OneMon API je možné postupne iterovať cez všetky nespracované pakety v paketovej fronte. Na nasledujúcom príklade je vidieť kostru analyzátora, ktorý vypisuje základné informácie, extrahované z paketov pomocou funkcie *print_packet_info(TPacket*)*:

```
// Call-back funkcia analyzátora PrintInfo
void PrintInfo(TQueueItem* start, TQueueItem* stop)
{
    TQueueItem* item = start;

    // spracovanie každého paketu vo fronte
    while(item != NULL)
```

```
    {
        // získanie kompletného payloadu paketu
        TPacket* packet = (TPacket*)item->packet;

        // volanie pomocnej funkcie, ktorá paket analyzuje
        print_packet_info(packet);

        // získanie nasledujúcej položky,
        // až pokiaľ cyklus nenarazí na zarážku stop
        item = GetNextItem(item, stop);
    }
}
```

Vytvorený analyzátor je ešte potrebné zaregistrovať v súbore `processing/modules/modules.c`. Registrácia spočíva v pridaní novej vetvy *else if(...)* na vyznačené miesto v súbore. Pre ilustráciu, kód, ktorý je nutné doplniť na vyznačené miesto v súbore `processing/modules/modules.c` pre registráciu analyzátora *PrintInfo*, je nasledovný:

```
...
else if(strcmp(name, "AnalyzerPrint") == 0)
{
    callback = PrintInfo;
}
...
```

Vyššie uvedený kód zabezpečí volanie call-back funkcie *PrintInfo* implementujúcej analyzátor. V databáze sa na tento analyzátor bude odkazovať identifikátorom *AnalyzerPrint*.

Posledným krokom je zahrnutie zdrojových súborov analyzátora do prekladu pridaním do súboru Makefile. Na záver je potrebný preklad aplikácie OneMon s novým analyzátorom a jeho opätovné spustenie.

Kapitola 6

Analyzátor VoIP prevádzky a kvality

Súčasťou tejto diplomovej práce je analyzátor, ktorý sa zameriava na monitorovanie VoIP komunikácie a jej kvality. Tento analyzátor má predovšetkým ukázať možnosti monitorovacej platformy OneMon a jej real-time analýzy, kedže platforma umožňuje aj ukladanie dát pre neskôršiu off-line analýzu pomocou nástrojov tretej strany.

Princípy používané vo VoIP telefónii sú vysvetlené v kapitole 4. K ustanoveniu telefónneho hovoru po IP paketovej sieti je potrebná súčasná participácia niekoľkých protokolov. V tejto diplomovej práci sa pri vývoji analyzátora kvality VoIP prevádzky zameriam na protokoly SIP a RTP. SIP je signalizačný protokol (kapitola 4.1), ktorý zaisťuje ustanovenie telefónneho spojenia, vyjednanie parametrov telefónneho spojenia aj k jeho korektnému ukončeniu. Protokol SIP však neprenáša žiadne multimediálne dáta. K tomu slúži protokol RTP, ktorého základné princípy boli predstavené v kapitole 4.4.

Analyzátor VoIP prevádzky a jej kvality, vyvinutý v rámci tejto diplomovej práce využíva neintruzívne metódy merania a hodnotenia kvality VoIP hovorov. To znamená, že analyzátor iba sleduje prechádzajúcu multimediálnu komunikáciu prenášanú protokolom RTP, ktorá je z pohľadu hodnotenia kvality telefónnych hovorov podstatná. Z paketov protokolu RTP analyzátor extrahuje potrebné informácie, predovšetkým časovú značku a sekvenčné číslo. Tento analyzátor sa zameriava iba na aspekty, ktoré vplývajú na kvalitu telefónneho hovoru z pohľadu prenosu jeho multimediálneho obsahu cez počítačovú sieť. Ostatné činitele, ktoré síce môžu mať podstatný vplyv na kvalitu hovoru, nie sú z pohľadu tohto analyzátora brané do úvahy.

Tento analyzátor pracuje v real-time režime (architektúra OneMon vysvetluje princíp tohto monitorovania v kapitole 5.6.3). To znamená, že analyzuje postupne všetky zaujímavé pakety prenášané protokolmi podielajúcimi sa na VoIP komunikácii v poradí, v akom boli do monitorovacieho prostredia OneMon doručené. Pri implementácii analyzátora je preto nutné brať ohľad aj na jeho výkonnosť, kedže výpočetné prostriedky počítača, na ktorom monitorovací nástroj OneMon beží, sú obmedzené. Ak by bolo spracovanie každého paketu príliš náročné, mohlo by sa stať, že nástroj OneMon by pri vyššom počte aktívnych monitorovaných telefónnych hovorov nezvládal real-time analýzou počítať parametre kvality. Pakety by sa hromadili v paketovej fronte nástroja OneMon, pokiaľ by nedošla voľná operačná pamäť. Potom by sa začali zahadzovať, čo by spôsobilo chybu analyzátora, ktorý je navrhnutý tak, aby analyzoval každý korektne doručený paket a zahrnul výsledky z jeho analýzy do celkového hodnotenia kvality VoIP komunikácie. Tento problém by bolo možné

do istej miery eliminovať vzorkovaním komunikácie. Pri tomto prístupe by sa spracovával iba každý N-tý paket dátového toku telefónneho hovoru. Mohlo by to viesť k približne N-násobnému zníženiu záťaže na analyzátor, avšak poskytnuté výsledky by mohli obsahovať veľkú chybu. Toto riešenie by bolo vhodné pre nasadenie vo veľkých systémoch a vyžadovalo by si použitie odlišných algoritmických prístupov, ako budú prezentované v tejto práci. Ďalšou možnosťou, ako zvýšiť výkonnosť je load-balancing medzi paralelne bežiacimi analyzátormi, kedže každý telefónny hovor je samostatne sledovaná jednotka a nie je potrebná náročná synchronizácia. O možnostiach vylepšenia sa píše v kapitole 8 zameranej na budúci vývoj monitorovacej platformy OneMon.

6.1 Princíp monitorovania hovorov a sledovania ich kvality

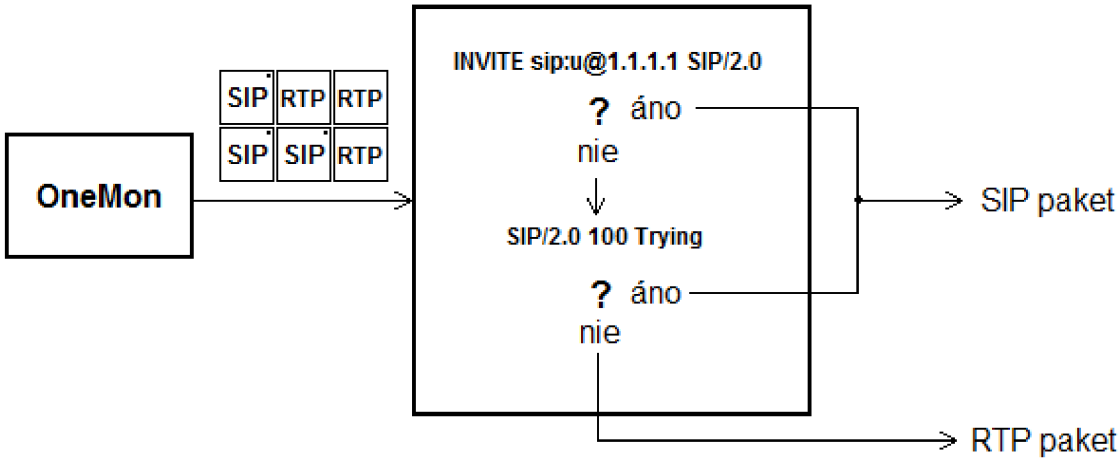
Implementovaný analyzátor VoIP kvality pracuje nad protokolmi SIP a RTP a postupne analyzuje každý takýto paket. Protokolom SIP sleduje analyzátor predovšetkým prebiehajúce telefónne hovory, teda od procesu ich ustanovenia, vyjednania parametrov spojenia, typu použitého kodeku až po ich ukončenie. Telefónne hovory, ktoré sú detekované zo sledovanej SIP komunikácie medzi užívateľskými agentmi komunikujúcich staníc, sú zaznamenávané v tabuľke *Call-Table*. Túto tabuľku si analyzátor interne udržuje a aktualizuje na základe prijatých SIP paketov. Princípy využité pri sledovaní telefónnych hovorov protokolom SIP sú popísané v sekcii 6.2.

Druhá časť analyzátora sa zameriava na pakety RTP, ktorými je prenášaný multimediálny obsah telefónnych hovorov, teda predovšetkým hlas. RTP pakety sú vysielané obvykle oboma komunikujúcimi stranami počas trvania telefónneho hovoru. Každý RTP paket obsahuje v hlavičkách nižších protokolov, ktorými je prenášaný, identifikátor zdrojovej a cieľovej užívateľskej stanice a čísla portov, na ktorých komunikácia prebieha. Tieto parametre je možné zistiť pri ustanovení telefónneho hovoru protokolom SIP a údaje o nich sú udržiavané v *Call-Table*.

Monitorovacia platforma OneMon doručuje do analyzátora VoIP prevádzky teda rovnako pakety SIP ako aj RTP. Prvým krokom, ktorý analyzátor vykoná pri analýze každého paketu, je zistenie, ktorému z týchto aplikačných protokolov daný paket patrí. K tomu analyzátor využíva najmä skutočnosť, že SIP je textový protokol. Analyzátor rozbalí paket na aplikačnú úroveň a pokúsi sa nahliadnuť na prvý riadok obsahu aplikačnej vrstvy paketu. Pri paketoch SIP môže byť tento paket typu *Request*, alebo typu *Response*. Prvý riadok pri týchto správach sa síce mierne odlišuje, ako vysvetľuje princíp žiadostí a odpovedí protokolu SIP v kapitole 4.1. Pre analyzátor však nájdenie riadku na aplikačnej vrstve paketu patriacemu protokolu SIP znamená jednoznačnú informáciu o jeho protokole. Ak paket na tomto riadku neobsahuje dáta, ktoré zodpovedajú špecifikácii SIP, analyzátor spracuje paket ako RTP. Toto riešenie sa javí byť rýchle a dostatočne kvalitné. Princíp je znázornený na obrázku 6.1.

6.2 Analýza paketov protokolu SIP

V prípade, že analyzátor detekuje spracovanie SIP paketu, filter z predchádzajúceho kroku vráti okrem protokolu SIP aj typ SIP správy. Prezradí teda, či sa jedná o žiadosť, alebo odpoveď, tak ako ich definuje štandard protokolu SIP. Z prijatého SIP paketu analyzátor extrahuje hlavičkové informácie, predovšetkým polia *From* a *To*, ktoré definujú URI adresy komunikujúcich užívateľských agentov a pole *Call-ID*. Táto trojica polí z hlavičky



Obrázek 6.1: Schéma filtra rozlišujúceho SIP a RTP pakety.

paketu jednoznačne identifikuje prebiehajúci dialóg protokolu SIP. Analyzátor SIP paketov udržiava všetky informácie o prebiehajúcich spojeniach v tabuľke Call-Table. Táto tabuľka umožňuje vyhľadávanie na základe pola *Call-ID*, ktoré je súčasťou hlavičky každého SIP paketu.

6.2.1 Tabuľka aktívnych hovorov - Call-Table

Call-Table je tabuľka, v ktorej VoIP analyzátor udržuje informácie o prebiehajúcich hovoroch detekovaných prostredníctvom zachytených paketov protokolu SIP. Call-Table taktiež obsahuje informácie o parametroch hovoru a o jeho priebehu z pohľadu monitorovania kvality. Každý záznam v Call-Table obsahuje:

Polia From a To z hlavičky protokolu SIP
: URI adresy užívateľských agentov participujúcich na telefónnom hovore.

Call-ID
: V spojení s identifikátormi zdroja a cieľa SIP komunikácie jedinečne popisuje dialóg medzi užívateľskými agentmi.

State
: Aktuálny stav, v ktorom sa hovor nachádza, podľa stavového automatu na obrázku 6.2 .

IP adresy zdrojovej a cieľovej stanice
: IP adresy, medzi ktorými prebieha prenos multimediálnych hlasových dát protokolom RTP.

Čísla portov protokolu RTP
: Porty, na ktorých prebieha prenos RTP paketov pomocou transportného protokolu UDP.

Kodek a dojednané parametre kódovania
: Obsahuje informáciu o použitom kodeku, ktoré môžu slúžiť pre výpočet kvality telefónneho hovoru. Obsahuje tiež informáciu o používanej vzorkovacej frekvencii kodeku.

Jitter
: Aktuálna hodnota iteratívne vypočítavaného jitteru po poslednom spracovanom pakete vzhľadom k prenosu RTP paketov.

Packet-loss
: Počet stratených paketov protokolu RTP.

Interné riadiace a stavové informácie
: Iné informácie, ktoré sú potrebné pre výpočty analyzátora a jeho činnosť, súvisiace s týmto telefónnym hovorom.

Tabuľka Call-Table je implementovaná pomocou jednosmerného lineárne zreťazeného zoznamu. Táto implementácia bola zvolená z dôvodu jednoduchosti a je postačujúca k monitorovaniu telefónnych hovorov v menších sieťach. Pre nasadenie vo veľkých topológiách by však mohla prispieť k zrýchleniu vyhľadávania v tejto tabuľke voľba inej abstraktnej dátovej štruktúry pre implementáciu Call-Table. Súčasná implementácia Call-Table s jednosmerným lineárnym zoznamom bola vylepšená tým, že informácie o nových dialógoch sa vkladajú vždy na začiatok tabuľky. Nemusia sa teda prehľadávať staré a ukončené dialógy, ktoré v Call-Table zostávajú aj po ukončení SIP dialógu medzi užívateľskými agentmi.

6.2.2 Informácie o prebiehajúcich spojeniach protokolu SIP

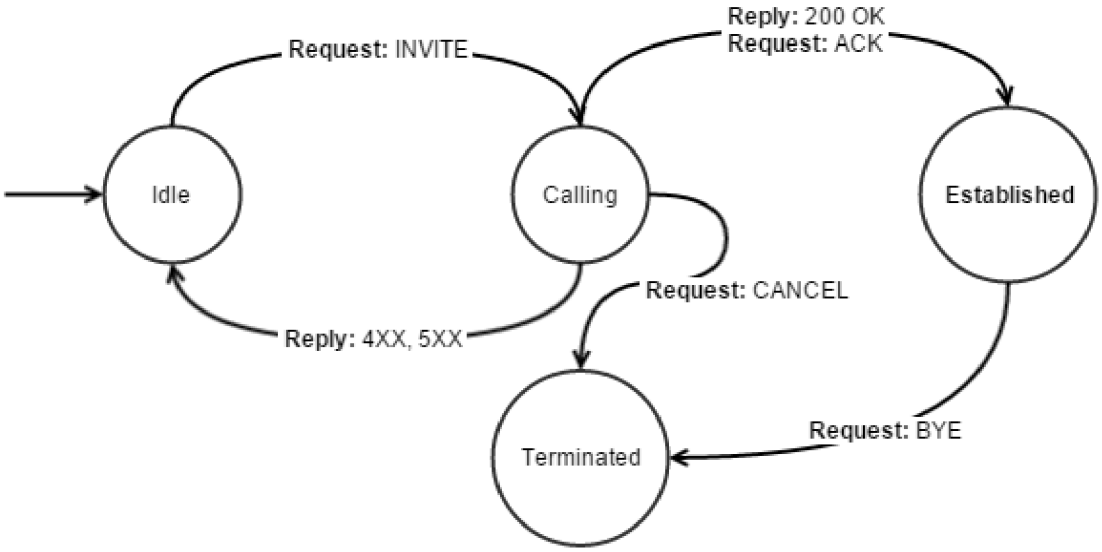
Po zistení typu SIP paketu a extrakcii základných informácií z jeho hlavičky analyzátor vyhľadá v Call-Table záznam zodpovedajúci dialógu, do ktorého paket patrí. Ak takýto záznam zatiaľ neexistuje, analyzátor vytvorí nový záznam v Call-Table s parametrami zodpovedajúcimi tomuto dialógu a nastaví jeho stav na hodnotu *Idle*.

V závislosti na type prijatého SIP paketu sa zmení stav prislúchajúceho záznamu v tabuľke Call-Table, čím si analyzátor udržuje stále aktuálne informácie o prebiehjúcich telefónnych hovoroch, ustanovených protokolom SIP. Prechody medzi stavmi telefónneho hovoru počas jeho trvania v závislosti na prijatých SIP správach je možné vidieť na stavovom diagrame 6.2. Tento stavový diagram je len zjednodušenou verziou a nepokrýva všetky možné stavy a prechody medzi nimi. Väčšina z nich však pre potreby tohto analyzátora nie je potrebná. V prípade, že analyzátor prijme SIP paket, ktorý nemá zásadný vplyv na sledovanie priebehu telefónneho hovoru analyzátorom, ignoruje tento paket.

Aktuálne informácie o prebiehajúcich hovoroch slúžia ako základ pre monitorovanie kvality týchto hovorov. V nasledujúcej kapitole 6.3 zaoberajúcej sa analýzou RTP paketov bude pri spracovaní každého RTP paketu nutné poznať parametre tohto spojenia, ktoré boli dojednané prostredníctvom protokolu SIP.

6.3 Analýza RTP paketov a určenie kvality telefónneho hovoru

Na IP telefóniu sú kladené mimoriadne vysoké nároky, pretože jej cieľom je nahradiť klasickú PSTN telefóniu. Tá umožňuje účastníkom komunikovať a počuť sa iba s minimálnym oneskorením v rádoch desiatok až stoviek milisekúnd. V IP telefónii je potrebné dosiahnuť minimálne rovnaké, ak nie ešte lepšie parametre odozvy a kvality hovoru ako pri klasickej PSTN technológii. IP telefónia sa však od PSTN odlišuje najmä v spôsobe prenosu hlasu, ktorý je fragmentovaný na časti zabalené prostredníctvom protokolu RTP a prenášané jednotlivo po paketových sieťach.



Obrázek 6.2: Zjednodušený stavový diagram používaný pri ustanovovaní a rušení telefónneho hovoru protokolom SIP.

Pakety RTP sú prenášané obvykle transportným protokolom UDP. Každý paket RTP nesie určitú, presne stanovenú dĺžku hlasovej informácie. Na základe sekvenčného čísla, ktoré je súčasťou každého RTP paketu (kapitola 4.4 pojednáva o RTP) sa na prijímajúcej strane telefónneho hovoru pakety zoradia do správneho poradia, čím vznikne súvislá zvuková stopa, ktorá je prehraná účastníkovi.

Pri prenose po IP paketových sieťach prúdia dáta RTP spolu so všetkými inými dátami po počítačovej sieti. Aby nedochádzalo k oneskoreniu, je nevyhnutné, aby boli pakety obsahujúce multimediálne informácie zabalené v protokole RTP prenesené s čo najkratším možným oneskorením. Pri prenose paketov cez počítačovú sieť však môže dochádzať k preťaženiu niektorých komunikačných liniek inými dátami a tým k pozdržaniu RTP paketov v sieti. Preto sa využívajú služby QoS na rôznych úrovniach, ktorými sú označované dáta, akými sú aj pakety RTP, vyžadujúce najrýchlejšie možné doručenie.

Aby však bolo možné služby QoS a prenos IP telefónie cez počítačovú sieť správne nastaviť, je potrebné vedieť zmerať kvalitu telefónnych hovorov, ktoré sieťou prechádzajú. Túto kvalitu je rovnako potrebné priebežne a pokiaľ možno nepretržite monitorovať, aby sa predišlo neočakávaným problémom napríklad pri zmenách sieťovej konfigurácie.

6.3.1 Výpočet stratovosti RTP paketov

Stratovosť RTP paketov (v angličtine nazývaná Packet-loss) je parameter, udávajúci pomer počtu stratených paketov k počtu očakávaných paketov. Protokol RTP je prenášaný prostredníctvom nespojovaného transportného protokolu UDP. Je to preto, že pre dáta, prenášané protokolom RTP je najdôležitejšie bezodkladné doručenie. RTP pakety obsahujú hlasové dáta zakódované určitým kodekom a väčšina kodekov sa do istej miery dokáže vysporiadať so stratovosťou. Ako bolo spomenuté v predchádzajúcom texte, každý RTP paket prenáša krátku vzorku hlasovej informácie. Ak sa stratí RTP paket, ktorý obsahuje napríklad 20 milisekúnd hlasovej informácie, kodek túto stratu vykompenzuje a táto strata je pre ľudské ucho takmer nepostrehnuteľná. Problém nastáva, pokiaľ sa stratí väčší počet

paketov nasledujúcich po sebe. Hovoríme o zhlukových stratách. Takéto straty je už veľmi ťažké vykompenzovať a skoro vždy majú vplyv na výslednú kvalitu telefónneho hovoru. Záleží taktiež od použitého kodeku, akú stratovosť pripúšťa, aby naďalej zostali zachované kvalitatívne parametre hlasovej komunikácie [20].

Výpočet parametru stratovosti L je realizovaný podľa nasledujúceho vzorca, kde N_{lost} je počet stratených paketov, $N_{expected}$ je počet očakávaných paketov a $N_{delivered}$ je počet korektne doručených paketov:

$$L = \frac{N_{lost}}{N_{expected}} = \frac{N_{expected} - N_{delivered}}{N_{expected}}$$

V prípade RTP komunikácie je počet doručených paketov možné jednoducho sledovať a inkrementovať čítač v Call-table príslušného telefónneho hovoru pre daný smer (keďže telefónny hovor je tvorený obvykle dvoma multimediálnymi reláciami RTP).

Počet očakávaných paketov je v RTP možné presne stanoviť na základe sekvenčných čísiel, ktoré sú uvedené v hlavičke každého RTP paketu. Sekvenčné číslo nasledujúceho RTP paketu je v korektnej postupnosti vždy o jedničku vyššie oproti predchádzajúcemu. K určeniu počtu očakávaných paketov je potrebné poznať sekvenčné číslo prvého RTP paketu v toku. Každý nový RTP paket inkrementuje sekvenčné číslo toku o hodnotu jedna. Rozdielom sekvenčného čísla posledného prijatého paketu a sekvenčného čísla prvého paketu v toku získa analyzátor počet očakávaných paketov, ktoré by mali byť pri nulovej stratovosti prijaté.

Ďalším zaujímavým parametrom pri sledovaní kvality VoIP sú zhlukové straty. Za zhlukovú stratu je považovaná strata dvoch a viacerých po sebe idúcich paketov z RTP toku. Triviálny spôsob indikácie, či došlo k zhlukovej strate, je sledovanie postupnosti sekvenčných čísiel RTP paketov po každom korektne prijatom pakete. Poradie RTP paketov v akom boli zachytené však nemusí tvoriť postupnosť, pretože RTP pakety môžu prísť do analyzátora prehádzané. Preto je vhodné výpočet zhlukovej straty oneskoriť o N paketov. Za N je vhodné zvoliť počet paketov, po ktorých doručení už paket predchádzajúci týchto N paketov bude považovaný za stratený.

Pre úplne korektný výpočet stratovosti na základe sledovania sekvenčných čisiel RTP paketov je nutné zaviesť sledovanie duplicitných sekvenčných čísiel. Môže sa stať, že v RTP toku sa niekoľko paketov stratí, avšak iný RTP paket s rovnakým sekvenčným číslom dorazí duplicitne niekoľkokrát. Ak nie sú sledované duplicitné sekvenčné čísla, spôsobí to chybu pri výpočte stratovosti RTP paketov. Taktiež je potrebné sledovať pretečenie sekvenčných čísiel v RTP toku, keďže pole so sekvenčným číslom má dĺžku iba 16 bitov.

6.3.2 Výpočet jitter-u

Hodnota jitter udáva rozdiel medzi očakávaným a skutočným príchodom paketu, ako bolo vysvetlené v kapitole 4.7.4. Hodnota jitteru je často prenášaná kontrolným protokolom RTCP v prípade, že RTCP je vysielané komunikujúcimi protistranami ako súčasť RTP komunikácie. Ak však nie sú k dispozícii pakety RTCP, je nutné hodnotu jitter spočítať analýzou zachytených RTP paketov. Použitá bude iteratívna metóda výpočtu hodnoty jitter, ktorá počíta túto hodnotu pre každý paket RTP toku na základe predchádzajúcich vypočítaných hodnôt.

K výpočtu jitteru je potrebné najskôr vedieť spočítať okamžitý rozptyl RTP paketov. Pre výpočet okamžitého rozptylu D dvoch RTP paketov v čase i je použitý nasledujúci vzorec:

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

R_X je skutočný nameraný čas, kedy bol RTP paket X zachytený a S_X je hodnota časového razítka z hlavičky RTP paketu (RTP timestamp). K výpočtu hodnoty jitter J pre prijatý paket i sa použije vzorec:

$$J_i = J_{i-1} + \frac{|D(i,i-1)| - J_{i-1}}{16}$$

Delenie hodnotou 16 pri výpočte jitteru pomáha redukovať šum zanesený do tohto výpočtu pri krátkodobej výraznej zmene okamžitého oneskorenia paketu. Zmena sa v takomto prípade prejaví až po prijatí niekoľkých paketov, keď sa hodnota jitter-u ustáli [18].

V nasledujúcej tabuľke 6.1 je na niekoľkých prijatých RTP paketoch ilustrovaný tento princíp výpočtu hodnoty jitter.

i	S_i	R_i	$D(i,i-1)$	$J(i)$
0	0	10	0	0
1	20	30	0	0
2	40	49	-1	0.0625
3	60	74	5	0.3711
4	80	90	-4	0.5979
5	100	111	1	0.6230
6	120	139	8	1.0841
7	140	150	-9	1.5788
8	160	170	0	1.4802
9	180	191	1	1.4501
10	200	210	-1	1.4220
11	220	229	-1	1.3956

Tabulka 6.1: Príklad iteratívneho výpočtu hodnoty jitter z prijatých RTP paketov.

6.3.3 Výpočet zjednodušeného E-Modelu

Znalosť hodnôt jitteru, stratovosti paketov a používaného kodeku, ktorým sú spracované hlasové dáta prenášané protokolom RTP, je možné využiť k výpočtu R-faktoru podľa zjednodušeného E-Modelu [17]. Tento zjednodušený model je špeciálne upravený tak, aby bolo možné spočítač hodnotu R-faktoru iba na základe prenosových charakteristík, ktoré môžu byť zmerané monitorovaním prechádzajúcich RTP paketov bez prítomnosti kontrolných paketov RTCP.

K výpočtu zjednodušeného modelu je však nutné poznať kodek, ktorým protistrany komunikujú a zahrnúť parametre tohto kodeku pri výpočte R-faktoru. Je nutné mať k dispozícii informácie o dojednaných parametroch RTP prenosu prostredníctvom protokolu SDP pred jeho začiatkom. Ak takéto informácie nie sú dostupné, napríklad ak je použitý neštandardný signalizačný protokol, alebo dáta signalizačného protokolu nie je možné z iného dôvodu odchytiť, je vhodné použiť techniky, ktoré dokážu pomerne presne odhadnúť vlastnosti používaného kodeku priamo z RTP streamu [16].

Kapitola 7

Testovanie a vyhodnotenie platformy OneMon

Táto kapitola sa bude venovať predovšetkým zhodnoteniu prínosu a výkonnosti monitorovacej platformy OneMon z hľadiska možností, ktoré poskytuje technológia Cisco OnePK, na ktorej je OneMon platforma vybudovaná. Pozornosť bude zameraná na priepustnosť medzi smerovačmi, ktoré sieťovú prevádzku zachytávajú a jadrom platformy OneMon. V závere sa nachádza zhodnotenie dosiahnutých výsledkov.

Na začiatku vývoja boli známe niektoré obmedzenia súvisiace s použitím technológie Cisco OnePK. Je to predovšetkým obmedzenie Datapath Service Set-u, ktorý je používaný pre prenos zachytených dát medzi smerovačmi, teda zariadeniami, na ktorých je sieťová prevádzka zachytávaná, a jadrom monitorovacej platformy OneMon. Dáta, ktoré prúdia do jadra platformy sú na strane smerovača zabalené do GRE tunelu. S týmto je spojenná určitá réžia, tak ako aj so samotným kopírovaním sieťovej prevádzky v smerovači. Priepustnosť dát odosielaných prostredníctvom Datapath by mala podľa spoločnosti Cisco predstavovať približne 25% výkonnosti sieťového rozhrania smerovača, cez ktoré sa export realizuje. Na priepustnosť by mohli mať vplyv aj ďalšie faktory, ktorými sú filtrácia paketov odosielaných prostredníctvom Datapath a tiež rozpoznávanie aplikačných protokolov funkciou NBAR (sekcia 3.3).

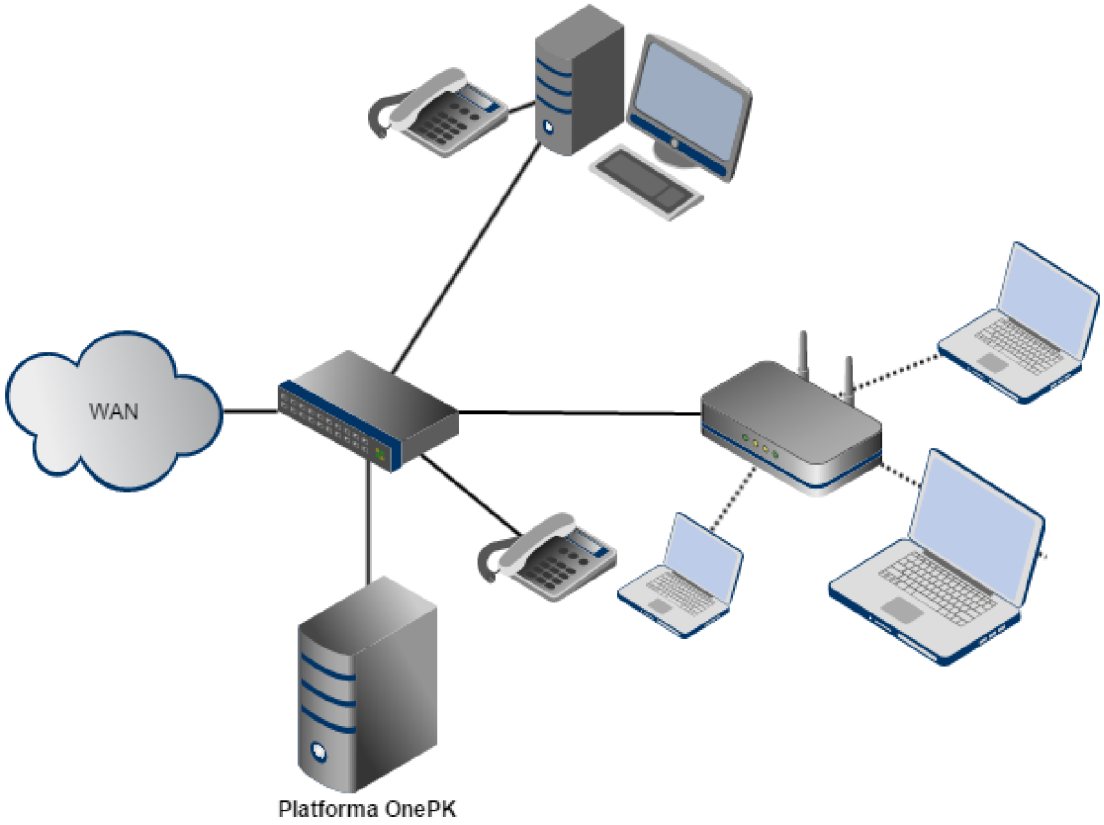
7.1 Zameranie platformy OneMon

Technológia Cisco OnePK nie je primárne určená na vývoj vysokorýchlostných systémov k práci s dátami. Používa sa skôr ako konfiguračná nadstavba nad sieťami zloženými zo sieťových zariadení Cisco. Napriek tomu vďaka Datapath Service Setu je možné zbierať určité rozumné množstvá zachytených dát a vykonávať nad nimi monitorovanie. Dátami vhodnými na monitorovanie môžu byť správy protokolov ARP a DHCP, ktorých analýzou sa dajú odhaliť útoky na linkovú a sieťovú vrstvu podvrhnutím falošných dát. Pri odhalení takéhoto správania by mohla riadiaca vrstva (spomenutá v kapitole o budúcnosti vývoja platformy OneMon 8) s využitím prostriedkov, ktoré poskytuje technológia Cisco OnePK napríklad vypnúť port, z ktorého podvrhnuté dáta prichádzajú. Ďalšou možnosťou využitia tohto monitorovacieho systému je sledovanie kvality služieb na sieti (napríklad telefónnych služieb VoIP). Včasnou detekciou zhoršenia kvality týchto služieb a zjednaním nápravy možno predísť úplnému zlyhaniu, nedostupnosti služby a nespokojnosti užívateľov.

Možnosť vývoja vlastných analyzátorov pre platformu OneMon poskytuje tiež priestor

na vývoj analyzátorov zabezpečujúcich účtovanie za služby prenášané po sieti.

Cieľom pri vývoji platformy OneMon bolo nahradiť nákladné a zložité riešenia monitorovania vyžadujúce zavedenie špeciálnych kolektorov pre menšie počítačové siete. Môžu to byť napríklad koncové podsiete s užívateľskými stanicami v menších organizáciách s jednotkami až desiatkami koncových sieťových zariadení (pracovných staníc, IP telefónov, sieťových tlačiarní). V prípade, že tieto siete už využívajú v hraničných bodoch smerovače Cisco s podporou technológie Cisco OnePK, nie je potrebné ani vynakladať ďalšie prostriedky na zariadenia zachytávajúce dáta zo sieťovej prevádzky.



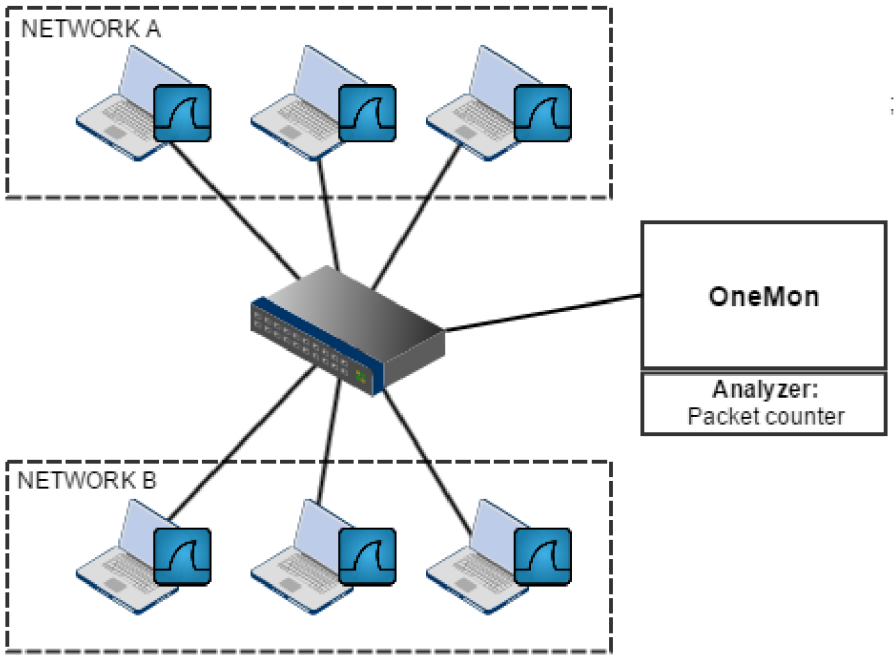
Obrázek 7.1: Príklad nasadenia monitorovacej platformy OneMon v koncovej podsieti.

7.2 Výkonnosť a spoľahlivost zachytávania dát sieťovej prevádzky

V tejto podkapitole budú predstavené dva zaujímavé testy, ktoré preverujú vlastnosti technológie Cisco OnePK a jej schopnosť doručovať pakety prostredníctvom Datapath Service Set-u do monitorovacej platformy OneMon. Prvý test v podkapitole 7.2.1 sleduje úspešnosť zachytávania komunikácie a jej doručovania do aplikácie OneMon. Druhý test bol zameraný na čas doručenia paketov do aplikácie OneMon. To hrá významnú úlohu pri niektorých druhoch monitorovania, napríklad aj pre monitorovanie kvality VoIP hovorov a určovanie parametrov akými je oneskorenie alebo jitter.

7.2.1 Stratovosť paketov doručených do platformy OneMon

Prvý test bol zameraný na zmeranie spoľahlivosti z pohľadu stratovosti paketov, teda či technológia Cisco OnePK je schopná zachytiť všetky pakety, ktoré zodpovedajú nastaveným filtrovacím pravidlám v súlade s požiadavkami správcu na monitorovanie. Test prebiehal nad virtuálnym routrom emulovaným prostredníctvom nástroja IOU. Schéma testovacej topológie je ilustrovaná na obrázku 7.2.



Obrázek 7.2: Testovacia topológia.

Pri teste boli nadväzované spojenia aplikačnými protokolmi HTTP, DNS, SIP a RTP medzi koncovými podsieťami *NETWORK A* a *NETWORK B* tak, aby komunikácia prechádzala smerovačom. Smerovač bol korektne nakonfigurovaný pre monitorovanie platformou OneMon. OneMon bol pripojený k smerovaču s aktívnym Datapath spojením pre doručovanie paketov zo smerovača do jadra systému OneMon. Bol vytvorený jednoduchý analyzátor, ktorého jediným cieľom bolo počítanie paketov doručených smerovačom do systému OneMon. Súčasne bola na všetkých virtuálnych staniciach, ktoré boli zaradené do testu spustená aplikácia Wireshark. Wireshark sledoval a filtroval ten istý typ paketov, ako aplikácia OneMon. Výsledky týchto testov sú zapísané v tabuľke 7.1. Stĺpec *Protokoly* špecifikuje aplikačné protokoly, ktorých pakety boli filtrované. V tabuľke je uvedená tiež približná dĺžka testu a počty paketov detekovaných nástrojom Wireshark a platformou OneMon.

Z výsledkov tohto testu vidno, že v niektorých variantoch bolo v skutočnosti medzi komunikujúcimi stranami prenesených viac paketov, ako smerovač doručil do monitorovacej aplikácie OneMon k analýze. Porovnaním zachytených paketov programom Wireshark a paketov, doručených do platformy OneMon sa zistilo, že pakety, ktoré neboli doručené do platformy OneMon boli prenášané transportným protokolom UDP. Nedoručenie týchto paketov mohlo byť spôsobené nesprávnou detekciou aplikačného protokolu funkciou NBAR, alebo inými dôvodmi, pre ktoré Cisco OnePK tieto pakety nedoručilo do OneMon prostredníctvom Datapath. Tieto pakety však boli korektne doručené cieľovému zariadeniu, čo sa

Protokoly	Dĺžka testu	Wireshark	OneMon	Straty
HTTP, DNS	1 minúta	1172	1172	0 paketov
HTTP, DNS	15 minút	7884	7881	3 pakety
HTTP, DNS	60 minút	19112	19112	0 paketov
SIP	krátky telefónny hovor	17	17	0 paketov
SIP	3 krátke telefónne hovory	41	41	0 paketov
RTP	telefónny hovor(60 sec)	6174	6173	1 paket
RTP	2 súbežné telefónne hovory(60 sec)	12811	12791	20 paketov
SIP, RTP	2 súbežné telefónne hovory(60 sec)	12902	12899	3 pakety

Tabulka 7.1: Tabuľka s výsledkami testovania stratovosti paketov zachytávanych platformou OneMon.

zistilo porovnaním zachytených výstupov programu Wireshark na obidvoch komunikujúcich protistrán.

7.2.2 Časovanie paketov doručených do aplikácie OneMon

Druhý test bol realizovaný s ohľadom na monitorovanie kvality VoIP telefónnych hovorov. Analýza kvality VoIP z hľadiska prenosu hlasových dát cez počítačovú sieť je realizovaná zo zachytených RTP paketov, ktoré sú nositeľmi hlasovej informácie. RTP pakety obsahujú vo svojej hlavičke iba informáciu s časovou značkou, ktorá udáva pozíciu prenášaného vzorku v multimediálnej sekvencii. Pre výpočet kvality prenosu je potrebné poznať čas, kedy RTP paket prišiel na vstupné rozhranie smerovača zachytávajúceho komunikáciu, aby bolo možné spočítať kvalitatívne parametre tohto spojenia.

Takúto informáciu však prostredie OnePK vo svojom API neposkytuje. Bolo teda potrebné implementovať vlastný systém časových značiek, ktorými sú pakety označované po doručení do jadra monitorovacej platformy OneMon. Medzi časom doručenia paketu smerovaču a časom doručenia paketu do monitorovacej platformy OneMon vznikne určitá časová odchýlka. Cieľom tohto testu je ukázať, či je táto odchýlka aspoň približná konštantnej hodnote, alebo je premenlivá, čo môže byť spôsobené napríklad nepravidelným oneskorením paketov vo výstupných frontách smerovača pred odoslaním prostredníctvom Datapath.

Odchýlka doručenia každého paketu bola vypočítaná podľa vzorca

$$D = |T_{Wireshark} - T_{OneMon}|$$

D je odchýlka časov prijatia paketu medzi programom Wireshark a platformou OneMon, $T_{Wireshark}$ je časová značka, kedy bol paket prijatý programom Wireshark a T_{Onemon} je časová značka prijatia paketu do jadra monitorovacej platformy OneMon.

Testovanie prebehlo na množine RTP paketov, ktoré boli zachytávané programom Wireshark a rovnako tak doručované do monitorovacej platformy OneMon. V grafe na obrázku 7.3 sú vynesené odchýlky časov prijatia každého paketu programom Wireshark a monitorovacou platformou OneMon.

Z grafu je možné vidieť, že odchýlka, teda rozdiel časových značiek prijatia programom Wireshark a platformou OneMon, sa pohybuje v rozmedzí od 30 do 36 milisekúnd. Časové značky prijatia paketov boli zaznamenávané vzhľadom k prvému paketu v sekvencii. Prvý paket bol teda označený časovou značkou 0 v programe Wireshark aj v monitorovacej platforme OneMon. Spodná hranica tejto odchýlky 30 milisekúnd teda s najväčšou prav-

Obrázek 7.3: Odchýlka časových značiek paketov zachytených programom Wireshark a monitorovacou platformou OneMon.

depodobnosťou vyjadruje čas potrebný na klasifikáciu paketu na smerovači prostriedkami Cisco OnePK, NBAR a jeho prenos cez Datapath do monitorovacej platformy OneMon.

Z tohto merania je možné vidieť aj maximálnu odchýlku, s ktorou je nutné počítať pri využívaní monitorovacej platformy OneMon. Túto odchýlku možno spočítať ako absolútnu hodnotu medzi najvyššou a najnižšou nameranou odchýlkou sledovaných paketov, teda $T_{dif} = |36ms - 30ms| = 6ms$.

Pri monitorovaní platformou OneMon je teda vhodné brať do úvahy možnú odchýlku 6 milisekúnd medzi časom príchodu paketu na vstupné rozhranie smerovača a časom zachyteným monitorovacou platformou OneMon. Tiež je nutné počítať s oneskorením 30 milisekúnd medzi preposlaním zachyteného paketu zo smerovača a prijatím do monitorovacej platformy OneMon.

7.3 Testovanie analyzátora VoIP

Analyzátor implementovaný v tejto práci primárne sleduje telefónne spojenia dojednané protokolom SIP (sekcia 6.2). Udržiava pritom informácie o týchto SIP spojeniach a ich stave. Problém nastane vtedy, keď dôjde k strate signalizačného SIP paketu, informujúceho napríklad o ukončení hovoru (Request: BYE). Analyzátor nemá informáciu o tom, že hovor skončil a stále ho považuje za prebiehajúci, pričom sa snaží zachytávať prislúchajúce RTP pakety a analyzovať komunikáciu. Keďže došlo k ukončeniu hovoru, RTP tok už neprebieha, čo môže viesť k chybným výpočtom kvality a môže byť interpretované ako jej zhoršenie. Mohlo by dôjsť napríklad k aplikovaniu bezpečnostnej politiky na takéto falošné zhoršenie kvality VoIP. Je preto nutné počítať s možnou stratou paketov, ktoré do analyzátora nedorazia. V tomto konkrétnom prípade by bolo možné túto situáciu ošetriť napríklad zavedením časovačov [15], ktoré by označili prebiehajúci SIP hovor za neaktívny, ak by nejakú dobu nedorazili žiadne pakety, súvisiace s týmto telefónnym hovorom.

Kapitola 8

Budúci vývoj a možnosti rozširovania

Monitorovacia platforma OneMon bola od svojho začiatku vyvíjaná s dôrazom na všestrannosť použitia a rozšíriteľnosť. Tieto kritériá sú dosahované vytvorením rozhrania OneMon API, ktoré umožňuje vývoj a registráciu ľubovolných real-time analyzátorov dátovej prevádzky prechádzajúcej monitorovaným segmentom počítačovej siete. Rovnako tak zvyšuje flexibilitu tejto platformy zavedenie databázovej vrstvy. Databázová vrstva, ktorá uchováva konfiguračné informácie prostredia OneMon, informácie o jeho behu a tiež slúži potrebám analyzátorov prevádzky, bola zavedená do platformy OneMon kvôli budúcemu rozvoju platformy a plánovanej grafickej nádstavbe WebAPI.

8.1 Grafické užívateľské prostredie OneMon - WebAPI

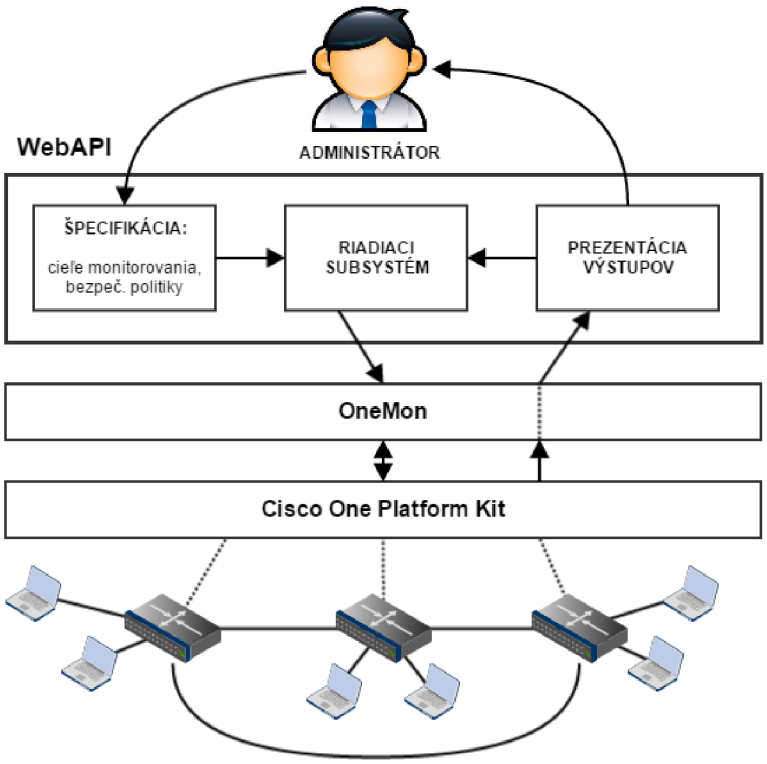
Nad platformou OneMon, ktorá je schopná v súčasnosti bežať ako démon v prostredí operačného systému Linux, vzniká nadstavbové grafické užívateľské rozhranie WebAPI, ktoré umožní pracovať s platformou OneMon aj užívateľovi bez znalosti technického pozadia platformy OneMon.

Súčasná verzia monitorovacej platformy OneMon je otestovaná a plne funkčná pri nasadení v menšej sieťovej topológii. Vyžaduje však zvýšené úsilie a dobré znalosti platformy a jej architektúry pre konfiguráciu a nastavenie nového monitorovania, vývoj analyzátorov a využitie všetkých prostriedkov technológie Cisco OnePK pre správu siete. Pre bežného správcu siete, ktorý by potreboval nasadiť platformu OneMon by to znamenalo vynaloženie väčšieho úsilia, kým by sa s platformou oboznámil.

Nová grafická nadstavba WebAPI by mala správcovi okrem jednoduchej špecifikácie cieľov monitorovania poskytnúť aj možnosť definovať automatické akcie, ktoré sa majú na sieť aplikovať v zavislosti na získaných výstupoch z analýzy zachytených dát. Koncept funkcií, ktoré bude WebAPI poskytovať, je znázornený obrázkom 8.1.

8.2 Riadenie sieťovej infraštruktúry

Prostredie OneMon neslúži iba pre monitorovanie sieťovej prevádzky, ale tiež pre jej riadenie. Prostredníctvom technológie Cisco OnePK, na ktorej je platforma OneMon vybudovaná, je možné priamo meniť konfiguráciu sieťových smerovačov. Na základe výsledkov



Obrázek 8.1: Základné ciele WebAPI a ich vzájomné prepojenie.

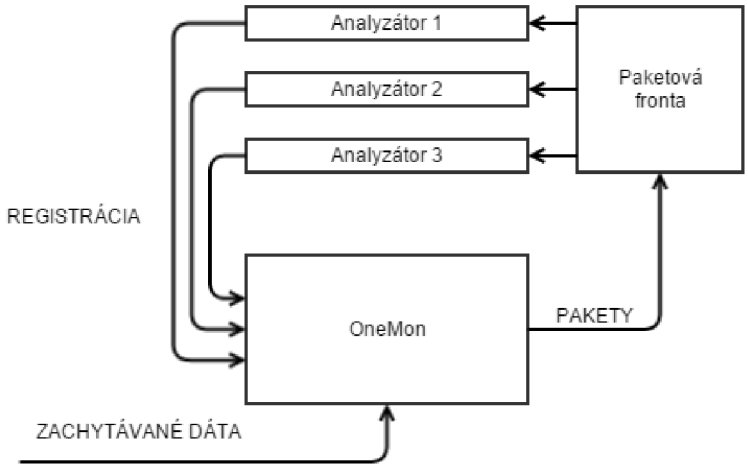
analýzy môžu byť sieťové zariadenia prekonfigurované. Je tu priestor aj na tvorbu a aplikáciu sieťových a bezpečnostných politík. Inšpiráciu pre spôsob ich definovania je možné nájsť napríklad v práci [2], kde bol definovaný jazyk OF-NCL pre riadenie softvérovo definovaných sietí založených na architektúre OpenFlow. Rovnaké princípy sú však uplatniteľné aj v architektúre Cisco OnePK. Sledovanie činnosti užívateľov a detekcia anomálií môžu v kombinácii s vhodne nastavenou bezpečnostnou politikou pomôcť minimalizovať škody spôsobené únikom citlivých informácií alebo šírením spamu. Platforma OneMon je vyvinutá tak, aby mohla fungovať v režime podobnom systémom IDS[1] a IPS[2].

8.3 Súčasný beh viacerých analyzátorov

Jedným z obmedzení súčasnej verzie platformy OneMon je možnosť používať iba jeden analyzátor nad zachytenými dátami. Toto obmedzenie vzniklo z dôvodu sústredenia sa na vývoj jadra platformy OneMon a jej prepojenie s technológiou Cisco OnePK. Pri vývoji jadra však bola zakomponovaná aj požiadavka na beh viacerých analyzátorov paralelne. Použitie viacerých analyzátorov súčasne sa môže na prvý pohľad javiť triviálne, ale je potrebné klásť dôraz na zachovanie synchronizácie pri odoberaní paketov jednotlivými analyzátormi z paketovej fronty a súčasne neznižovať rýchlosť pomalými synchronizačnými prostriedkami.

[1] IDS (Intrusion Detection Systems) sú systémy, ktoré monitorujú sieťovú prevádzku s cieľom odhaliť podozrivé aktivity.

[2] IPS (Intrusion Prevention Systems) sú systémy, ktoré sledujú sieťovú prevádzku a pri zistení škodlivej činnosti podniknú kroky k jej potlačeniu v súlade so sieťovou bezpečnostnou politikou.



Obrázek 8.2: Koncept registrácie analyzátorov k odberu dát monitorovacou platformou OneMon.

Tiež je nutné mať na pamäti, že dáta zo sieťovej prevádzky sú po zachytení ukladané iba do jednej paketovej fronty, ktorá je spoločná pre všetky analyzátory. Je potrebné, aby tieto dáta boli dostupné v paketovej fronte, až pokiaľ ich nespracujú všetky analyzátory. K tomuto účelu bol zavedený princíp registrácie analyzátorov, ilustrovaný obrázkom 8.2 , ktoré majú záujem v spustenej instancii platformy OneMon dostávať zachytené dáta zo sieťovej prevádzky. Analyzátory používajú funkciu z OneMon API *GetNextItem()*, ktorá im doručí nasledujúci paket z paketovej fronty. Všetky analyzátory bežiace v samostatných vláknach takto postupne spracovávajú dáta z fronty paketov. Fronta paketov uvoľní paket až potom, čo každý zaregistrovaný analyzátor signalizuje spracovanie tohto paketu.

8.4 Analýza v clustroch

Pre monitorovacie prostredie OneMon sú vhodné najmä rýchle analyzátory, ktoré extrahujú informácie zo zachytených dát a vykonávajú nad nimi časovo nenáročné výpočty. V prípade, že analýza dát dosahuje vyššiu časovú zložitosť ako dovoľujú vypočetné možnosti servera, na ktorom OneMon beží, je možné spracovanie dát prerozdeliť medzi ďalšie servery špeciálne určené pre zložitejšie analýzy a vytvoriť tak cluster. Príkladom by mohlo byť vyhľadávanie vzorov v zachytenej VoIP komunikácii, automatická identifikácia účastníka telefónneho hovoru na základe charakteristík jeho hlasu [14] a podobne.

Rozdelenie do clustrov je vhodné najmä pre analýzy, ktoré nepotrebujú veľké množstvo vstupných dát, ale majú vyššie časové, pamäťové alebo výkonnostné nároky, ako je prípustné pre spracovanie v rámci analyzátora v aplikácii OneMon. Odovzdávanie dát do iných aplikácií, ktoré bežia v tomto clustri, je vhodné realizovať v réžii analyzátora, zavedeného do aplikácie OneMon za týmto účelom. Tento analyzátor bude zabezpečovať prerozdeľovanie vstupných dát medzi výpočetné jednotky v clustri, zber výsledkov, ich sumarizáciu a prezentáciu správcovi siete alebo vykonanie preddefinovaných akcií na základe týchto výsledkov. Pri takomto riešení by bolo vhodné použiť niektoré z existujúcich frameworkov pre paralelné spracovanie a výpočty v clustroch, napríklad Open-MPI [3].

[3] http://www.open-mpi.org/

Kapitola 9

Záver

V diplomovej práci boli preskúmané možnosti softvérovo definovaného monitorovania na platforme Cisco OnePK. Sú tu predstavené princípy fungovania softvérovo definovaných sietí a architektúra Cisco OnePK. Pozornosť bola zameraná najmä na monitorovanie kvality VoIP hovorov. Popísané boli základné techniky využívané v súčasných sieťach VoIP aj princípy hodnotenia kvality, používané v IP telefónii. Hlavnou časťou tejto diplomovej práce bol návrh architektúry monitorovacieho prostredia OneMon a jeho implementácia predstavená v kapitole 5. Prostredie OneMon bolo implementované nad technológiou spoločnosti Cisco pre softvérovo definované siete Cisco One Platform Kit.

Pri implementácii monitorovacej platformy OneMon boli kladené požiadavky na maximálne využitie všetkých možností, ktoré ponúka súčasná verzia Cisco OnePK. Kľúčovým pri implementácii bolo tiež použitie funkcie NBAR 3.3, ktorá umožní filtrovanie dát na základe aplikačných protokolov už na sieťových zariadeniach zachytávajúcich monitorované dáta určené k analýze.

Platforma OneMon bola vyvíjaná s dôrazom na budúcu rozšíriteľnosť. Tento cieľ bol dosiahnutý zavedením konceptu analyzátorov, ktoré môžu byť zavedené do platformy OneMon. K tomu slúži rozhranie pre vývoj a registráciu analyzátorov popísaných v kapitole 5.6.3. V Kapitole 8 zameranej na budúci vývoj platformy sú spomenuté aj princípy použitia paralelných analyzátorov, ich registrácie do prostredia OneMon a možnosti zefektívnenia činnosti analyzátorov v prostredí OneMon pre náročnejšie typy analýz vykonávaných nad zachytenou sieťovou komunikáciou. Tieto princípy budú implementované v nasledujúcich iteráciách vývoja prostredia OneMon.

Najväčšou výzvou pri vývoji monitorovacieho prostredia OneMon bolo oboznámenie sa s funkciami, ktoré ponúka vývojové prostredie Cisco OnePK SDK a ich správne použitie. V čase, keď sa s vývojom platformy OneMon začínalo, neboli k dispozícii takmer žiadne komplexnejšie materiály vysvetľujúce pokročilejšie funkcie prostredia Cisco OnePK. Funkcie boli často testované iba metódou pokus-omyl. Dochádzalo aj k zlyhaniu operačného systému Cisco IOS smerovačov, na ktorých bolo testované prostredie OnePK. Problémy s funkciou NBAR v operačnom systéme Cisco IOS boli vyriešené až príchodom novej verzie tohto operačného systému. Funkcia NBAR predtým nefungovala správne a pri požiadavke na filtrovanie konkrétneho aplikačného protokolu nechodili do platformy OneMon žiadne dáta. Ak by nebolo možné využiť funkciu NBAR, filtrovanie na základe aplikačných protokolov by musela vykonávať až platforma OneMon, do ktorej by prúdili všetky dáta prechádzajúce smerovačom a filtrované iba na základe adries a čísiel portov. To by výrazne zvýšilo záťaž najmä na analyzátory platformy OneMon, ktoré by museli odbúravať množstvo dát iných aplikačných protokolov.

Pre monitorovaciu platformu OneMon bol implementovaný analyzátor VoIP prevádzky. Tento analyzátor spracováva pakety signalizačného protokolu SIP a sleduje k nim prislúchajúce multimediálne spojenia prenášané protokolom RTP. Funkčnosť tohto analyzátora bola otestovaná v malej sieti s niekoľkými vzájomne komunikujúcimi softvérovými telefónmi. Detekcia prebiehajúcich spojení funguje spoľahlivo, je však závislá na korektnom doručení všetkých signalizačných paketov SIP do analyzátora. Strata niektorého z nich by pre analyzátor znamenala nepresnú informáciu o SIP relácii. Problém môže nastať aj v prípade rozpadnutia hovoru z dôvodu nekorektného odpojenia jedného z účastníkov a následnej chybnej reakcii stavového automatu sledujúceho priebeh SIP komunikácie medzi užívateľskými agentmi. Riešením týchto situácií je súbežné sledovanie paketov RTP prenášajúcich hlasové informácie k reláciám SIP.

V predposlednej kapitole tejto diplomovej práce sú prezentované myšlienky budúceho rozširovania platformy OneMon. Monitorovacia platforma OneMon má svoje uplatnenie pre nasadenie v menších sieťach, alebo pre monitorovanie primeraného množstva dát, ktoré bude platforma OneMon a jej analyzátory schopné spracovať s ohľadom na výpočetné možnosti servera na ktorom bude nasadená. Jej výhodou je, že pre nasadenie v sieťach postavených na zariadeniach spoločnosti Cisco podporujúcich technológiu OnePK nevyžaduje okrem servera (na ktorom bude spustená) žiadne dodatočné nákladné investície.

Pre budúci rozvoj je kľúčovým najmä vývoj grafickej nadstavby WebAPI (predstavená v podkapitole 8.1) platformy OneMon, ktorá poskytne užívateľsky prívetivú možnosť pre konfiguráciu monitorovania, prezentáciu jej výsledkov a krátkodobých aj dlhodobých analýz nad monitorovanými dátami. Významný prínos bude mať aj možnosť konfigurovať bezpečnostné politiky na vzniknuté situácie zistené monitorovaním a analýzou siete. To umožní zvýšiť flexibilitu v počítačových sieťach v súlade s konceptom softvérovo definovaných sietí.

Na vývoji monitorovacieho prostredia OneMon pracujem v rámci projektu "Moderní prostředky pro boj s kybernetickou kriminalitou na Internetu nové generace"vo výskumnej skupine Sec6Net na Fakulte informačních technologií Vysokého učení technického v Brně so snahou zahrnúť toto prostredie ako funkčnú vzorku do výsledkov projektu.

Literatura

[1] Cisco DevNet: Cisco onePK 1.3 API reference [online]. [cit. 2015-05-14]. Dostupné z: https://developer.cisco.com/site/onepk/documents/api-reference/c/.

[2] Dávid Antolík: *ŘÍDICÍ JAZYK PRO OPENFLOW SÍTE. Bakalárska práca.* Brno: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, 2013.

[3] Floriano De Rango, Mauro Tropea, Peppino Fazio, Salvatore Marano: Overview on VoIP: Subjective and Objective Measurement Methods. *IJCSNS*, ročník 6, č. 1B, 2006, [cit. 2015-05-14].

[4] Hyojoon Kim, Nick Feamster: Improving Network Management with Software Defined Networking. *IEEE Communications Magazine*, 2013, [cit. 2015-03-10]. URL http://www.cc.gatech.edu/~hkim368/publication/SDN_ieeemagazine_Kim.pdf

[5] InfoNet: Kodeky pre kódovanie hlasu v telekomunikáciách. http://www.infonet.as/index.php?idp=3,20,47.

[6] INTERNET ENGINEERING TASK FORCE (IETF): RTP: A Transport Protocol for Real-Time Applications . RFC 1889, January 1996. URL https://www.ietf.org/rfc/rfc1889.txt

[7] INTERNET ENGINEERING TASK FORCE (IETF): SIP: Session Initiation Protocol. RFC 3261, June 2002. URL https://www.ietf.org/rfc/rfc3261.txt

[8] INTERNET ENGINEERING TASK FORCE (IETF): RTP: A Transport Protocol for Real-Time Applications. RFC 3550, July 2003. URL https://www.ietf.org/rfc/rfc3550.txt

[9] INTERNET ENGINEERING TASK FORCE (IETF): SDP: Session Description Protocol. RFC 4566, July 2006. URL https://www.ietf.org/rfc/rfc4566.txt

[10] ITU: Telecommunication Standardization Sector [online]. [cit. 2015-05-14]. http://www.itu.int/en/ITU-T/Pages/default.aspx.

[11] ITU-T Recommendation G.107: The E-model: a computational model for use in transmission planning. http://www.itu.int/rec/T-REC-G.107.

[12] ITU-T Recommendation P.910: Subjective video quality assessment methods for multimedia applications. http://www.itu.int/rec/T-REC-P.910-200804-I/en.

[13] Koistinen, T.: Protocol overview: RTP and RTCP. [cit. 2015-03-10].

[14] Limin Nie, Xuan Wang, Xiaorong Yi, Jiancheng Lin: The Implementation of Speaker Recognition on VoIP Auditing in Gigabit High-speed Environment. *IWISA 2009*, 2009, [cit. 2015-05-09].
URL
http://www.academypublisher.com/proc/iwisa09/papers/iwisa09p396.pdf

[15] Martin Basel: *ANALYZÁTOR KVALITY HOVORU VOIP. Bakalárska práca.* Brno: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, 2014.

[16] Martin Kmeť: *ANALÝZA A DETEKCE TYPU MULTIMEDIÁLNÍCH DAT V PROVOZU RTP. Diplomová práca.* Brno: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, 2014.

[17] Petr Matousek, Martin Kmet, Martin Basel: On-line Monitoring of VoIP Quality Using IPFIX. *INFORMATION AND COMMUNICATION TECHNOLOGIES AND SERVICES*, ročník 12, č. 4, 2014, [cit. 2015-04-27].

[18] TonCar.cz: VoIP Basics: About Jitter.
http://toncar.cz/Tutorials/VoIP/VoIP_Basics_Jitter.html.

[19] Vozňák, M.: *Spojovací systémy.* Ostrava: VŠB - TECHNICKÁ UNIVERZITA OSTRAVA, 2009, ISBN 978-80-248-1961-7.

[20] Wenyu Jiang, H. S.: Perceived Quality of Packet Audio under Bursty Losses. *IEEE INFOCOM*, 2002, [cit. 2015-05-15].
URL http://www.cs.columbia.edu/techreports/cucs-009-01.pdf

[21] Wikipedia: CISCO Generic Routing Encapsulation [online]. [cit. 2015-05-14]. Dostupné z: http://en.wikipedia.org/wiki/Generic_Routing_Encapsulation.

[22] Wikipedia: Protokol H.323 [online]. [cit. 2015-05-14]. Dostupné z: http://cs.wikipedia.org/wiki/H.323.

[23] Wikipedia: Protokol SIP [online]. [cit. 2015-05-14]. Dostupné z: http://cs.wikipedia.org/wiki/Session_Initiation_Protocol.

[24] Wikipedia: QoS metriky paketových sietí: Subjektívne a objektívne testovanie [online]. [cit. 2015-05-14]. Dostupné z: http://qos-diplomka.webzdarma.cz/3_METRIKY/325.HTM.

[25] Wikipedia: RTP Control Protocol [online]. [cit. 2015-05-14]. Dostupné z: http://en.wikipedia.org/wiki/RTP_Control_Protocol.

Příloha A

Obsah optického média

Na CD médiu priloženom k tejto práci sú uložené zdrojové kódy implementovanej monitorovacej platformy OneMon. Súčasťou je aj ukážkový analyzátor VoIP prevádzky a analyzátor pre ukladanie zachytených dát do PCAP súborov. CD médium obsahuje tiež zdrojové kódy najnovšej verzie Cisco OnePK 1.3 a skript createNEp12.sh k vygenerovaniu potrebných certifikátov pre ustanovenie OnePK relácie.

`./src`
Zložka so zdrojovými kódmi platformy OneMon. Obsahuje zdrojový kód jadra platformy, pozostávajúceho z rozhrania pre konfiguráciu monitorovania, subsystému pre zber zachytených dát a konektora pre registráciu analyzátorov prevádzky. V súbore `src/onemon.sql` sa nachádza skript pre vytvorenie databázy platformy OneMon.

`./onep`
Zdrojové kódy Cisco One Platform Kit SDK verzie 1.3 v jazyku C.

`./doc`
Zložka obsahuje zdrojové kódy diplomovej práce pre program LaTeX. Preložená textová časť diplomovej práce sa nachádza v súbore `doc/projekt.pdf`.

`./scripts/createNEp12.sh`
Skript pre vygenerovanie certifikátov OnePK relácie medzi sieťovým prvkom a platformou OneMon.

`./README.txt`
Textový súbor s popisom obsahu CD média.

Příloha B

Analyzátor pre export do súborov PCAP

```
#include "pcap.h"

pcap_t* pcap_handle = NULL;
pcap_dumper_t* pcap_dumpfile = NULL;

typedef struct pcap_pkthdr pcap_pkthdr_t;

// Call-back funkcia analyzátora Pcap
void Pcap(TQueueItem* start, TQueueItem* stop)
{
    // získanie konfiguračnej entity pcap_filename z databázy
    char* pcap_filename = get_config_value("pcap_filename");
    if(pcap_filename==NULL)
        return;
    open_pcap(pcap_filename);

    // spracovanie každého paketu z frontu, až po zarážku STOP
    TQueueItem* item = start;
    while(item != NULL)
    {
     // získanie ADT reprezentujúcej paket v OnePK SDK
        TPacket* packet = item->packet;

        uint8_t* l2_start;
        uint32_t l2_len;
        // získanie L2 payloadu paketu a jeho dĺžky
        onep_dpss_pkt_get_l2_start(packet, &l2_start, &l2_len);

        // prepočet časových značiek pre uloženie do PCAP
        pcap_pkthdr_t x = {{(uint32_t)item->timestamp.tv_sec,
        (uint32_t)item->timestamp.tv_nsec/1000}, l2_len, l2_len};
```

```
        pcap_dump((u_char*)pcap_dumpfile, &x, l2_start);

        // posun na ďalší paket vo fronte
        item = GetNextItem(item, stop);
    }
}

// pomocná funkcia pre otvorenie PCAP súboru
void open_pcap(char* filename)
{
    if(pcap_handle==NULL)
    {
        pcap_handle = pcap_open_dead(1, 1500);
        if (pcap_handle == NULL) {
            fprintf(stderr, "Couldn't create PCAP handle.\n");
            exit(EXIT_FAILURE);
        }
    }
    if(pcap_dumpfile==NULL)
    {
        pcap_dumpfile = pcap_dump_open(pcap_handle, filename);
        if (pcap_dumpfile == NULL) {
            fprintf(stderr, "Couldn't create PCAP dumpfile.\n");
            exit(EXIT_FAILURE);
        }
    }
}
```

Příloha C

Schéma MySQL databázy pre platformu OneMon

```
-- Table structure for table `access_list`

CREATE TABLE IF NOT EXISTS `access_list` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `number` int(11) DEFAULT NULL,
  `action` enum('permit','deny') DEFAULT 'permit',
  `protocol` varchar(32) DEFAULT NULL,
  `ip_source` int(11) NOT NULL,
  `ip_destination` int(11) NOT NULL,
  `ttl` int(11) DEFAULT NULL,
  `filter_id` int(11) NOT NULL,
  `pn_source` int(11) NOT NULL,
  `pn_destination` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_access_list_ip_networks_idx` (`ip_source`),
  KEY `fk_access_list_ip_networks1_idx` (`ip_destination`),
  KEY `fk_access_list_filter1_idx` (`filter_id`),
  KEY `fk_access_list_ports1_idx` (`pn_source`),
  KEY `fk_access_list_ports2_idx` (`pn_destination`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------
-- Table structure for table `analyzer`

CREATE TABLE IF NOT EXISTS `analyzer` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(45) DEFAULT NULL,
  `description` varchar(45) DEFAULT NULL,
  `src` varchar(128) DEFAULT NULL,
  `args` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```

```
-- ----------------------------------------------------------
-- Table structure for table `application`

CREATE TABLE IF NOT EXISTS `application` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(45) DEFAULT NULL,
  `active_flag` tinyint(1) DEFAULT NULL,
  `certificate_id` int(11) NOT NULL,
  `analyzer_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_application_certificates1_idx` (`certificate_id`),
  KEY `fk_application_analyzer1_idx` (`analyzer_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;

-- ----------------------------------------------------------
-- Table structure for table `application_config`

CREATE TABLE IF NOT EXISTS `application_config` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `application_id` int(11) NOT NULL,
  `config_name` varchar(255) NOT NULL,
  `config_value` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;

-- ----------------------------------------------------------
-- Table structure for table `certificate`

CREATE TABLE IF NOT EXISTS `certificate` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(45) DEFAULT NULL,
  `root_cert_path` varchar(128) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;

-- ----------------------------------------------------------
-- Table structure for table `filter`

CREATE TABLE IF NOT EXISTS `filter` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(45) DEFAULT NULL,
  `type` varchar(45) DEFAULT NULL,
  `application_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_policy_map_policy_map1_idx` (`application_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```

```
-- ------------------------------------------------------
-- Table structure for table `filter_has_nbar_protocol`

CREATE TABLE IF NOT EXISTS `filter_has_nbar_protocol` (
  `filter_id` int(11) NOT NULL,
  `nbar_protocol_id` int(11) NOT NULL,
  PRIMARY KEY (`filter_id`,`nbar_protocol_id`),
  KEY `fk_filter_has_nbar_protocol_nbar_protocol1_idx` (`nbar_protocol_id`),
  KEY `fk_filter_has_nbar_protocol_filter1_idx` (`filter_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- ------------------------------------------------------
-- Table structure for table `interface_list`


CREATE TABLE IF NOT EXISTS `interface_list` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `intf_name` varchar(100) DEFAULT NULL,
  `intf_type` varchar(100) DEFAULT NULL,
  `router_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_interface_list_router1_idx` (`router_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- ------------------------------------------------------
-- Table structure for table `ip_network`

CREATE TABLE IF NOT EXISTS `ip_network` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `address` varchar(45) DEFAULT NULL,
  `mask` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- ------------------------------------------------------
-- Table structure for table `nbar_protocol`


CREATE TABLE IF NOT EXISTS `nbar_protocol` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `protocol_name` varchar(45) DEFAULT NULL,
  `protocol_description` varchar(255) DEFAULT NULL,
  `protocol_id` varchar(32) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;

-- ------------------------------------------------------
-- Table structure for table `ports`
```

```
CREATE TABLE IF NOT EXISTS `ports` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `greater_or_equal` int(11) DEFAULT NULL,
  `less_or_equal` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- ----------------------------------------------------------
-- Table structure for table `router`

CREATE TABLE IF NOT EXISTS `router` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `management_ip` varchar(45) DEFAULT NULL,
  `name` varchar(45) DEFAULT NULL,
  `application_id` int(11) NOT NULL,
  `username` varchar(45) DEFAULT NULL,
  `password` varchar(45) DEFAULT NULL,
  `interfaces` text,
  PRIMARY KEY (`id`),
  KEY `fk_router_application1_idx` (`application_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```