Jihočeská univerzita

Přírodovědecká fakulta

Ústav aplikované informatiky

# Cross-platform Software Defined Networking controller v multi-vendor prostředí

Autor: Petr Nebáznivý

Vedoucí práce: Rudolf Vohnout, Ing., Ph.D.

České Budějovice 2016

**Jihočeská univerzita v Českých Budějovicích**
**Přírodovědecká fakulta**

# ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE

**Student: Petr NEBÁZNIVÝ**
*(jméno, příjmení, tituly)*

**Obor – zaměření studia**: Aplikovaná Informatika

**Katedra:** Ústav aplikované informatiky

**Školitel:** Rudolf Vohnout, Ing., Ph.D.
*(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)*

**Garant z PřF:** .....
*(jméno, příjmení, tituly, katedra – jen v případě externího školitele)*

**Školitel – specialista, konzultant:** .....
*(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)*

**Téma bakalářské práce:**

*Cross-platform Software Defined Networking controller v multi-vendor prostředí*

Cíle práce:

Hlavní cíl:

- Navrhnout a demonstrovat funkcionalitu zvoleného open-source SDN kontroléru v multi-vendor síťovém prostředí.

Popis práce:

Práce se bude zavývat problemtikou softwarově definovaných sítí, kdy její hlavnm cílem bude demonstrovat management a kontrolu síťové topologie tvořenou aktivními prvky od různých výrobců. Navíc by tato topologie měla být tvořena prvky, které jsou tzv. cross-platform, tj. s různými operačními systémy a navíc různých kategorií (pracujícími na L2+L3 ISO/OSI modelu).

Základní doporučená literatura :

Paul GORANSSON a Chuck BLACK: *Software Defined Networks: A Comprehensive Approach.* Massachusetts: Morgan Kaufmann; 1 edition (June 6, 2014). ISBN: 978-0124166752

Financování práce :.....

Vedoucí práce : Rudolf Vohnout podpis : .....

U externích vedoucích fakultní garant práce.....podpis : .....

Garant oboru bak.. studia (nepožaduje se u zaměření „příprava na mag. studium biologie) .....

podpis : .....

Vedoucí katedry : Libor Dostálek podpis .....

Případný souhlas vedoucího ústavu AV .....podpis : .....

V Českých Budějovicích dne .....

Převzal/a dne..25.11.2015..... podpis : .....

**Bibliografické údaje**

Nebáznivý P., 2016: Cross-platform Software Defined Networking controller v multi-vendor prostředí [Cross-platform Software Defined Networking controller in multi-vendor enviroment Bc. Thesis, in Czech.] – 69 p., Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

**Abstrakt**

Tato bakalářská práce se zabývá problematikou softwarově definovaných sítí a jedním protokolem, který podporuje tuto technologii. V textu je obsažen stručný popis funkcionality OpenFlow protokolu a popis zprovoznění open source kontroléru, který spravuje aktivní prvky různých výrobců a s různými operačními systémy.

**Klíčová slova:**

OpenFlow, SDN, Floodlight, OpenSource

**In English**

This thesis deals with software defined network and with protocol which supports this technology. In the text bellow is a brief OpenFlow functionality description. Thesis also contains instructions on how to deploy OpenFlow controller which manages active network devices from different vendors and with different operating systems.

**Keywords**

OpenFlow, SDN, Floodlight, OpenSource

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č.111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích, dne 21. dubna 2016.

Podpis: _____

**Poděkování**

Rád bych poděkoval panu Ing. Rudolfu Vohnoutovi, Ph.D. za odborné rady, vyjednání aktivních prvků a pomoc při zpracování této bakalářské práce. Dále bych chtěl poděkovat panu Bc. Petru Mackovi a firmě Kostax za ochotu a zapůjčení aktivních prvků od firmy HP. Chtěl bych také poděkovat firmě Medisize za zapůjčení přístrojů Mikrotik pro prvotní testování OpenFlow.

# Obsah

# 1 Úvod

S nástupem nových technologií jako jsou zařízení IoT[1], chytré telefony, cloudové služby vznikly větší nároky na síťové infrastruktury. Infrastruktury s necentralizovanou správou jednotlivých síťových prvků, začínají být složité na správu a údržbu. Zároveň tyto infrastruktury nelze dynamicky měnit, což je v dnešní době internetových aplikací a virtuálních strojů velikou nevýhodou. Řešením všech těchto problémů by měly být softwarově definované sítě.

Modifikace infrastruktury sítě v reálném čase je užitečná například pro nastavování nových bezpečnostních opatření, změn směrování síťových toků, load-balancingu. V každé větší síťové infrastruktuře, kde je každé zařízení autonomní a samostatně konfigurovaným zařízením[1], je tato úloha skoro neproveditelná.

Tato práce popisuje základní vlastnosti virtuálních sítí a softwarově definovaných sítí. Hlavní část práce je zaměřena na zprovoznění a nastavení softwarově definované sítě pomocí protokolu OpenFlow, který zprostředkovává komunikaci mezi jednotlivými aktivními prvky a kontrolérem. Hlavním cílem je ukázat, zda je OpenFlow kontrolér schopný pracovat s různými aktivními prvky od různých výrobců.

[1] Internet of Things

# 2 Cíle práce

Cílem práce je vytvoření funkčního zapojení aktivních síťových prvků, které budou spravovány jedním kontrolérem. Tento kontrolér musí podporovat protokol OpenFlow.

Práce má také několik podcílů, které musejí být splněny.

- Instalace a nastavení open-source kontroléru, který podporuje OpenFlow protokol
- Nastavení komunikace jednotlivých aktivních prvků od různých výrobců k OpenFlow kontroléru.
- Nastavení flow tabulek pomocí OpenFlow kontroléru

# 3 Metodika

1. Sběr a analýza dostupné literatury a dostupných technologií.
2. Provedení rešerše pro technologie SDN, kontrolérů a zařízení podporující OpenFlow.
3. Sestavení přehledu SDN technologií a kontrolérů pro řízení síťové infrastruktury.
4. Nastavení aktivních prvků a instalace potřebného softwarového vybavení.
5. Provedení experimentu s aktivními prvky připojenými na OpenFlow kontrolér.

# 4 Analýza současných technologií

Kapitola obsahuje popis jazyka P4, který slouží pro definici síťové infrastruktury, protokolu OpenFlow, použitelných kontrolérů a zvoleného kontroléru.

## 4.1 SDN

Jak již bylo zmíněno v úvodu, softwarově definované sítě mají v této době smysl, protože umožňují dynamicky upravovat síťovou infrastrukturu. V této době se tyto sítě používají u cloudových řešení, kdy zákazníci požadují přístup k pronajímané infrastruktuře. Jejich dalším bonusem je rychlejší nasazení bezpečnostních opatření, které je u velkých infrastruktur velice obtížné a také časově náročné.[1]

SDN se často zaměňuje s pojmem OpenFlow, což je pouze jedna implementace konceptu softwarově definovaných sítí. Avšak aktuální implementace OpenFlow si našla cestu do mnoha reálných sítí, má však i určité nevýhody, které jsou spojeny s rozšiřitelností dalšího zpracování paketů nebo počtem podporovaných protokolů. Nedostatky OpenFlow by měl vyřešit programovací jazyk P4, který slouží pro definování sítí.[1]

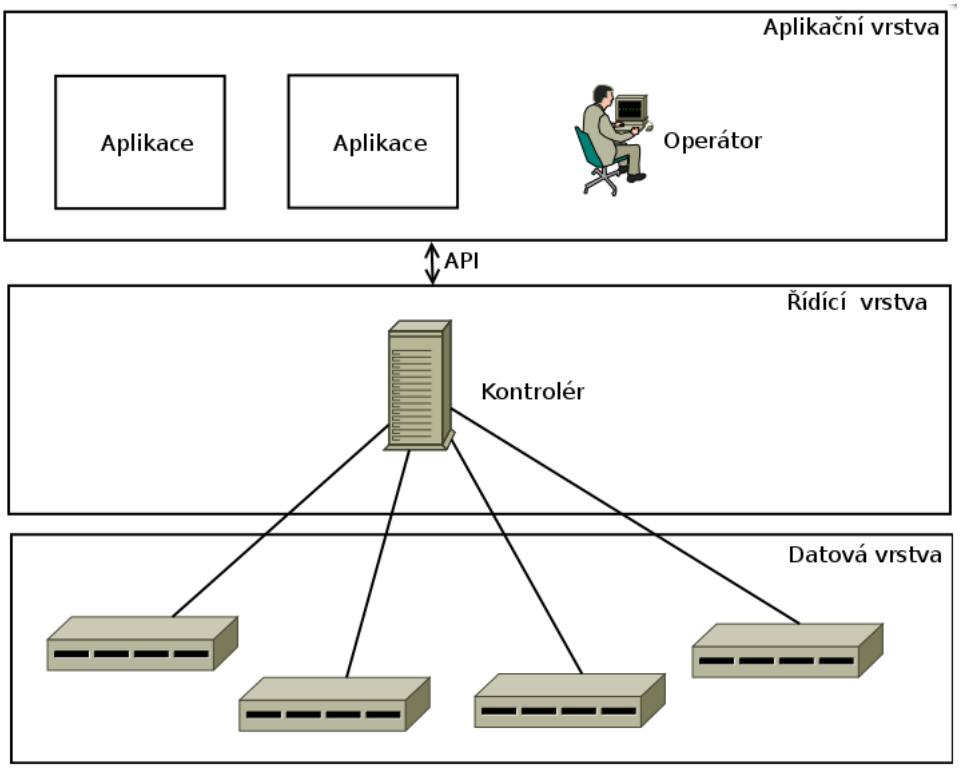
SDN také umožňují inteligentně směrovat datový provoz, včasně alokovat síťové prostředky pro zvýšení propustnosti. Tuto službu využijí nejvíce velké služby jako je Google, Facebook, Twitter, apod.[1]

Největší výhodou SDN sítí je jejich nezávislost na značce a použitém operačním systému. Tímto se také zmenšuje riziko závislosti zákazníků na řešení jednoho výrobce.[1] Ale není podmínkou, že všechny operační systémy a všichni výrobci tuto technologii podporují.

Koncept softwarově definovaných sítí striktně odděluje řídící a datovou vrstvu síťové infrastruktury. To znamená, že síťový hardware je velice rychlé zařízení, které neobsahuje žádný řídící software. Síťový hardware se připojuje k řídící vrstvě, která se v SDN nazývá kontrolér.[1]

SDN kontrolér má na starosti více SDN zařízení, kterým určuje jejich chování. Tímto je docíleno přenosu z distribuovaného chování síťových prvků k centralizovanému.

Centralizovaný pohled na síťovou infrastrukturu je velice výhodný pro plánování a řízení, protože kontrolér pohlíží na síť jako na jeden celek. Tento pohled ulehčí práci správcům, protože komunikují pouze s jedním zařízením a to s kontrolérem a ne s několika zařízeními.[1]




Obr. 1. SDN architektura[1]

Výhody konceptu SDN:[1]

- **Přímá konfigurovatelnost** – možné díky oddělení datové a řídící vrstvy
- **Abstrakce** – pro většinu nastavení není třeba znalost celé sítě
- **Centralizace** – správa celé sítě z jednoho místa
- **Otevřený standart** – přináší nezávislost na výrobci hardware
- **API** – komunikační rozhraní s kontrolérem na aplikační úrovni

### 4.1.1 Jazyk P4

Jazyk P4 byl vytvořen ve spolupráci firem Barefoot Networks, Google, Intel, Microsoft, a Princetonské University. Byl navržen jako vysokoúrovňový jazyk, umožňující programovaní protokolově nezávislých paketových procesorů. P4 je schopný spolupracovat s kontrolními protokoly SDN sítí jako je OpenFlow.[2]

Jazyk P4 má tři cíle:[2]

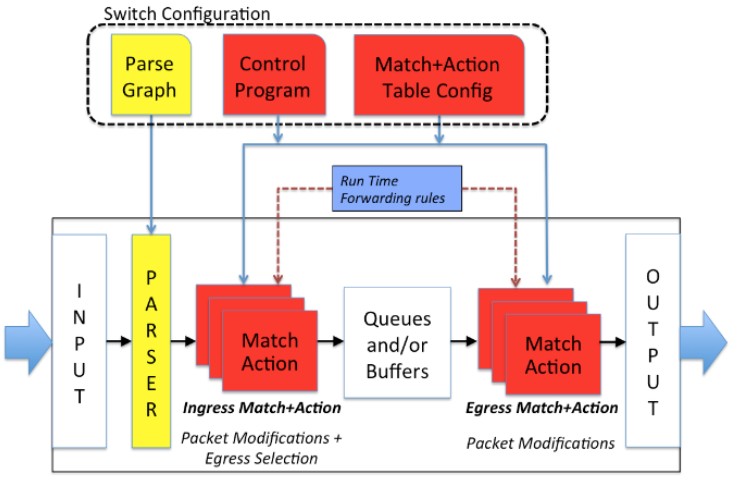
- **Rekonfiguraci konfiguračních polí** - Umožnit programátorovi měnit způsob zpracování paketů na přepínačích jakmile je nasazen.
- **Protokolovou nezávislost** – Přepínače by neměly být vázány k žádnému určitému protokolu. Přepínače by měly mít možnost kompletní rekonfigurace.
- **Cílovou nezávislost** – Programátor by měl být schopen popsat zpracování paketu nezávisle na základním hardwaru.

P4 je deklarovaný jazyk, který říká aktivním síťovým prvkům (přepínače, síťové karty, firewally, filtry, atd.) jakým způsobem mají zpracovávat pakety. P4 chce zásadně změnit způsob návrhu počítačových sítí.[3]

Při použití P4 se musí začít návrhem požadavků, které budou vytvářeny na síťovou infrastrukturu. Druhým krokem je napsání P4 programu, který popisuje jak má systém zpracovávat pakety a posledním krokem je kompilace programu s instrukcemi pro aktivní prvek. P4 dovoluje síťovým programátorům využít výhody, které jsou známé z vývoje softwaru (skládání programů, debuggování, prokazatelnost chování, kontrolu modelů, atd.) přenést do návrhu síťových systému.[3]

Výhodou P4 je možnost napsat P4 program, který přidá vlastní proces zpracování paketu nebo omezí počet protokolů, které bude síť podporovat. Díky tomu je možné provést rychlé nasazení nového protokolu, formátu hlavičky v řádu několika dní. Odpadá tak letité čekání na výrobce čipu, který bude nový protokol podporovat.[3]

Na jedné straně má P4 sloužit jako jednoduchý jazyk pro určení způsobu zpracování paketu, ale zároveň chce kompletně změnit způsob návrhu síťových infrastruktur. Autoři jazyka P4 doufají, že pokročilé sítě, které jsou již dnes softwarově definované, se jednou stanou plně programovatelnými.[3]




Obr. 2. Abstraktní model zpracování paketů P4[4]

Jazyk P4 je stále ve vývoji a všechny zdrojové kódy a překladače jsou ke stažení na stránkách p4.org. Veškeré kódy jsou ke stažení zadarmo a jsou šířené pod licencí Appache.[3]

### 4.1.2 OpenFlow

OpenFlow byl původně představen a implementován jako součást výzkumu sítě na Stanfordské Univerzitě. Jeho původním účelem bylo umožnit vytvoření experimentálních protokolů v síti kampusu, které by mohly být využity k výzkumu a experimentům. Před tím musela univerzita vytvořit jejich vlastní experimentální platformu od základů. To, co se vyvinulo z původní myšlenky, je pohled, že OpenFlow může kompletně nahradit funkcionalitu L2 a L3 protokolů v podnikových směrovačích a přepínačích.[5]

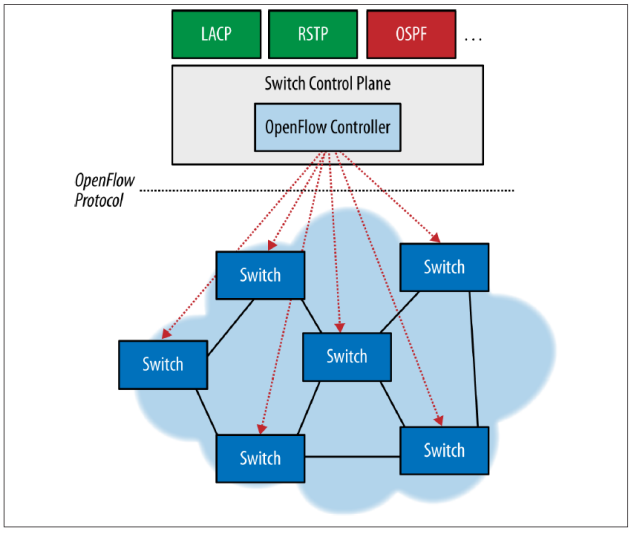
V roce 2011 bylo založeno neziskové konsorcium nazývané Open Networking Foundation skupinou poskytovatelů služeb za účelem komercializace, standardizace a podpory používání OpenFlow v reálném provozu. ONF[2] je nový typ stanfordské organizace pro vývoj technologií spojených s SDN. V organizaci mají velice aktivní marketingové oddělení, které je určeno k propagaci OpenFlow protokolu a dalších technologií spojených s SDN.[5]

[2] ONF - Open Networking Foundation

Klíčové komponenty OpenFlow modelu zobrazené na Obr. 3 se později staly součástí obecně známé definice SDN.[5]

Tři hlavní komponenty SDN jsou:[5]

- Oddělení kontrolní a datové vrstvy (podle ONF je kontrolní vrstva spravována logicky centralizovaným kontrolérem).
- Používání standardizovaných protokolů mezi kontrolérem a agentem instalovaným na síťovém zařízení.
- Poskytnutí schopnosti programování sítě z centralizovaného pohledu, přes moderní a rozšiřitelné API.




Obr. 3. Architektura OpenFlow[5]

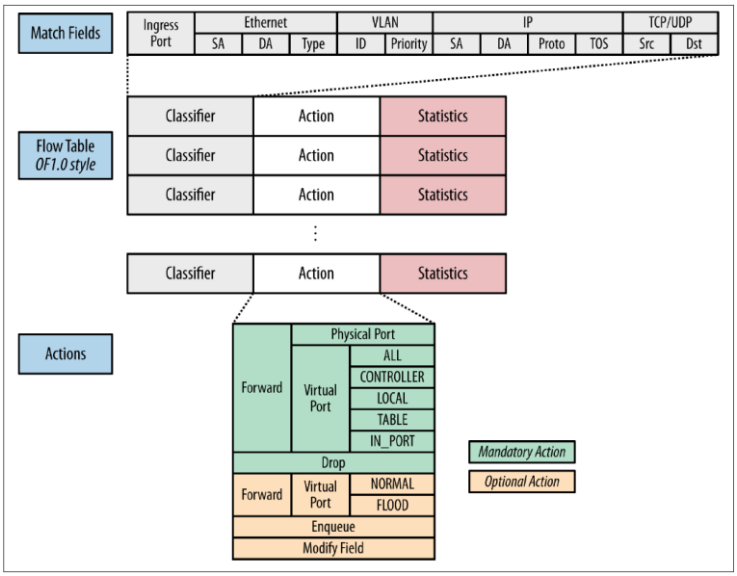
OpenFlow protokol je nyní rozdělen do dvou částí:[5]

- Síťový protokol pro navázání relace řízení, která definuje strukturu zpráv pro výměnu modifikací flow tabulek, sbírání statistik a definování základních struktur přepínače (tabulky a porty).
- Protokoly pro konfiguraci a management. OF-Config založený na NETCONF[3] pro alokaci fyzických portů přepínače v kontroléru, definující vysokou dostupnost (aktivní/neaktivní) a chování při výpadku připojení ke kontroléru.

[3] Network Configuration Protocol – protokol pro správu síťových zařízení

OpenFlow protokol přímo nepodporuje rozdělení sítě na jednotlivé skupiny spravované stejnými pravidly nebo rozdělené do domén. Ačkoli nástroje jako je FlowVisor a specifická implementace protokolu jednotlivými výrobci to umožnuje.[5]

V OpenFlow flow záznamech je celá hlavička paketu přístupná pro porovnání a modifikaci. Možnosti porovnávání a modifikací se vyvíjely s jednotlivými verzemi OpenFlow protokolu. Jak je vidět na Obr. 4 protokol implementuje L2+L3+ACL funkcionality.[5]




Obr. 4. OpenFlow protokol verze 1.0[5]

### 4.1.3 Komunikace kontrolér zařízení

OpenFlow protokol podporuje tři druhy zpráv:[6]

- Kontrolér → přepínač zprávy jsou iniciovány kontrolérem a jsou využívány k přímé správě nebo inspekci stavu přepínače
- Asynchronní zprávy jsou iniciovány přepínačem a jsou používány pro aktualizaci kontroléru. Jsou předávány informace o událostech na síti a změnách stavu přepínače.
- Symetrické zprávy jsou iniciovány buď přepínačem, nebo kontrolérem a jsou odesílány bez předchozí žádosti.

**Kontrolér → přepínač**

Kontrolér/přepínač zprávy jsou iniciovány kontrolérem a mohou nebo nemusí vyžadovat odezvu z přepínačů.[6]

**Features**: po navázání komunikace, kontrolér pošle zprávu s požadavkem služby do přepínače. Přepínač musí odpovědět odpovědí funkce, která specifikuje možnosti podporované přepínačem.[6]

**Configuration**: Kontrolér je schopen nastavit nebo získat konfiguraci parametrů přepínače. Přepínač pouze odpovídá na dotazy kontroléru.[6]

**Modify-State**: Zprávy jsou zasílány kontrolérem pro správu stavů přepínače. Jejich hlavním účelem je přidání/odebrání a úprava flow záznamů ve flow tabulkách a nastavování vlastností portů.[6]

**Read-State**: Zprávy jsou používány kontrolérem pro sběr statistických informací z flow tabulek, portů a jednotlivých flow záznamů na přepínači.[6]

**Send-Packet**: Jsou užívány kontrolérem pro odesílání paketů do určitých portů na přepínači.[6]

**Barrier**: Barrier požadavky/odpovědi jsou užívány kontrolérem pro ujištění, že existují všechny potřebné závislosti nebo pro potvrzení úspěšného dokončení operace.[6]

**Asynchronní zprávy**

Asynchronní zprávy jsou vysílány bez žádosti od kontroléru z přepínače. Přepínač zašle asynchronní zprávu do kontroléru s oznámením o příchozím paketu, změně stavu přepínače nebo pří chybě na přepínači. OpenFlow protokol obsahuje 4 hlavní asynchronní zprávy.[6]

**Packet-in**: Je zasílána při každém příchozím paketu, pro který nebyl nalezen záznam ve flow tabulce nebo pokud je nastavená akce zaslání paketu kontroléru. Pokud má přepínač dostatečnou paměť pro uchování paketu, paket-in obsahuje pouze část hlavičky. Původní nastavení je 128 bajtů a buffer ID, které je použito, pokud je kontrolér připraven předat paket dále. Přepínač, který nepodporuje nebo nemá možnost uložení paketu v paměti, zasílá celý paket do kontroléru jako část zprávy.[6]

**Flow-Removed**: Je zasílána pokud dojde k odebrání flow záznamu z tabulky, který byl vložen do přepínače flow modify zprávou a jeho čas nečinnosti, který indikuje, že by měl být záznam odebrán kvůli nedostatečné aktivitě. Nebo byl odebrán z důvodu skončení platnosti, tento flow záznam je okamžitě odebrán bez ohledu na to, zda je využíván.[6]

**Port-Status**: Je očekávána od přepínače, pokud dojde ke změně stavu konfigurace portu a stavu portu, nebo dojde ke změně stavu portu podle protokolu 802.1D[6]

**Error**: Je použita pokud přepínač potřebuje oznámit kontroléru problém.[6]

**Symetrické zprávy**

Symetrické zprávy jsou zasílány bez požadavku a jsou zasílány oběma směry.[6]

**Hello**: Je vyměňována mezi kontrolérem a přepínačem při započetí komunikace.[6]

**Echo**: Může být použita pro zjištění odezvy, propustnosti a testu připojení kontroléru k přepínači. Po echo request zprávě vždy musí následovat echo reply zpráva.[6]

**Vendor**: Poskytuje standartní cestu pro OpenFlow přepínače k nabídce dalších funkcionalit pomocí OpenFlow standartní zprávy.[6]

### 4.1.4 Navázaní spojení s kontrolérem

Přepínač musí být schopen navázat připojení na fixní IP adresu a port, který definoval uživatel. Poté dochází k pokusu o navázání zabezpečeného kanálu mezi kontrolérem a přepínačem. Pokud se nepovede, naváže se nešifrované spojení. Ihned po navázání spojení musí obě strany vyslat OFPT_HELLO zprávu s nejvyšší podporovanou verzí OpenFlow protokolu. Po přijetí zprávy může příjemce určit, jaká verze protokolu bude použita z odeslaných a přijatých zpráv.[6]

Pokud je daná verze podporovaná zařízením, komunikace pokračuje. Pokud není, musí příjemce odeslat OFPT_ERROR zprávu. Zpráva může obsahovat ASCII zprávu s vysvětlením dané situace. Po odeslání této zprávy dochází k ukončení připojení.[6]

### 4.1.5 Flow tabulky

Flow tabulky obsahují informace a příkazy o tom, jak má přepínač naložit s daným paketem, pokud paket nevyhovuje žádnému pravidlu, je odeslán do kontroléru, který rozhodne co s ním.

Flow záznamy v tabulkách obsahují tři pole:

- **Header field** – pro porovnávání pravidel
- **Counter** – počet shodujících se paketů
- **Actions** - akce, která se použije pro pakety vyhovující pravidlu

| Header Fields | Counters | Actions |
|---|---|---|

Obr. 5. Pole flow tabulky[6]

**Header field**

Obr. 6 ukazuje hlavičku paketu, která se používá pro porovnání se záznamy ve flow tabulce. Flow tabulka obsahuje hodnoty, které jsou porovnávány s hodnotami v jednotlivých polích hlavičky paketu.[6]

| Ingress Port | Ether source | Ether dst | Ether type | VLAN id | VLAN pri-or-ity | IP src | IP dst | IP proto | IP ToS bits | TCP/ UDP src port | TCP/ UDP dst port |
|---|---|---|---|---|---|---|---|---|---|---|---|

Obr. 6. Hlavička paketu, používaná při porovnání[6]

**Counters**

Počítadla jsou udržovány pro tabulky, porty, flow záznamy a pro frontu. OpenFlow kompatibilní počítadla mohou být vytvořena softwarově a spravována hardwarovým počítadlem s omezeným oborem hodnot.[6]

**Action**

Každý záznam ve flow tabulce nemusí být svázán pouze jednou akcí, ale může jich mít i několik. Tyto akce určují, jak se naloží s paketem při porovnávání. Pokud není nastavena žádná akce, paket je zahozen. Přepínač může odmítnout flow záznam, pokud jej nemůže provést. V případě, že dojde k odmítnutí zpracování flow záznamu přepínačem, musí okamžitě oznámit, že tabulka obsahuje nepodporovaný flow záznam (unsupported flow error).[6]

Přepínač nemusí podporovat všechny typy akcí, které OpenFlow protokol podporuje. Přepínač musí podporovat všechny akce, které jsou uvedené jako požadované. Ihned po připojení ke kontroléru, přepínač oznámí, jaké další akce podporuje.[6]

Přepínače podporující OpenFlow protokol se rozdělují do dvou typů:[6]

- **OpenFlow-only** – podporují pouze akce, které jsou vyžadované OpenFlow protokolem.
- **OpenFlow-enable** – Tato zařízení (přepínače, směrovače, přístupové body) také podporují „NORMAL“ akce.

**Požadované akce (Předávání) –** OpenFlow přepínač musí podporovat předávání paketů na fyzický port nebo na virtuální.[6]

- **ALL –** posílá paket všemi porty, kromě toho, kterým paket přišel.
- **CONTROLLER –** všechny pakety jsou zabaleny a odeslány do kontroléru.
- **LOCAL –** Zašle paket do lokálního síťového zásobníku přepínače.
- **TABLE –** Porovná paket s flow záznamy v tabulce.
- **IN_PORT –** zašle paket zpět na port, ze kterého přišel.

**Nepovinné akce (Předávání) -** Přepínač může, ale nemusí podporovat tyto akce nad virtuálními porty:

- **NORMAL –** Zpracování paketů použitím standartních způsobů podporovaných přepínačem (zpracování na L2, VLAN a L3). Přepínač může zkontrolovat pole obsahující VLAN ID, k určení zda dojde k předání paketu pomocí standartních způsobů. Pokud přepínač není schopen předat paket z OpenFlow definované sítě VLAN do normální, musí oznámit kontroléru, že tato funkce není podporovaná.[6]
- **FLOOD**[4] **–** Zaslání paketu na všechny aktivní porty, kromě portu, ze kterého paket přišel.[6]

[4] Flooding se může například použít pro rozesílání aktualizací směrovacích tabulek v rozsáhlých sítích.[7]

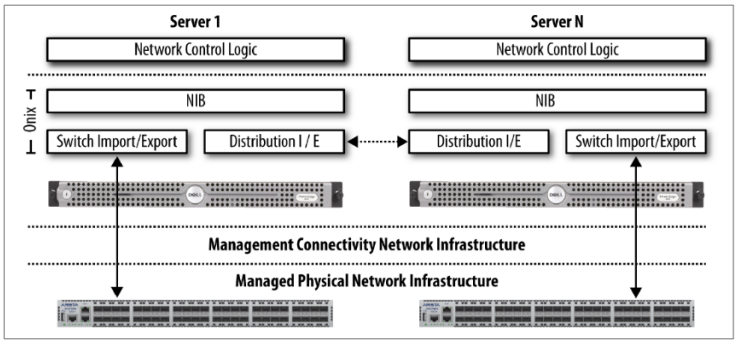
**Nepovinná akce (řazení)** – Řadící akce předá paket do fronty, která je přidružená k portu. Předávání paketů z fronty je řízeno nastavením fronty. Tato funkce se používá pro základní podporu QoS[5].[6]

**Požadovaná akce (zahození)** – Flow záznam neobsahující žádnou akci znamená, že všechny pakety vyhovující tomuto pravidlu by měly být zahozeny.[6]

**Nepovinná akce (modifikace pole)** – i když nejsou striktně vyžadovány akce pro nastavené VLAN, změny zdrojových a cílových IP/MAC adres, zvyšují užitnost OpenFlow protokolu.[6]

### 4.1.6 OpenFlow kontroléry

Nejvíce SDN kontrolérů se vyvinulo okolo protokolu OpenFlow. Většina z nich používá jako základ Onix design, který je znázorněn na Obr. 7. Pouze pár firem integrovalo OpenFlow protokol do svých komerčních produktů jako jediný podporovaný protokol. Spíše se můžeme setkat s komerčními kontroléry, které integrují OpenFlow v kombinaci s dalšími protokoly.[5]




Obr. 7. Model kontroléru Onix[5]

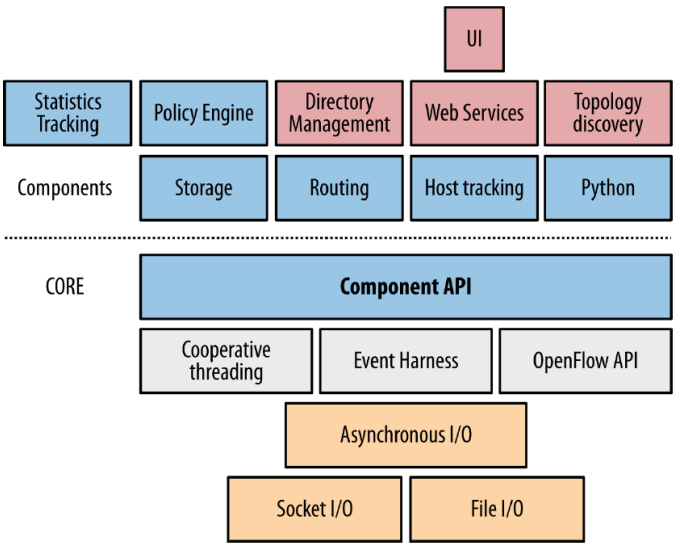
Na rozdíl od VMware/Nicra nebo L3VPN/PCE a dalších, kontrolér postavený na protokolu OpenFlow nepotřebuje žádné další zabalování paketů a brány. Ale pokud budou dvě různé sítě s různými protokoly, bude vyžadován hybridní provoz na portu, který spojuje OpenFlow síť a síť, na které OpenFlow protokol není. Tento hybridní model funkčnosti bude vzhledem k růstu často nasazovaným modelem.[5]

[5] QoS – Quality of Services – slouží k upřednostňování služeb na sítí. Například web má většinou nejnižší prioritu.

**NOX/POX**

NOX kontrolér byl vyvinut začínající firmou Nicra. Kontrolér byl věnován výzkumné komunitě a tím se v roce 2008 stal open source projektem. Tento krok z něj udělal první open source OpenFlow kontrolér. NOX byl následně rozšiřován a podporován ON.LAB[6] aktivitami na Stanfordské univerzitě s významným přispěním z UC Berkeley a ICSI[7]. NOX poskytuje C++ API k OpenFlow protokolu (OF v1.0) a asynchronní, událostní programový model.[5]

NOX se dá využít jako kontrolér a framework založený na komponentách pro vytváření SDN aplikací. Poskytuje podporu modulům specifických pro OpenFlow, ale může být a je rozšiřován. Jádro NOX poskytuje pomocné metody a API, které mohou být použity pro interakci s OpenFlow přepínačem.[5]




Obr. 8. Architektura NOX[5]

POX je novější verze NOX založená na pythonu. Důvod vývoje nového kontroléru byl návrat NOX k C++ a vývoj oddělené platformy založené na pythonu verze 2.7. Kontrolér obsahuje vysokoúrovňové SDN API obsahující podporu pro virtualizaci.[5]

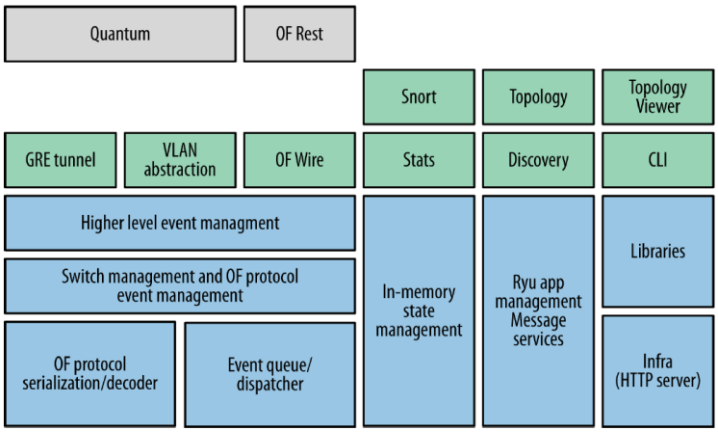
NOX i POX kontrolér je možný stáhnout ze stránky github.com/noxrepo, kde se nachází i jejich poslední verze.

[6] Open Networking Lab – byla založena jako nezisková organizace k prosazování vize, co by networking mohl znamenat pro veřejné blaho.[8]

[7] International Computer Science Institute

**Ryu**

Ryu je založeno na systému komponent a je vydáván jako open source framework napsaný v pythonu. Služba zpráv Ryu podporuje vývoj komponent i v jiných jazycích. Komponenty obsahují podporu protokolu OpenFlow až do verze 1.5 a Nicra rozšíření. Dále obsahuje management událostí, oznámení, management stavů in-memory, aplikační management, služby infrastruktury a další znovupoužitelné knihovny.[5]




Obr. 9. Architektura aplikace Ryu[5]

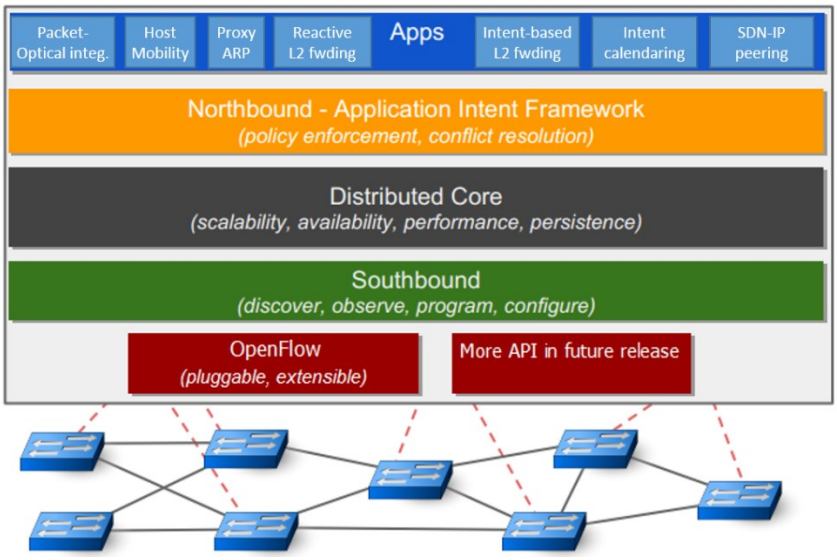
Ryu lze stáhnout ze stránky github.com/osrg/ryu, počítač musí mít nainstalovaný python ve verzi 2.7. Po stažení se musí spustit skript, který Ryu nainstaluje.

Pokud je na počítači nainstalovaný balíčkovací systém pythonu (pip), lze celý kontrolér naistalovat jednoduchým příkazem *pip install ryu* a celá instalace se provede automaticky.

**ONOS**

ONOS je operační systém pro SDN. ONOS je zkratka pro **O**pen **N**etwork **O**perating **S**ystem, celý projekt je vydáván jako open source. V roce 2014 nezisková organizace ON.LAB, která operační systém vyvíjela spolu s dalšími partnery, uvolnila zdrojové kódy. Celý operační systém je psaný v jazyce Java a používá OSGi pro správu funkcí. Jednotlivé funkce systému jsou načítány pomocí OSGi implementovaným v programu Karaf.[18]

Systém má modulární architekturu, která je složená z aplikační a síťové vrstvy, která slouží jako nejvyšší stupeň abstrakce síťové topologie. Tuto vrstvu používají aplikace pro nastavování a spravovaní aktivních prvků. Všechny příkazy z této vrstvy jsou překládány do OpenFlow příkazů, které se poté vkládají na příslušné aktivní prvky za použití „Southbound“ pluginu. Na Obr. 10 je vidět architektura ONOS kontroléru, v nejvyšší vrstvě architektury jsou aplikace, které jsou dostupné v jednotlivých verzích.[18]




Obr. 10. Architektura ONOS kontroléru

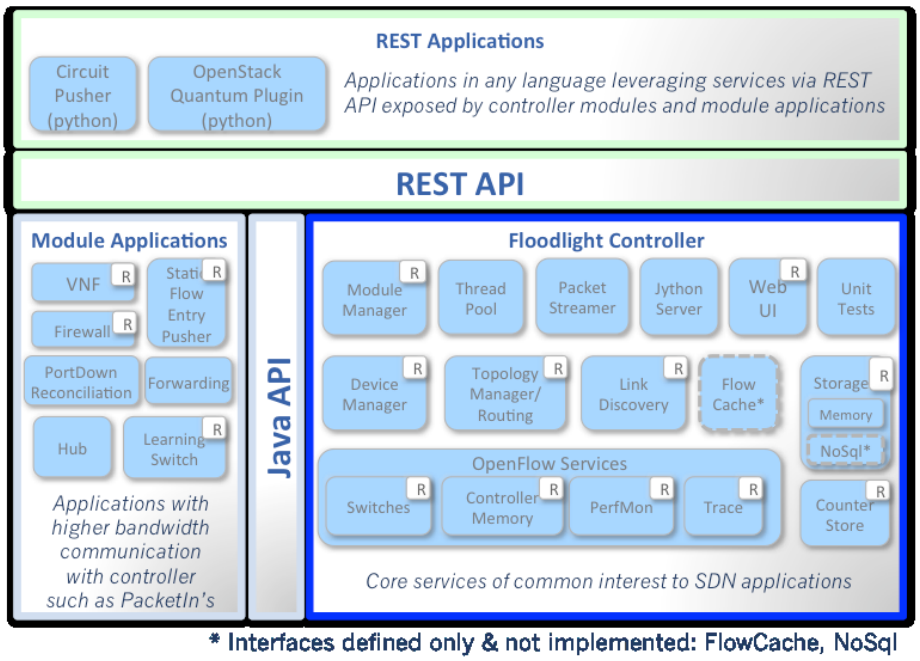
Zdrojové kódy projektu jsou dostupné na platformě github https://github.com/opennetworkinglab/onos. K projektu ONOS je vypracován i manuál, podle kterého lze ONOS zprovoznit, nevýhodou tohoto manuálu je jeho neúplnost, pokud jsou potřeba kompletní informace, je potřeba dát dohromady několik verzí.

## 4.2 Floodlight kontrolér

Floodlight je velice populární SDN kontrolér, který byl věnován open source komunitě firmou Big Switch Network. Floodlight je postaven na kontroléru Beacon, který byl vyvinut na Stanfordské univerzitě. Kontrolér je vydáván pod Appache licencí a je celý napsaný v Javě. Architektura kontroléru stejně jako aplikační rozhraní je sdílené s komerční verzí kontroléru Big Network Controller (BNC), který poskytuje firma Big Switch Network.[5]

Jádro Floodlight kontroléru je modulární a obsahuje komponenty pro management topologie, správu zařízení (sledování MAC a IP), výpočet trasy, infrastrukturu pro webový přístup, uložení počítadel (OpenFlow počítadla) a úložiště pro stavy. Úložiště stavů je standardně uloženo v paměti, ale umožnuje implementaci ukládání do SQL nebo NoSQL databází.[5]

Komponenty jsou brány jako služby s rozhraním umožňující export stravu. Kontrolér jako takový je soubor nestavových aplikačních rozhraní. Skrze aplikační rozhraní aplikace mohou nastavit chování kontroléru nebo se mohou přihlásit k odběru událostí, které kontrolér generuje. Pro přihlášení k odběru se používá Java Event Listener.[5]




Obr. 11. Architektura Floodlight kontroléru[10]

Hlavním modulem OpenFlow kontroléru je Floodlight Provider, který obstarává vstupní a výstupní informace pro přepínače a překládá OpenFlow zprávy do Floodlight událostí. Floodlight implementuje vláknový model, který umožňuje modulům sdílet vlákna s ostatními moduly. Zpracování událostí se provádí pomocí předávání kontextu z modulů běžících ve vlákně. Data jednotlivých modulů jsou chráněna synchronními zámky. Závislosti jednotlivých komponent se řeší během spuštění kontroléru přes konfiguraci.[5]

Správce topologie používá k objevování nových zařízení v sítí LLDP[8], správce vyhledává všechna zařízení a to bez ohledu na to, zda podporují OpenFlow. V kontroléru jsou obsažené vzorové aplikace jako learning switch. Další aplikace jsou hub a static flow pusher. Floodlight také nabízí OpenStack Quantum plug-in.[5]

Floodlight OpenFlow kontrolér může spolupracovat s jakýmkoliv zařízením, které má agenta podporujícího verzi OpenFlow, která je podporována Floodlight protokolem. Big Switch také poskytuje open source agenta Indigo, který byl zahrnut do komerčních produktů. Jako další Big Switch poskytuje open source knihovnu Loxi, která podporuje mnoho jazyků. Knihovna dokáže zpracovávat různé verze OpenFlow protokolu.

### 4.2.1 Možnosti kontroléru

Floodlight kontrolér obsahuje REST API, které umožňuje nastavování kontroléru a definici chování jednotlivých aktivních prvků. Vývojáři implementovali grafické rozhraní, které by mělo usnadnit správu aktivních prvků a mělo by zvýšit přehled administrátora o topologii sítě. Grafické rozhraní je dostupné z webového prohlížeče na adrese http://<controller>:8080/ui/index.html.[14] Grafické rozhraní si lze prohlédnout v příloze 6 a příloze 7.

**ACL**

Kontrolér obsahuje aplikaci firewall, která vynucuje ACL, která jsou aplikovaná na jednotlivé pakety. Kontrolér sleduje jednotlivé pakety a na základě nastavených ACL vkládá do flow tabulek potřebné záznamy. Zmíněný postup platí pro reaktivní jednání firewallu. V proaktivním režimu jsou do přepínačů vložené potřebné flow záznamy a to bez ohledu na to zda je přepínač od kontroléru požadoval.[13] Zmíněné postupy fungují pouze, pokud jsou naučena zařízení s IP adresami, které jsou použity při nastavování ACL.

[8] Link Layer Discovery Protocol – standardizovaný protokol, který se používá pro zjišťování pouze přímo připojených zařízení.[11]

Při vytváření ACL mají přednost pravidla, která byla do přepínače přidána nejdříve. To znamená, že dříve přidaná pravidla mají na přepínači větší prioritu. Jednotlivá pravidla mohou mít prioritu od 0 do 30 000. Všechny ACL záznamy generované firewall aplikací jsou vkládány do přepínačů jako statické záznamy. Tyto záznamy jsou většinou pojmenované „ACLRule_[Rule Id]_[Switch_DPID]“. Pod tímto jménem se dá s tímto statickým záznamem pracovat.[13]

Aplikace je používaná pro filtraci IP paketů mezi různými koncovými zařízeními. To znamená, že uživatel, který přidává záznam, musí uvést zdrojovou i cílovou adresu. Aplikace je také schopna vynutit pravidla, která neobsahují žádnou zdrojovou adresu, takové pravidlo je aplikováno na všechny zdrojové IP adresy v síti. V tomto případě je vložen patřičný záznam do přepínače, na kterém je připojeno zařízení s IP adresou, která je nastavena v cílové adrese. Akce s parametrem „ALLOW“ pouze povolují zakázaná flow. Pokud zakážeme 10.0.0.0/24 příkazem „ALLOW“ můžeme povolit jednu adresu například 10.0.0.2/32.[13]

**Firewall**

Bezstavové rozhraní pro firewall nabízí rozšíření aplikace ACL, která pracuje pouze s IP adresami. Firewall dovoluje vydávat pravidla, která mohou filtrovat pakety například pomocí MAC adres, IP adres, podle protokolu 3. a 4. vrstvy ISO/OSI modelu. Tento modul není plně funkční, pokud nejsou doručovány odpovědi na LLDP zprávy.

| Port | Zdroj. MAC | Cíl. MAC | Eth. Typ | VLAN ID | Zdroj. IP | Cíl. IP | IP prot. | Zdroj. port | Cíl. port | Akce |
|---|---|---|---|---|---|---|---|---|---|---|
| * | * | * | * | * | * | * | * | * | 22 | DROP |

Obr. 12. Funkce firewall OpenFlow

**Static Flow Pusher**

Modul je automaticky načítaný a je obsluhován pomocí bezstavového aplikačního rozhraní. Modul dovoluje vkládání statických flow záznamů do OpenFlow přepínačů. OpenFlow podporuje dva způsoby vkládání flow záznamů. Jsou to způsoby reaktivní a proaktivní. [15]

Reaktivní způsob vkládání flow záznamů se provádí, pokud přepínač nenašel odpovídající flow záznam ve své tabulce. Paket, který nemohl být zpracován na přepínači

je odeslán do kontroléru, který ho vyhodnotí a vloží patřičný flow záznam a nechá přepínač dokončit předání paketu. [15]

Proaktivní způsob vkládání flow vkládá flow záznam do tabulky přepínače dříve, než paket dorazí k přepínači. Při tomto způsobu vkládání záznamů není paket nikdy odeslán do kontroléru k vyhodnocení. [15]

Floodlight kontrolér podporuje oba způsoby vkládání záznamů. Static flow pusher je většinou používaný pro proaktivní vkládání flow záznamů. Kontrolér v původní konfiguraci spouští modul Fowarding, který používá reaktivní metodu vkládání záznamů.

| Port | Zdroj. MAC | Cíl. MAC | Eth. Typ | VLAN ID | Zdroj. IP | Cíl. IP | IP prot. | Zdroj. port | Cíl. port | Akce |
|---|---|---|---|---|---|---|---|---|---|---|
| * | * | AA:BB:... | * | * | * | * | * | * | * | port 6 |

Obr. 13. Příklad flow parametrů pro přepínání

| Port | Zdroj. MAC | Cíl. MAC | Eth. Typ | VLAN ID | Zdroj. IP | Cíl. IP | IP prot. | Zdroj. port | Cíl. port | Akce |
|---|---|---|---|---|---|---|---|---|---|---|
| * | * | * | * | * | * | 8.8.8.8 | * | * | * | port 6 |

Obr. 14. Příklad flow parametrů pro směrování

| Port | Zdroj. MAC | Cíl. MAC | Eth. Typ | VLAN ID | Zdroj. IP | Cíl. IP | IP prot. | Zdroj. port | Cíl. port | Akce |
|---|---|---|---|---|---|---|---|---|---|---|
| * | * | AA:BB:... | * | 1 | * | * | * | * | * | port 6<br>port 7<br>port 8 |

Obr. 15. Příklad flow parametrů pro virtuální sítě

**Filtr virtuálních sítí**

Tento modul slouží pro definici a správu virtuálních sítí v síti OpenFlow. Pomocí tohoto modulu lze logicky uskupovat jednotlivá zařízení.

### 4.2.2 Příkazy pro práci s kontrolérem

Chování Floodlight kontroléru se nastavuje pomocí bezstavového aplikačního rozhraní. V této kapitoly jsou popsána jednotlivá aplikační rozhraní a jejich ovládání, jsou zmíněny pouze příklady použití, pro podrobnější informace doporučuji použít oficiální dokumentaci.

### ACL

Jak již bylo zmíněno v minulé kapitole, modul ACL slouží k nastavování přístupových oprávnění. Pro práci s modulem ACL existují čtyři příkazy, které se odesílají na adresu webového rozhraní kontroléru. Veškeré odpovědi od kontroléru jsou formátované jako JSON. JSON je také očekáván pro příkazy ovlivňující kontrolér.

**Přidání ACL:**

```
curl -X POST -d "{\"src-ip\":\"10.0.0.1/32\",
                  \"dst-ip\":\"10.0.0.2/32\",
                  \"action\":\"deny\"}"
http://<kontrolér_ip>:8080/wm/acl/rules/json
```

**Výpis všech nastavených ACL:**

```
curl http://<kontrolér_ip>:8080/wm/acl/rules/json
```

**Odebrání ACL:**

```
curl -X DELETE -d "{\"ruleid\":\"1\" }"
http://<kontrolér_ip>:8080/wm/acl/rules/json
```

**Odebrání všech ACL:**

```
curl http://<kontrolér_ip>:8080/wm/acl/clear/json
```

### Firewall

Modul firewallu je při spuštění načítán automaticky, ale jeho funkce jsou vypnuté. Pro konfiguraci modulu se používá 5 základních příkazů. Data mezi kontrolérem a přepínačem jsou přenášena v datovém formátu JSON. Tento modul aplikuje pravidla, která nastavíte, pouze pokud je využíván modul pro směrování paketů. Konfigurační příkazy firewallu jsou v příloze 8.

### Filtr virtuálních sítí

Při testování tohoto modulu se nepodařilo vytvořit JSON patřičné struktury, který by parser kontroléru přijal a nastavil požadovaný filtr virtuální sítě. Pro ovládání tohoto modulu jsou celkem tři URL, přes které jde filtr nastavit. Položky {tenant} jsou prozatím ignorované. Konfigurační příkazy pro filtr virtuálních sítí jsou umístěné v příloze 9.

### Static Flow Pusher

Tento modul je přístupný přes REST API. Pro práci s tímto modulem je potřeba znát tři základní příkazy, které vkládají flow záznamy do aktivních prvků, získávají vložené flow záznamy z aktivních prvků a mažou flow záznamy.

**Nastavení / Smazání flow záznamů**

```
curl -X POST -d "{\"switch\": \"00:00:00:00:00:00:00:00\",
                  \"name\":\"flow1\",
                  \"cookie\":\"0\",
                  \"priority\":\"100\",
                  \"in_port\":\"6\",
                  \"active\":\"true\",
                  \"actions\":\"output=8\"}"
http://<kontrolér_ip>:8080/wm/staticflowpusher/json
curl -X DELETE -d "{\"name\":\"flow1\"}"
http://<kontrolér_ip>:8080/wm/staticflowpusher/json
```

| Port | Zdr. MAC | Cíl. MAC | Eth. Typ | VLAN ID | Zdr. IP | Cíl. IP | IP Prot. | Zdr. port | Cíl. port | Akce |
|---|---|---|---|---|---|---|---|---|---|---|
| Port 6 | * | * | * | * | * | * | * | * | * | Port 8 |

Obr. 16. Porovnání paketu podle vloženého flow

**Výpis flow záznamů v aktivním prvku**

```
curl http://<kontrolér_ip>:8080/wm/staticflowpusher/list/
     <datapath_id[9]>/json
```

**Smazání všech flow záznamů**

```
curl http://<kontrolér_ip>:8080/wm/staticflowpusher/clear/
     <datapath_id>/json
```

Kompletní výčet parametrů, které se dají použít, lze nalézt v dokumentaci kontroléru.

[9] datapath_id – jedinečný identifikátor aktivního prvku v kontroléru

# 5 Problémy nutné k řešení

V kapitole jsou rozebrány jednotlivé problémy, které bylo nutné řešit před nasazením OpenFlow kontroléru. Jsou rozebrány problémy, které vznikaly buď na straně kontroléru, nebo aktivního prvku. V poslední kapitole je popsaná topologie, která byla použita pro pokus.

## 5.1 Implementace OpenFlow protokolu

Největším problémem OpenFlow protokolu je implementace verzí jednotlivými výrobci. Některá zařízení, která byla testovaná s open source kontrolérem, havarovala už při prvotním párování. Většinou se nepovedla výměna Hello zprávy. Protože kontrolér dodržuje instrukce OpenFlow protokolu a propaguje nejvyšší verzi OF. Ale zařízení, které přijímá OF Hello zprávu, tuto verzi nepodporuje a odpoví kontroléru chybou. Pokud by byl protokol správně implementován, zařízení by vybralo takovou verzi, kterou podporuje jak kontrolér, tak samotné zařízení. Bohužel realita je jiná, a proto je potřeba upravit kontrolér.

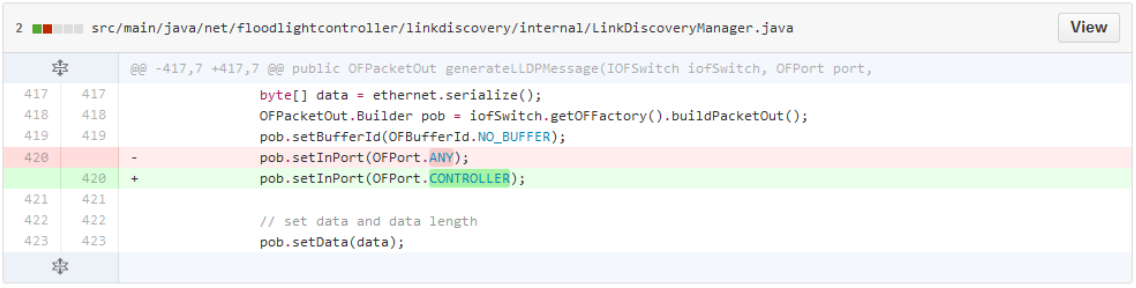
Protože není možné aby OpenFlow protokol byl takto hloupě navržen a kontroléry nebyly schopné připojit zařízení, která mají problém s vyššími verzemi protokolu, byl vyzkoušen HP VAN SDN kontrolér. Prvotní instalace je sice o něco složitější než zprovoznění většiny open source kontrolérů. Tato složitější instalace je ale vyvážená okamžitou funkčností OF kontroléru bez problémů, který byly zmiňovány výše. Na většině OF zařízení se ihned rozběhne základní síťový provoz a povolí se základní protokoly jako je ARP a IP protokol. V tomto případě je vidět, že společnost HP to myslí s SDN vážně.

Text níže bude obsahovat informace o implementaci OF protokolu jednotlivými výrobci v různých zařízeních. U jednotlivých zařízení jsou také zmíněny chyby, které vznikly v průběhu testování. Většina chyb, které vznikly, byly spíše způsobené implementací OF kontroléru. Většina informací o implementaci OF protokolu v jednotlivých zařízeních je převzata z oficiální dokumentace výrobce zařízení.

### 5.1.1 Zařízení H3C (HP) Comware

V topologii je zařazen přepínač HP 5130-48G se systéme Comware 7. Tento aktivní prvek podporuje OpenFlow verze 1.3. Přepínač šel připojit ke kontroléru pomocí několika příkazů. Vše proběhlo rychle a také to vypadalo, že bude vše fungovat jak má. Ihned po připojení zařízení do přepínače, by měl aktivní prvek nahlásit připojené zařízení kontroléru. Bohužel se kontrolér z tohoto přepínače nenaučil žádná zařízení.

Další chybou, kterou kontrolér vykazoval, byla zpráva o nemožnosti odeslání paketu na port, který kontrolér vyžadoval. Přepínač vracel chybovou hlášku s parametrem „BAD_PORT“. Po bližším přezkoumání problému bylo zřejmé, že není chyba na straně operačního systému Comware. Chyba samotná byla v kontroléru, který do parametru pro vstupní port přidával hodnotu „ANY“. Podle specifikace protokolu musí kontrolér do těchto zpráv dávat jako vstupní port hodnotu „Controller“. Chyba byla v kontroléru opravena a žádost o začlenění opravy do zdrojových kódů Floodlight kontroléru a byla úspěšná.




Obr. 17. Část opraveného kódu kontroléru

### 5.1.2 Zařízení HP ProCurve ProVision

V topologii je zařazen přepínač HP 5406zl se systémem ProVision. Tento aktivní prvek podporuje OpenFlow verze 1.0 a 1.3. Při použití OF verze 1.3 měl kontrolér problém s vkládáním flow záznamů do tabulky přepínače. Opět se vyskytl problém s učením zařízení připojených k přepínači. To znamenalo, že modul, který vkládá automaticky flow záznamy, opět nefungoval. Naštěstí operační systém dovoluje výběr verze OF protokolu. Po nastavení verze 1.0 se přepínač začal chovat správně, začal oznamovat, která zařízení byla připojená k portům, a začal přijímat flow záznamy.

Stejně jako Comware přepínač obsahuje tabulku, do které si ukládá naučené MAC adresy. Díku tomu jsou oba přepínače schopny pracovat jako L2 zařízení při vložení pouze jednoho flow záznamu.

### 5.1.3 Zařízení Mikrotik RouterOS

V topologii je zařazen Mikrotik RB2011UiAS-2HnD-IN se systémem RouterOS 6.34.2. Přepínač podporuje OpenFlow verze 1.0. Mikrotik byl prvním aktivním prvkem, který byl testován s Floodligt kontrolérem. S Floodlight verze 0.90 nebyl problém s připojením prvku ke kontroléru, ale v novějších verzích už problém je. Protože kontrolér podle předpisu protokolu vždy propaguje nejvyšší verzi protokolu, kterou podporuje. Proto nastává situace, kdy prvek není schopný rozpoznat Hello zprávu kontroléru a odpoví chybou. Proto bylo nutné najít část kódu, která je zodpovědná za připojení prvku ke kontroléru a změnit toto chování. Proto byl kontrolér upraven tak, že čeká, až přijme Hello zprávu od aktivního prvku a až poté odešle Hello zprávu aktivnímu prvku ve verzi, kterou podporuje.

```
357             }
358             sendHelloMessage();
359             setState(new WaitFeaturesReplyState());
360         }
397         @Override
398         void enterState() throws IOException {
399             // sendHelloMessage();
400         }
401     }
```

Obr. 18. Upravená část kódu pro Mikrotik

### 5.1.4 Zařízení TP-Link OpenWRT

Operační systém dodávaný firmou TP-Link protokol OpenFlow nepodporuje. Ale díky aktivní komunitě vyvíjející operační systém pro síťové prvky s názvem OpenWRT, lze podporu OpenFlow přidat. Aktivní prvek od firmy TP-Link patří mezi nejlevnější aktivní prvky, které byly testovány a je primárně určen pro domácnosti a malé firmy. Ale do topologie byl zařazen kvůli zjištění, zda i malé firmy s levnějšími aktivními prvky mohou využívat výhody SDN.

Do topologie byl připojen TP-Link WR1043ND v2, na který je možné naistaloval OpenWRT. OpenWRT jako takové nenabízí podporu OpenFlow protokolu, ale může být doinstalována nebo přidána. Jednodušší cestou je instalace aplikace Open vSwitch, který je dostupný jako balíček a je možné jej naistaloval jedním příkazem. Aplikace je vydávána pod licencí Appache 2.0. Open vSwitch podporuje OpenFlow verze 1.3. Velkou nevýhodou této aplikace je, že nelze vkládat flow záznamy z kontroléru. Aplikace zprostředkuje připojení k OF kontroléru, poskytne data o flow záznamech a portech, které jsou zařazené do OpenFlow přepínače. Tím to ale končí, protože kontrolér není schopný se naučit připojená zařízení a nedokáže určit propojení mezi jednotlivými aktivními prvky.

Lepší cestou pro instalaci OpenFlow do OpenWRT je využití zdrojových kódů, které uvolnila Stanfordská univerzita. Zdrojové kódy je možné stáhnout platformy git, kterou vlastní sama univerzita. Po integraci zdrojových kódů do OpenWRT bude aktivní prvek podporovat OF verze 1.0. Největším problémem integrace OF1.0 jsou samotné návody, které jsou ve většině případů vydané kolem roku 2013. Při postupu podle těchto návodů se kompilace zdrojových kódů nikdy nezdaří.

Po chvíli hledání lze naleznout zdrojové kódy pro klienta s podporou verze 1.3, která je postavená na zdrojových kódech výše zmíněného klienta.

Při procházení návodů, byly nalezeny i zkompilované verze s podporou OF, ale v reálném provozu je riziko takový operační systém použít, protože nikdo neví, co do něj autor mohl přidat.

## 5.2 Kompatibilita OpenFlow kontroléru

Pouze jeden aktivní prvek měl psanou podporu ze strany kontroléru. Proto se dalo očekávat, že ostatní aktivní prvky budou mít při komunikaci s OF kontrolérem nějaký problém. Většina problémů se dala vyřešit zásahem do zdrojových kódů kontroléru.

Při připojování jednotlivých aktivních prvků k různým verzím OF kontroléru bylo zjištěno, že největší problém s komunikací má právě podporovaný aktivní prvek s operačním systémem ProVision. Při pokusech o vložení flow záznamu, byl kontrolér vždy neúspěšný. V seznamu s podporovanými zařízeními se nezmiňují o verzi, se kterou bylo zařízení testováno a tak byla ponechána původní nastavení s nejvyšší podporovanou verzí 1.3. Po neúspěšných pokusech o vložení flow záznamu proběhla modifikace nastavení a změna podpory OF z 1.3 na 1.0. Přepnutí na aktivním prvku není nic složitého, složitější to bylo pro kontrolér, který měl uložené identifikační číslo aktivního prvku ve své interní databázi a při pokusech o připojení, kontrolér vyhazoval výjimky. V reálném životě to znamená vypnout kontrolér, smazat jeho záznamy a znovu jej spustit. Tato možnost je v laboratorních podmínkách přijatelná, ale ne pro větší sítě. Po všech těchto úpravách se začal prvek chovat standardně, začal přijímat flow záznamy a propagovat zařízení připojená na jeho portech.

Floodlight podporuje OF do verze 1.4, s touto verzí je také vysílána zpráva Hello. Ale jak již bylo zmíněno v kapitole 5.1.3, některá zařízení mají problém porozumět této zprávě a vrací kontroléru chybovou zprávu. To způsobí neustále připojování a odpojování aktivního prvku. Toto chování lze upravit specifikací podporovaných OF verzí, ale aktivní prvky podporující vyšší verze, než jsou podporovány kontrolérem, jsou odmítnuty.

### 5.2.1 HP VAN SDN kontrolér

Tento kontrolér nabízí poměrně to samé jako Floodlight, ale podle mého názoru je vše dotažené a funkční. Kontrolér nabízí podporu OpenFlow veze 1.0 a 1.3. Kontrolér stejně jako jeho open source protějšek nabízí REST API pro aplikace, které chtějí ovlivňovat chod a provoz síťové infrastruktury.[16]

Architektura kontroléru je rozdělena na tři části:[16]

- **North Bound APIs** – poskytuje HTTPS RESTful rozhraní, které umožňuje aplikacím ovlivňovat chování kontroléru a sítě.
- **Native APIs** – poskytuje Java aplikacím možnost spolupracovat s kontrolérem. Tyto aplikace mají přístup k paketům, které jsou zasílány do kontroléru a mohou s nimi manipulovat.
- **Southbound API** – je používané pro komunikaci s aktivními prvky OF sítě.

Ihned po instalaci je kontrolér schopen začít spolupracovat s OpenFlow přepínači. Po připojení aktivního prvku jsou vloženy statické flow záznamy, které umožní normální funkci sítě. Kontrolér obsahuje pěkné grafické rozhraní, které poskytuje základní informace o topologii, připojených zařízeních a informace o aktivních prvcích, které s kontrolérem komunikují. Jako první začnou fungovat aktivní prvky od firmy HP. Na ProVision přepínači byla opět nastaven podpora OF 1.3 a nebyl problém se vkládáním flow záznamů.

Připojení proběhlo úspěšně i s Mikrotikem, který podporuje pouze OF 1.0. Nevznikl žádný problém s odpojováním a připojováním přepínače. Jediný problém byl s aplikací Open vSwitch, která nebyla funkční ani s tímto HP kontrolérem.

Po naučení zařízení, která jsou připojená k jednotlivým přepínačům, je kontrolér schopný vkládat flow záznamy do tabulek automaticky. Opět ale platí, jakmile připojíte jeden prvek, který není schopný propagovat zařízení připojená k portům, modul přestane fungovat.

HP VAN SDN kontrolér není vydáván jako open source a není poskytován s licencí, která by umožňovala jeho bezplatné použití. Pro správnou funkčnost a legální využívání je potřeba zakoupit základní licenci, která dovolí správu 50 aktivních zařízení. Pro zaměstnance HP a HP partnery je k dispozici 60-ti denní demo licence. Postup instalace a zprovoznění HP VAN SDN kontroléru bude zmíněn v příloze 3.

### 5.2.2 ONOS

Tento kontrolér má velice jednoduchou instalaci, stačí postupovat krok za krokem podle manuálu a kontrolér je během 15 minut v provozu. Kontrolér nemá problém s jednotlivými verzemi OpenFlow protokolu. Ihned po nastavení aktivního prvku se vše správně zobrazí ve webovém grafickém rozhraní.

Grafické rozhraní kontroléru je jedno z nejlepších mezi kontroléry, které jsou zmíněné v této práci. Grafické rozhraní umožňuje správu jednotlivých aplikací, které se dají za chodu kontroléru vypínat nebo zapínat. Součástí instalace kontroléru je plno aplikací, které jsou užitečné pro správu sítě (OpenFlow provider, LLDP provider, Reactive forwarding, atd). Zmíněné aplikace jsou zařazené jako test aplikace, ale během testování fungovaly dobře. Nabídka nastavení je trochu zavádějící, nedá se zde nic nastavit, tato sekce slouží pouze pro přehled nastavení jednotlivých aplikací.

Nejzajímavější částí webového rozhraní je přehled topologie sítě, který funguje výborně. Na rozdíl od testovaných kontrolérů, je toto rozhraní aktuální a nezpůsobuje zamrznutí webového prohlížeče. V tomto náhledu si lze prohlédnout potřebné informace o aktivních prvcích a připojených zařízeních.

V náhledu na porty jednotlivých aktivních prvků, ONOS jako jediný ukazuje aktuální stav portu i s počtem přenesených dat. Při testování tohoto kontroléru byly použity pouze dva aktivní prvky a to Mikrotik a TP-Link. Po připojení hostů ke kontroléru se ihned projeví funkčnost reaktivního vkládání, která umožní komunikaci mezi hosty a také do internetu.

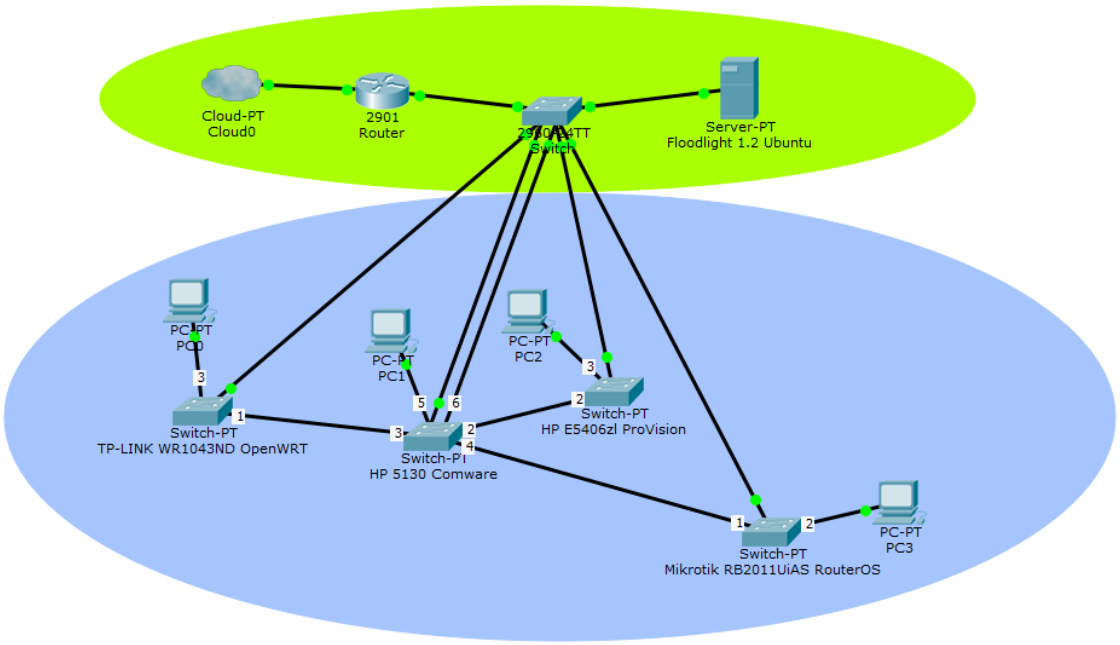

Obr. 19. Logo projetku ONOS[19]

## 5.3 Návrh síťové topologie

Na Obr. 20 je znázorněna topologie sítě, na které byl protokol OpenFlow testován. Fyzické zapojení topologie je zobrazeno v příloze 4 a příloze 5. Aktivní prvky v zeleném poli nejsou spravovány pomocí OpenFlow příkazů. Prvky v modrém poli jsou součástí OpenFlow sítě.

Aktivní prvky v zeleném poli slouží pro spojení aktivních prvků a kontroléru. Pro tuto síť je na přepínači vytvořena virtuální síť s identifikátorem 10, do ní jsou přiřazeny porty 1-5. Těchto 5 portů podporuje standard Fast Ethernet. Z OF sítě existuje jeden spoj pro internetové připojení, pro tuto síť je vytvořena virtuální síť 20 a je použit gigabitový port. Směrovač má dva gigabitové porty, jeden je připojen do sítě JČU a druhý do druhého gigabitového portu přepínače, který je v módu trunk.

Prvky v modrém poli obsahují čtyři testované aktivní prvky s podporou OpenFlow protokolu a 4 počítače, na kterých byly ověřovány jednotlivé flow záznamy. Vždy první port aktivního prvku slouží pro spojení s kontrolérem. Do druhého portu jsou zapojovány jednotlivé spoje mezi aktivními prvky, koncové stanice jsou většinou ve třetích portech. Čísla u jednotlivých aktivních prvků neoznačují číslo fyzického portu, ale číslo, které je přiděleno portu v sítí OpenFlow.




Obr. 20. Síťová topologie

# 6 Instalace a ověření topologie

Kapitola obsahuje nutné kroky pro úspěšné zprovoznění sítě podporující OpenFlow protokol. Je zde popsán kompletní postup pro instalaci OpenFlow kontroléru, připojení jednotlivých aktivních prvků včetně instalace a kompilace operačního systému pro TP- Link.

## 6.1 Výběr OS a instalace prerekvizit

Pro běh kontroléru byla vybrána linuxová distribuce Ubuntu. Tato distribuce byla zvolena z několika důvodů. Prvním důvodem je velké množství ovladačů pro různý hardware. Dalším důvodem je balíčkovací systém, který obsahuje všechny potřebné programy pro provoz SDN kontroléru.

V první fázi testování byl využíván Windows 7 od Microsoftu. Bohužel tento systém nevyhovoval. Floodlight využívá k běhu open source verzi Java Virtual Machine a oproti Javě, kterou vydává Oracle, obsahuje balíčky, které nejsou ve standardní verzi Javy dostupné. Další nevýhodu je složité nastavování kompilačního programu make.

Ze serverů Ubuntu byl stažen Ubuntu 15.10. Je jedno, zda je zvolena verze pro desktop nebo pro server. Pokud je účelem testovat a upravovat zdrojový kód kontroléru, je lepší použít desktopovou verzi. Desktopová verze Ubuntu také obsahuje jednoduchého průvodce instalací a proto je instalace velice jednoduchá.

Po instalaci OS je nutné naistalovat tyto balíčky, které jsou potřeba pro chod kontroléru a kompilaci OS pro TP-Link.

**Instalace potřebných balíčků**

```
sudo apt-get update

sudo apt-get install build-essential default-jdk ant python-dev
eclipse autoconf binutils bison build-essential ccache flex gawk
gettext git libncurses5-dev libssl-dev ncurses-term quilt
sharutils subversion texinfo xsltproc zlib1g-dev
```

## 6.2 Spuštění OpenFlow kontroléru

Před samotným spuštěním kontroléru je nutné stáhnout jeho zdrojové kódy, které jsou dostupné na platformě github nebo na stránkách Floodlight projektu. Doporučuji stáhnout zdrojové kódy z githubu, který obsahuje vždy poslední verzi kontroléru.

Po stažení zdrojových kódů je potřeba provést kompilaci zdrojových kódů pomocí nástroje ant. Při úspěšné kompilaci je vytvořena složka target, kde jsou umístěny binární balíčky.

**Stažení a instalace kontroléru**

```
git clone git://github.com/floodlight/floodlight.git
cd floodlight
ant
```

Pokud je potřeba zdrojový kód upravovat, existuje možnost vytvoření projektu pro Eclipse IDE.

**Vytvoření projektu pro Eclipse**

```
sudo chmod 777 setup-eclipse.sh
ant eclipse
```

Po úspěšném sestavení, lze projekt importovat do Eclipse IDE a začít s úpravami. V src/main/java/net/floodlightcontroller/core/internal/OFChanelHandler.java je nutné v metodě enterState zrušit volání metody sendHelloMessage() viz Obr. 21.

```
362        @Override
363        void enterState() throws IOException {
364            //sendHelloMessage();
365        }
366    }
```

Obr. 21. Upravený řádek

Tímto zásahem se zakáže odeslání Hello zprávy do aktivního prvku. Proto se musí ve stejném souboru zajistit odeslání zprávy, po přijetí Hello zprávy z aktivního prvku.

```
358            sendHelloMessage();
359            setState(new WaitFeaturesReplyState());
360        }
```

Obr. 22. Vložený příkaz pro odeslání Hello zprávy

Po těchto úpravách je nutné nakonfigurovat základní vlastnosti Floodlight kontroléru. Konfigurační soubor je uložený ve složce /src/java/resources/floodlight /floodlightdefault.properties, kde je nutné specifikovat port, na který se budou aktivní prvky připojovat. Protože Mikrotik není schopný měnit port, musí se nastavit port 6633. Další úpravou je definice podporovaných OF verzí, v aktuálním případě nebyly potřebné verze 1.4 a 1.2.

**Úpravy konfiguračního souboru**

```
net.floodlightcontroller.core.internal.FloodlightProvider.openFlowPort=6633

net.floodlightcontroller.core.internal.OFSwitchManager.supportedOpenFlowVersions=1.0, 1.1, 1.3
```

Všechna ostatní nastavení jsou ponechaná v původním nastavení. Pokud není žádoucí, aby byl funkční modul pro automatické vkládání flow záznamu, musí se tento modul vypnout. Mohou se také definovat klíče, podle kterých mají být pakety předávané.

**Nastavení automatického vkládání flow**

```
net.floodlightcontroller.forwarding.Forwarding.match=vlan, mac, ip, transport
```

Pokud byla provedena pouze kompilace zdrojového kódu bez jeho úprav v Eclipse IDE, stačí kontrolér pouze spustit.

**Spuštění kontroléru**

```
sudo java -jar target/Floodlight.jar
```

Pokud je vše správně nainstalované kontrolér začne vypisovat do konzole logovací zprávy. Mezi prvními výpisy jsou zprávy o spuštění služeb kontroléru s výpisy jejich portů nebo stavů. Pokud se neobjeví žádné chyby, je kontrolér připravený pro přijímání připojení od aktivních prvků. Podle nastavení, které bylo zmíněné výše, bude kontrolér naslouchat na portu 6633 a podporovat OF verze 1.0, 1.1 a 1.3.

## 6.3 Nastavení aktivních prvků

Kapitola obsahuje popis nastavení jednotlivých aktivních prvků a stavových informací, které jsou vypisovány. Pokud bylo třeba udělat nějaký speciální krok pro zprovoznění OpenFlow protokolu, bude zde popsán. Nejjednodušší nastavení mají certifikovaná zařízení.


Obr. 23. Certifikát OpenFlow[17]

### 6.3.1 Zařízení HP (H3C) 5130 Comware

Nejjednodušší konfiguraci OpenFlow protokolu má právě zařízení HP 5130 se systémem Comware. Aktivní prvek je nutné konfigurovat pomocí sériového portu.

Prvním krokem je nastavení IP adresy pro umožnění komunikace s kontrolérem. Druhým krokem je vytvoření VLANy pro porty, které budou součástí sítě OpenFlow.

**Základní konfigurace systému Comware**

```
system-view
hostname OpenFlow_switch
vlan 10
name Management_vlan
int vlan 10
ip add dhcp
int g1/0/1
port access vlan 10
vlan 20
name OpenFlow
int ran g1/0/2 to g1/0/20
port access vlan 20
```

| **Konfigurace OpenFlow** |
| --- |
| **system-view**<br>**openflow** instance 1<br>**classification** vlan 20<br>**controller** 1 address ip <IPv4_adresa_kontroléru><br>**active instance** //zapnutí instance<br>**un active instance** //vypnutí instance |

Výše uvedené příkazy nastavují pouze OpenFlow a neřeší bezpečnost aktivního prvku. Výpis událostí OpenFlow je v původní konfiguraci vypnut a není dobré to měnit., pokud bude použito reaktivní vkládání flow záznamů. Při tomto typu vkládání flow záznamu se konzole plně zahltí a nelze do ní již psát příkazy.

| **Zapnutí výpisu událostí OpenFlow** |
| --- |
| **debugging** openflow event<br>**un debugging** openflow event //vypnutí výpisů |

Po sadě těchto příkazů je Comware schopný připojit se ke kontroléru. Zařízení, které bylo zapůjčené, patří mezi základní modely podporující SDN. Při proaktivním vkládání flow záznamů je aktivní prvek vytěžován na 70 – 80% jeho výpočetní kapacity.

### 6.3.2 Zařízení HP ProCurve E5406zl ProVision

Aktivní prvky se systémem ProVision musejí mít definovanou VLANu pro komunikaci s kontrolérem. Aktivní prvky s operačním systémem Comware toto mít nemusejí.

Opět jako u Comwaru je dobré začít s konfigurací aktivního prvku přes sériový port, ale není to nutné, protože ihned pro zapnutí a připojení do sítě s DHCP serverem si přednastavená VLAN 1 získá adresu. Pro povolení SSH je nutné jej povolit pomocí webového rozhraní.

**Základní konfigurace ProVision**

```
system-view
hostname OpenFlow_swhitch2
vlan 10
name Management
int vlan 10
ip add dhcp
int g1/0/1
port access vlan 10
vlan 20
name OpenFlow
int ran g1/0/2 to g1/0/2
port access vlan 20
```

**Konfigurace OpenFlow ProVision**

```
system-view
openflow
controller-id 1 ip <IPv4_adresa_kontroléru>  controller-
interface vlan 10
instance 1
controller-id 1
member-vlan 20
version 1.0 //1.3 zapne podporu OF1.3
enable //zapne instanci OpenFlow disable vypne instanci OF
quit
enable // zapne OF
```

S tímto aktivním prvkem nebyly žádné větší potíže při nastavování, kromě problému vkládání flow záznamů při zapnuté podpoře OF 1.3.

### 6.3.3 Zařízení Mikrotik RB2011UiAS RouterOS

Zařízení Mikrotik v původní konfiguraci nepodporuje OpenFlow protokol, ale lze jej doinstalovat. Protože aktivní prvky Mikrotik lze spravovat pomocí grafického rozhraní nebo příkazové řádky, jsou zmíněny obě metody.

Pro správu pomocí grafického rozhraní se využívá aplikace Winbox, kterou lze stáhnou z oficiálních stránek výrobce www.mikrotik.com/download. Na stejné stránce je potřeba stáhnout extra balíčky, ve kterých je obsažen i OpenFlow instalační balíček. Pro tento typ aktivního prvku je potřeba stáhnout balíček ze sekce MIPSBE. Po stažení balíčku je potřeba se připojit k aktivnímu prvku a vložit instalační balíček do „Files“. Po úspěšném nahrání se musí aktivní prvek restartovat, při načítání systému se balíček automaticky naistaluje.

Po provedení předchozích kroků je aktivní prvek připraven pro konfiguraci OpenFlow. V aplikaci Winbox přibyde další záložka s názvem OpenFlow. Po otevření této záložky se zobrazí okno, kde je možné provádět jednotlivá nastavení. Pod záložkou „Ports“ lze přidávat porty do OpenFlow přepínače. Podle pořadí vložení, se portu přiděluje číslo, pod kterým ho lze najít v OF kontroléru. Pokud je potřeba změnit tato čísla, musí být odebrána instance OF přepínače a znovu vytvořena.

Jako u předchozích aktivních prvků je potřeba nastavit klienta DHCP na portu, který je připojen ke kontroléru. Po úspěšném nastavení a nastaveném reaktivním vkládání flow záznamů lze jednotlivé flow záznamy vidět pod záložkou „Flows“. Celý postup si lze prohlédnout v příloze 10.

Nastavení Mikrotiku pomocí konzole je oproti Winboxu delší. Příkaz pro přidání portu se musí opakovat podle počtu portů, které se mají přidat do OF přepínače.

**Nastavení OpenFlow RouterOS**

```
openflow add name=OpenFlow controller=<IPv4_adresa_kontroléru>
passive-port=6633

openflow port add interface=<název_portu> switch=OpenFlow

ip dhcp-client add interface=ether1

openflow enable OpenFlow

openflow disable OpenFlow //pro vypnutí OF
```

### 6.3.4 Zařízení TP-LINK WR1043ND OpenWRT

Originální operační systém dodávaný firmou TP-Link nepodporuje OpenFlow protokol, protože celý produkt je mířený na domácí uživatele a malé firmy, kde se s využitím SDN protokolů nepočítá. Tento typ aktivního prvku má dostatečný výkon i paměť pro nasazení linuxového operačního systému OpenWRT. Tento OS se dá rozšířit o podporu OpenFlow protokolu dvěma způsoby. První z těchto dvou způsobů je instalace open VSwitch aplikace a druhý je integrace podpory OpenFlow protokolu přímo do binárního souboru systému. V textu níže budou zmíněny oba způsoby jak přidat podporu protokolu.

**Instalace open VSwitch**

Přechod z operačního systému TP-Link na OpenWRT je díky binárním souborům, které vytvořili autoři tohoto systému velice jednoduché. Na stránkách projektu lze stáhnout aktuální verzi systému. Je důležité vědět revizi aktivního prvku, tento model se prodává již ve své třetí revizi. Stránku produktu naleznete na wiki.openwrt.org/toh/tp-link/tl-wr1043nd. Při přechodu z původního operačního systému na OpenWRT se doporučuje instalace verze factory. Po stažení binárního souboru se provede reinstalace operačního systému pomocí webového rozhraní.

Po úspěšném upgradu je aktivní prvek připraven a webové rozhraní Luci je dostupné na adrese 192.168.1.1/24, existuje také možnost použít telnet. SSH není funkční, dokud není nastaveno heslo pro administrátora. Po nastavení hesla se vypne telnet a zapne SSH. Po připojení k terminálu lze začít s instalací Open vSwitch.

| **Instalace open VSwitch** |
| --- |
| **opkg update**<br>**opkg install** openvswitch |

Po instalaci aplikace je nutné provést konfiguraci síťových rozhraní, která lze udělat pomocí webového rozhraní Luci nebo editací souboru /etc/config/network. Příklad konfigurace je v příloze 1. Je potřeba také upravit firewall /etc/config/firewall příklad v příloze 2.

**Konfigurace openVSwitch**

```
ovs-vsctl add-br oflan \
-- set bridge oflan protocols=OpenFlow10,OpenFlow13 \
-- set-controller oflan tcp:<IPv4_adresa_kontroléru>:6633 \
-- add-port oflan eth1.1 \
-- add-port oflan eth1.3 \
-- add-port oflan eth1.4 \
-- add-port oflan eth1.5
```

Ihned po tomto nastavení se zařízení připojí k OF kontroléru. Floodlight kontrolér není schopný vložit jakýkoliv flow záznam do zařízení, takže tato aplikace není použitelná s tímto kontrolérem. Do zařízení se dá OpenFlow integrovat kompilací zdrojových kódů. Zdrojové kódy OpenFlow se musí kompilovat se zdrojovými kódy systému OpenWRT. Tato operace je časově náročná, na novějších počítačích zabere něco kolem 30 minut. Tato doba čekání je vyvážená plnou funkčností OpenFlow. V níže zmíněném postupu je ukázána kompilace OpenFlow verze 1.0 a 1.3. Nelze zkompilovat verze 1.0 a 1.3 do jednoho binárního balíčku, proto je nutné vybrat, která verze se má kompilovat. Ke kompilaci je dobré používat poslední verzi OpenWRT ve verzi branch. Při použití verze trunk může docházet k neúspěšné kompilaci. Odkazy na zdrojové kódy je možné najít na dev.openwrt.org/wiki/GetSource. Kompletní postup kompilace je v příloze 11.

Po kompilaci jsou binární balíčky uložené v ~/openwrt/openwrt/bin/ar71xx/. Pokud je na zařízení stále OS od TP-Linku, postup je stejný, jak byl zmiňován na začátku kapitoly. Pokud je na prvku OpenWRT, je nutné nahrát binární balíček do složky tmp pomocí SCP.

**Instalace OS openWRT**

```
sysupgrade /tmp/<název_binárního_balíčku>.bin
```

Po úspěšném upgradu je nutné provést instalaci balíčku tc a přidání odkazů na tento program pro OpenFlow, který bez něj nefunguje.

**Instalace potřebných balíčků**

```
opkg update
opkg install luci tc
cd /sbin
ln -s /usr/sbin/tc
cd /etc
ln -s /lib/functions.sh
```

Nastavení samotného OF klienta je v /etc/config/openflow.

**Nastavení OpenFlow OpenWRT**

```
config 'ofswitch'
        option 'dp' 'dp0'
        option 'dpid' '000000000001'
        option 'ofports' 'eth1.1 eth1.3 eth1.4 eth1.5'
        option 'ofctl' 'tcp:192.168.104.253:6633'
        option 'mode'  'outofband'
```

Po této konfiguraci už zbývá spuštění OF klienta.

**Spuštění OpenFlow OpenWRT**

```
/etc/init.d/openflow start
/etc/init.d/openflow stop //vypnutí OpenFlow
```

## 6.4 Testování topologie

Do topologie byla připojena čtyři zařízení. V první fázi testování byl problém s aktivním prvkem HP 5130. Jako jediný nepředával pakety kontroléru, při prvním spuštění. To znamenalo vytvoření potřebných flow záznamů, které by umožnily komunikaci jednotlivých zařízení mezi sebou a internetem. Při vytváření takovýchto flow záznamů jsou jednotlivá zařízení vázaná k určitému portu, MAC adrese a IP adrese. Největší problém je vytvoření flow záznamu pro ARP protokol. Protože každý počítač má jinou MAC adresu, musely by se vytvářet záznamy na každém přepínači, aby bylo zaručeno úspěšné doručení. Částečným řešením problému je veškerý ARP provoz poslat na všechny porty. Z hlediska bezpečnosti a vytížení sítě to není ta nejlepší volba, ale sníží se počet statických flow na přepínačích. Při testování různých flow záznamů byl vyřešen problém s HP 5130. Nakonec stačí pouze jeden flow záznam, který se přidá do tohoto zařízení a začne fungovat reaktivní vkládání flow záznamů.

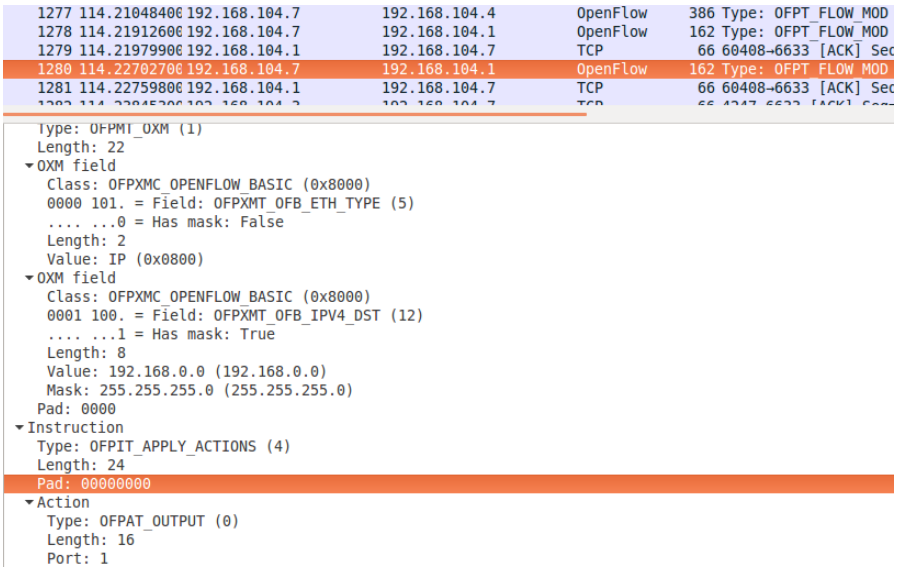
V druhé fázi po úspěšném zprovoznění reaktivního vkládání flow byly otestovány funkce firewall a ACL. Obě dvě fungovaly, tak jak jsou popsány v dokumentaci. Celý provoz byl řízen pomocí reaktivního vkládání flow záznamů. Tato metoda řízení sítě je dobrá při první spuštění kontroléru. Během pár sekund běhu kontroléru je síť plně funkční.

Při reaktivním vkládání záznamů, dochází ke značnému vytížení jednotlivých aktivních prvků. Toto vytížení je způsobené neustálým vkládáním a odebíráním flow záznamů z jejich tabulek. Nejvytíženějším prvkem celé topologie byl HP 5130, který byl neustále vytěžován na 70 – 80% své výpočetní kapacity. Ostatní aktivní prvky byly vytížené okolo 10% jejich výpočetní kapacity. Důvodem tak velkého vytížení HP zařízení mohlo být to, že sloužil jako centrální uzel, který všechny počítače připojoval do internetu. Ostatní aktivní prvky měly připojeného pouze jednoho hosta. Zatížení HP 5130 lze také zmenšit vypnutím výpisu logovacích zpráv do konzole. Tyto zprávy se nedají použít k diagnostice, protože jsou generovány ve velkém počtu a velkou rychlostí. Tyto zprávy mají pouze jedinou informační hodnotu, která ukazuje, zda je funkční reaktivní vkládání flow záznamů.

Neúměrné vytížení nebylo jenom na straně síťového prvku, ale také na straně kontroléru, který běžel ve virtualizovaném prostředí. Prvotní konfigurace se 4 GB RAM a jedním jádrem se ukázala pro reaktivní vkládání flow záznamů nedostačující. Poslední testovaná konfigurace měla 6 GB RAM a tři jádra. Tato konfigurace se ukázala jako dostačující pro testovanou síťovou topologii.

Při testování aplikace Open vSwitch, která přidává podporu OpenFlow do systému OpenWRT, nebylo možné vkládat flow záznamy do flow tabulek aktivního prvku. Tato aplikace také nepodporovala předávání neznámého paketu do kontroléru. Z toho důvodu nedocházelo k předávání LLDP zpráv zpět ke kontroléru. Jediné co tato aplikace dovolila, bylo připojení aktivního prvku ke kontroléru, kterému předala všechny potřebné informace o aktivním prvku na rozdíl od RouterOS.

Pro snížení zátěže lze využít statické flow záznamy. Při specifikaci jednotlivých flow záznamů pro všechny hosty na sítí, se snížilo i zatížení HP 5130 ze zmiňovaných 70 – 80% na 10%. Nejlepších výsledků síť dosahoval právě při kombinaci reaktivního vkládání a statických flow záznamů. Na Obr. 24 je vidět OpenFlow paket, který vkládá záznam do flow tabulky přepínače. Paket obsahuje informace, které slouží k porovnávání, v tomto případě jsou hledány všechny pakety s cílovou IP adresou z rozsahu 192.168.0.0/24. Pokud nějaký takový paket přijde do přepínače, přepínač má nastaveno předání paketu na port 1.




Obr. 24. Vložení flow záznamu

Při testování byla testována rychlost připojení k internetu. Jako referenční vzorek byla brána rychlost naměřená na počítači, který nebyl připojen do OpenFlow sítě, ale používal stejnou výchozí bránu jako počítače v sítí OpenFlow. Hodnoty se od sebe příliš nelišily, kromě TP-Linku, který byl nejpomalejší s největší odezvou. Tímto testem je dokázáno, že odesílání neznámých do kontrolérů paketů nesnižuje rychlost připojení.

| Prvek | Rychlost (download/upload/odezva) |
|---|---|
| Referenční | 95,47 Mbps / 94,60 Mbps / 9 ms |
| HP5130 Comware | 95,59 Mbps / 94,56 Mbps / 9 ms |
| HP E5406zl ProVision | 95,23 Mbps / 94,51 Mbps / 9 ms |
| Mikrotik RB2011UiAS RouterOS | 94,75 Mbps / 94,56 Mbps / 9 ms |
| TP-Link WR1043ND ver. 2 | 47,92 Mbps / 47,09 Mbps / 12 ms |

Tabulka. 1. Rychlosti naměřené na jednotlivých zařízeních

Zajímavá je reakce na změnu topologie, rychlost aplikování změn je otázkou pár sekund. Podivné chování kontrolérů se projeví při odpojení jedné z redundantních tras. Spojení mezi počítači je přibližně na 5 sekund spojení přerušeno. Tato chyba je způsobena chybou kontroléru, který neodvolá špatný flow záznam z přepínače po oznámení vypnutí portu. Změna topologie a propagace změn je nezávislá na výrobci a operačním systému přepínače.

Při testování aplikace firewall se později ukázalo, že dochází ke špatnému porovnávání cílových MAC adres. Pro porovnání dvou hodnot, které obsahují MAC adresy, bylo použito „=“. Takto docházelo k porovnání referencí datových typů a ne k porovnání obsahu proměnné. Oprava tohoto chování nebyla složitá, stačilo nahradit „=“ funkcí equals(). Tuto chybu lze najít ve verzi 1.2, která byla testovaná. Popsaná chyba byla v FirewallRule.java v metodě isSame(FirewallRule r).

# 7 Diskuze

Největším problémem Floodlight kontroléru je chybějící grafické rozhraní, které by usnadnilo správu celé sítě. Prozatím musí administrátor používat jednoduché rozhraní, které je pouze informativní. Existují dvě grafická rozhraní, která byla vytvořena pro práci s kontrolérem, ale nejsou funkční.

První grafické rozhraní umožnovalo pouze nastavování flow záznamů a jejich editaci. Tento program byl použitelný do verze 0.90. Od této verze proběhly změny v REST API kontroléru, které tento program odstavily. Otázkou je, zda je toto grafické rozhraní potřeba opravovat.

Druhé grafické rozhraní je postavené na nodejs a využívá ke svému běhu sails. Instalace prostředí pro správnou funkčnost je obtížné. Jsou vyžadované komponenty, které nejsou dostupné v balíčcích. Když už se povede sehnat a naistalovat tyto balíčky, rozhraní selže na výjimku. Při pokusech o upozornění na chybu autor nereagoval. Při založení chybového tiketu na platformě github, autor nereagoval, ale navrženou opravu vložil. Naneštěstí to není jediná chyba tohoto rozhraní.

Samotné grafické rozhraní integrované ve Floodlight kontroléru obsahuje také chyby, kvůli kterým se neaktualizuje v reálném čase a nevypisuje kompletní informace o aktivních prvcích.

Velkou nepříjemností je nefunkčnost definice virtuálních sítí. Během pokusu se nepodařilo najít správný formát příkazu, který by kontrolér přijal. Tyto sítě nelze vytvářet a není tak možnost testování moduly, který se stará o směrování paketů do správných VLAN.

Neřešeným problémem, který vznikl při testování síťové topologie, byla nefunkčnost aplikace Open vSwitch. Tato nefunkčnost nemusela být způsobena samotnou aplikací, mohla být způsobena samotným operačním systémem OpenWRT nebo aktivním prvkem.

# 8 Závěr

Floodlight kontrolér je dobrý kontrolér pro testování OpenFlow sítí. Po opravení všech chyb, které jsou zmíněny v textu výše, je kontrolér lepší než kontrolér od firmy HP. Na rozdíl od kontroléru HP, Floodlight reaktivně vkládá flow záznamy do aktivních prvků. HP vloží do jednotlivých prvků statické flow záznamy, které síť zprovozní, ale dále je vše na administrátorovi. Uživatelé bez znalostí programovacího jazyka Java, Linuxu a OpenFlow nebudou schopni Floodlight kontrolér zprovoznit, protože některé verze obsahují tak zásadní chyby, že znemožňují správné fungování komunikace zařízení s kontrolérem.

Značnou nevýhodou Floodlight kontroléru je chybějící grafické rozhraní, které by umožnilo správu celé síťové topologie. Prozatím má administrátor k dispozici informativní grafické rozhraní, které obsahuje základní informace o zařízeních a hostech. Naneštěstí i toto grafické rozhraní obsahuje chybu, která neumožňuje aktualizaci v reálném čase. Pokud je potřeba vidět aktuální data, je potřeba znovu zadat adresu grafického rozhraní kontroléru. Toto je zase lépe vyřešené u kontroléru firmy HP. Jejich grafické rozhraní bylo plně funkční a ukazovalo všechna potřebná data. Přehled síťové topologie je také lépe vyřešený, protože se nekříží jednotlivá zařízení přes sebe.

Většina času věnovaná OpenFlow, byla zaměřená na zprovoznění komunikace jednotlivých zařízení s Floodlight kontrolérem a poté na řešení jednotlivých chyb při komunikaci. Po vyřešení těchto závad již nezbyl čas na podrobné otestování všech modulů. Konkrétně modul forwarding nebyl otestován úplně. Podle dokumentace je schopný předávat pakety jako přepínač mezi jednotlivé virtuální sítě, ale to nebylo otestováno. A to zejména z toho důvodu, že se nepovedlo do kontroléru zavést virtuální sítě.

Floodlight je vhodný spíče pro nadšence, kteří chtějí otestovat OpenFlow protokol v menších až středních sítích. A to zejména proto, že po úpravách, které byly zmíněné v textu, je celá síť během pár minut plně funkční a i v omezeném grafickém rozhraní si lze prohlédnout topologii sítě.

Jediný kontrolér, který je natolik dotažený, aby běžel a byl připraven spravovat síť, je ONOS, který v původní rešerši nebyl zahrnut, protože nebyl tak velký, jako je dnes. V posledním roce se do projektu zapojily velké firmy, které projektu ONOS velice pomohly. ONOS jsem nemohl otestovat stejně, jako jsem testoval HP VAN SDN kontrolér a Floodlight, protože v době, kdy mi byl ONOS představen, již nebyly k dispozici prvky od firmy HP. I přesto tento kontrolér vykazoval nejlepší vlastnosti a funkcionalitu. ONOS má nejlepší grafické rozhraní napříč kontroléry, které byly testovány. Jediné co grafickému rozhraní chybí, je možnost nastavování jednotlivých služeb, tato nastavení se musejí provádět stále pomocí REST API nebo CLI.

# 9 Seznam literatury


[1] Jazyk P4 jako budoucnost SDN. *Root.cz* [online]. Praha: Internet Info, s.r.o., 2016 [cit. 2016-01-21]. Dostupné z: http://www.root.cz/clanky/jazyk-p4-jako-budoucnost-sdn/

[2] P4: high-level language for programming protocol-independent packet processors. *ONCR Research* [online]. Stanford: Stanford University, 2014 [cit. 2016-01-25]. Dostupné z: http://onrc.stanford.edu

[3] REXFORD, Jen a Nick MCKEOWN. Let's get started. In: *P4* [online]. p4.org: p4.org, 2015 [cit. 2016-01-25]. Dostupné z: http://p4.org/p4/lets-get-started/

[4] *The P4 Language Specification* [online]. p4.org, 2015 [cit. 2016-01-25]. Dostupné z: http://p4.org/wp-content/uploads/2015/04/p4-latest.pdf

[5] NADEAU, Thomas D a Kenneth GRAY. *SDN: software defined networks*. 1st ed. Sebastopol, CA: O'Reilly Media, 2013, xxvii, 352 s. ISBN 978-1-4493-4230-2.

[6] *OpenFlow Switch Specification* [online]. [cit. 2016-02-03]. Dostupné z: http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf

[7] ROUSE, Margaret. What is flooding?: Flooding. In: *TechTarget* [online]. 2007 [cit. 2016-02-09]. Dostupné z: http://searchnetworking.techtarget.com/definition/flooding

[8] Mission|ON.Lab. *ON.Lab* [online]. Menlo Park, California [cit. 2016-02-14]. Dostupné z: http://onlab.us/mission/

[9] Getting Started. *RYU the Network Operating* [online]. [cit. 2016-02-14]. Dostupné z: http://ryu.readthedocs.org/en/latest/getting_started.html#what-s-ryu

[10] Architecture. *Floodlight Documentation* [online]. [cit. 2016-02-15]. Dostupné z: https://Floodlight.atlassian.net/wiki/display/Floodlightcontroller/Architecture

[11] MACEK, Petr. Zjišťování síťového okolí protokolem LLDP s využitím PHP. In: *Samuraj-CZ* [online]. 2013 [cit. 2016-02-15]. Dostupné z: http://www.samuraj-cz.com/clanek/zjistovani-sitoveho-okoli-protokolem-lldp-s-vyuzitim-php/

[12] Floodlight. *Project Floodlight* [online]. [cit. 2016-02-16]. Dostupné z: http://www.projectfloodlight.org/floodlight/

[13] ACL (Access Control List) REST API. *Floodlight Documentation* [online]. Sydney, 2015 [cit. 2016-02-20]. Dostupné z: https://floodlight.atlassian.net/wiki/display/floodlightcontroller/ACL+%28Access+Control+List%29+REST+API

[14] The Controller. *Floodlight Documentation* [online]. Sydney, 2015 [cit. 2016-02-20]. Dostupné z: https://floodlight.atlassian.net/wiki/display/floodlightcontroller/The+Controller

[15] Static Flow Pusher API (New). *Floodlight Documentation* [online]. Sydney, 2016 [cit. 2016-02-23]. Dostupné z: https://floodlight.atlassian.net/wiki/pages/viewpage.action?pageId=1343518

[16] HP VAN SDN Controller Software. *HP* [online]. Hewlett-Packard Development Company, L.P. The information cont, 2015 [cit. 2016-03-05]. Dostupné z: h17007.www1.hp.com/docs/networking/solutions/sdn/4AA4-8807ENW.PDF

[17] OpenFlow® Conformance Certification. *Open Networking Foundation* [online]. [cit. 2016-03-09]. Dostupné z: https://www.opennetworking.org/certification/product

[18] ONOS Tutorial. SDN Hub [online]. [cit. 2016-04-09]. Dostupné z: http://sdnhub.org/tutorials/onos/

[19] *ONOS* [online]. [cit. 2016-04-11]. Dostupné z: http://onosproject.org/

# 10 Seznam obrázků

# 11 Přílohy

## Příloha 1. Nastavení síťové konfigurace openWRT

```
config switch
        option name 'switch0'
        option reset '1'
        option enable_vlan '1'
        option enable_learning '0
config switch_vlan
        option device 'switch0'
        option vlan '1'
        option ports '0t 1'
        option vid '1'
config switch_vlan
        option device 'switch0'
         option vlan '3'
         option ports '0t 4'
         option vid '3'
config switch_vlan
        option device 'switch0'
        option vlan '4'
        option ports '0t 3'
        option vid '4'
config switch_vlan
        option device 'switch0'
        option vlan '2'
        option ports '5 6'
        option vid '2'
```

```
config switch_vlan
        option device 'switch0'
        option vlan '5'
        option ports '0t 2'
        option vid '5'
config interface 'loopback'
        option ifname 'lo'
        option proto 'static'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'
config interface
        option ifname 'eth1.1'
        option proto 'static'
config interface
        option ifname 'eth1.3'
        option proto 'static'
config interface 'lan3'
        option ifname 'eth1.4'
        option proto 'static'
config interface 'lan4'
        option ifname 'eth1.5'
        option proto 'static'
config interface 'wan'
        option ifname 'eth0'
        option type 'bridge'
        option proto 'dhcp'
```

## Příloha 2. Nastavení firewallu

**config defaults**

option syn_flood 1

option input ACCEPT

option output ACCEPT

option forward ACCEPT

## Příloha 3. Instalace HP VAN SDN kontrolér

HP VAN SDN kontrolér vyžaduje k instalaci Ubuntu, je jedno jestli je použita desktopová nebo serverová verze. Je však doporučeno použít serverovou. Instalovaný bude HP VAN SDN kontrolér verze 2.7.10. HP vydalo instalační manuál, který dobré si před instalací přečíst. Samotný kontrolér je možné stáhnout ze stránky www.hp.com/networking/support, kde je nutné zadat kód J9863AAE**.**

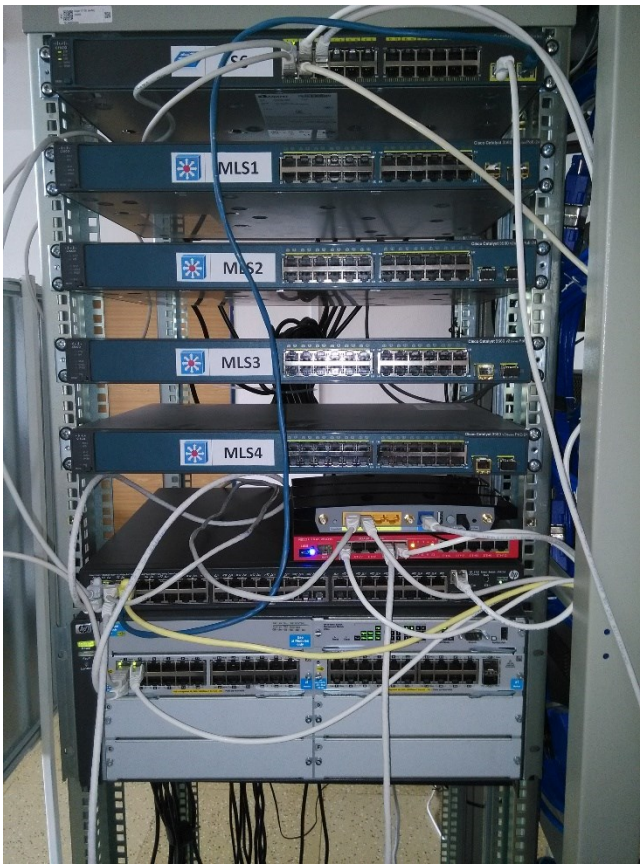
| **Postup instalace HP VAN SDN kontroléru** |
| --- |
| **sudo apt-get update** |
| **sudo apt-get install** python-software-properties |
| **sudo apt-get install** ubuntu-cloud-keyring |
| **sudo add-apt-repository** cloud-archive:icehouse |
| //Zmáčknout [Enter] |
| **sudo apt-get update** |
| **sudo apt-get install** keystone |
| //pokud není HW v doporučené konfiguraci, je potřeba |
| //vytvořit soubor override.txt |
| **touch** /tmp/override.txt |
| **sudo dpkg** --unpack hp-sdn-ctl_2.7.10.0289_amd64.deb |
| **sudo apt-get install** -f |

**Pokračování instalace**

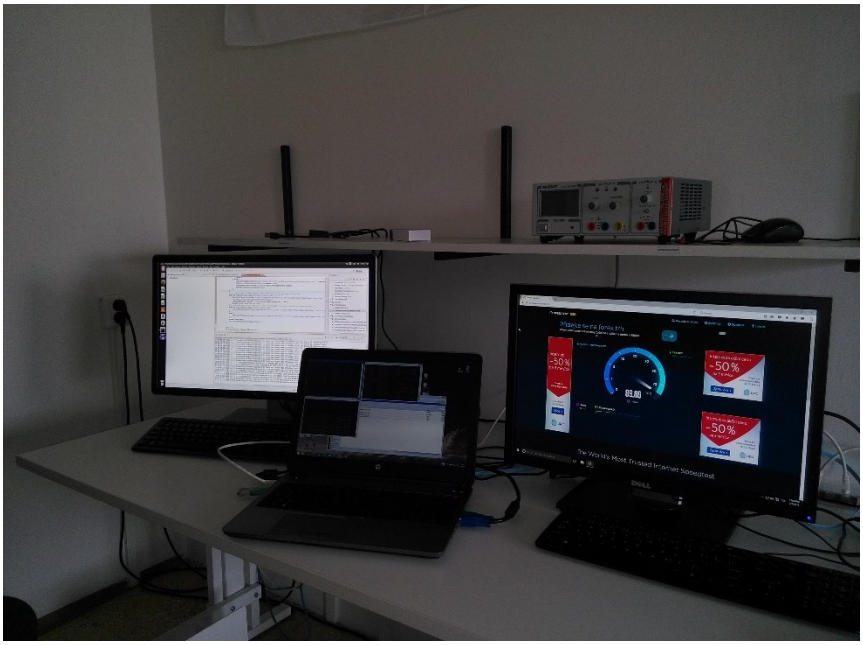
```
//Vytvoření účtu pro přihlášení
sudo nano /etc/keystone/keystone.conf
//pod sekcí [token] je nutné přidat řádek
provider=keystone.token.providers.uuid.Provider
sudo service keystone restart
export OS_SERVICE_TOKEN=ADMIN
export OS_SERVICE_ENDPOINT=http://127.0.0.1:35357/v2.0
keystone tenant-create --name sdn
keystone user-create --name admin --pass heslo1 --tenant sdn
keystone role-create --name sdn-admin
keystone role-create --name sdn-user
keystone user-role-add --user admin --role sdn-admin --tenant
sdn
keystone user-role-add --user admin --role sdn-user --tenant sdn
```

Po provedení všech instalačních kroků je přístupné grafické rozhraní na adrese https://<IP_adresa_kontroléru>:8443/sdn/ui. Na této adrese je přihlašovací stránka. Pro přihlášení je potřeba použít přihlašovací údaje nastavené v keystone v tomto případě je „User Name“ admin a „Password“ heslo1.

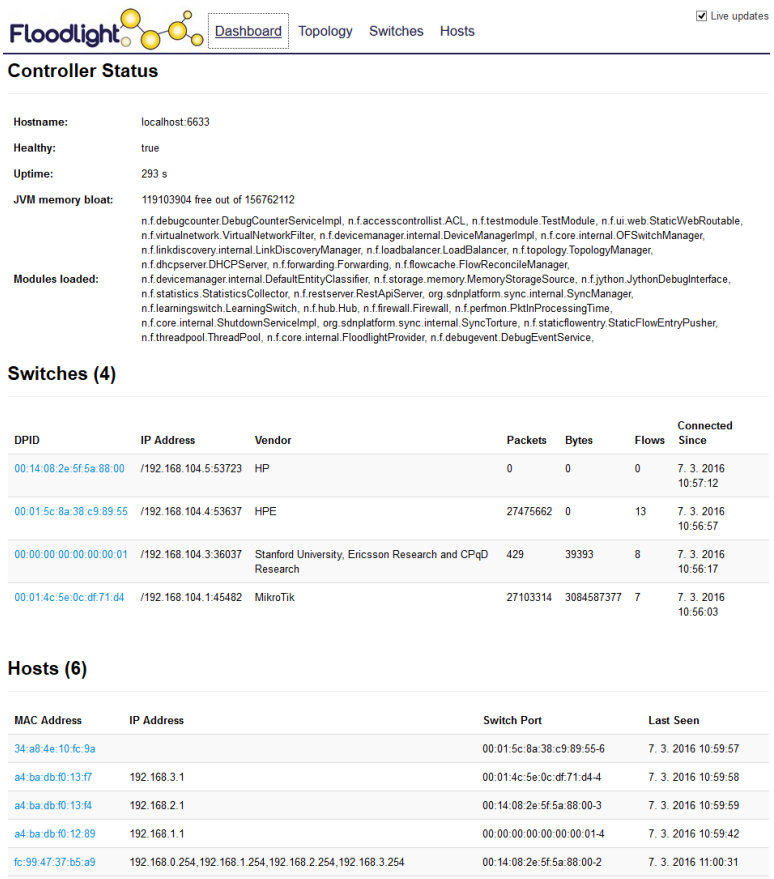
## Příloha 4. Fyzické zapojení topologie



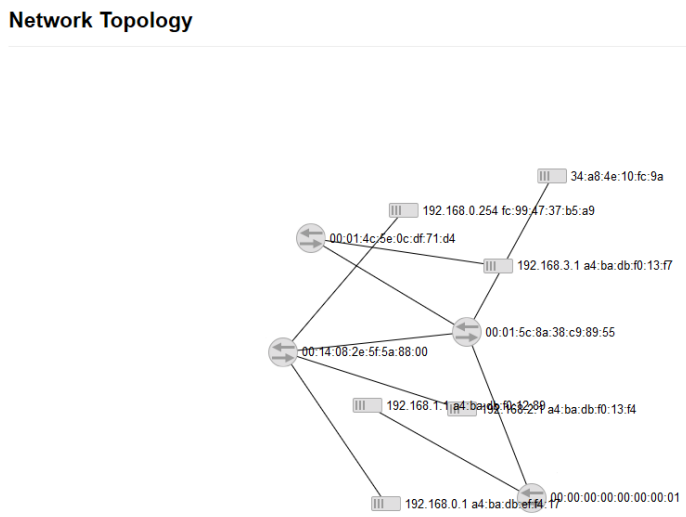

## Příloha 5. Kontrolér a fyzická stanice



# Příloha 6. Floodlight Dashboard




# Příloha 7. Floodlight topology



# Příloha 8. Konfigurační příkazy Firewallu Floodlight

**Status firewallu**

```
curl http://<kontrolér_ip>:8080/wm/firewall/module/status/json
```

**Zapnutí firewallu**

```
curl -X PUT -d ""
http://<kontrolér_ip>:8080/wm/firewall/module/enable/json
```

**Vypnutí firewallu**

```
curl -X PUT -d ""
http://<kontrolér_ip>:8080/wm/firewall/module/disable/json
```

**Získání / Nastavení subnetu firewallu**

```
curl http://<kontrolér_ip>:8080/wm/firewall/module/subnet-
     mask/json

curl -X POST -d "{\"subnet-mask\":\"X.X.X.X"}"
http://<kontrolér_ip>:8080/wm/firewall/module/subnet-mask/json
```

**Získání / Nastavení / Smazání pravidla firewallu**

**curl Chyba! Odkaz není platný.**

```
curl -X POST -d "{\"src-ip\": \"10.0.0.4/32\",
                  \"dst-ip\": \"10.0.0.10/32\",
                  \"dl- type\":\"ICMP\"}"
```

**Chyba! Odkaz není platný.**

**curl** -X DELETE -d "{\"rule-id\":\"id\"}" **Chyba! Odkaz není platný.**

| Port | Zdr. MAC | Cíl. MAC | Eth. Typ | VLAN ID | Zdr. IP | Cíl. IP | IP Prot. | Zdr. port | Cíl. port | Akce |
|---|---|---|---|---|---|---|---|---|---|---|
| * | * | * | * | * | 10.0.0.4/32 | 10.0.0.10/32 | ICMP | * | * | předat |

Obr. 8.1 Příklad porovnání hlavičky

## Příloha 9. Konfigurační příkazy filtru VLAN

**Nastavení / Smazání filtru sítě**

```
curl -X PUT/POST -d "{ "network":
                                { "gateway": "10.0.0.7",
                                 "name": "virtualNetwork1" }
                        }"
http://<kontrolér_ip>:8080/networkService/v1.1/tenants/default/networks/NetworkId1

curl -X DELETE -d ""
http://<kontrolér_ip>:8080/networkService/v1.1/tenants/default/networks/{unikatní_jméno_sítě}
```

**Přidání / Odebrání hosta**

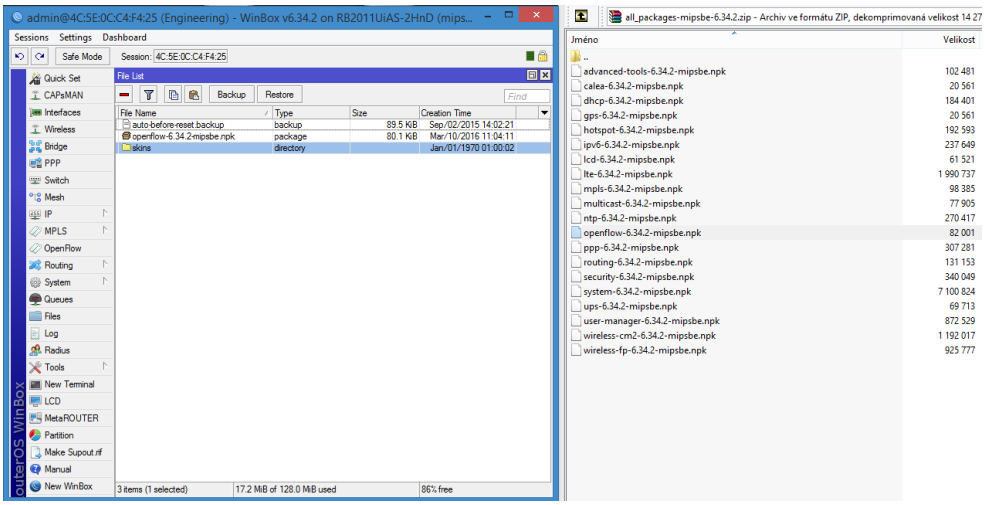
```
curl -X PUT -d "{"attachment":
                                {"id":"NetworkId1",
                                "mac":"00:00:00:00:00:08"}
                    }"
http://localhost:8080/networkService/v1.1/tenants/{tenant}/networks//{unikatní_jméno_sítě}/{port}/port1/attachment

Za položku {tenant} se vloží default.
```

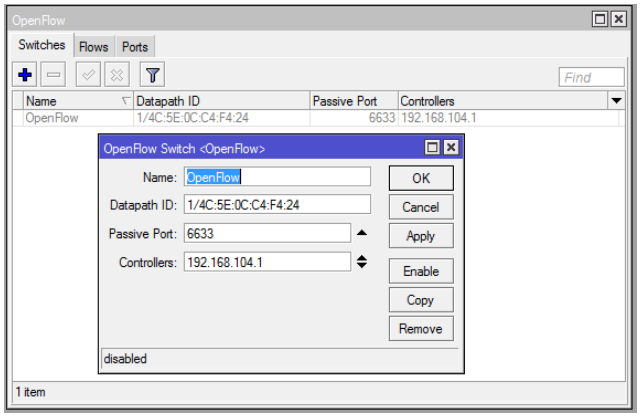
**Získání všech sítí a jejich bran**

```
curl http://<kontrolér_ip>:8080/
     networkService/v1.1/tenants/{tenant}/networks
```

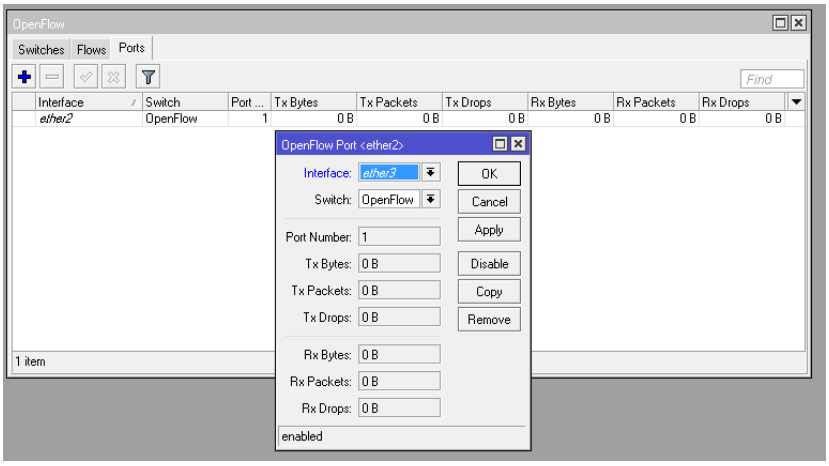
## Příloha 10. Nastavení zařízení Mikrotik



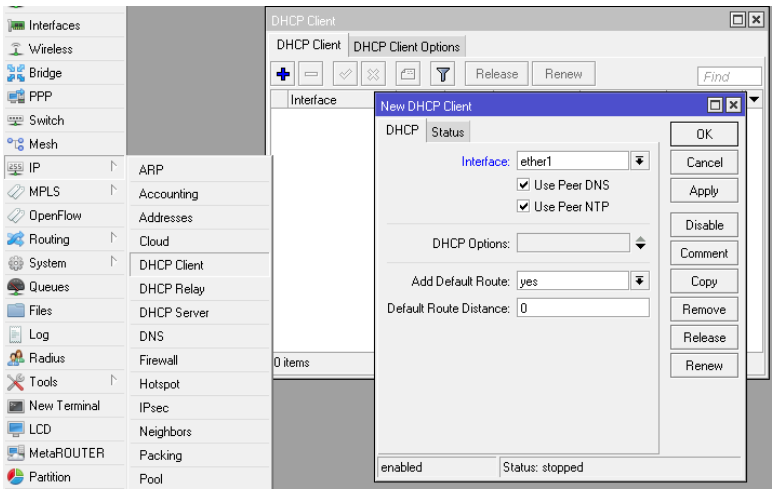

Obr. 10.1 Instalace OpenFlow rozšíření




Obr. 10.2 Nastavení OpenFlow přepínače




Obr. 10.3 Přidání portu do OpenFlow přepínače




Obr. 10.4 Nastavení DHCP klienta

## Příloha 11. Instalace OpenFlow do TP-Link směrovače

**Kompilace OpenWRT s OpenFlow 1.0 nebo 1.3**

```
mkdir openwrt
cd openwrt
git clone git://git.openwrt.org/15.05/openwrt.git
cd openwrt
./scripts/feed update -a
./scripts/feed install –a
make menuconfig
Target system -> Atheros AR7xxx/AR9xxxx
Target profile -> TP-LINK TL-WR1043ND
make prereq
cd ..
//stažení a linkování OpenFlow 1.0
git clone git://gitosis.stanford.edu/openflow-openwrt
cd openflow-openwrt
git checkout -b openflow-1.0/tplink origin/openflow-1.0/tplink
cd ~/openwrt/openwrt/package/
ln -s ~/openwrt/openflow-openwrt/openflow-1.0/
cd ~/openwrt/openwrt/
ln -s ~/openwrt/openflow-openwrt/openflow-1.0/files
//stažení a linkování OpenFlow 1.3
git clone https://github.com/CPqD/openflow-openwrt.git
cd ~/openwrt/openwrt/package/
ln -s ~/openwrt/openflow-openwrt/openflow-1.3/
cd ~/openwrt/openwrt/
ln -s ~/openwrt/openflow-openwrt/openflow-1.3/files
cd ~/openwrt
make menuconfig
//Je nutné vybrat tyto balíky pod záložkou Network
<*> openflow
<*> kmod-tun
```

```
//Je nutné vybrat tyto balíky pod záložkou

//Kernel Modules -> Network support

<*> kmod-sched-core

<*> kmod-sched

save and exit

make kernel_menuconfig

//Je nutné vybrat tyto balíky pod záložkou
//Networking support -> Networking options

<*> Hierarchical Token Bucket (HTB)

//Kompilace zdrojových kódů

make V=s //(V=s je dobrý pro debug)
```