VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ZÁKONNÉ ODPOSLECHY V SDN

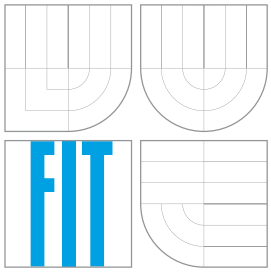
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. BARBORA FRANKOVÁ

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ZÁKONNÉ ODPOSLECHY V SDN

LAWFUL INTERCEPTION IN SOFTWARE DEFINED NETWORKS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. BARBORA FRANKOVÁ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. LIBOR POLČÁK

BRNO 2015

Abstrakt

Práce se zabývá využitím softwarově definovaných sítí v oblasti zákonných odposlechů. Staví na konkrétní implementaci systému pro zákonné odposlechy – SLIS, který byl vyvinut v rámci projektu Sec6Net. Navrhuje rozšíření v několika oblastech, ve kterých SDN nabízí potenciál ke spolehlivější identifikaci odposlouchávaných uživatelů a efektivnějšímu využití sítě. První zmíněný cíl je realizován prostřednictvím modulu funkce dynamické identity, druhý pak pomocí konfigurace síťových sond a OpenFlow přepínačů.

Abstract

This thesis covers utilization of software defined networks for lawful interception purposes. Based on specific implementation of lawful interception system SLIS developed by Sec6Net group, suggests improvements aiming at more precise identification of intercepted users and better effectivity of system resources. First aim is achieved by implementation of a new module for dynamic identification component while the other one alters configuration mechanism for probes and OpenFlow switches.

Klíčová slova

softwarově definované sítě, SDN, OpenFlow, OpenDaylight, POX, zákonné odposlechy, Sec6Net

Keywords

software defined networking, SDN, OpenFlow, OpenDaylight, POX, lawful interception, Sec6Net

Citace

Barbora Franková: Zákonné odposlechy v SDN, diplomová práce, Brno, FIT VUT v Brně, 2015

Zákonné odposlechy v SDN

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně pod vedením pana Ing. Libora Polčáka.

.....
Barbora Franková
27. května 2015

Poděkování

Ráda bych poděkovala Ing. Liboru Polčákovi za cenné rady, věcné připomínky a vstřícnost při konzultacích.

Obsah

Kapitola 1

Úvod

Jedním z rysů moderní společnosti je značný technologický pokrok v oblasti informačních technologií a výpočetní techniky. S tím souvisí rychlé pronikání těchto technologií do téměř všech oblastí lidské činnosti a každodenního života. Vytváří tak obrovský prostor pro útočníky, kteří se mohou dostat k více či méně chráněným soukromým informacím.

Počítačová kriminalita každým rokem roste. Znalosti či schopnosti nutné ke kybernetickým útokům se stále snižují a zabránit počítačové kriminalitě není možné. Pachatelé za sebou nezanechávají fyzické stopy a ve velké míře zneužívají anonymitu, kterou Internet poskytuje. Některé z útoků, např. neoprávněný přístup, odposlouchávání, narušení systému nebo počítačové padělání, již byly zaneseny do zákona jako trestná činnost a přesné znění lze nalézt v trestním zákoníku (zákon č. 40/2009 Sb.), v zákoně o elektronických komunikacích (zákon č. 127/2005 Sb.), v Evropském právu (Úmluva Rady Evropy o počítačové kriminalitě – Convention on Cybercrime, ETS No. 185) a dalších.

Tato práce je zaměřena na softwarově definované sítě. Softwarově definované sítě fungují s využitím *operačního systému*, který umožňuje abstrakci prostředků i informací v počítačových sítích. Jedná se o přístup, který je dynamický, lehce ovladatelný, adaptivní a s nízkými náklady.

Jednou z možností, jak bojovat proti počítačové kriminalitě, jsou systémy pro zákonné odposlechy. Návrh těchto systémů byl vytvořen úřadem ETSI pro všechny země Evropské unie. Systémy pro zákonné odposlechy umožňují oprávněným orgánům sledovat komunikaci podezřelých subjektů v počítačové či telefonní síti [1]. Sledování je možné provádět pouze na základě soudního příkazu. V rámci projektu Sec6Net vznikla konkrétní implementace systému pro zákonné odposlechy – *Sec6Net Lawful Interception System* (SLIS) [19]. Systém je určen pro nasazení v sítích poskytovatelů Internetu.

Cílem této práce je návrh, implementace a testování systémů pro zákonné odposlechy v prostředí softwarově definovaných sítí. Jedná se o získávání informací o identitě odposlouchávaného uživatele a dynamická rekonfigurace hardware na základě pozice v topologii.

Práce je rozdělena do několika kapitol. V kapitole 2 je popsán princip a architektura softwarově definovaných sítí. Kapitola 3 se zabývá architekturou systému SLIS. Jsou zde popsány jednotlivé bloky, ze kterých se systém skládá, jejich zapojení a rozhraní pro komunikaci. V kapitole 4 jsou uvedeny cíle práce a návrhy rozšíření systému SLIS tak, aby mohl provádět odposlechy v softwarově definovaných sítích. V kapitole 5 je popsána implementace těchto rozšíření a kapitola 6 je věnována ověření funkčnosti a testování.

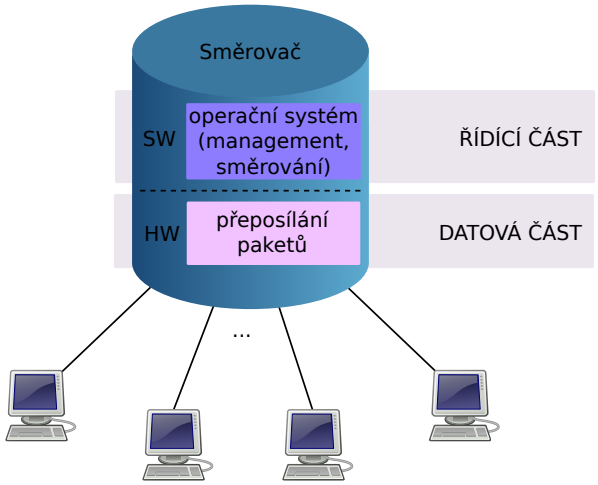
Kapitola 2

Softwarově definované sítě

Cílem této kapitoly je vysvětlit pojem softwarově definované sítě. V sekci 2.1 je popsána obecná architektura síťových zařízení. Sekce 2.2 vysvětluje princip softwarově definovaných sítí (SDN) a sekce 2.3 je věnována protokolu OpenFlow. V sekci 2.4 jsou popsány kontrolery, které byly použity pro implementaci rozšíření.

2.1 Architektura síťových zařízení

Klasické síťové zařízení, jak je znázorněno na obrázku 2.1, obsahuje vlastní hardware pro přeposílání paketů (*datová část, data plane*) a operační systém v software pro management a logiku (*řídící část, control plane*). Nad operačním systémem mohou být implementovány další moduly, u managmentu jde například o podporu protokolů CLI nebo SNMP, logika ze strany uživatele je dána moduly podporujícími směrovací protokoly, přepínání nebo QoS. Všechna tato zařízení ale mají pevně danou funkcionalitu od výrobce, složitý operační systém a jsou uzavřená k inovacím.



Obrázek 2.1: Architektura klasického síťového zařízení.

Tradiční hardwarová zařízení pracují na různých vrstvách ISO/OSI architektury. Přepínače využívají informace z druhé vrstvy a přepínání paketů probíhá na základě cílové MAC adresy. Záznamy o MAC adresách spolu s VLAN a rozhraním jsou uloženy v CAM tabulce. Cílová MAC adresa každého přijatého paketu je porovnána se záznamy v této tabulce a paket je odeslán na uvedené rozhraní [3].

2.1.1 Řídící část (Control Plane)

Funkce řídící části zahrnují konfiguraci systému, vysokoúrovňový management a statistiky. Ve směrovačích slouží ke správě směrovací tabulky (Routing Information Base – RIB) a přepínací tabulky (Forwarding Information Base – FIB). Informace ve směrovací tabulce se aktualizují pomocí směrovacích protokolů, jako je např. RIP, OSPF nebo BGP [20]. Vzhledem k tomu, že funkce v řídící části se neprovádí nad každým paketem, nejsou obvykle limitovány časem a proto mohou být implementovány v software [3].

2.1.2 Datová část (Data Plane)

Funkce datové části provádí menší množství operací a týkají se především přijímání, zpracování a směrování paketů. Ve chvíli, kdy je paket přijat na vstupním rozhraní, vyhledá se ve směrovací tabulce cílová adresa. Na základě zdrojové a cílové IP adresy, čísla portu transportní vrstvy a typu protokolu se paket klasifikuje a zahodí (např. firewall) nebo odešle na výstupní rozhraní. Poté se sníží time-to-live (TTL) a přepočítá se kontrolní součet hlavičky [20]. Funkce datové části se provádí nad každým paketem, což vyžaduje vysokou výkonnost v reálném čase a implementaci v hardware [3].

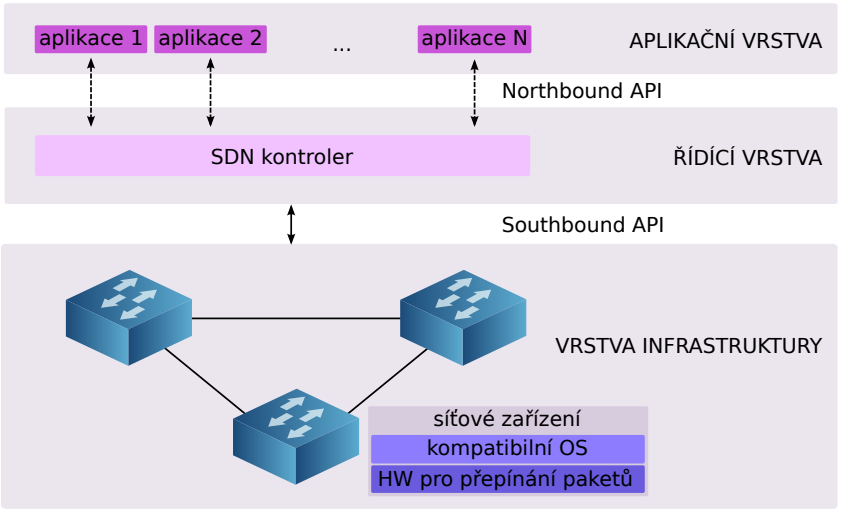
2.2 Princip softwarově definovaných sítí

Softwarově definované sítě (Software Defined Networking – SDN) oddělují rozhodovací logiku a rychlé přepínání paketů v hardware. Zatímco vrstva infrastruktury zůstává na síťovém zařízení, řídící část a vyšší logika se přesouvá do tzv. kontroleru. Kontroler je oddělené zařízení, které má přehled o topologii celé sítě a o prostředcích k směrování a přepínání paketů, které fyzicky umožňují jednotlivá síťová zařízení. S využitím těchto informací je kontroler schopný určit cesty pro toky v síti a podle toho posílat pokyny jednotlivým síťovým zařízením.

Architektura softwarově definovaných sítí se tedy skládá z vrstvy infrastruktury, což je datová část původní architektury a řídící vrstvy, ve které je oddělená kontrolní část ve formě kontroleru. Nad řídící vrstvou lze vytvářet různé aplikace, které tvoří aplikační vrstvu. Jednotlivé vrstvy jsou propojeny programovacím rozhraním, tzv. northbound a southbound API. Architektura SDN je znázorněna na obrázku 2.2.

2.2.1 Síťové zařízení

V SDN je síťové zařízení abstrahováno od specifické činnosti jako je router, směrovač, firewall nebo load-balancer. SDN přepínač obsahuje rychlý hardware pro přeposílání paketů a velmi jednoduchý operační systém, který sám o sobě musí umět pouze komunikovat s kontrolerem a rychle přeposílat pakety mezi porty na základě na základě pokynů od kontroleru [10]. Je možné nakonfigurovat zařízení tak, aby samostatně odpovídalo na některé události, např. výpadky sítě nebo podpůrné funkce poskytované LLDP, STP nebo ICMP [2].



Obrázek 2.2: Architektura softwarově definovaných sítí.

V SDN se síťová zařízení dělí podle způsobu, kterým mohou zpracovávat pakety. Prvním typem je přepínač, u kterého všechny funkce tradičního přepínače (např. správa směrovací a přepínací tabulky) řeší centrální kontroler. Funkcionalita samotného přepínače je omezena pouze na datovou část, která přeposílá pakety na základě příkazů kontroleru [13].

Druhý typ zařízení je hybridní SDN přepínač, který obsahuje vlastní kontrolní část a funkcionalitu tradičního switche. Zároveň ale tato zařízení obsahují modul v operačním systému pro propojení s externím kontrolerem a umožňují tak běh klasického i softwarově definovaného přeposílání paketů. Při využívání hybridních přepínačů není nutná změna infrastruktury sítě pro využívání SDN přístupu.

2.2.2 SDN kontroler

SDN kontroler je software, který provádí kontrolu nad množinou zdrojů jednotlivých datových částí. Může být implementován jako větší množství programových částí, které mohou být rozmístěny v několika fyzických zařízeních. Všechny části si udržují synchronizovaný a konzistentní pohled na topologii a stav sítě. Funkcionalita kontroleru obsahuje [14]:

- správu stavu sítě (informace o síťových uzlech a koncových zařízeních, konfigurace, statistiky apod.) a jeho případnou distribuci ostatním kontrolerům;
- vysokoúrovňový model pro zachycení vztahů mezi spravovanými zdroji, pravidly a dalšími poskytovanými službami (často se používá Yang[1] model);
- programové rozhraní, které umožní ovládat kontroler z externí aplikace;
- správu toků;
- zjišťování topologie, směrování (algoritmy pro výpočet cesty).

[1] jazyk pro modelování dat, který se využívá k modelaci konfigurace a stavu dat získaných pomocí Network Configuration Protocol (NETCONF), https://tools.ietf.org/html/rfc6020

2.2.3 Komunikační kanál

Komunikační kanál je rozhraní, které propojuje síťové zařízení s kontrolerem. Přes komunikační kanál kontroler konfiguruje síťová zařízení a přijímá nebo zasílá pakety.

2.2.4 Programové rozhraní

Síťová zařízení jsou konfigurována kontrolerem přes programové rozhraní (API). V rámci SDN se využívají Southbound a Northbound API.

Southbound

Southbound API je nízkoúrovňové programovací rozhraní, které obvykle obsahuje nezbytnou funkcionalitu na programování parametrů poskytovaných operačním systémem síťových zařízení. Mezi limitující vlastnosti toho rozhraní patří [4]

- race conditions – správné pořadí využívání operací Southbound API;
- nízká úroveň abstrakce – náročné programování složitějších funkcionalit sítě;
- složité pro několik nezávislých úloh (směrování, load-balacer a QoS zároveň).

Typickým příkladem je OpenFlow, I2RS, NETCONF nebo OnePK.

Northbound

Northbound API je programovací rozhraní, které zapouzdřuje nízkou úroveň instrukcí southbound API a umožňuje programovat složitější síťové funkce. V Northbound API je možné implemetovat funkce jako výpočet cesty, STP, směrování, hloubková inspekce paketů a další [14]. V současné chvíli toto rozhraní není standardizováno [7].

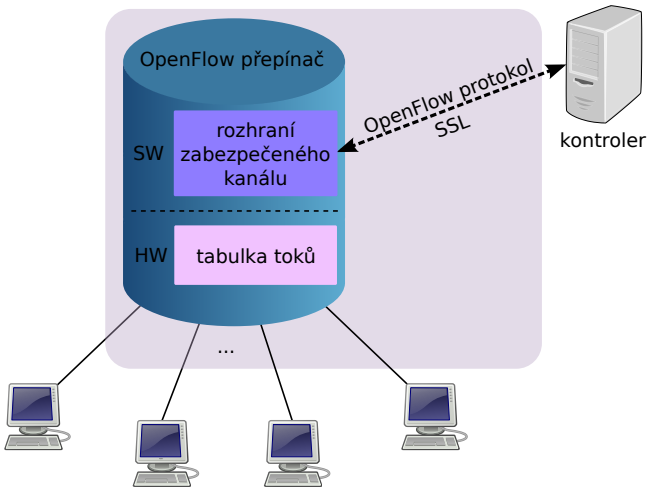
2.3 OpenFlow

OpenFlow je první standard pro komunikační rozhraní mezi řídící vrstvou a síťovými zařízeními. Nejedná se o produkt nebo určitou vlastnost, ale o množinu protokolů, programové rozhraní a jeden z možných modelů softwarově definovaných sítí. Architektura OpenFlow je znáznorněna na obrázku 2.3. Skládá se z OpenFlow kontroleru (controller), OpenFlow přepínače (switch) a OpenFlow protokolu [8, 16].

2.3.1 OpenFlow Wire Protocol

OpenFlow přepínač využívá koncept datových toků k identifikaci síťového provozu na základě pravidel, která jsou staticky nebo dynamicky naprogramována z centrálního kontroleru [16]. OpenFlow přepínač se skládá minimálně z následujících položek:

- Rozhraní zabezpečeného kanálu, které propojuje přepínač s kontrolerem, umožňuje posílání příkazů i paketů a obecně odpovídá definici southbound rozhraní.
- Tabulka toků, která obsahuje množinu záznamů o datových tocích a je znázorněna na obrázku 2.4. Každý záznam se skládá z pole hlaviček, počítadla a množiny pravidel.



Obrázek 2.3: Architektura OpenFlow.

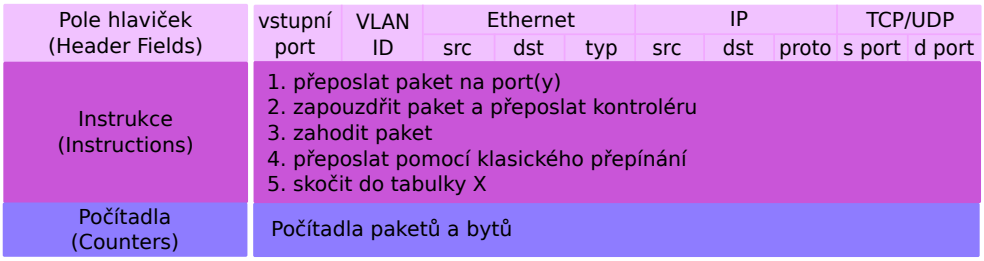
– *Pole hlaviček (Header Fields)* se využívá k porovnání příchozích paketů a existujících záznamů v tabulce toků. V OpenFlow v1.0 je tok specifikován jako 10-tice, která se skládá ze vstupního portu, VLAN ID, zdrojové a cílové MAC adresy, typu ethernetového rámce, zdrojové a cílové IP adresy, protokolu IP a zdrojového a cílového TCP/UDP portu. Každá z těchto položek může být nahrazena zástupným znakem (wildcard), což umožňuje agregaci toků.
– *Počítadla (Counters)* aktualizují se s každým přijatým paketem, pro který byl nalezen záznam v tabulce toků a počítají se pro danou tabulku toků, tok, port a frontu.
– *Množina instrukcí (Instructions)* určuje, jak bude přepínač zacházet s pakety, pro které byl nalezen záznam v tabulce toků. V OpenFlow protokolu verze 1.0 jsou definovány povinné akce přeposlat (*forward*) a zahodit (*drop*). Při přeposlání paketu je nutné specifikovat fyzický, virtuální nebo některý z rezervovaných portů
  * `ALL` – odeslání paketu na všechna rozhraní kromě příchozího;
  * `CONTROLLER` – odeslání paketu na kontroler;
  * `LOCAL` – odeslání paketu na lokální stack přepínače (paket musí být zpracován pomocí operačního systému přepínače – pouze u hybridních přepínačů);
  * `TABLE` – provedení pravidla z tabulky toků;
  * `IN_PORT` – odeslání paketu zpět na vstupní port.

  OpenFlow v1.3 umožňuje využití více tabulek toků a rozšiřuje množinu instrukcí o položku skoč do tabulky X (*go-to table X*). Tabulky se prochází sekvenčně až do chvíle, kdy v seznamu instrukcí není další příkaz skoč do tabulky. Při průchodu tabulkami je možné paket modifikovat. Změnit lze jakoukoliv položku v paketu, která se dá porovnávat v rámci pravidel. Je také možné zvýšit nebo snížit TTL, případně vložit nebo odstranit tag (VLAN, MPLS, PBB).

  V každé tabulce se mohou zadané akce provést okamžitě, pokud se vloží do

množiny akcí v *apply-actions*. V případě, že se vloží do množiny *write-actions*, provedou se až po průchodu paketu poslední tabulkou.

- OpenFlow protokol, který je naimplementován na straně kontroleru i přepínače a používá se pro komunikaci přes zabezpečený kanál [11].



Obrázek 2.4: Tabulka toků v OpenFlow přepínačích.

Všechny přijaté pakety se v přepínači porovnají s tabulkou toků. Pokud je v tabulce nalezen odpovídající záznam, přepínač provede všechny akce, které jsou pro daný tok specifikovány. Pokud pro paket nebyla nalezena shoda, je přeposlán přes zabezpečený kanál do kontroleru. Kontroler pak pomocí přidávání, upravování a odstraňování záznamů v tabulce toků rozhoduje, jak se budou takové pakety zpracovávat [8].

Proaktivní a reaktivní přístup

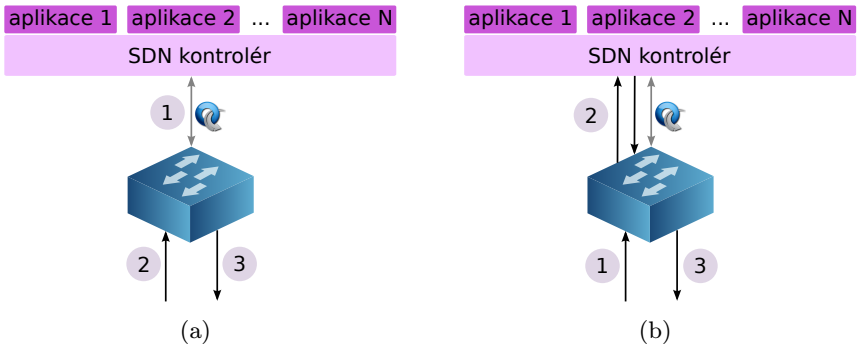
Proaktivní a reaktivní přístup se rozlišuje na základě způsobu nahrávání pravidel do síťového zařízení. Rozdíl v těchto přístupech je znázorněn na obrázku 2.5. Proaktivní přístup je založen na předvyplnění tabulky toků v každém přepínači. Výhodou je nulové zpoždění pro nové toky, protože pravidlo je v přepínači už definováno. Zároveň přerušení spojení s kontrolerem neovlivní samotný proces přeposílání paketů. Nevýhodou pak může být nutnost přesně definovat všechna pravidla, což vyžaduje například agregaci (wildcard) tak, aby byly pokryty všechny cesty [6].

U reaktivního přístupu je paket, který nemá záznam v tabulce toků, odeslán kontroleru. Kontroler na základě informací z paketu vytvoří nové pravidlo pro všechny OpenFlow přepínače v síti. Tento přístup má výhodu v efektivním využívání paměti, ale pro každý nový tok znamená určité počáteční zpoždění. V případě, že dojde k přerušení spojení kontroleru a přepínače, pakety neznámých toků budou zahozeny.

2.3.2 OpenFlow Management and Configuration Protocol (OF-Config)

The OpenFlow Management and Configuration Protocol (OF-Config) definuje způsob, jakým lze vzdáleně získávat a upravovat konfiguraci OpenFlow přepínače. Je založen na datovém modelu XML a využívá transportní protokol NETCONF. Mezi základní funkce tohoto protokolu (verze 1.0) patří:

- přiřazení jednoho nebo více kontrolerů danému přepínači;
- konfigurace portů a front.



Obrázek 2.5: Znázornění pořadí kroků u (a) proaktivního a (b) reaktivního vkládání pravidel do tabulky toků. U proaktivního přístupu je nejdříve vyplněna tabulka toků (1). Každý přijatý paket (2) se pak zahodí nebo přepošle (3) podle pravidel v tabulce. Reaktivní přístup musí nejdříve přijmout paket (1), který odešle kontroleru (2). Kontroler na základě tohoto paketu vytvoří záznam v tabulce toků, podle které se paket a všechny následující ve stejném toku zahodí nebo odešlou (3).

Hlavním rozdílem mezi OpenFlow a OF-Config je časový rámec podporovaných operací. Zatímco OpenFlow pracuje s kratšími časovými úseky, které jsou vázány na datové toky (např. vytváření směrovacích tabulek), OF-Config definuje operace, které mají obecně delší platnost a přímo nesouvisí s datovými toky (např. zapnutí a vypnutí portu) [15, 17].

2.3.3 Rozšíření vyšších verzí OpenFlow

Specifikace protokolu OpenFlow se neustále rozšiřuje. Aktuální verze z roku 2014 je OpenFlow v1.5. Rozšíření, které se v jednotlivých verzích protokolu objevily, jsou následující:

- OpenFlow v1.1
  - více tabulek toků;
  - skupiny – povolují OpenFlow reprezentovat množinu portů jako jeden celek pro přeposílání paketů, jsou k dispozici různé druhy skupin pro různé abstrakce (např. multicasting nebo multipathing);
  - podpora MPLS a VLAN tagů;
  - virtuální porty – složitější přeposílání paketů, např. tunely;
  - kontrola spojení přepínače a kontroleru – využití pohotovostní cache toků k vyrovnání se se ztrátou spojení s kontrolerem.
- OpenFlow v1.2
  - více kontrolerů.
- OpenFlow v1.3
  - podpora IPv6;

  - počítadla toků – mohou být připojena do tabulky toků a měřit nebo kontrolovat rychlost paketů.

- OpenFlow v1.4
  - optické porty a synchronizované tabulky.

2.4 Použité kontrolery

2.4.1 OpenDaylight

Pro rozšíření systému pro zákonné odposlechy je nutné vědět, jakým způsobem funguje L2 přepínač řízený kontrolerem OpenDaylight. OpenDaylight je open source projekt, který má podporu i v komerční sféře (např. Cisco XNC[2] je postaveno přímo nad OpenDaylight). Jeho součástí je GUI a poskytuje northbound rozhraní pro Java a REST API. Pracuje s protokolem OpenFlow v1.0 až v1.3 [12].

Kontroler zjišťuje topologii sítě pomocí LLDP. Ve chvíli, kdy detekuje jakoukoliv změnu, je spuštěn modul *LoopRemover*. Tento modul odstraní z topologie smyčky tím, že vytvoří kostru grafu a zablokuje oba konce linek, které v kostře grafu nejsou. Tyto linky jsou nepoužívané a mají dostupnou plnou kapacitu, kterou lze využít.

Pro zjištění pozice koncových stanic v síti musí kontroler počkat, až začnou samy komunikovat. Pokud na přepínač přijde paket, L2 přepínač zjistí jeho zdrojovou MAC adresu. Pokud pro tuto adresu existuje pravidlo, je paket odeslán na výstupní port uvedený v pravidle. V opačném případě je paket odeslán na všechna výstupní rozhraní kromě vstupního. OpenFlow L2 přepínač se tedy při přijetí paketu s neznámou zdrojovou MAC adresou chová stejně jako by se choval klasický přepínač.

Důležité je, že OpenDaylight si udržuje informace o párování MAC-port pro všechny adresy v síti. Jedná se o globální pohled na síť a ne lokální pro každý přepínač. Port spárovaný s danou MAC adresu je port, kam je koncové zařízení s MAC adresou přímo připojeno. Aby párování zůstalo korektní, kontroler nikdy nesmí dostat duplikované pakety se stejnou MAC adresou ze dvou různých přepínačů. Z tohoto důvodu OpenDaylight nepodporuje přesouvání koncových stanic mezi jednotlivými přepínači.

2.4.2 POX

POX je open source kontroler implementovaný v Pythonu, který byl vyvinut ze staršího NOXu. Poskytuje asynchronní, událostmi řízené programovací rozhraní. Jádro kontroleru obsahuje pomocné metody a API pro interakci s OpenFlow přepínači (zahrnuje např. správu připojení a sledování událostí). K dispozici jsou i další komponenty, jako je detekce koncových zařízení, směrování nebo zjišťování topologie (LLDP). V porovnání s dalšími kontrolery je pomalejší a méně výkonný. NOX a POX v současné chvíli podporují protokol OpenFlow v1.2 [14].

[2] http://www.cisco.com/go/xnc

Kapitola 3

Systém pro zákonné odposlechy

Tato kapitola se věnuje systémům pro zákonné odposlechy. V sekci 3.1 je popsána obecná architektura systémů pro zákonné odposlechy definovaná úřadem ETSI. Sekce 3.2 je věnována stručnému popisu projektu *Sec6Net* a sekce 3.3 systému pro zákonné odposlechy SLIS, který v rámci projektu vznikl.

3.1 Architektura systémů pro zákonné odposlechy

Pro Evropskou unii jsou systémy pro zákonné odposlechy (Lawful Interception – LI) standardizovány úřadem ETSI[1]. Každý odposlech musí být potvrzen soudním příkazem [19]. ETSI vytvořilo referenční model pro architekturu systému pro zákonné odposlechy (Lawful Interception System – LIS). Obecné schéma LIS je znázorněno na obrázku 3.1.

Dle norem ETSI se na zákonném odposlechu podílí poskytovatel komunikačních služeb a orgány činné v trestním řízení [1].

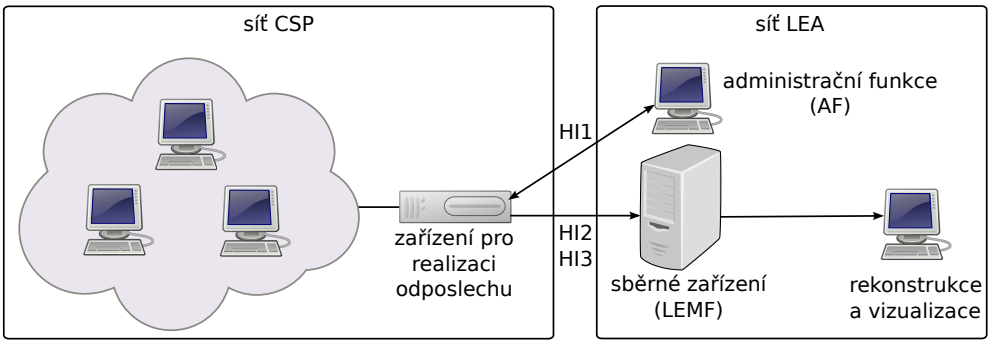
Poskytovatelem komunikačních služeb (Communications service provider – CSP) může být například poskytovatel internetu (Internet service provider – ISP) nebo telekomunikační operátor. V rámci sítě CSP se pak nachází specializované zařízení pro realizaci odposlechu (sonda nebo jiný hadrware).

Orgány činné v trestním řízení (Law Enforcement Agency – LEA) jsou oprávněny odposlouchávat a zaznamenávat data. Součástí LEA je sběrné zařízení (Law Enforcement Monitoring Facitily – LEMF), do kterého jsou zachycená data přenášena. Na straně ogránů činných v trestním řízení pak vyšetřovatelé provádí analýzu, rekonstrukci a vizualizaci zachyceného provozu.

Součástí definice LIS je rozhraní Handover Interface (HI) určené pro komunikaci mezi CSP a vyšetřující LEA, které se skládá z následujících komponent:

- HI1 – přenášení požadavků na zahájení odposlechů ze strany LEA;
- HI2 – přenášení metadat o aktivitě sledovaných uživatelů do LEMF (např. změny IP adres, připojení/odpojení ze sítě, odeslání zprávy apod.);
- HI3 – přenášení obsahu zachycené komunikace uživatele do LEMF.

[1] www.etsi.org



Obrázek 3.1: Systém pro zákonné odposlechy.

3.2 Projekt Sec6Net

Sec6Net je zkratkou pro projekt *Moderní prostředky pro boj s kybernetickou kriminalitou na Internetu nové generace*[2]. Tento projekt je zaměřen na výzkum a vývoj prostředků monitorování provozu sítí, analýzu záznamů provozu sítí a metod prostředků zabezpečení lokálních sítí s důrazem na sítě využívající protokol IPv6.

3.3 Systém SLIS

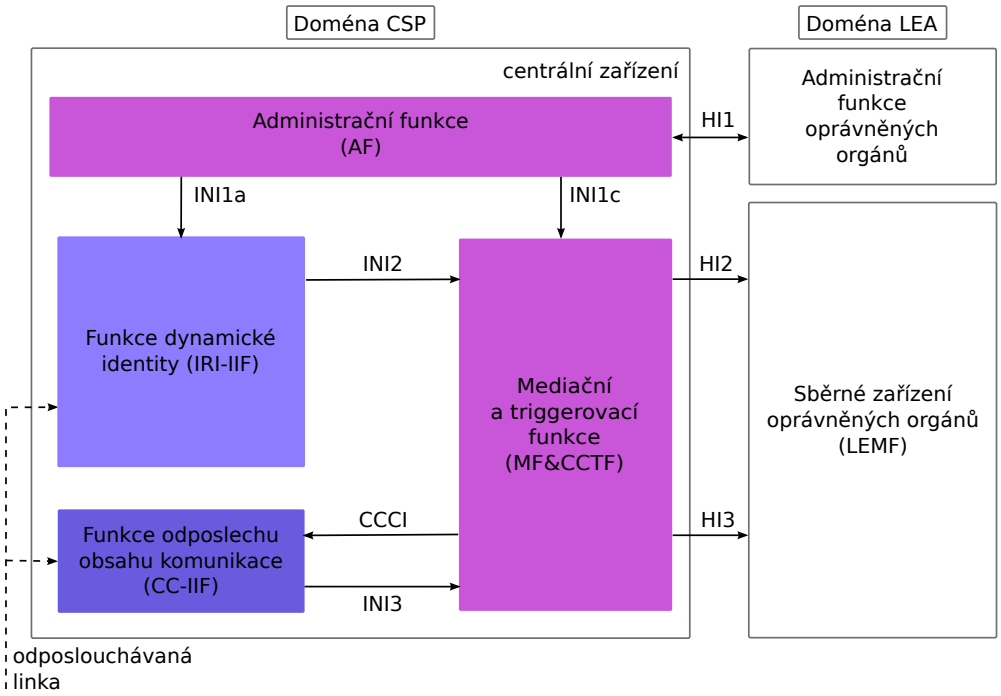
V rámci projektu Sec6Net byl vytvořen prototyp systému pro zákonné odposlechy *Sec6Net Lawful Interception System* (SLIS). SLIS je konkrétní implementace LIS založená na referenčním modelu, který byl vytvořen úřadem ETSI. Architektura systému SLIS je znázorněna na obrázku 3.2.

Klíčovou součástí systému je centrální zařízení připojené k odposlouchávané lince v doméně CSP. Skládá se ze čtyř částí, které jsou propojeny těmito rozhraními:

- **INI1** – rozhraní, které se využívá ke konfiguraci jednotlivých bloků centrálního zařízení;
- **INI2** – rozhraní pro přenost metadat komunikace odposlouchávaného uživatele;
- **INI3** – rozhraní pro přenos zachycených dat;
- **CCCI** – rozhraní, které se využívá ke konfiguraci CC-IIF sondy.

Požadavky na odposlechy ze strany LEA jsou vkládány do administrační funkce (Administration Function – AF), která pak podle těchto požadavků konfiguruje ostatní bloky systému. Funkce dynamické identity (Intercepted Related Information – Internal Interception Function – IRI-IIF) sleduje události na síti a při detekci nového nebo změně už známého indentifikátoru odposlouchávaného uživatele (IP adresy, MAC adresy, E-mailové adresy apod.) informuje mediační funkci rozhraním INI2. Funkce odposlechu obsahu komunikace (Content of Communication – Internal Interception Function – CC-IIF) zachytává

[2] http://www.fit.vutbr.cz/~matousp/grants.php?id=517



Obrázek 3.2: Architektura systému SLIS.

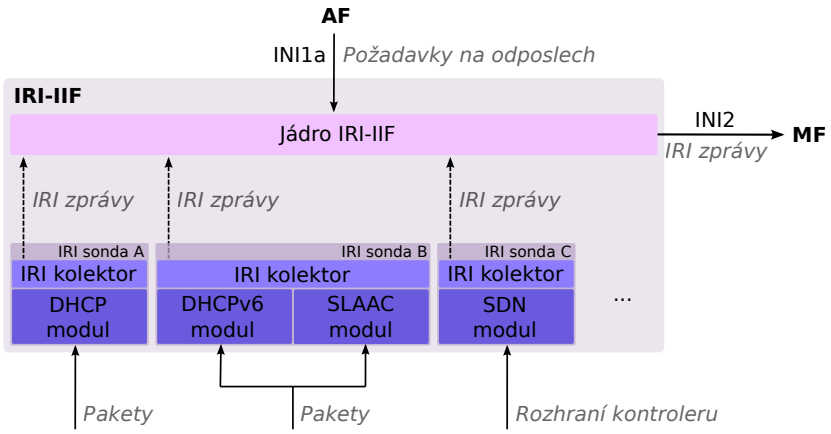
veškerý obsah komunikace odposlouchávaného uživatele, který přeposílá přes rozhraní INI3 mediační funkci.

Mediační funkce (Mediation Function – MF) přijímá informace z funkce dynamické identity a funkce odposlechu obsahu komunikace, které následně kombinuje a přeposílá LEA. Triggerovací funkce (Content of Communication Trigger Function – CCTF), která je v systému naimplementována jako součást mediační funkce, má za úkol konfiguraci CC-IIF sond přes CCCI rozhraní. V následujících sekcích jsou jednotlivé části systému popsány podrobněji.

3.3.1 Administrační funkce (AF)

Úkolem administrační funkce je přijímat požadavky na odposlechy od libovolného množství LEA. Požadavek na odposlech obsahuje položky Lawful Interception Identifier (LIID), který identifikuje daný odposlech, dále datum a čas zahájení a ukončení odposlechu, síťový identifikátor odposlouchávaného uživatele (např. IP adresa) a informaci o tom, zda se budou ukládat pouze metadata nebo veškerý obsah komunikace.

Blok AF při přijetí nového požadavku zkontroluje všechny položky a pokud je vše v pořádku, vloží odposlech do fronty čekajících odposlechů. Ve chvíli, kdy údaj o začátku odposlechu odpovídá aktuálnímu času, je požadavek vložen do fronty aktivních odposlechů. AF poté na základě této fronty konfiguruje ostatní části systému tak, aby bylo zajištěno zachycení všech dat v povoleném intervalu pro odposlech a zároveň nebyla zachycena žádná data mimo tento interval [19].



Obrázek 3.3: Architektura bloku IRI-IIF.

3.3.2 Funkce dynamické identity (IRI-IIF)

Odposlouchávaný uživatel musí být jednoznačně identifikovatelný v síti. Jeho identita (IP adresa, případně další identifikátory používané v síťovém prostředí) se ale může na straně poskytovatele dynamicky měnit např. skrze protokoly DHCP, RADIUS nebo ND [18]. Úlohou IRI-IIF je zjišťování dynamické identity sledováním probíhajících komunikací (relace, hovory, spojení apod.) a analýzou protokolů. Informace o tom, kdy a komu byl identifikátor určený k odposlechu přidělen a o případných změnách předává MF&CCTF tak, aby mohly být včas překonfigurovány připojené sondy CC-IIF [19].

IRI-IIF zjišťuje identity na základě požadavků na odposlechy z administrační funkce získaných přes rozhraní INI1a. Požadavek na odposlech obsahuje jednoznačný identifikátor odposlouchávaného uživatele (např. IP adresa nebo SIP URI). Výstupem bloku jsou tzv. IRI zprávy související s konkrétním odposlechem, které se odesílají přes rozhraní INI2 mediační funkci a následně jsou předány LEMF přes HI2.

Součástí bloku IRI-IIF je seznam typů identfikátorů, které funkce umí rozpoznat. Jednotlivé typy identifikátorů se nazývají NID. V současné chvíli jsou naimplementovány NIDy pro IPv4, IPv6, TCP trojice (IP adresa, port, protokol) a pětice (zdrojová a cílová IP adresa, zdrojový a cílový port, protokol) a další. Na každý NID lze v systému SLIS zadat odposlech.

Architektura bloku IRI-IIF

Blok IRI-IIF je navržen modulárně a je znázorněn na obrázku 3.3. Skládá se z tzv. jádra IRI-IIF a modulů pro analýzu jednotlivých protokolů. Mezi jádrem a moduly existuje mezivrstva – IRI kolektor, která má za úkol přijímat zprávy od jednotlivých modulů a přeposílat je jádru. Díky tomu lze jednotlivé části IRI-IIF provozovat i jako samostatná zařízení v síti – tzv. IRI-IIF sondy. Každá IRI-IIF sonda se skládá z kolektoru a libovolného počtu modulů a odesílá zprávy do centrálného zařízení, kde se nachází jádro IRI-IIF [19].

Typ zprávy	Událost
BEGIN	detekce začátku spojení (např. přiřazení IP adresy koncovému zařízení)
CONTINUE	detekce nových nebo doplňujících informací po začátku spojení a odeslání zprávy IRI BEGIN nebo po odeslání zprávy END
END	detekce ukončení spojení po odeslání zprávy IRI BEGIN
REPORT	detekce komunikace na síti před vytvořením spojení a odesláním IRI BEGIN (např. autentifikace uživatele)

Tabulka 3.1: Tabulka IRI zpráv a událostí, které vedou k jejich vytvoření.

```
('SLAAC',                                                          název modulu
 1367187912.15,                                                    aktuální čas
  'BEGIN',                                                         typ IRI zprávy
    'User has generated new IP address',                           popis IRI zprávy
      [('MAC','C8:4E:19:2C:33:C7'), ('IPv6','2001:DB8::1')])       seznam identifikátorů
```

Obrázek 3.4: Příklad IRI zprávy [9].

Zprávy IRI-IIF

Komunikace modulů s jádrem IRI-IIF probíhá pomocí specializovaných IRI zpráv, které informují o identitě odposlouchávaného uživatele. Typy a formát zpráv vychází z návrhu ETSI. Zprávy mají formát uspořádané n-tice, která se skládá z jednoznačného názvu modulu, aktuálního času, typu zprávy, popisu zprávy a seznamu identifikátorů. Typy zpráv spolu s událostmi, které je generují, jsou uvedeny v tabulce 3.1. Příklad zprávy je uveden na obrázku 3.4.

3.3.3 Mediační a triggerovací funkce (MF&CCTF)

Úlohou mediační funkce je centrální správa aktuálně probíhajících odposlechů. MF je konfigurována AF přes rozhraní INI1c. Tímto rozhraním se posílají pouze základní informace (identifikátor odposlechu, čas začátku a konce odposlechu a informace, zda se má zaznamenávat i obsah komunikace). Mediační funkce na základě těchto příkazů zpracovává metadata z IRI-IIF a data z CC-IIF, informace navzájem kombinuje a zasílá LEMF skrz HI2 a HI3 rozhraní.

Mediační funkce je zkombinována s triggerovací funkcí, která je zodpovědná za konfiguraci CC-IIF sond. Konfigurace probíhá přes CCCI rozhraní formou odeslílání zpráv typu žádost (ze strany MF&CCTF) a odpověď (ze strany CC-IIF).

3.3.4 Funkce odposlechu komunikace (CC-IIF)

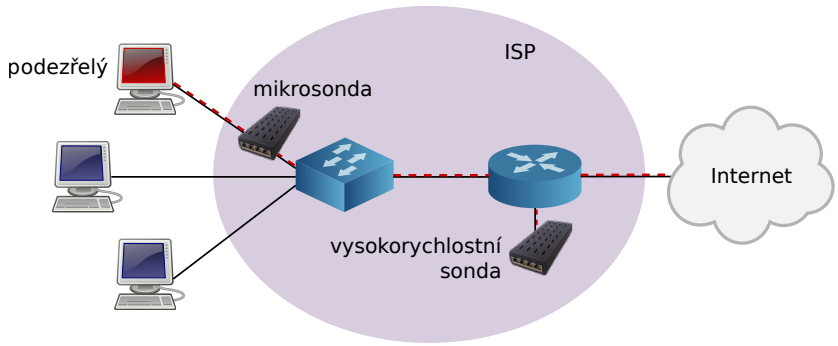
Úkolem bloku CC-IIF je sledování síťového provozu a kopírování veškerého obsahu komunikace, která se vztahuje k odposlouchávanému uživateli. Narozdíl od IRI-IIF nejsou součástí předáváných dat pouze metadata (identifikátory), CC-IIF realizuje samotný odposlech sledovaného uživatele. Konfigurace (požadavky na zahájení nebo ukončení odposlechu) je prováděna z mediační funkce rozhraním CCCI. Zachycená data se pak předávají mediační funkci přes zabezpečené INI3 rozhraní.

Blok CC-IIF může být řešen softwarově jako součást systému SLIS nebo jako samostatné zařízení (síťová sonda). Sondy jsou realizovány jako specializovaná hardwarová zařízení, která jsou schopna zaznamenat veškerou komunikaci beze ztráty jakékoliv informace.

V rámci projektu Sec6Net byly vyvinuty dva prototypy CC-IIF sond. Příklad rozmístění těchto protoypů je znázorněn na obrázku 3.5.

Prvním z nich je vysokorychlostní sonda, která je určena pro nasazení k velkým poskytovatelům Internetu (ISP, Internet Service Provider) a na páteřní linky, které mají velmi vysokou přenosovou rychlost. Tyto sondy jsou určeny pro případy, kde není možné sledovat komunikaci blízko k odposlouchávanému uživateli, např. protože přistupuje ze sítě v jiné jurisdikci, či existuje podezření prozrazení odposlechu. V takovém případě je ale nutné rozmístit sondy na všechna místa, přes která mohou být odposlouchávaná data směrována [19].

Druhým prototypem je mikrosonda, která slouží k nasazení u menších ISP nebo ideálně přímo do infrastruktury mezi ISP a koncového uživatele. V takovém případě je možné zachytit všechna data, která odposlouchávaný uživatel odeslal, bez ohledu na adresáta, a předejít tak možným pochybnostem o zdrojové IP adrese. V sítích založených na protokolu IP totiž sice lze podvrhnout IP adresu, ale odposloucháváním dat co nejblíže podezřenému je možné předejít pozdějším nejasnostem při uznávání odposlechnutých dat jako soudního důkazu [5].



Obrázek 3.5: Znázornění možného rozmístění CC-IIF sond v topologii ISP.

Kapitola 4

Uplatnění SDN v systému pro zákonné odposlechy

V rámci diplomové práce jsem navrhla rozšíření, která využijí výhod SDN a vylepší systém pro zákonné odposlechy:

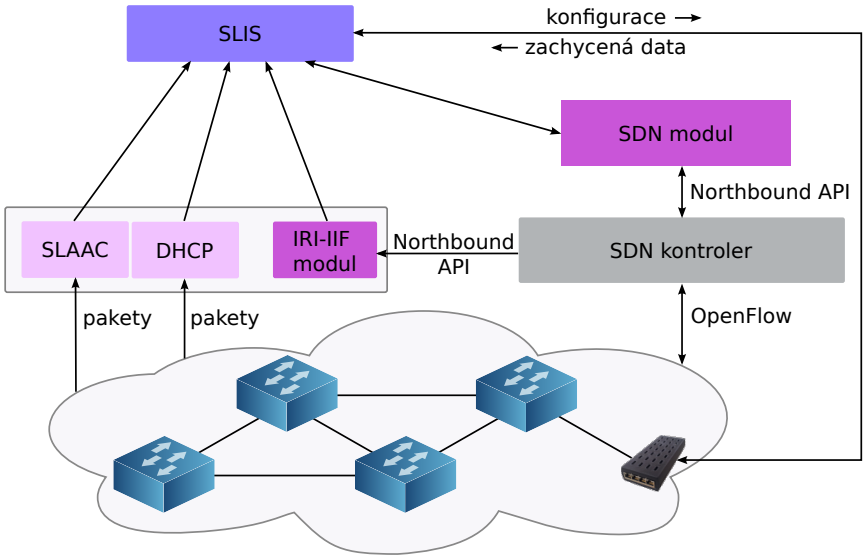
- Moduly pro IRI-IIF, které jsou určeny k získávání dynamické identity. Modul pro OpenDaylight se periodicky dotazuje kontroleru na topologii sítě a změny hlásí IRI-IIF. Modul pro POX využívá automaticky vytvářených událostí ke zjišťování topologie a změny pak také oznamuje IRI-IIF.
- Konfigurace CC-IIF sond. Jedná se především o rozšíření triggerovací funkce tak, aby odlišovala jednotlivé CC-IIF sondy a jejich pozice v topologii. S využitím SDN pak bude možné nastavit každou sondu tak, aby nedocházelo k odposlechu uživatele více sondami. Bude také možné předejít zahlcení jedné sondy přesměrováním toku odposlouchávaných dat k jiné sondě, která bude v danou chvíli méně vytížená.

V této kapitole jsou podrobně pospány cíle práce a návrh řešení. Návrhy SDN modulů pro IRI-IIF jsou uvedeny v sekci 4.1. V sekci 4.2 je popsána dynamická rekonfigurace CC-IIF sond a přepínačů v prostředí SDN. Tento návrh byl publikován ve sborníku studentské konference Excel@FIT 2015[1].

4.1 SDN moduly pro IRI-IIF

Prvním způsobem uplatnění softwarově definovaných sítí v systému pro zákonné odposlechy je návrh a implementace SDN modulů pro IRI-IIF, které získávají částečné identity koncových stanic z kontroleru. Částečná identita stroje zahrnuje identitu na síťové (L2) a linkové vrstvě (L3). Identifikátory získané z SDN kontroleru (MAC adresa, IP adresa a přepínač, ke kterému je zařízení připojeno) IRI-IIF propojuje s částečnými identitami uživatelů využívající detekované stroje (získaných z jiných modulů systému SLIS). Jádru IRI-IIF se informace o částečných identitách odesílají pomocí IRI-IIF zpráv, které byly definovány v tabulce 3.1. Zapojení modulu (`IRI-IIF modul`) do systému pro zákonné odposlechy je znázorněno na obrázku 4.1.

[1] http://excel.fit.vutbr.cz/2015/submissions/083/83.pdf



Obrázek 4.1: Schéma zapojení systému pro zákonné odposlechy, SDN kontroleru, modulu pro zjišťování dynamické identity – IRI-IIF modul (4.1), a modulu pro sledování topologie – SDN modul (4.2).

4.2 Dynamická rekonfigurace přepínačů a CC-IIF sond

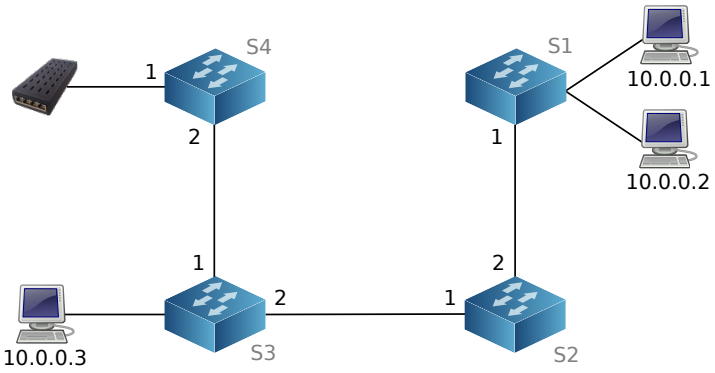
Druhým uplatněním SDN v systému pro zákonné odposlechy je využití znalosti kompletní topologie sítě a zlepšení nastavování jednotlivých CC-IIF sond. CC-IIF sondy jsou konfigurovány z bloku MF&CCTF tzv. triggerovací funkcí. Pokud přijde požadavek na zahájení odposlechu, musí tato funkce na základě topologie rozhodnout, kterou sondu nastaví a jaká data k ní bude přeposílat. V původní verzi však MF&CCTF neměla přehled o topologii sítě a proto byly všechny sondy konfigurovány stejně.

Z toho důvodu byl vytvořen nový modul do systému pro zákonné odposlechy (`SDN_modul`). Jeho zapojení do systému je znázorněno na obrázku 4.1. Modul `SDN_modul` se v pravidelných intervalech dotazuje kontroleru na aktuální topologii, vkládá pravidla pro směrování kopií odposlouchávaných dat k sondám a triggerovací funkce ve SLIS na základě informací od tohoto modulu konfiguruje jednotlivé sondy.

Na rozdíl od funkce dynamické identity, která se zajímá o identifikátory koncových zařízení, je pro triggerovací funkci nezbytné znát kompletní topologii. Jedinou informaci, kterou nejsme schopni získat dynamicky, je pozice CC-IIF sond v síti. Součástí modulu proto musí být konfigurační soubor, který specifikuje, na kterém rozhraní jsou připojeny.

4.2.1 Princip rekonfigurace zařízení

Kombinací topologie získané z kontroleru a pozice sond z konfiguračního souboru modul vytváří grafovou reprezentaci, kde vrcholy grafu jsou jednotlivá zařízení a hrany odpovádají linkám. Ve chvíli, kdy přijde požadavek na odposlech, začíná modul s konfigurací síťových zařízení. Konfigurace spočívá ve využití tří tabulek toků. Do první tabulky modul ukládá pravidla, která porovnávají procházející hlavičky paketů s IP adresou, která má být odpo-



Obrázek 4.2: Ukázková topologie se zapojenou CC-IIF sondou.

slouchávána. Pokud zdrojová nebo cílová adresa paketu odpovídají, je paket označen VLAN tagem a odeslán na výstupní port směrem k CC-IIF sondě. Následně je původní paket (bez VLAN tagu) předán třetí tabulce.

Druhá tabulka toků je na všech přepínačích stejná. Má za úkol porovnávat pakety s VLAN tagem a odesílat je směrem k CC-IIF sondě. Pravidla se prochází postupně od nejvyšší priority, proto musí být v první tabulce pravidlo s vysokou prioritou, které bude také porovnávat VLAN tag. Pakety, které budou takto označeny, pak nebude zpracovávat a pouze je předá druhé tabulce.

Třetí tabulka je plně pod správou kontroleru a přeposílá pakety k cílovým zařízením bez ohledu na pravidla v předchozích tabulkách. Tímto způsobem se tedy vytvoří duplikát paketu s VLAN tagem a původní nezměněný paket se přepošle podle pravidel z kontroleru.

4.2.2 Ukázka rekonfigurace zařízení

Modul zná aktuální topologii sítě a tak může jednoduše zjistit, ke kterému přepínači je koncové zařízení s danou IP adresou přímo připojeno. Uvažujme například topologii uvedenou na obrázku 4.2. Předpokládejme, že přišel požadavek na vložení odposlechu IP adresy 10.0.0.1. Zařízení s touto IP adresou je připojeno k přepínači S1. Na tento přepínač se vloží dvě pravidla s vysokou prioritou, která budou porovnávat danou zdrojovou a cílovou adresu v paketu. V případě, že jedna z těchto adres bude rovna 10.0.0.1, vloží se do paketu VLAN hlavička a odešle se na výstupní port 1. Ukázka pravidel je uvedena v tabulce 4.1 (porovnávání cílové IP adresy probíhá obdobně jako porovnávání zdrojové IP adresy). Na tomto i všech ostatních přepínačích se pak všechny pakety s VLAN hlavičkou budou přeposílat na rozhraní 1. Tato pravidla jsou uložena ve druhé tabulce a ukázka je uvedena v tabulce 4.2. Na přepínači S4 bude uloženo pravidlo, které ze všech paketů odesílaných na rozhraní 1 VLAN odstraní.

Sytém SLIS podporuje pravidla odposlechu konkrétní IP adresy, trojice (IP adresa, port, protokol) a pětice (zdrojová IP adresa, port, cílová adresa, port a protokol). Při zahájení odposlechu na trojici nebo pětici identifikátorů se na přepínače vloží pravidla, která budou porovnávat všechny tyto identifikátory. U pětice pak můžeme libovolně rozhodnout, zda pravidlo na označování paketů VLAN tagem vložíme na přepínač, ke kterému je připojen iniciátor komunikace, nebo na přepínač, ke kterému je připojen iniciovaný.

Prio	VLAN	IP zdroj	IP cíl	Ostatní	Akce
20	1	*	*	*	go-to tab 2
10	*	10.0.0.1	*	*	push VLAN outport 1 pop VLAN go-to tab 3
1	*	*	*	*	go-to tab 3

Tabulka 4.1: Ukázka pravidel pro odposlech v první tabulce toků. Porovnávání s hvězdičkou znamená, že na daném místě může být cokoliv. *Push/pop VLAN* značí přidání/odstranění VLAN tagu, *go-to table* znamená skoč do tabulky a *outport* odeslání paketu na výstupní port.

Prio	VLAN	IP zdroj	IP cíl	Ostatní	Akce
10	1	*	*	*	outport 1

Tabulka 4.2: Ukázka pravidel v druhé tabulce toků.

V případě, že je v topologii více CC-IIF sond, lze jednoduchým způsobem nastavit vhodnou sondu nebo rozdělovat zátěž. Každá CC-IIF sonda bude mít vlastní VLAN tag. Při přidávání odposlechu můžeme z grafu topologie zjistit, která sonda je nejblíž koncovému zařízení s danou IP adresou, a při duplikování paketů vložit VLAN tag nejbližší CC-IIF sondy. V druhé tabulce všech přepínačů pak budou pravidla, která pakety s VLAN hlavičkou odešlou směrem k této CC-IIF sondě.

V reálných zařízeních nemusí být k dispozici více tabulek toků. V takových případech je možné použít i alternativní přístupy. Jedním z nich je využití jednoho z fyzických portů přepínače, na který se bude duplikovat komunikace odposlouchávaného uživatele. Všechny pakety přijaté na tomto portu pak budou označeny a přeposlány směrem k sondě. K implementaci tohoto řešení stačí pouze jedna tabulka toků, ale nevýhodou je permanentní zablokování jednoho portu a nepřehlednost tabulky toků.

Kapitola 5

Implementace

Tato kapitola je věnována popisu implementace podle návrhu uvedeného v kapitole 4. V sekci 5.1 je uvedena implementace modulů pro získávání částečné identity a v sekci 5.2 dynamická rekonfigurace sond a síťových prvků.

5.1 Získávání částečné identity

Z kontroleru je možné získat tři typy identifikátorů pro každé zařízení: IP adresu, MAC adresu a identifikátor přepínače, ke kterému je toto zařízení připojeno. Ve chvíli, kdy je detekován začátek spojení, je nutné odeslat funkci dynamické identity IRI zprávu. Součástí zprávy je uvedená trojice identifikátorů. IRI-IIF odpovídající identifikátory propojí a tím rozšíří zjištěnou identitu tohoto zařízení.

IP adresa i MAC adresa jsou pro IRI-IIF známé identifikátory, se kterými umí pracovat. Identifikátor přepínače je ale pro systém neznámý. Z toho důvodu jsem rozšířila modul `nid.py`, kde je uložen seznam všech známých typů identifikátorů. Nově je přidán identifikátor `sdnConnector`, který obsahuje ID přepínače a rozhraní, na kterém je zařízení připojeno.

Pro implementaci jsem zvolila kontrolery OpenDaylight a POX. Nový modul pro OpenDaylight se jmenuje `odl_iri` a je napsán v jazyce Python 3. Modul pro POX `pox_iri` je napsán v jazyce Python 2. Oba tyto moduly jsou rozepsány v následujících sekcích.

5.1.1 OpenDaylight

OpenDaylight poskytuje velké množství modulů. Ve výchozím nastavení ale není žádný nainstalovaný. Při prvním spuštění je nutné doinstalovat moduly, které pak zajistí korektní chování L2 přepínačů a zjišťování topologie sítě. Jedná se o:

- `odl-dlux-core`
  GUI, ve kterém lze přehledně zobrazit topologii a nainstalované toky na jednotlivých přepínačích.
- `odl-openflowplugin-all`
  Rozšíření, které umožňuje používat OpenFlow pro manipulaci s toky.
- `odl-l2switch-all`
  Modul má na starosti chování přepínačů v topologii podle (popsáno v sekci 2.4.1).

Součástí L2 přepínače je také `address tracker` (učí se IP adresu a MAC adresu koncových stanic) a `host tracker` (sleduje pozici koncových zařízení v topologii)

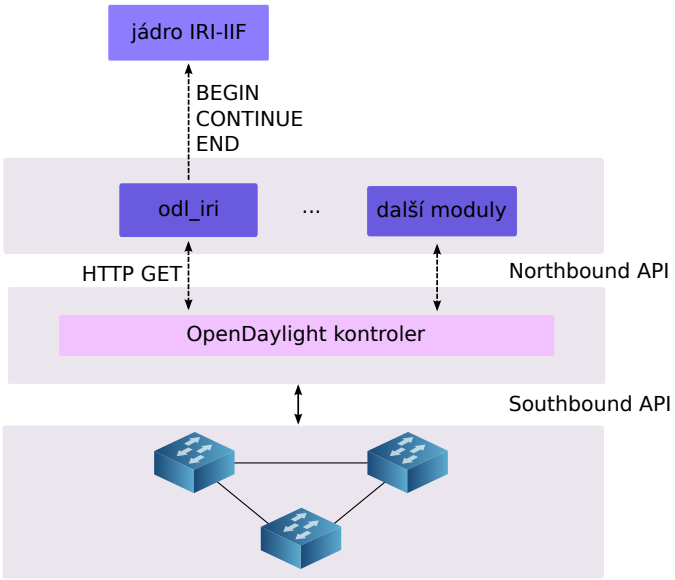
- `odl-restconf`
  Rozšíření, které poskytuje RESTCONF (HTTP) rozhraní, pomocí kterého lze komunikovat s kontrolerem bez GUI

Důležitou součástí `odl-l2switch` je tzv. *topology manager*, který poskytuje HTTP rozhraní pro zjištění topologie sítě. Topology manager je přístupný na adrese

`http://localhost:8181/restconf/operational/network-topology:network-topology/topology/flow:1`

V nově vytvořeném modulu `odl_iri` se pravidelně odesílá topology manageru HTTP žádost. Kontroler odpovídá zasláním XML, ve kterém jsou mimo jiné uvedena všechna koncová zařízení spolu se třemi typy identifikátorů: IP adresou, MAC adresou a identifikátorem přepínače, ke kterému je toto zařízení připojeno.

Modul `odl_iri` vytvoří seznam aktuálně zjištěných koncových zařízení a ten porovná s přechozím stavem. Pokud se v seznamu objevilo nové zařízení, je detekován začátek spojení a jádru IRI-IIF se odešle zpráva IRI BEGIN. Jestliže naopak nějaké zařízení v novém seznamu není, detekuje se konec spojení a je odeslána zpráva IRI END. Ve všech zprávách je uvedena trojice částečných identifikátorů daného zařízení. Propojení kontroleru, modulu `odl_iri` a IRI-IIF je znázorněno na obrázku 5.1.



Obrázek 5.1: Znázornění propojení modulu `odl_iri`, kontroleru OpenDaylight a systému pro zákonné odposlechy.

5.1.2 POX

POX je navržen modulárně a je tedy možné vytvořit samostatný modul, který bude komunikovat s moduly pro řízení sítě. Všechny moduly jsou řízeny událostmi, které se vytváří automaticky při změně topologie nebo manuálně. Odposloucháváním těchto událostí spolu s vytvořením některých vlastních je pak možné okamžitě získat informace o změnách v topologii. Aby bylo možné sledovat změny v topologii, je nutné spustit POX se třemi moduly:

- `openflow.discovery`
  Modul zjišťuje topologii pomocí LLDP paketů. Při změně stavu linky (link up / link down) vytvoří událost LinkEvent, při zjištění nového přepínače vytvoří události *ConnectionUp* a *ConnectionDown*.

- `host_tracker`
  Modul, který sleduje koncová zařízení v síti (vždy minimálně IP adresu, MAC adresu a přepínač, ke kterému je zařízení připojeno). Při změně stavu (zjištění nového koncového zařízení / timeout) je vytvořena událost *HostEvent*.

- `forwarding.l2_learning`
  Modul, který vytváří z přepínačů v topologii L2 přepínače. Pokud do přepínače přijde paket, který nemá záznam v tabulce toků, je odeslán do kontroleru. Kontroler pak upraví informace o koncových stanicích (pokud jde o nové zařízení) a vloží nové pravidlo do přepínače.

K získávání informací o topologii jsem vytvořila modul `pox_iri`. Na obrázku 5.2 je znázorněno propojení jedotlivých modulů v rámci POX kontroleru a IRI-IIF. Tento modul odposlouchává události HostEvent a UpdateIpEvent. Při zachycení události HostEvent mohou nastat situace *join*, *move* a *leave*. *Join* značí zjištění nového koncového zařízení a ihned po zachycení této události se IRI-IIF odešle zpráva BEGIN. *Move* znamená přemístění známého koncového zařízení v topologii a IRI-IIF jsou odeslány zprávy END a BEGIN. Událost *leave* značí, že dané koncové zařízení už není v síti aktivní a při zachycení této události se odesílá zpráva END.

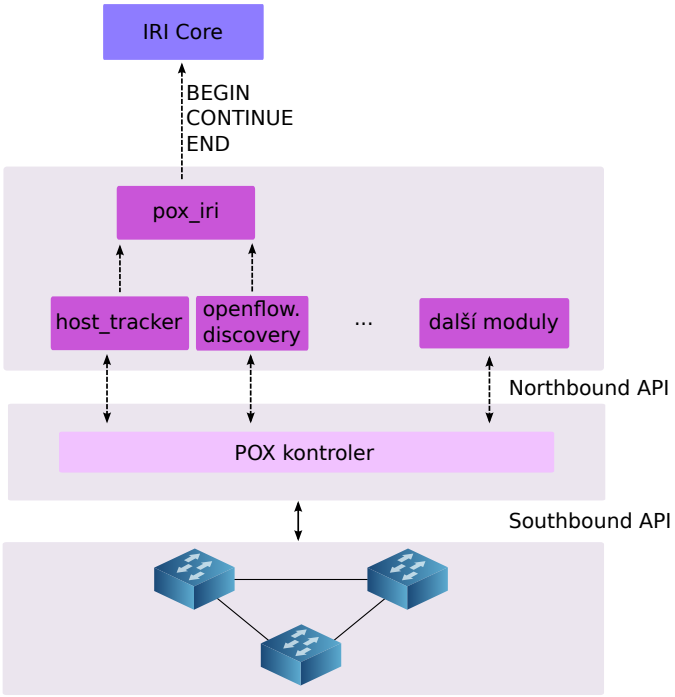
POX má proti OpenDaylight nespornou výhodu v tom, že změny v topologii samy vytváří události. Můžeme tak pouze čekat na událost a nezatěžovat kontroler opakovanými dotazy jako v případě OpenDaylight, kde se v nejhorším případě dozvíme změnu až po uplynutí intervalu mezi jednotlivými dotazy (ve výchozím nastavení 1 s).

5.2 Dynamická rekonfigurace přepínačů a sond

Dynamická rekonfigurace přepínačů a sond je postavena na označování duplikovaných paketů VLAN tagem sondy, ke které bude paket směrován. Aby bylo možné nezávisle na sobě vkládat a odstraňovat pravidla pro odposlechy a zároveň v pravidelných intervalech kontrolovat topologii, je nutné dynamické zjišťování stavu přepínačů a pravidel. Před každým vložením nebo změnou pravidla se provede sekvence kroků, které zajistí co nejlepší směrování paketů označených VLAN tagem.

5.2.1 Vkládání odposlechu ze SLIS

Pro vkládání a odstraňování pravidel pro označování zájmových paketů jsem vytvořila modul `ODL_trigger`. Tento modul má za úkol poskytovat rozhraní mezi MF&CCTF a OpenDaylight. Komunikace mezi jednotlivými částmi systému je uvedena na obrázku 5.3. Ve chvíli,

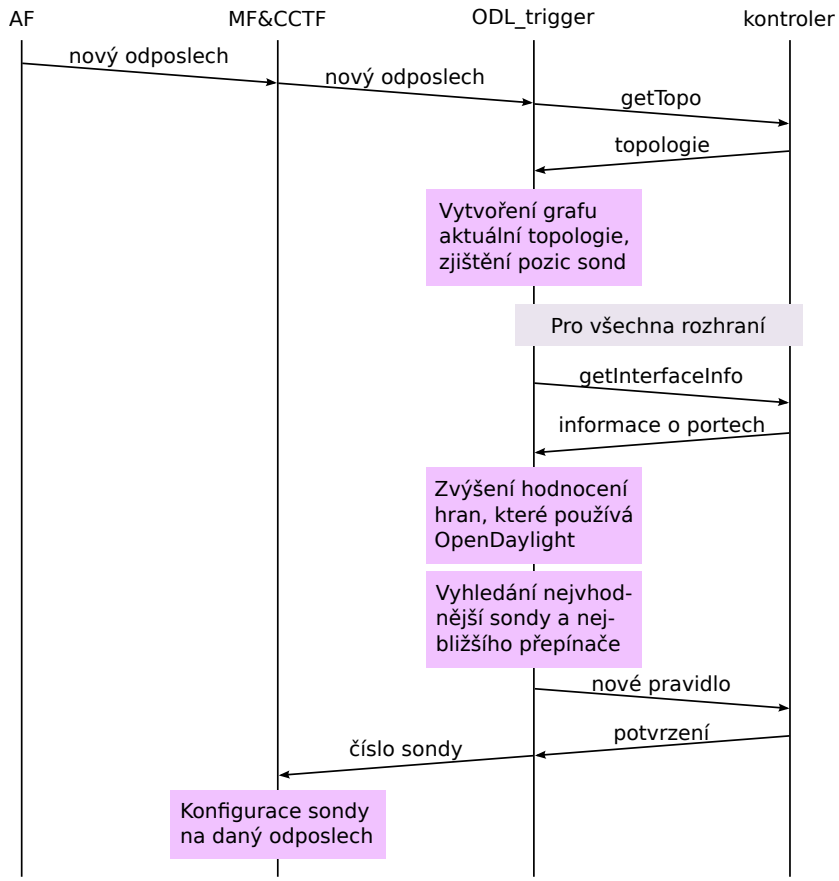


Obrázek 5.2: Znázornění propojení modulu `pox_iri`, kontroleru POX a systému pro zákonné odposlechy.

kdy do mediační funkce přijde z administrační funkce požadavek na zahájení odposlechu, zavolá CCTF funkci `InsertIntercept` z modulu `ODL_trigger`. Funkce `InsertIntercept` zjistí aktuální topologii, vytvoří odpovídající graf a zvýší hodnocení hran, které používá OpenDaylight.

Při výběru sondy, která bude zachytávat komunikaci daného zařízení, se provádí jednoduché vyvažování zátěže. Nejdříve se z kontroleru zjistí informace o topologii a pozice sond (`getTopology`, `getLoopFreeTopo`, `getProbePosition`). Ve chvíli, kdy je vytvořen graf topologie, se najdou nejkratší cesty mezi zařízením, které chceme odposlouchávat, a všemi sondami. Pravidla pro odposlech se implicitně vkládají na sondu, která má nejnižší součet hodnocení hran na cestě k odposlouchávanému zařízení. Pokud je ale rozdíl v počtu odposlechů některých sond více než trojnásobný, vybere se sonda s méně odposlechy nezávisle na pozici v topologii. Tím zajistíme, že se budou častěji využívat linky, které OpenDaylight nevyužívá pro provoz neodposlouchávaných paketů.

Pokud program zná nejvhodnější sondu a nejbližší přepínač, může nahrát pravidla pro odposlech. Tvar pravidel pro odposlech je uveden v tabulce 5.3, přičemž bude vložen VLAN tag dané sondy. `InsertIntercept` vrací ID nejvhodnější sondy. Triggerovací funkce na základě návratové hodnoty nastaví CC-IIF sondu na odposlech přes rozhraní CCCI.



Obrázek 5.3: Komunikace mezi jednotlivými částmi systému při vkládání odposlechu.

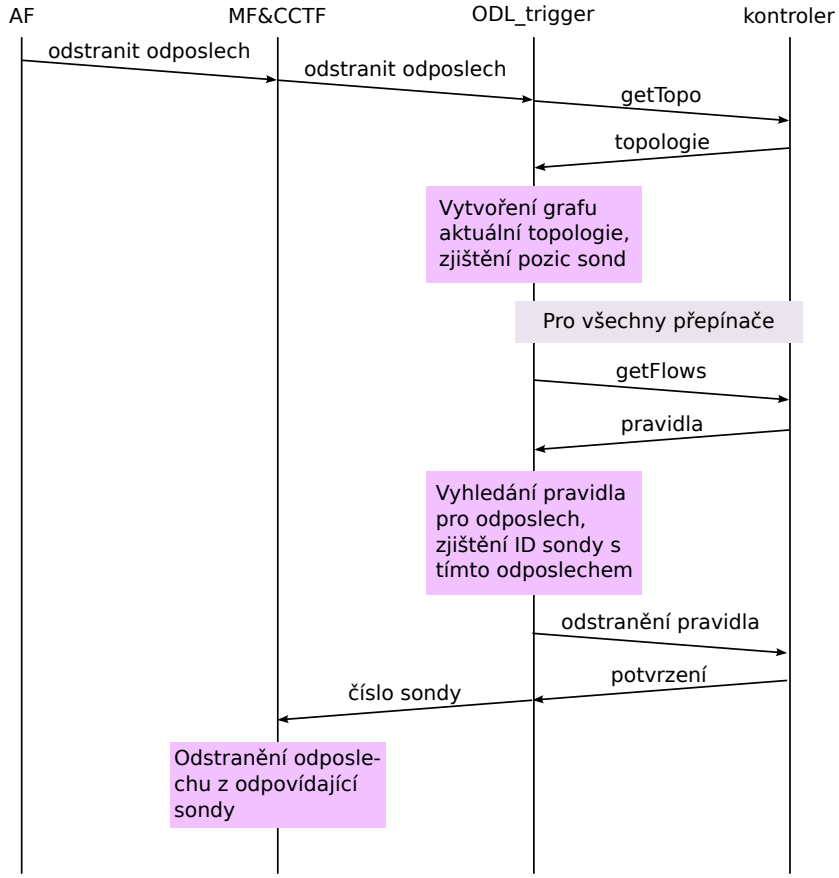
5.2.2 Odstraňování odposlechu ze SLIS

Odstranění odposlechu ze systému je jednodušší než vkládání. Komunikace mezi SLIS, modulem `ODL_trigger` a kontrolerem je znázorněna na obrázku 5.4. Ve chvíli, kde mediační funkci přijde požadavek na ukončení odposlechu, spustí se funkce `DeleteIntercept` z modulu `ODL_trigger`. `DeleteIntercept` načte z kontroleru topologii sítě a zjistí pozici sond (`getTopology`, `getProbePosition`). Poté získá všechny aktuální pravidla v první tabulce (`getFlows`).

V načtených pravidlech nalezne odpovídající dvě pravidla (v případě odstraňování pětice pouze jedno). Ze seznamu akcí v pravidle zjistí VLAN tag sondy, ke které byly zasílány označené pakety. `DeleteIntercept` pak odstraní pravidla z přepínače a číslo sondy vrátí MF&CCTF. Triggerovací funkce na základě návratové hodnoty odstraní odposlech i z odpovídající CC-IIF sondy.

5.2.3 Dynamická rekonfigurace sond

Pro zjišťování změn v topologii jsem vytvořila modul pro MF&CCTF `ODL_trigger`, který je implementován v jazyce Python. Má za úkol v pravidelných intervalech zjišťovat topo-



Obrázek 5.4: Komunikace mezi jednotlivými částmi systému při odstraňování odposlechu.

logii sítě a v případě změny rekonfigurovat přepínače a zajistit, aby triggerovací funkce překonfigurovala CC-IIF sondy.

`ODL_trigger` se spouští na samostatném vlákně v rámci MF&CCTF funkce. Periodicky zjišťuje aktuální topologii a pozici sond (`getTopology`, `getProbesPosition`). Pokud nebyla nalezena žádná sonda, daný běh se ukončí. Pokud je nalezena alespoň jedna sonda, vytvoří se z topologie *networkx* graf. V prvním běhu program nahraje inicializační pravidla do první tabulky toků (`setInitialFlowRules`) a pravidla pro přeposílání paketů označených VLAN tagem (`setVlanForwadringFlows`). Nakonec uloží graf topologie do souboru JSON (`saveGraph`).

Všechny další běhy pak porovnávají aktuální graf s tím, který byl při poslední změně uložen do JSON souboru. V případě, že došlo k jakékoliv změně topologie, démon postupně provede následující kroky:

1. Zjistí, zda jsou na všech přepínačích nahrána inicializační pravidla v první tabulce. V případě, že byl připojen nový přepínač a pravidla neobsahuje, nahrají se (`setInitialFlowRules`).

2. Aktualizuje pravidla v druhé tabulce toků (`renewVlanForwardingFlows`). Pokud by

byla připojena nová sonda, nahraje se nové pravidlo, které bude směrovat pakety označené VLAN tagem k této sondě. Pokud byla některá sonda odpojena, odstraní se pravidlo na přeposílání paketů.

3. Zkontroluje pravidla na označování paketů v první tabulce toků (`renewInterceptFlows`). Pro každé pravidlo pro odposlech je nutné zkontrolovat VLAN tag a výstupní rozhraní, na které se označené pakety odesílají. V případě, že se změní pouze rozhraní a VLAN tag zůstane stejný, upraví se pravidlo a sonda zůstane nastavená pořád stejně. Pokud se ale pakety mají začít zasílat na jinou sondu, musí se kromě úpravy pravidla také odstranit odposlech z původní sondy a vložit na novou sondu. `ODL_trigger` odešle triggerovací funkci zprávu, ze které sondy se má odposlech odstranit a na kterou se má nahrát. Triggerovací funkce na základě těchto informací odstraní odposlech z první sondy a vloží ho na druhou.

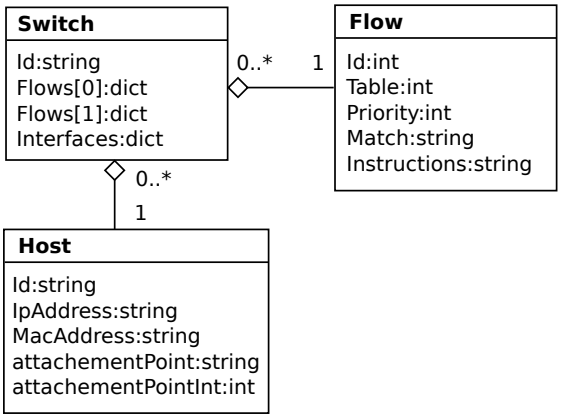
Pravidla v první i druhé tabulce toků ovlivňuje způsob změny topologie. Obecně může nastat jedna nebo více z následujících situací:

- přidání linky
  - Pravidla se nemění, ale při přidávání nového odposlechu se bude brát v úvahu i nová linka.
- výpadek linky
  - Zkontrolují se všechna aktuální pravidla, která odesílají paket na některé výstupní rozhraní a také cesty z jednotlivých přepínačů k sondám.
  - Výpadek linky, která jako jediná vede přímo k sondě, způsobí odstranění pravidel pro odposlech s VLAN tagem dané sondy. Všechna tato pravidla se aktualizují, změní VLAN tag a budou přeposílat pakety na rozhraní k některé jiné sondě.
  - Výpadek linky mezi přepínači ovlivní pouze pravidla, která jsou nahraná na těchto přepínačích. V případě, že se linka používala pro přeposílání označených paketů, musí se upravit všechna pravidla pro odposlech v první tabulce toků a pravidla pro přeposílání paketů s VLAN tagem ve druhé tabulce toků.
- přidání přepínače
  - Pokud přepínač ještě nebyl zapojen v topologii, nahrají se inicializační pravidla a pravidla pro přeposílání paketů s VLAN tagem.
  - Pokud přepínač byl zapojen v topologii a obsahuje nějaká pravidla na označování paketů, porovnají se se seznamem aktuálních odposlechů. Pokud byl odposlech mezitím zrušen, odstraní se i z přepínače. Pravidla pro přeposílání paketů s VLAN tagem se zkontrolují a případně aktualizují.
- odstranění přepínače
  - Podobně jako u výpadku linky se zkontrolují všechna pravidla a případně se aktualizují výstupní porty.
- přidání a odstranění koncového zařízení
  - Pravidla se nemění.

- přidání CC-IIF sondy
  - Do všech přepínačů se nahraje pravidlo pro přeposílání paketů s VLAN tagem nové sondy.
- odstranění CC-IIF sondy
  - Ze všech přepínačů se odstraní pravidlo pro přeposílání paketů s VLAN tagem dané sondy.
  - Pravidla pro odposlech, které označovaly pakety VLAN tagem této sondy, se aktualizují (změní se VLAN tag na tag jedné z dostupných CC-IIF sond a aktualizuje se výstupní rozhraní).

5.2.4 Použité třídy

V kódu je nutné často manipulovat s přepínači, koncovými zařízeními a toky. Pro zjednodušení manipulace s těmito prvky a jednodušší přístup k jejich vlastnostem jsou použity třídy Switch, Host a Flow, které jsou znázorněny na UML diagramu 5.5.



Obrázek 5.5: UML diagram použitých tříd.

Ve slovníku *Flows[0]* ve třídě Switch jsou uložena všechna pravidla z první tabulky toků tohoto přepínače a ve *Flows[1]* pravidla z druhé tabulky. *Interfaces* je slovník, ve kterém jsou uvedena všechna rozhraní tohoto přepínače spolu s identifikátorem zařízení, ke kterému vedou. Položka *attachementPoint* ve třídě Host značí ID přepínače, ke kterému je zařízení připojeno a *attachementPointInt* specifikuje rozhraní tohoto přepínače.

5.2.5 Funkce pro zjišťování topologie

Modul `OpenDaylight.py` obsahuje několik funkcí na zjišťování aktuální topologie. Tyto funkce jsou používány jak při vkládání nového odposlechu, tak při zjišťování změn v topologii. Nejdůležitější z nich jsou následující:

- `getTopology`
  Funkce `getTopology` slouží k získání informací o přepínačích, koncových zařízeních a

propojujích linkách. Odesílá HTTP GET požadavek kontroleru, který odpoví zasláním XML s požadovanými informacemi. V XML jsou uvedeny všechny přepínače, koncová zařízení a propojení mezi nimi. Podle této zprávy vytvoří instance tříd Switch a Host, kterým přiřadí zjištěné vlastnosti. Zároveň se vytvoří grafová reprezentace topologie pomocí *networkx*. URL HTTP žádosti je stejná jako při zjišťování koncových zařízení v modulu `odl_iri` (sekce 5.1.1).

- `getLoopFreeTopology`
  Tato funkce má za úkol zjistit, které porty `LoopRemover` zablokoval a OpenDaylight je nepoužívá. Z kontroleru je nutné dalšími HTTP pořadavky získat XML s detaily jednotlivých rozhraní přepínačů. Pro zjištění stavu rozhraní `1` na přepínači `openflow:1` je nutné odeslat požadavek GET na uvedenou adresu.

  `http://localhost:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:1/node-connector/openflow:1:1`

  V XML je u každého portu uveden status:

  - *forwarding* – značí, že linka je aktivní a používá se pro zasílání paketů
  - *discarding* – značí, že linka je neaktivní a pakety se přes ni neposílají

  V případě, že linka má na obou koncích stav portu *forwarding*, je odpovídající hraně v grafu zvýšena váha na hodnotu 2. Při hledání nejkratší cesty v grafu pak budou preferovány linky, které nejsou zatíženy žádným provozem.

  Váhu hran v grafu lze zvýšit také v konfiguračním souboru `ProbeConfig`. Váha hrany mezi přepínačem 1 a 2 se zvýší na 1000 vložením řádku:

  ```
  weight openflow:1 openflow:2 1000
  ```

  Při hledání cesty se pak bude kromě kostry grafu brát ohled také na administrátorem zadaný požadavek. Zvýšení hrany v grafu může sloužit například pro označení nespolehlivé nebo jinak vytížené linky. V případě, že je váha hrany vyšší než 1000, bude se používat pouze v případě, že neexistuje žádná jiná cesta.

- `getProbePosition`
  Tato funkce načítá polohy CC-IIF sond z konfiguračního souboru `ProbeConfig`. Záznam o jedné sondě má následující tvar:

  ```
  probe:1 openflow:2:2 1
  ```

  Ke každé sondě je uveden přepínač a port, ke kterému je připojena. Každá sonda má vlastní číselný identifikátor (první sloupec) a vlastní VLAN tag (třetí sloupec), kterým budou označovány všechny pakety pro tuto sondu. Údaje v `ProbeConfig` má na starosti správce sítě nebo programátor, protože program nemá možnost kontroly zapojení sond. Pokud je pozice nebo identifikační číslo uvedeno chybně, budou se pakety označené VLAN tagem odesílat na špatný přepínač, případně se bude rekonfigurovat jiná sonda.

5.2.6 Funkce pro zjišťování pravidel

Funkce pro zjišťování pravidel v tabulkách toků jsou naimplementovány jako součást modulu `OpenDaylight.py`. Mezi nejdůležitější z nich patří tyto funkce:

- `getFlows`
  Funkce má za úkol zjistit pravidla toků v prvních a druhých tabulkách všech přepínačů. Opět se využívá HTTP GET požadavek, který je nutný vytvořit pro každý přepínač, ze kterého chceme zjistit pravidla v tabulce toků. Příkladem je požadavek na uvedenou URL, kdy kontroler odpoví zaláním XML se všemi pravidly v tabulce 0 přepínače `openflow:1`.

  `http://localhost:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:1/table/0`

  Pro každý tok je vytvořena instance třídy Flow. V upravené formě jsou pak zpracována pole pro porovnávání paketu (*match*) a akce, které se mají s daným paketem provést (*apply-actions*, *go-to table*). Každá instance je uložena do `Switch.Flows[0]` nebo `Switch.Flows[1]` podle toho, zda se jedná o pravidla v první nebo druhé tabulce.

- `setInitialFlowRules`
  Tato funkce vloží inicializační pravidla do tabulky 0 v každém přepínači. První pravidlo má nejnižší prioritu, neobsahuje žádné porovnání a předává pakety rovnou třetí tabulce, kde jsou zpracovány podle pravidel nastavených OpenDaylight kontrolerem.

  Druhé pravidlo zjišťuje, zda paket obsahuje VLAN tag a pokud ano, je paket předán druhé tabulce. Každé přidání nebo odebrání pravidla znamená jeden HTTP PUT požadavek. Inicializační pravidla se nahrávají pouze v případě, že na přepínačích neexistují a i při změně topologie zůstávají stejné. Přesný tvar obou pravidel je uveden v tabulce 5.1.

ID	Prio	VLAN	IP zdroj	IP cíl	Ostatní	Akce
1	2	*	*	*	*	go-to tab 3
2	10	1	*	*	*	go-to tab 2

Tabulka 5.1: Inicializační pravidla v první tabulce.

  Všechna pravidla, která se mají nahrát do přepínačů, je nutné odeslat kontroleru na odpovídající URL. V URL je uveden přepínač, tabulka a ID pravidla. Číslo tabulky a ID toku se musí shodovat s údaji v zasílaném XML, ve kterém se nachází informace z tabulky 5.1. Aby bylo pravilo opravdu nahráno na přepínač, musí se tok nastavit jako striktní.

  `http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0`

- `setVlanForwardingFlow`
  Funkce `setVlanForwardingFlow` má za úkol vkládat pravidla do druhé tabulky toků. Pro každý přepínač se spočte nejkratší cesta ke všem dostupným sondám a zjistí se port, který vede směrem na následující přepínač na této cestě. Pro každou sondu se pak vloží samostatné pravidlo, které bude pakety s určeným VLAN tagem odesílat na výstupní port směrem k sondě. Speciálním případem jsou přepínače, ke kterým je přímo připojena CC-IIF sonda. Pro danou sondu je nutné vložit pravidlo, které před odesláním paketu na port odstraní VLAN tag.

ID	Prio	VLAN	IP zdroj	IP cíl	Ostatní	Akce
1	10	1	*	*	*	outport 3
2	10	2	*	*	*	pop VLAN outport 4

Tabulka 5.2: Pravidla ve druhé tabulce.

- `renewVlanForwardingFlows`
  Funkce renewVlanForwardingFlows má za úkol přenastavit pravidla v druhé tabulce v případě, že se změní topologie. Tato funkce nejdřív odstraní původní pravidla z tabulek a nahradí je aktuálními. Pro každý switch je zapotřebí jeden HTTP DELETE a jeden HTTP PUT požadavek.

- `setInterceptFlow`
  Odposlech v systému SLIS může být cílený na IP adresu, trojici (IP adresa, port, transportní protokol) nebo pětici (zdrovová a cílová IP adresa, zdrojový a cílový port, transportní protokol). Úkolem funkce `setInterceptFlow` je vložit na určený přepínač pravidla, která rozpoznají pakety sledovaného uživatele, vytvoří duplikát a označí ho VLAN tagem jedné z CC-IIF sond.

  Pravidla se vkládají vždy na přepínač, ke kterému je zařízení připojeno. Pro IP adresu a trojici se vždy vkládají dvě pravidla. První porovnává zdrojovou IP adresu (případně zdrojovou IP adresu, zdrojový port a protokol) a druhé porovnává cílovou IP adresu (případně cílovou IP adresu, cílový port a protokol). V případě pětice se vkládá pouze jedno pravidlo na přepínač, ke kterému je připojeno zařízení se zdrojovou IP adresou.

  Pokud je paket pozitivně porovnán s některým z pravidel pro odposlech, je nutné označit ho VLAN tagem. Po vložení VLAN tagu je paket odeslán na výstupní rozhraní, které vede směrem k odpovídající sondě. V dalším kroku se VLAN tag odstraní a paket se předá poslední tabulce, která se postará, aby byl nezměněný doručen původnímu příjemci. Díky tomu je odposlouchávání paketů nepozorovatelné pro zdrojovou i cílovou stanici.

- `renewInterceptFlow`
  Tato funkce se stejně jako `renewVlanForwardingFlows` volá v případě, že se změnila topologie sítě. Postupně projde všechna pravidla pro odposlechy v první tabulce a zkontroluje, jestli je VLAN tag a výstupní rozhraní stále aktuální. V případě, že se jedna z těchto položek změní, je nutné pravidlo změnit. Do OpenFlow přepínačů není možné nahrávat pravidla se stejným porovnáním, takže je nutné neaktuální pravidlo odstranit a nahradit ho novým. Zpoždění, vzniklé opětovným nahráním pravidla, je důkladněji zkoumáno v kapitole 6.

ID	Prio	IP zdroj	IP cíl	Zdroj port	Cíl port	Proto	Ostatní	Akce
3	5	10.0.0.1	*	*	*	*	*	push VLAN outport 1 pop VLAN go-to tab 3
5	5	10.0.0.1	*	80	*	TCP	*	push VLAN outport 1 pop VLAN go-to tab 3
7	5	10.0.0.3	10.0.0.4	80	35698	TCP	*	outport 3 go-to tab 3

Tabulka 5.3: Pravidla na označování zájmového provozu VLAN tagem. První pravidlo je zaměřeno pouze na IP adresu, druhé na trojici a třetí na pětici. K prvnímu a druhému pravidlu bude vždy existovat analogické, které ale bude porovnávat cílovou IP adresu (a případně cílový port a protokol).

Kapitola 6

Testování

Tato kapitola je věnována ověření implementace navržených rozšíření systému pro zákonné odposlechy. V sekci 6.1 je popsáno spuštění všech součástí systému. Sekce 6.2 popisuje testování modulů pro IRI-IIF a sekce 6.3 testování dynamické rekonfigurace přepínačů a sond. V sekci 6.4 jsou uvedeny výsledky měření výkonnosti systému.

6.1 Příprava a spuštění systému

Pro účely ladění a testování propojení SDN a SLIS byl využíván program *mininet*[1]. Mininet umožňuje vytvořit libovolnou virtuální síť s OpenFlow přepínači a koncovými stanicemi. Dovoluje připojení přepínačů k externímu kontroleru a případně i k reálné síti. V mininetu je možné také vytvářet vlastní topologie. Mininet s vlastní topologií se spouští příkazem

```
root# mn --custom MyTopo.py --topo MyTopo --mac --controller remote,
ip=127.0.0.1,port=6653 --switch ovsk,protocols=OpenFlow13
```

Po zadání příkazu se spustí jednotlivé přepínače a pokusí připojit ke kontroleru na adrese `127.0.0.1:6653`. Důležité je specifikovat verzi OpenFlow, kterou budou přepínače používat. Bez parametru `OpenFlow13` se spustí virtuální přepínače, které ale budou komunikovat pouze pomocí OpenFlow v1.0 a neumožní používat více tabulek toků.

V rámci testování jsem vytvořila několik topologií. V těchto topologiích jsem virtuální síť propojila s fyzickým zařízením, na kterém běžel systém pro zákonné odposlechy. Pro přidání reálného zařízení se na vybraný přepínač vloží fyzický port, který k tomuto zařízení vede.

```
root# ovs-vsctl add-port s1 eth0
```

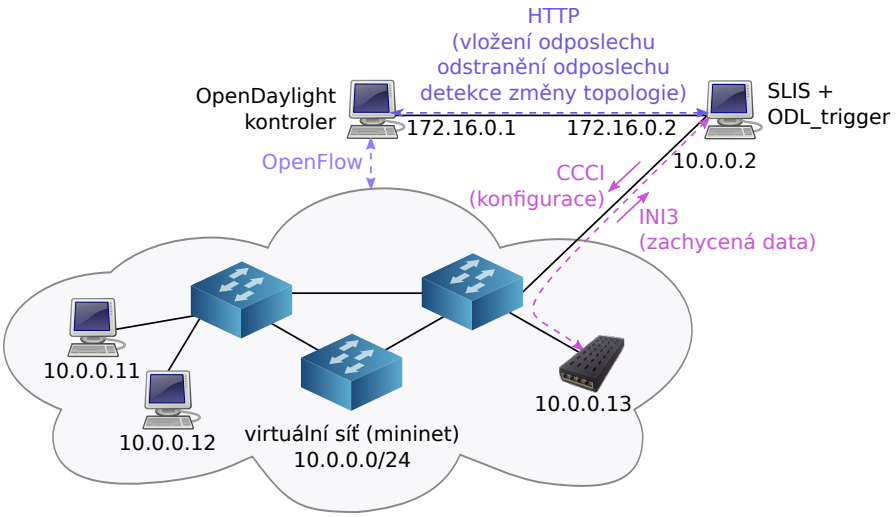
Ladění a testování vytvořených programů probíhalo s využitím kontroleru OpenDaylight a POX. Kontroler POX se spouští příkazem

```
root# sudo ./pox.py samples.pretty_log openflow.discovery host_tracker
forwarding.l2_learning misc.pox_iri
```

Poté je nutné spustit SLIS podle návodu níže. Ve chvíli, kdy jsou všechny části spuštěny, by modul `pox_iri` měl začít odesílat IRI zpávy.

Pro spuštění kontroleru OpenDaylight je nutné mít nainstalovaný balík JRE (Java Runtime Enviroment 7). OpenDaylight kontroler se spouští příkazem

[1] http://mininet.org/



Obrázek 6.1: Propojení mininetu, OpenDaylight a SLIS.

```
root# sudo ./bin/karaf
```

Ve chvíli, kdy OpenDaylight běží, je nutné nainstalovat další součásti (tento krok stačí provést pouze při prvním spuštění). Instalace potřebných součástí se provede v příkazové řádce kontroleru

```
root# feature:install odl-dlux-core odl-openflowplugin-all odl-l2switch-all
odl-restconf odl-nsf-all odl-adsal-compatibility odl-adsal-all
```

Po instalaci vytvoří OpenDaylight XML soubory, kterými je možné specifikovat chování těchto součástí. V souboru `58-l2switchmain.xml` se musí změnit mód L2 přepínačů na reaktivní a výchozí tabulku, do které OpenDaylight vkládá pravidla, je nutné upravit na 2. V souboru `54-arphandler.xml` se také musí změnit výchozí tabulka pro vkládání pravidel na 2.

OpenDaylight detekuje koncová zařízení až ve chvíli, kdy začnout komunikovat. Pokud chceme zjistit celou topologii, nejlepším krokem je zadat příkaz `pingall` v mininetu. Tímto příkazem se všechny koncové stanice pokusí zjistit stav ostatních zařízení pomocí příkazu `ping`. Kontroler se naučí topologii a do tabulky toků vloží pravidla, která budou porovnávat MAC adresy zdrojového a cílového zařízení. Každé pravidlo má specifikovaný výstupní port, na který se odpovídající paket zašle.

Centrální zařízení SLIS se spouští na fyzickém stroji, který má spojení s kontrolerem i s přepínačem ve virtuální síti (znázorněno na obrázku 6.1). První rozhraní (`172.16.0.2`) je nutné pro vkládání a odstraňování pravidel, která označují odposlouchávané pakety VLAN tagem, a zjišťování změn v topologii. Druhé rozhraní (`10.0.0.2`) je potřebné pro nastavování CC-IIF sond a ukládání zachycených dat. SLIS se spouští příkazem:

```
root# /etc/init.d/slis start
```

Posledním krokem je zprovoznění sond. Hardwarové sondy lze do mininetu vložit podobně jako je připojen SLIS. V experimentech jsem však použila softwarové sondy, které

byly spuštěny na virtuálních koncových zařízeních, protože mininet umožňuje spustit libovolný skript na jednotlivých zařízeních. Každé sondě je nutné upravit konfigurační soubor a specifikovat ID sondy a IP adresu zařízení, na kterém je spuštěn SLIS. Softwarové sondy se pak spustí příkazy:

```
root# ./cciif.py
root# ./cc-input.py
```

V tuto chvíli by měly běžet všechny části systému. Ve webovém rozhraní SLIS lze vkládat a odstraňovat odposlechy. Modul `ODL_trigger` kontroluje topologii mininetu a v případě, že nastala změna, kontaktuje SLIS s informací, kterého odposlechu se změna týká a na kterou sondu se má odposlech přenastavit.

6.1.1 Omezení systému

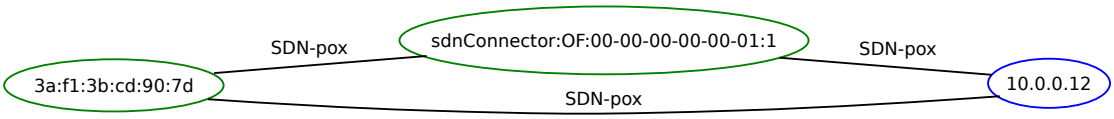
Omezení systému vychází z vlastností OpenDaylight, které jsou popsány v kapitole 2.4.1. Mezi hlavní omezení patří nemožnost přesouvání koncových stanic mezi různými přepínači. IP adresa, MAC adresa a přepínač, ke kterému je zařízení připojeno, jsou tedy v podstatě perzistentní identifkátory.

Dále nelze propojit ne-SDN síť s OpenDaylight. Systém je možné propojit se směrovačem, který bude fungovat jako výchozí brána, ale kontroler nemá přehled o zařízeních za bránou. Z toho vyplývá omezení vkládání odposlechů pro SLIS. V původním stavu totiž systém umožňoval vložit odposlech na jakoukoliv IP adresu. Při použití OpenDaylight a SLIS je ale možné vložit pouze odposlechy na IP adresy, které jsou aktuálně připojeny v lokální síti.

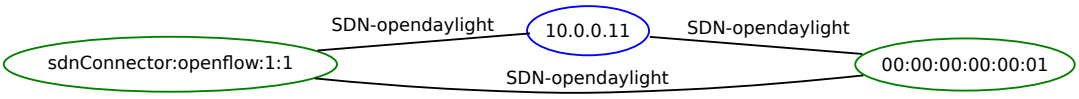
6.2 Rozšíření částečné identity

Cílem tohoto experimentu bylo ověřit funkčnost nových modulů pro IRI-IIF. Testování nejdříve probíhalo na zařízeních zapojených podle postupu v 6.1. Oba moduly pro IRI-IIF ale lze spouštět i bez modulu `ODL_trigger`. Druhá část experimentu ověřila chování bez modulu `ODL_trigger`.

Funkčnost modulů se dá ověřit ve webovém rozhraní SLIS. Po spuštění systému se rozšíří identita zařízení, která OpenDaylight nebo POX zná. IRI-IIF jsou odeslány zprávy s identifikátory těchto zařízení (IP adresa, MAC adresa a sdnConnector – přepínač, ke kterému je zařízení připojeno). Na obrázku 6.2 a 6.3 je ukázán záznam jedné zprávy, kterou IRI-IIF rozšířila identitu zařízení.



Obrázek 6.2: Identita zařízení s IP adresou 10.0.0.12 a MAC adresou 3a:f1:3b:cd:90:7d. Identifikátor sdnConnector značí přepínač a rozhraní, ke kterému je toto zařízení připojeno.



Obrázek 6.3: Identita zařízení s IP adresou 10.0.0.11 a MAC adresou 00:00:00:00:00:01, které je připojeno k přepínači openflow:1 na rozhraní 1.

6.3 Dynamická rekonfigurace

Experimenty v této sekci mají za cíl ověřit chování modulu `ODL_trigger` při vkládání, odstraňování nebo modifikaci odposlechu při změně topologie. Vytvořila jsem několik skriptů pro mininet s různými topologiemi, které zjišťovaly následující chování systému:

- reakce systému na změnu topologie (výpadek linek);
- vyvažování zátěže mezi CC-IIF sondami;
- kombinace vyvažování zátěže, kdy váhy některých hran jsou zadány administrátorem, a změny topologie.

6.3.1 Výpadek linky

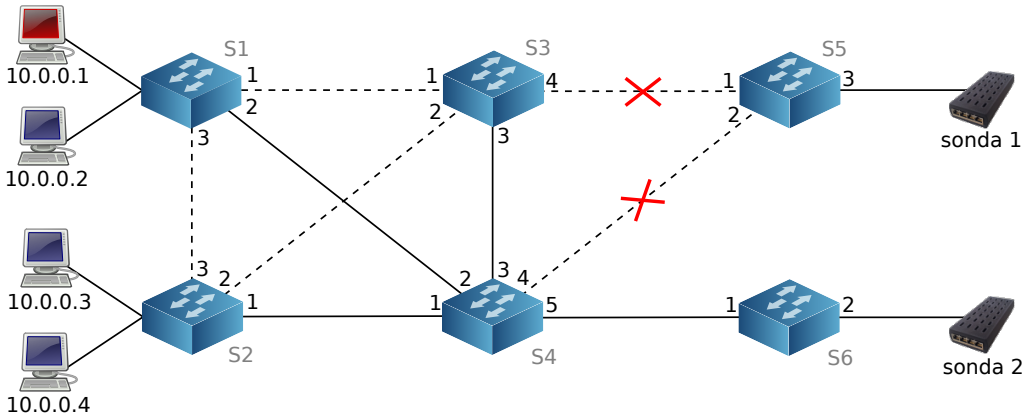
Cílem prvního experimentu bylo ověřit chování systému při výpadku linky, která se využívá pro přeposílání paketů označených VLAN tagem. Použitá topologie je znázorněna na obrázku 6.4. Linky znázorněné plnými čarami označují kostru grafu, kterou používá OpenDaylight pro zasílání neodposlouchávaného provozu. Při vkládání odposlechů i při rekonfiguraci by systém měl brát ohled na využití linek. Pokud je to možné, systém použije k zachytávání dat sondu, ke které se označené pakety dostanou po linkách, které nejsou součástí kostry grafu.

Předpokládejme, že se bude odposlouchávat zařízení s IP adresou `10.0.0.1`. V tabulkách jsou uvedena pravidla, která se nastaví na přepínače v případě, že dojde ke změně topologie (výpadku linky). Experiment proběhl v několika krocích:

1. Spuštění systému, který vytvořil iniciační pravidla v první tabulce toků a pravidla na přeposílání paketů s VLAN tagem v druhé tabulce toků.

Přepínač	Tabulka	ID pravidla	Prio	VLAN	IP zdroj	IP cíl	Ostatní	Akce
S1	1	1	2	*	*	*	*	go-to tab 3
S1	1	2	5	any	*	*	*	go-to tab 2
S1	2	1	5	1	*	*	*	outport 1
S1	2	2	5	2	*	*	*	outport 2

Tabulka 6.1: Pravidla v první a druhé tabulce toků na přepínači S1. První dvě pravidla jsou na na ostatních přepínačích stejná, druhá dvě jsou analogická, ale s odpovídajícími výstupními porty.



Obrázek 6.4: Ukázková topologie použitá pro demonstraci chování systému při výpadku linky.

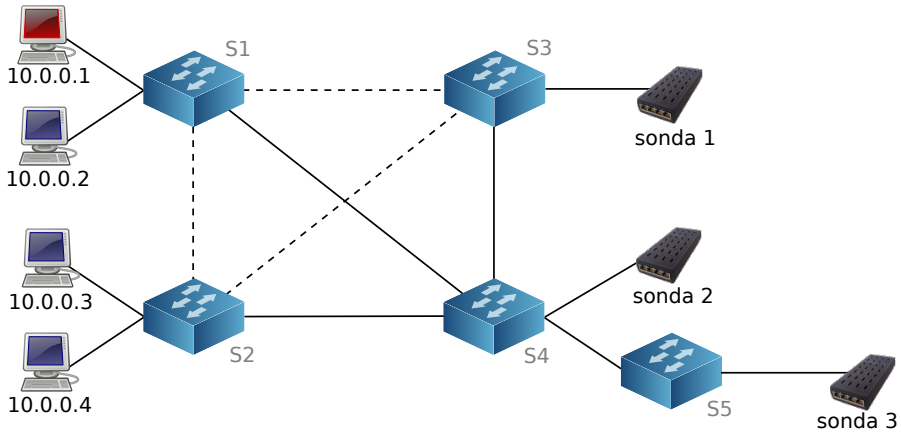
2. Vložení odposlechu IP adresy `10.0.0.1`. Do první tabulky na přepínači S1 systém vložil pravidlo, které bude označovat pakety se zdrojovou nebo cílovou adresou `10.0.0.1` VLAN tagem 1. Zároveň SLIS přes CCCI rozhraní nastavil pro odposlech sondu 1. Zachycená komunikace je přeposílána z přepínače S1 na přepínače S3, S5 a poté na sondu 1.

Přepínač	Tabulka	ID pravidla	Prio	IP zdroj	IP cíl	Ostatní	Akce
S1	1	3	10	10.0.0.1	*	*	push VLAN 1 outport 1
S1	1	4	10	*	10.0.0.1	*	pop VLAN 1 go-to tab 3

Tabulka 6.2: Pravidla pro označování paketů se zdrojovou nebo cílovou adresou `10.0.0.1` VLAN tagem 1. Toto pravidlo se nahraje pouze na přepínač S1.

3. V mininetu příkazem `link S3 S5 down` rozpojíme linku mezi přepínači S3 a S5. V té chvíli se detekuje změna topologie. Pro všechna pravidla na označování paketů a přeposílání paketů označených VLAN tagem systém přepočítá nejkratší cesty k jednotlivým sondám. V tomto případě se upraví pravidla na přepínači S1 a S3. S1 bude označené pakety přeposílat na přepínač S4 (výstupní port v pravidlech v tabulce 6.2 se změní z 1 na 2). S4 pak pakety odešle na přepínač S5. Nastavení sondy 1 na odposlech zůstane beze změny.

4. Příkazem `link S4 S5 down` rozpojíme i linku mezi přepínači S4 a S5. Opět se detekuje změna topologie a začnou se kontrolovat všechna pravidla. Sonda 1 je ale nedostupná, takže se ze všech přepínačů odstraní pravidlo pro směrování označených paketů s VLAN tagem 1 (na všech přepínačích tabulka 2 pravidlo s ID 1). Pravidlo pro odposlech IP adresy `10.0.0.1` se také změní a začne se označovat VLAN tagem



Obrázek 6.5: Topologie použitá pro demonstraci vyvažování zátěže mezi sondami na základě kostry grafu.

2. Systém SLIS se následně pokusí odstranit odposlech ze sondy 1 a nastavit pro odposlech dané IP adresy sondu 2.

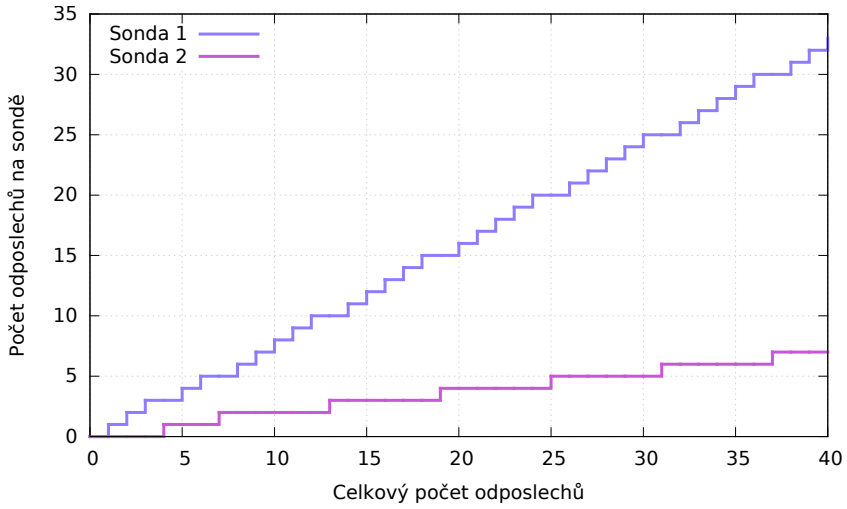
Během experimentu se systém choval podle očekávání. Při postupném rozpojení linek mezi odposlouchávaným zařízením a nejbližší sondou zjistil nejkratší cesty. Podle nich pak dokázal upravit výstupní porty stávajících pravidel a směrovat označenou komunikaci stále ke stejné sondě, ale i přehodit odposlech na jinou sondu v případě, že první se stane nedostupnou.

6.3.2 Vyvažování zátěže na základě používaných linek

Cílem tohoto experimentu je ověřit vyvažování zátěže mezi sondami. Vyvažování zátěže se řídí konstrou grafu, kterou OpenDaylight používá pro přeposílání paketů neodposlouchávané komunikace. Při vytváření grafové reprezentace topologie se váha hran kostry zvýší na 5, zatímco ostatní hrany mají hodnocení 1.

`ODL_trigger` při vkládání odposlechu vytvoří graf topologie. Pro každou sondu zjistí součet hodnocení hran nejkratší cesty, která vede ze sondy na přepínač, ke kterému je připojeno zařízení, které chceme odposlouchávat. Primárně se odposlechy vkládají na sondu, která má součet hodnocení hran nejnižší. Může se ovšem lehce stát, že se všechny odposlechy budou vkládat pouze na tuto sondu a ostatní sondy budou nevytížené. Při vkládání odposlechu se tedy kromě součtu hodnocení hran kontroluje i počet odposlechů na jednotlivých sondách. Pokud je rozdíl v počtu odposlechů mezi nejbližší sondou a jakoukoliv jinou moc velký, vloží se odposlech na vzdálenější sondu. Tímto způsobem lze zabránit přetěžování sond v síti.

Topologie, která byla použita v tomto experimentu, je znázorněna na obrázku 6.5. V prvním experimentu jsem využila pouze sondu 1 a sondu 2. Do systému jsem postupně vkládala požadavky na odposlech trojice IP adresa `10.0.0.1`, port a TCP protokol. Pravidla pro označování paketů VLAN tagem se vkládaly na přepínač S1 a v grafu 6.6 je znázorněno, jak se odposlechy vkládaly na jednotlivé sondy.



Obrázek 6.6: Ukázka vyvažování zátěže mezi sondou 1 a sondou 2 v topologii 6.5.

V současné chvíli je limit rozdílu mezi sondami nastaven na trojnásobek. Ve chvíli, kdy je sonda 1 zatížena třikrát více než sonda 2, vloží se nový odposlech na sondu 2. Tento limit lze měnit a případně i úplně ignorovat. Sonda 2 by pak sloužila jen jako záložní sonda pro případ, že by sonda 1 byla odpojena nebo nedostupná (tento případ užití je ukázán v experimentu 6.3.3).

V druhé části experimetu jsem využila i třetí sondu. Opět jsem postupně vkládala odposlechy na přepínač S1, ale tentokrát se zátěž vyvažovala mezi všechny tři sondy. V grafu 6.7 je znázorněno, kolik odposlechů se na jednotlivé sondy vložilo.

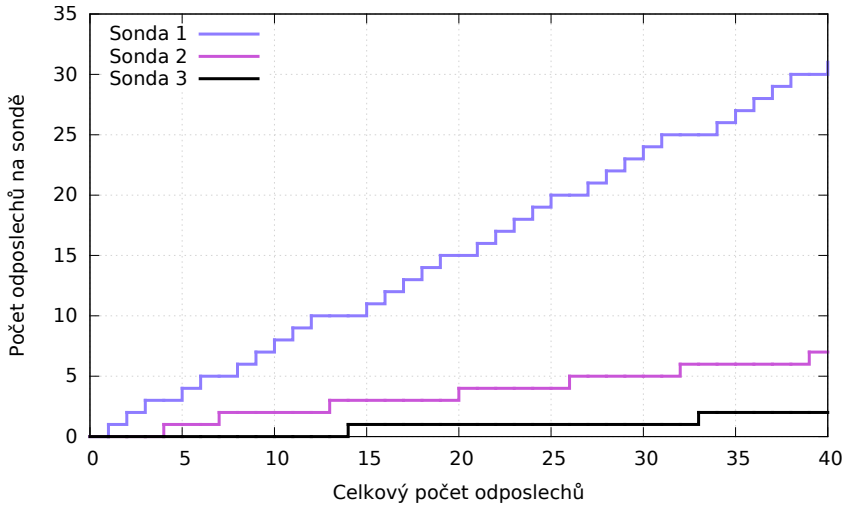
6.3.3 Hrany grafu specifikované administrátorem

Posledním experimentem zaměřeným na chování systému při dynamické rekonfiguraci je speciální případ, kdy hodnocení hran grafu může být specifikováno administrátorem. Tato situace může nastat například ve chvíli, kdy se síť ISP nachází na dvou lokacích a je propojená jednou nebo více linkami. Pokud budou na obou místech zapojeny CC-IIF sondy, bude nejvýhodnější vložit odposlech na sondu, která se nachází v dané lokaci. Z tohoto důvodu je možné vložit do konfiguračního souboru řádky, na kterých administrátor uvede linku, které se ohodnocení týká, a její váhu. V případě, že je váha linky vyšší nebo rovna 1000, přestane se linka využívat pro vyvažování zátěže.

Experiment proběhl na topologii znázorněné na obrázku 6.8. Do konfiguračního souboru jsem vložila řádek

```
weight openflow:3 openflow:4 1000
```

Tím se zvýšilo hodnocení hrany mezi přepínačem S3 a S4 na 1000. Odposlechy IP adresy `10.0.0.1`, portu a protokolu se vkládaly vždy na sondu 1. Naopak odposlechy IP adresy `10.0.0.101`, portu a protokolu se vložily vždy na sondu 2. Ve chvíli, kdy selhaly linky vedoucí k sondě 1 a tím pádem sonda přestala být dostupná, všechny odposlechy se přesunuly na vzdálenější sondu 2.



Obrázek 6.7: Vyvažování zátěže mezi všemi třemi sondami v topologii 6.5.

Zvýšením hodnocení hran může administrátor vypnout vyvažování zátěže mezi sondami, ale i specifikovat linky, které jsou třeba z jiného důvodu zatížené nebo nespolehlivé. Tyto linky se pak použijí pouze v případě, že už neexistuje jiná cesta mezi zařízením, které chceme odposlouchávat, a některou sondou.

6.4 Výkonnost systému

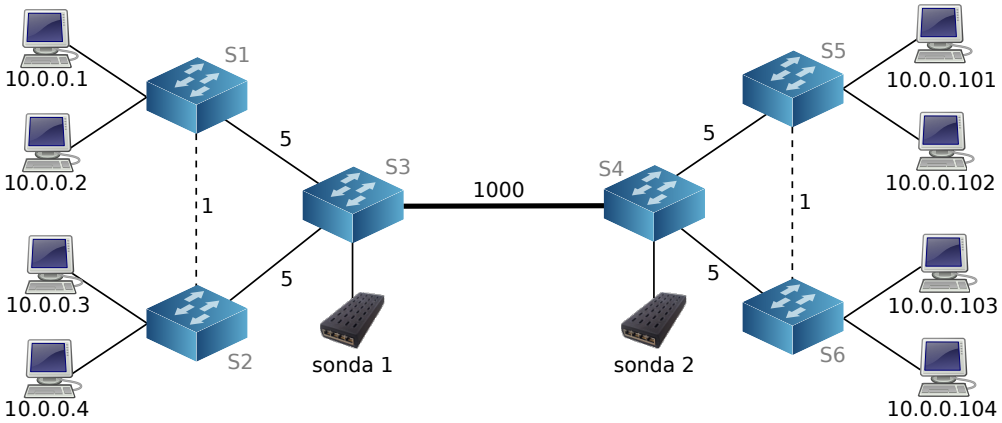
Experimenty v této sekci jsou zaměřeny na zjištění rychlosti systému při manipulaci s toky. Topologie v obou případech zůstala stejná jako v předcházejícím experimentu a je znázorněna na obrázku 6.8.

6.4.1 Zpoždění při vkládání odposlechu

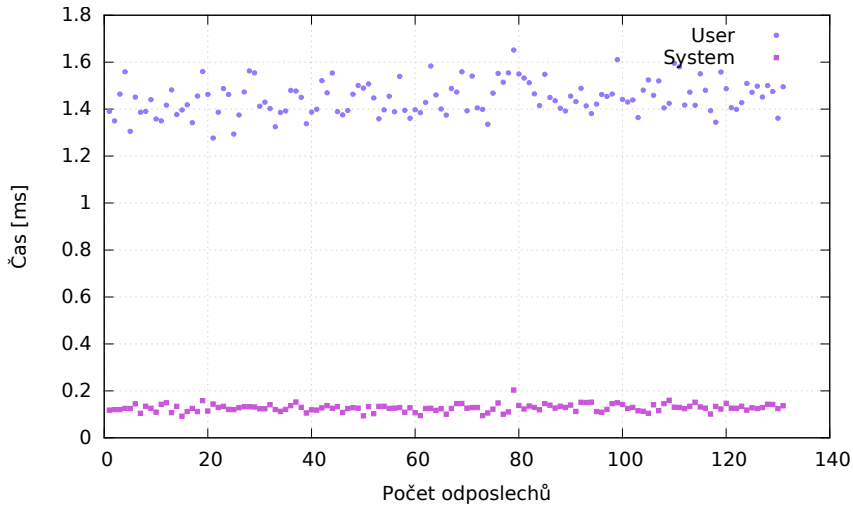
Cílem prvního experimentu bylo zjistit dobu, za kterou se nahraje nový odposlech do systému. Doba měření zahrnuje pouze činnosti, které provádí modul `ODL_trigger`.

V průběhu experimentu se do systému postupně zadávaly požadavky na odposlech trojice: IP adresa `10.0.0.1` (vždy stejná), číslo portu (každým novým pravidlem se zvýšilo) a protokol TCP. V grafu 6.9 jsou znázorněny naměřené výsledky. Uživatelský čas značí dobu, kterou počítač strávil výpočtem a systémový čas potom dobu, kdy čekal v rámci procesu. Z grafu je zřejmé, že počet pravidel v přepínačích nemá vliv na dobu vkládání nového odposlechu. Průměrná doba vložení je přibližně 1,4 sekundy.

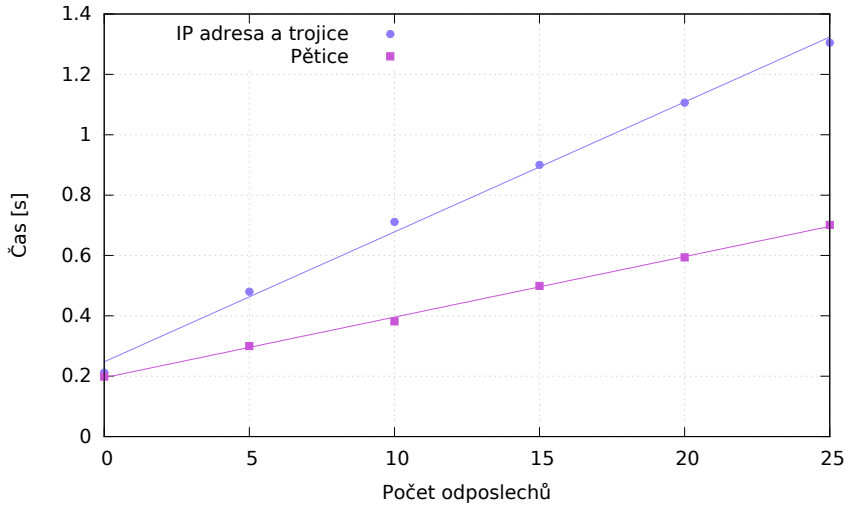
Při vkládání pravidla na odposlech IP adresy časy zůstanou stejné jako u trojic, protože u obou se musí vložit dvě pravidla do přepínače (pro porovnání zdrojové i cílové adresy). Vkládání pravidla na odposlech pětice by pak bylo provedeno rychleji, protože se na přepínače vkládá pouze jedno pravidlo. Počet pravidel, které jsou již nahrány na přepínačích, nemá na dobu vkládání nového odposlechu žádný vliv.



Obrázek 6.8: Topologie použitá pro demonstraci vkládání odposlechů na sondy, kde váha hrany mezi přepínačem S3 a S4 je specifikována administrátorem.



Obrázek 6.9: Doba vložení jednoho odposlechu (dvojice pravidel) v závislosti na počtu pravidel v tabulce.



Obrázek 6.10: Doba upravení všech pravidel pro odposlech (odstranění pravidla, vypočítání nové cesty v grafu a nejvhodnější sondy, vložení pravidla) v závislosti na celkovém počtu pravidel.

6.4.2 Zpoždění při změně topologie

Cílem druhého experimentu je zjistit, jak dlouho trvá modulu `ODL_trigger` přesunout pravidla z jedné sondy na druhou v případě, že první sonda bude nedostupná. Topologie opět zůstala stejná jako v předchozím případě. Doba měření zahrnuje jak zjišťování změn v topologii, tak následnou změnu pravidel na jednotlivých přepínačích.

Při experimentu se do systému vkládal vždy určitý počet odposlechů, které se označovaly VLAN tagem sondy 1. Poté byla sonda 1 odstraněna z topologie. Reakcí systému bylo odstranění pravidla pro přeposílání paketů označených VLAN tagem 1 z tabulky 2 a úprava všech pravidel pro označování odposlouchávaných dat. V grafu 6.10 jsou znázorněny naměřené hodnoty.

Zatímco v případě vkládání odposlechů nemá počet stávajících pravidel na přepínačích žádný vliv, u přesouvání toků je to logicky naopak. Čas roste lineárně s počtem odposlechů, přičemž u pětic je čas na úpravu jednoho odposlechu kratší, protože se upravuje pouze jedno pravidlo. Průměrný čas pro odstranění a vložení upraveného pravidla je přibližně 20 ms.

Kapitola 7

Závěr

V rámci diplomové práce jsem se seznámila s principem softwarově definovaných sítí a se systémem pro zákonné odposlechy SLIS, který vznikl v rámci projektu Sec6Net. Pro implementaci jsem zvolila dva kontrolery – OpenDaylight a POX. OpenDaylight má velké zastoupení v komerční sféře (je podporován firmami Cisco, Dell, HP a mnoha dalšími). POX zase patří mezi kontrolery, které lze velmi jednoduše rozšiřovat a testovat.

Pro systém SLIS jsem navrhla a implementovala rozšíření, která využijí výhod SDN:

- Moduly pro IRI-IIF, které jsou určeny k získávání částečné identity koncových zařízení a uživatelů. Modul pro OpenDaylight se periodicky dotazuje kontroleru na topologii sítě a informace o částečné identitě koncových zařízení zasílá IRI-IIF. Modul pro POX využívá automaticky vytvářených událostí ke zjišťování topologie a informace o identitě pak také oznamuje IRI-IIF.

- Dynamická rekonfigurace CC-IIF sond a OpenFlow přepínačů. Jedná se především o implementaci CCTF, ze které bylo v rámci SLIS implementováno jen nutné minimum. CCTF nyní rozlišuje jednotlivé CC-IIF sondy podle jejich pozice v topologii. Na základě těchto informací systém vytvoří grafovou reprezentaci topologie a zvýší váhu hran, které používá OpenDaylight pro zasílání neodposlouchávaných paketů. Při vložení odposlechu se pak vybere vhodná sonda. Vybírá se především podle vzdálenosti a zároveň se provádí jednoduché vyvažování zátěže, aby se předešlo zahlcení jedné sondy. Administrátor může také nastavit cenu linek v konfiguračním souboru. Pokud zvolí cenu větší než 1000, přestane se provádět vyvažování zátěže a daná linka se bude využívat pouze jako záložní. V případě, že se ze systému odpojí sonda s aktivními odposlechy, rozdělí se tyto odposlechy mezi ostatní sondy.

Nově implementované části systému SLIS jsem následně otestovala s využitím virtuální sítě v programu *mininet*. Moduly pro IRI-IIF rozšiřují částečnou identitu zařízení o IP adresu, MAC adresu a přepínač, ke kterému je koncové zařízení připojeno. U dynamické rekonfigurace sond a přepínačů jsem otestovala chování systému v případě vypadnutí linky, vyvažování zátěže mezi sondami a zadání váhy hrany administrátorem. Ve všech situacích se program choval podle očekávání. Poté jsem provedla několik experimentů, kterými jsem zjistila výkonnost nově naimplementovaných částí.

Výsledky této práce jsou součástí projektu *Sec6Net*. Možným navazujícím výzkumem by mohlo být například rozšíření systému tak, aby se OpenFlow přepínače chovaly jako CC-IIF sondy a samy zasílaly zachycená data systému pro zákonné odposlechy.

Literatura

[1] European Telecommunications Standards Institute: ETSI TR 101 943: Telecommunications security; Lawful Interception (LI); Concepts of Interception in a generic Network Architecture. 2001, verze 1.1.1.

[2] Betts, M.; Davis, N.; Dolin, R.; aj.: SDN architecture. Technická zpráva, Open Networking Foundation, 2014.

[3] Chao, H. J.; Liu, B.: *High performance switches and routers*. John Wiley & Sons, 2013.

[4] Chomjak, R.: *Demonstrace možností technologie Cisco OnePK*. Bakalářská práce, Brno, FIT VUT v Brne, 2014.

[5] Cronin, E.; Sherr, M.; Blaze, M.: On the (un)reliability of eavesdropping. In *International Journal of Secure Networking*, 2008, s. 103–113.

[6] Frenandez, M. P.: Comparing OpenFlow Controller Paradigms Scalability: Reactive and Proactive. In *AINA '13 Proceedings of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications*, IEEE, 2013, s. 1009–1016.

[7] Ganti, V.; Lubsey, V.; Shekhar, M.; aj.: Open Data Center Alliance: Software-Defined Networking Rev. 1.0. 2013.

[8] Heller, B.: OpenFlow Switch Specification, Version 1.0.0 (Wire Protocol 0x01) [online]. https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf, 2009.

[9] Holkovič, M.: *Detekce identity na různých vrstvách architektury TCP/IP*. Bakalářská práce, Brno, FIT VUT v Brně, 2013.

[10] Kaur, S.; Singh, J.; Ghumman, N. S.: Network Programmability Using POX Controller. In *ICCCS International Conference on Communication, Computing & Systems*, IEEE, 2014, s. 134–138.

[11] McKeown, N.; Anderson, T.; Balakrishnan, H.; aj.: OpenFlow: enabling innovation in campus networks. In *ACM SIGCOMM Computer Communication Review*, ACM, 2008, s. 69–74.

[12] Medved, J.; Tkacik, A.; Varga, R.; aj.: OpenDaylight: Towards a Model-Driven SDN Controller architecture. In *2014 IEEE 15th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2014, s. 1–6.

[13] Metzler, J.: What is SDN? And Why Should I Care? [online].
https://www.eiseverywhere.com/file_uploads/
458f97398bb66838e66ecb90c7a41eb7_Jim_Metzler.pdf, 2012.

[14] Nadeau, T.; Grey, K.: *SDN: Software Defined Networks*. O'Reilly Media, 2013.

[15] Open Networking Foundation: OF-CONFIG 1.0 OpenFlow Configuration and Management Protocol [online].
https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config1dot0-final.pdf,
2009.

[16] Open Networking Foundation: Software-Defined Networking: The New Norm for Networks [online]. https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf,
2012.

[17] Open Networking Foundation: OF-CONFIG 1.2 OpenFlow Management and Configuration Protocol [online].
https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1.2.pdf,
2014.

[18] Polčák, L.: Challenges in Identification in Future Computer Networks. In *ICETE 2014 Doctoral Consortium. Wien: SciTePress - Science and Technology Publications*, 2014, s. 15–24.

[19] Polčák, L.; Martínek, T.; Hranický, R.; aj.: Zákonné odposlechy v moderních sítích. Technická zpráva, FIT VUT v Brně, 2014.

[20] Rabaey, J. M.; Potkonjak, M.; Koushanfar, F.; aj.: Challenges and Opportunities in Broadband and Wireless Communication Designs. In *ICCAD-2000. IEEE/ACM International Conference on Computer Aided Design*, 2000, s. 76–82.

Seznam příloh

Příloha A

Obsah CD

app	kompletní balík software
src	zdrojové soubory vytvořených rozšíření
tex	zdrojové soubory technické zprávy
dp-xfrank08.pdf	technická zpráva ve formátu PDF
dp-xfrank08-print.pdf	technická zpráva ve formátu PDF pro tisk
README	nápověda k obsahu CD