VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

VIRTUALIZACE SÍTÍ - SOFTWAROVĚ DEFINOVANÉ SÍTĚ

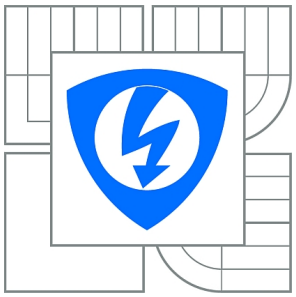
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE MAREK VESELÝ
AUTHOR

BRNO 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

VIRTUALIZACE SÍTÍ - SOFTWAROVĚ DEFINOVANÉ SÍTĚ

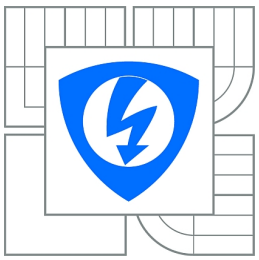
NETWORK VIRTUALIZATION - SOFTWARE DEFINED NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MAREK VESELÝ
AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. VÍT NOVOTNÝ, Ph.D.
SUPERVISOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Marek Veselý	***ID:***	134658
Ročník:	3	***Akademický rok:***	2012/2013

NÁZEV TÉMATU:

Virtualizace sítí - softwarově definované sítě

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s problematikou virtualizace v oblasti informačních technologií a datových sítí. Stručně rozeberte problematiku virtualizace počítačů a oblast virtualizace sítí LAN (VLAN). Podrobně se zaměřte na principy softwarově definovaných sítí (SDN), proberte možnosti, existující technologie a zhodnoťte výhody i slabiny tohoto řešení datových sítí. Zjistěte dostupnost zařízení s podporou SDN a rozeberte možnosti využití SDN jako infrastruktury pro flexibilní návrh pracovišť laboratorních úloh pro předmět BARS.

DOPORUČENÁ LITERATURA:

[1] Rich Seifert, James Edwards The All-New Switch Book: The Complete Guide to LAN Switching Technology. John Wiley & Sons, ISBN-13: 978-0470287156
[2] OpenNetworking.org: Software defined networking: The New Norm for Networks. https://www.opennetworking.org/images/stories/downloads/white-papers/wp-sdn-newnorm.pdf

Termín zadání:	11.2.2013	***Termín odevzdání:***	5.6.2013

Vedoucí práce: doc. Ing. Vít Novotný, Ph.D.
Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Cíl práce je zaměření se na nový trend síťové architektury a to softwarově definované sítě. Popisuje funkci kontroleru, jako hlavního zařízení, které ovládá celou síť. Dále je zde uvedena komunikace mezi přepínačem a kontrolerem, kterou řídí protokol OpenFlow.
První část práce se zabývá stručně principem VLAN sítí. Dále se více orientuje softwarově definovaným sítím, kde je probrán princip této sítě. Podrobně je zde rozebrán protokol OpenFlow, jako jeho funkce, výhody a nevýhody sítě a možnost využití této sítě na akademické půdě.

KLÍČOVÁ SLOVA

Softwarově definované sítě, protokol, OpenFlow, kontroler, přepínač, otevřená síť, VLAN, virtualizace, Open Network Foundation, Beacon

ABSTRACT

The aim of work is a new trend in network architecture, called Software defined networks. Describes function of controller as the main device, which controlling the whole network. Further the communication between switch and controller is here, which controlling protocol OpenFlow.
First chapter gently deal with principe of the VLAN networking. Further it is much more oriented with software defined networks, where the main principe is described. Protocol OpenFlow is described in deatail, as his functions are discussed too. There are also advantages and disadvantages and possibility to use this network in university campus.

KEYWORDS

Software defined networking, protocol, OpenFlow, controller, switch, open network, VLAN, virtualization, Open Network Foundation, Beacon

VESELÝ, Marek *Virtualizace sítí – softwarově definované sítě*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 45 s. Vedoucí práce byl doc. Ing. Vít Novotný, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Virtualizace sítí – softwarově definované sítě“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno ....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Vítu Novotnému, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Nové trendy, jako jsou mobilní zařízení, virtualizace serverů a nástup cloudových služeb, s sebou přináší stále větší nároky na síťovou architekturu. Současná architektura už začíná projevovat slabiny a nemožnost dalšího zrychlení sítě. Ocitáme se tedy na mrtvém bodě, kde rozšiřující služby, jako Cloud a služby v reálném čase, stále více zatěžují síť a znemožňují přístup službám s nízkou prioritou, jako protokolu HTTP. Tuto problematiku by měla mimo jiné řešit nová síťová architektura Softwarově definovaných sítí.

Tato práce popisuje princip nové síťové architektury a protokolu OpenFlow, který slouží pro komunikaci mezi kontrolerem s dalšími prvky v síti, přepínač, či směrovač. Kontroler je základní řídicí jednotka celé této sítě. Je zde rozebrána většina funkcí protokolu OpenFlow, jako tabulky toků sloužící k přepínání a směrování v síti. Tento protokol je zde i řádně analyzován a rozebrán, ovšem je stále ve vývoji, tudíž se zde nové verze s vylepšeními objevují každou chvíli.

Okrajově se tato práce zabírá i virtuálními sítěmi, kde je jen nastíněn jejich princip a popsána základní činnost.

1 VIRTUALIZACE LOKÁLNÍCH SÍTÍ

1.1 Úvod

Lokální síť (LAN[1]) umožňuje přímo Linkové vrstvě komunikaci mezi uzly patřícími do jedné lokální sítě. Typické střední až velké organizace mohou mít mnoho sítí LAN pro podporu velkého počtu uživatelů a síťových aplikací. Každý uživatel je připojen k jedné fyzické síti, to je ale omezeno v sítích LAN zařízením, např. počet portů na hubu. Fyzické připojení je zde stejné jako logické, což vyžaduje při logické změně spojení i změnu fyzického zapojení.

Virtuální síť LAN (VLAN[2]) je technologie, která umožňuje oddělit logické spojení od fyzického připojení. Uživatelé jsou připojeni přes fyzické kabely k fyzickému zařízení (nejčastěji přepínači), což znamená, že stanice a aplikace mohou přímo komunikovat na společné LAN. Síť je virtuální v tom, že stanice a aplikace se mohou chovat, jako by byly logicky odděleny, ale ve skutečnosti jsou připojeny k jediné fyzické LAN. K dosažení této flexibility je nutno použít přepínače místo opakovačů. Kromě toho musí i přepínače podporovat VLAN technologii. [1]

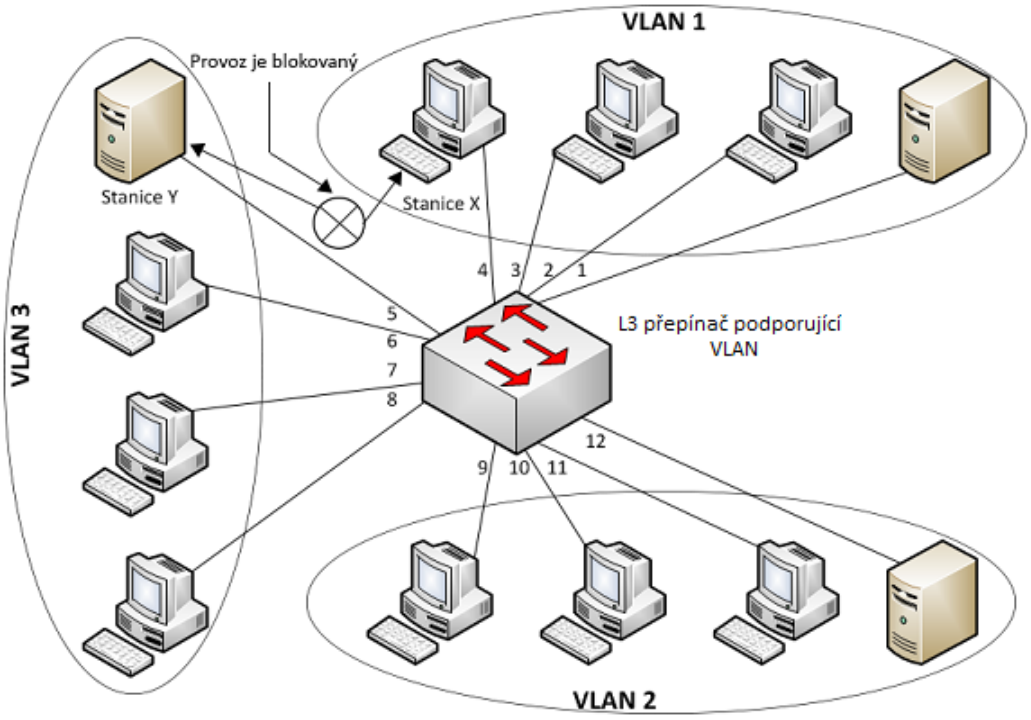
Klasické LAN sítě nemají tak rozsáhlé zabezpečení, informace se mohou dostat i k jiným uživatelům, než jsou určeny. U virtuálních sítí VLAN jsou informace dostupné jen pro členy dané VLAN sítě a jakékoliv narušení do sítě je lokalizováno, viz obr. 1.1, dá se zde tudíž nastavit lepší zabezpečení sítě. To ale nebrání všem útokům na síť. VLAN jen zkomplikuje práci útočníkovi a zabrání vniknutí jen těm útočníkům, kteří nejsou ochotni vynaložit dostatek úsilí nezbytného k prolomení ochrany sítě. [1]

1.2 Mobilita uživatelů

U virtuální sítě není mobilita uživatelů omezena jen na určité místo, patro budovy, ale mohou uživatelé přistupovat k síti i přes různé podlaží budovy, nebo na geograficky rozptýlených místech, pomocí VPN přístupu, což ale značně může omezit rychlost v důsledku pomalejší rychlosti WAN[3] připojení. Je zde tedy zvolen kompromis mezi výkonem, náklady (na přepínače) a flexibilitou.

Tento způsob mobility ovšem vytváří větší výkonové požadavky na VLAN přepínače. Přepínač musí určit VLAN členství na základě označení daného rámce, aby zjistil, kam mají rámce dorazit. [1]

[1] Local Area Network
[2] Virtual Local Area Network
[3] Wide Area Network



Obr. 1.1: Příklad VLAN konfigurace.

1.3 Základ virtuálních lokálních sítí

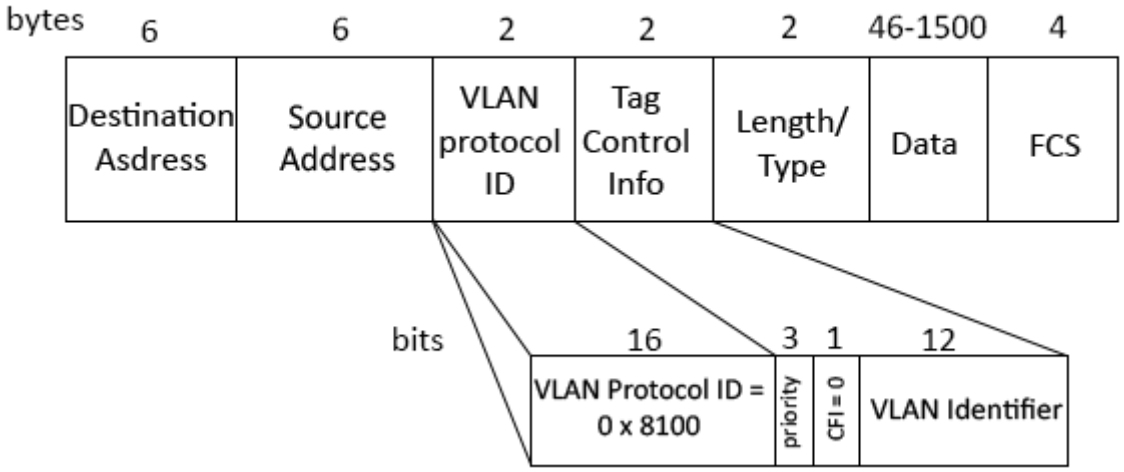
Je to logická skupina objektů v různých sítích. Vše potřebné pro toto logické spojení je obsaženo v rámcích, posílaných v síti. Přepínače se podívají do rámce a zjistí, ke které VLAN patří. Nemusí znát důvod pravidel, jen je uplatní a rozdělí rámce do jednotlivých VLAN. [1]

1.3.1 Označování rámců v síti

V tradičním prostředí bez VLAN technologie je toto triviální, každý rámec zde patří jen do jedné vnitřní sítě. U virtuální technologie, kde existuje více VLAN sítí, není jasné, že rámec, který dorazí, patří do jedné určité VLAN sítě. Existují proto dvě metody na identifikaci VLAN:

- **Analyzování rámce a uplatnění pravidla** – rámec je vždy zkontrolován a je zde uplatněna kompletní sada VLAN asociačních pravidel pro síť, typicky se toto provádí prostřednictvím přepínače, označované jako implicitní značkování.
- **VLAN identifikátor v samotném rámci** – označované jako explicitní značkování, cesta sítí se určí podle předem definovaného označení (tagu) v rámci, označené na obr. 1.2. [1]

Výhody a nevýhody značkování:



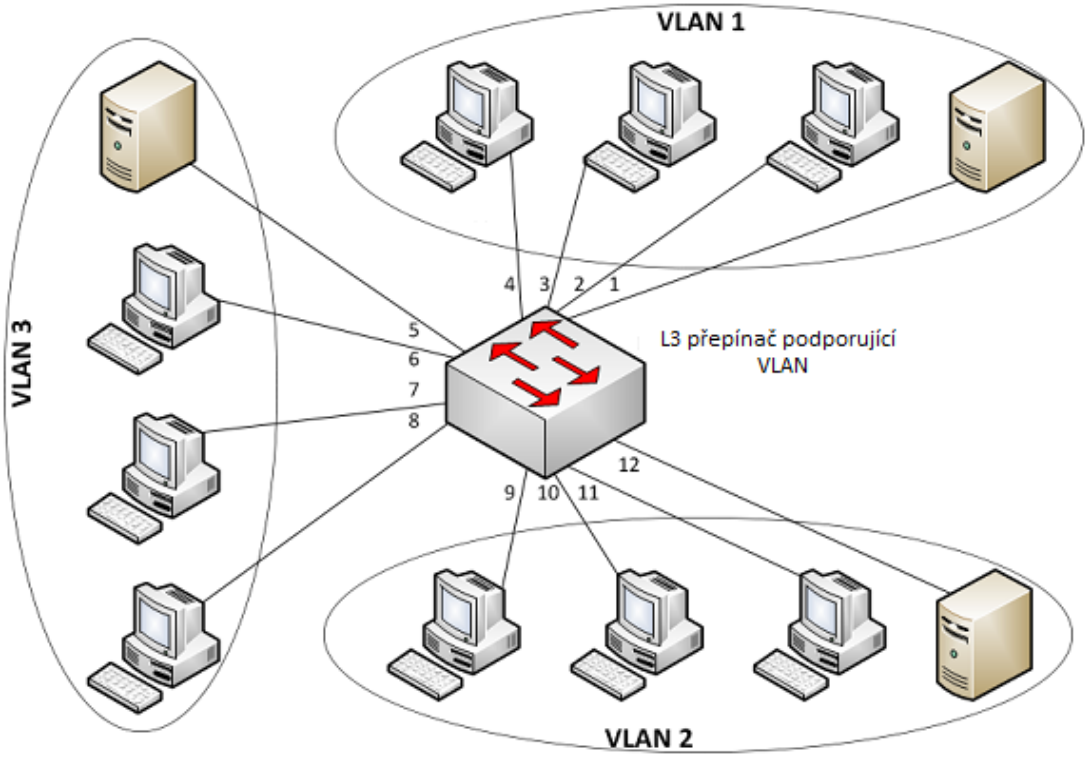
Obr. 1.2: Označený VLAN Ethernetový rámec. [1]

- **Výhody**: VLAN asociační pravidla musí být aplikovaná jen jednou. Jenom krajní přepínače musí znát asociační pravidla. Hlavní přepínač může dostat více potřebného výkonu díky explicitnímu VLAN identifikátoru.
- **Nevýhody**: Značky lze interpretovat pouze na zařízení podporující VLAN. Vložení nebo odstranění značky přepočtem FCS, případně ohrožení integrity rámce. Vložení značky může zvýšit délku rámce nad maximální povolenou velikost. [1]

1.3.2 Mapování rámců v síti

V případě označeného rámce je mapování jednoduché, značka (tag) obsahuje identifikátor VLAN rámce, tudíž je jednoduché ho přiřadit k určité VLAN. Mapování je založené pomocí:

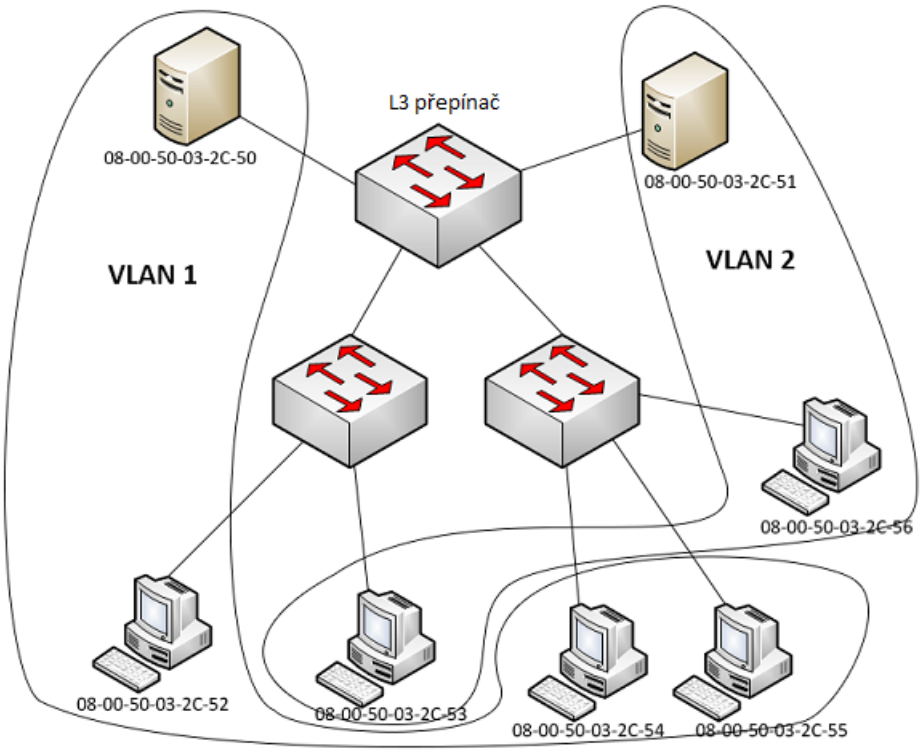
- **Portů** – nejjednodušší implicitní pravidlo mapování. Rámec je zde přiřazen k VLAN podle portu přepínače, viz obr. 1.3. VLAN 1 patří portům 1-4, VLAN 2 portům 9-12, ...
- **MAC adres** – složitější pravidlo, kde rámec je přiřazen k VLAN pomocí své MAC adresy, viz obr. 1.4. Má to značné výhody při přesunutí počítače k jinému přepínači, nebo do jiného portu v rámci stejného přepínače, zařízení pak dále patří do stejné VLAN sítě.
- **Protokolů** – pracuje na 3. vrstvě, kde přepínač přidělí rámec podle protokolu přenášeného paketu.
- **IP podsíťování** – založené na 3. vrstvě, kdy přepínač si v 3. vrstvě ISO/OSI modelu přečte v záhlaví IP adresu a přiřadí jednotlivou VLAN, znázorněno na



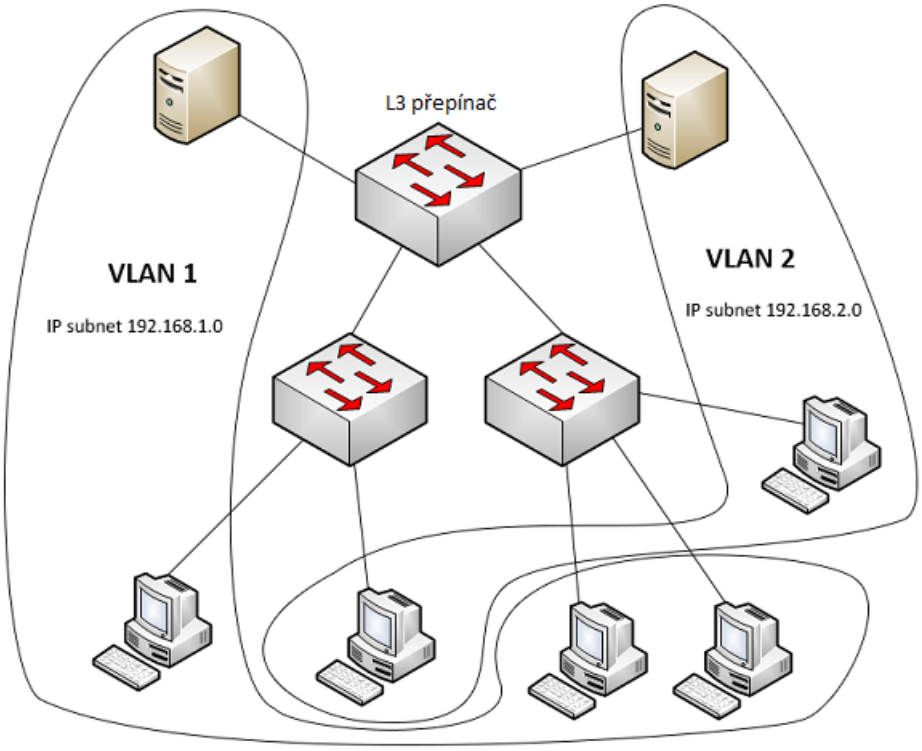
Obr. 1.3: VLAN mapování podle portů.

obr. 1.5. Přitom přepínač stále pracuje na 2. vrstvě, neprovádí žádné směrování.

- **Aplikací** – vyústění mapování založeného na protokolech. Určí VLAN podle vyšších vrstev aplikačních a určí VLAN podle určité aplikace. Slouží hlavně pro poskytování audio nebo videokonference. [1]



Obr. 1.4: VLAN mapování podle MAC adres.



Obr. 1.5: VLAN mapování podle IP podsíťování.

2 SOFTWAROVĚ DEFINOVANÉ SÍTĚ

2.1 Úvod

Tradiční síťové architektury nejsou dostatečně pružné, z hlediska své statické povahy, ke splnění požadavků nynějších firem a koncových uživatelů. Přeměnu síťové architektury skýtají softwarově definované sítě (SDN[1]). [2]

V SDN architektuře jsou oddělené řídicí a datové úrovně, síťová inteligence a stav sítě jsou logicky centralizované. Vlastní síťová infrastruktura je obsluhována aplikacemi, díky kterým firmy získají nebývalou programovatelnost, automatizaci a řízení nad sítí, které jim umožní budovat vysoce škálovatelné a flexibilní sítě, které se snadno přizpůsobí měnícím se potřebám firem a uživatelů. [2]

Softwarově definovaná síť má řidicí logiku sítě oddělenou od hardwaru síťového provozu. Přesouvá ovládání logiky sítě do programovatelného softwaru. Jedním z klíčových prvků je schopnost navrhnout a uvažovat o síťovém kontrolním plánu jako o centrálně řízené aplikaci pracující na celosvětovém síťovém pohledu (GNV – Global network view) jako jeho vstup. [3]

Komunikaci mezi řídicí a datovou úrovní podporovaných síťových zařízení strukturuje, v tomto případě, protokol OpenFlow, který přináší podstatné výhody pro firmy, zahrnující:

- Centralizovaná správa a řízení síťových zařízení od různých výrobců.
- Lepší automatizace a řízení pomocí společného API[2].
- Rychlé inovace a schopnost realizovat nové síťové funkce a služby bez nutnosti konfigurace u jednotlivých zařízení nebo nutnosti čekání na aktualizace od různých výrobců.
- Programovatelnost od všech stran i koncových uživatelů pomocí společného programovacího prostředí.
- Zvýšení spolehlivosti sítě a bezpečnosti v důsledku centralizovaného a automatizovaného řízení síťových prostředků, jednotná politika v nasazení a méně chyb při konfiguraci.
- Lepší zkušenosti koncových uživatelů s aplikacemi využívajícími centralizované sítě, které bezproblémově přizpůsobí chování sítě potřebám uživatelů. [2]

S SDN se dnešní statické sítě mohou vyvinout do rozšířitelné platformy pro poskytování služeb, které mohou rychle reagovat na měnící se potřeby obchodní, trhu, či potřeby koncových uživatelů. [2]

[1] SDN – Software defined networking
[2] Application Programming Interface

2.2 Nezbytnost nového síťového modelu

Velké rozšíření chytrých mobilních zařízení, virtualizace serverů a nástup cloudových služeb patří mezi trendy ovlivňující síťový průmysl k přehodnocení tradičních síťových architektur. Konvenční sítě jsou hierarchické, postavené z ethernetových přepínačů uspořádaných do stromové struktury. Toto ale mělo smysl, když komunikace klient – server byla dominantní. [2]

Stará, plně fyzicky centralizovaná kontrola je nedostatečná, protože omezuje odezvu, spolehlivost a škálovatelnost. Takováto architektura je nevhodná pro potřeby dnešních firemních datových center, areálů, ...

Potřebu nové sítě vyvolává:

- **Změna provozního modelu** – v rámci podnikového datového centra se provozní model podstatně změnil. Uživatelé chtějí mít přístup kdykoliv a kdekoliv k firemnímu obsahu a k aplikacím z jakéhokoliv zařízení. Toto přivádí správce k úvaze o modelu, který by mohl obsahovat privátní cloud, veřejný cloud, nebo nějaký mix, který by řešil komunikaci v celé síti.
- **Možnost vlastního přizpůsobení** – uživatelé stále častěji používají chytré telefony, tablety a notebooky pro přístup k firemní síti. Správce musí těmto zařízením umožnit přístup k síti a zároveň chránit soukromá i firemní data.
- **Vzestup cloudových služeb** – Podniky vřele přijaly cloudové služby, což vede k nebývalému nárůstu těchto služeb a nynější síť na ně nestačí kvůli jejich velkému zatížení statické sítě.
- **Větší objemy přenášených dat** – manipulace s těmito daty vyžaduje velké paralelní zpracování na tisíce serverů, u všech je potřeba přímé spojení. Vzestup obrovských (až gigabitových) datových souborů podceňuje konstantní poptávku po dostatečné kapacitě sítě v datových centrech. Provozovatelé sítí čelí novému úkolu, a to velkému růstu objemu přenášených dat, dříve nepředstavitelné velikosti. [2]

Tradiční síťová architektura nebyla navržena tak, aby splňovala požadavky dnešních uživatelů a firem, tudíž jsou síťový designéři omezeni současnými sítěmi:

- **Složitost**: Síťová technologie se doposud skládala z velké části z oddělených sad protokolů sloužících k připojení k počítači spolehlivě přes velké vzdálenosti, rychlosti připojení a topologie. V průběhu několika desetiletí průmysl vyvinul síťové protokoly, které poskytují vyšší výkon a spolehlivost, širší konektivitu a přísnější bezpečnostní opatření. Protokoly mají tendenci být definovány v izolaci s každým řešením specifického problému. To má za následek jedno z hlavních omezení sítě – složitost. Např. přidávání nebo přesunutí jakéhokoliv zařízení se musí dotknout více přepínačů, směrovačů, firewallů, atd. aktualizace ACL, VLAN, kvalita služeb (QoS) a další mechanismy, použí-

vající zařízení na úrovni nástroje pro správu, založené na protokolech. Síťová topologie, odlišnost výrobců a verze softwarů musí být vzata do úvahy. Vzhledem k této složitosti jsou dnešní sítě relativně statické. Statická povaha sítí je v ostrém kontrastu k dynamické povaze dnešního serverového prostředí, kde virtualizace serverů má značně zvýšit počet počítačů, využívající dané servery, a zásadně změnit předpoklady o fyzickém umístění hostů. Dříve aplikace byly na jednom serveru a umožňovaly provoz s vybranými klienty. Dnes jsou aplikace distribuovány přes více virtuálních strojů (VM), které si vyměňují jednotlivý tok navzájem. [2]

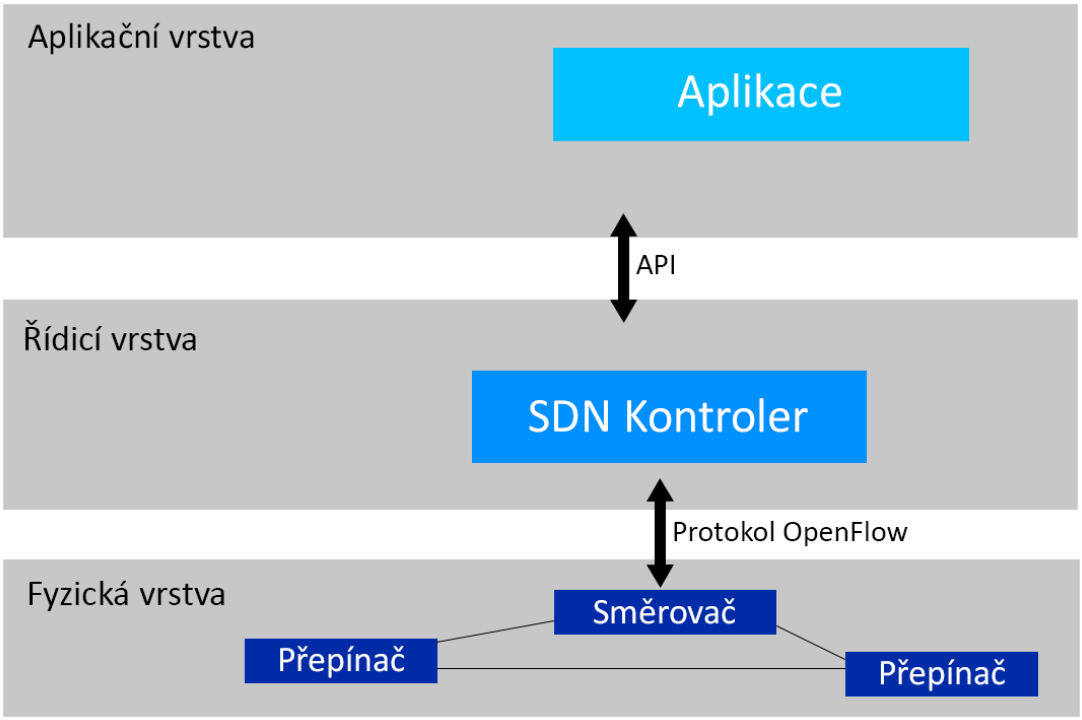
- **Nejednotná politika**: K implementaci celé síťové politiky musíte nastavit tisíce zařízení a mechanismů. Například, jednou za čas je potřeba vytvořit nový virtuální server, a to může trvat hodiny i dny k překonfigurování ACL záznamů v rámci celé sítě. V dnešních sítích je velmi obtížné pro správce kontrolovat přístup, zabezpečení, QoS a další stále mobilnější uživatele, kteří opustí podnik a jsou zranitelní vůči bezpečnosti. [2]
- **Problém v rozšíření sítě**: Nároky na datová centra rychle rostou, takže se síť také rozrůstá, ale stává se mnohem složitější s přidáváním nových zařízení. [2]
- **Závislost na výrobci**: Nedostatek standardního otevřeného rozhraní omezuje schopnost provozovatelů udělat síť podle svého individuálního prostředí. [2]

2.3 Základy softwarově definovaných sítí

SDN je nově vznikající architektura, kde je řízení sítě odděleno od přeposílání (forwarding) a je přímo programovatelné. Toto přemístění řízení, dříve pevně vázané v jednotlivých síťových zařízeních, do dostupných výpočetních zařízení umožňuje základní infrastruktuře řídit sítě jako logické virtuální jednotky. Princip je naznačen na obr. 2.1. [2]

Síť je logicky centralizovaná díky softwarovému použití SDN kontrolerů, které řídí globální pohled nad sítí. V důsledku toho se síť chová jako jeden logický přepínač. S SDN firmy a menší podniky získají kontrolu nad celou sítí z jednoho logického místa, které výrazně zjednoduší návrh sítě, provoz a podporu prvků od různých výrobců. SDN výrazně zjednoduší i síťová zařízení, která pak nebudou muset rozeznávat a zpracovávat tisíce protokolů, ale pouze přijímat pokyny od SDN kotroleru. Nejdůležitější je, že mohou provozovatelé sítí a správci programovat tento zjednodušený síťový provoz, aniž by museli ručně konfigurovat desítky tisíc řádků kódu rozptýleného mezi tisíce zařízení. Kromě toho využívá i centralizované inteligence a správce může změnit chování sítě v reálném čase, nasadit nové aplikace a síťové služby v řádu hodin nebo dnů. Se standardní sítí tohle může trvat i měsíce, a to je

dnes značně dlouhá doba. Centralizace sítě je obsažena v řídicí vrstvě a dává správcům sítě flexibilitu konfigurace, správy, zabezpečení a optimalizaci síťových zdrojů pomocí dynamických a automatizovaných programů pro SDN. Kromě toho mohou psát programátoři tyto programy sami a nemusí se omezovat na funkce, které jsou pevně nastaveny od výrobců v uzavřeném softwarovém prostředí. [2]



Obr. 2.1: Logický pohled na architekturu SDN. [2]

Podle Cariden vize tvoří jádro SDN software označovaný jako Network Services (NS-OS). NS-OS může být popsán ve třech vrstvách:

- **Síťová řidicí vrstva** – je ekvivalentní k ovladačům zařízení v počítačových operačních systémech poskytujících I/O a řídí specifické zařízení.
- **Vrstva síťového modelu** – poskytuje abstraktní pohled na síťové zdroje a řeší rezervaci po určitý čas.
- **Vrstva řidicích služeb** – poskytuje inteligenci k ovládání síťových cest, síťový přístup a orchestraci úkolů. [4]

Software byl vždy důležitým prvkem sítí a nyní je zaměřen na inovace v sítích jako nikdy předtím. Je zde silné hnutí k vytvoření programovatelného síťového standardu, který umožní vývojářům softwaru spoléhat na síťové zdroje se stejnou lehkostí, jako je tomu u výpočetních a úložných zařízení. [4]

Na jedné straně je SDN zabývající se oddělením řídicího síťového softwaru od síťového hardwaru. Na straně druhé je SDN zabývající se poskytováním standar-

dizovaného programového rozhraní pro vývojáře aplikací. Třídění SDN aplikací je rozděleno podle hodnoty jejich zdroje, které přispívají SDN:

- Prvním zdrojem hodnoty je bezpečnost a další služby povolující tok na úrovni programovatelnosti, to je Flow Services SDN (Služby toku SDN sítě).
- Druhým zdrojem je virtualizace, označená Virtualization SDN.
- Třetí je programovatelnost síťových zdrojů podle aplikací, Infrastructure SDN (Infrastruktura SDN). [4]

SDN architektura dále podporuje sadu API, která umožňuje provádět běžné síťové služby, včetně směrování, multicastu, bezpečnosti, řízení přístupu, řízení šířky pásma, přepínání, kvality služeb, procesoru a optimalizace úložného prostoru, využití energie a všechny formy zásad pro správu, vytvořené na míru pro splnění obchodních cílů. Například tato architektura umožňuje snadno definovat a prosazovat stejnou politiku v celé drátové i bezdrátové technologii na akademické půdě. Umožňuje řídit celou síť prostřednictvím inteligentní skladby a opravných systémů. The Open Networking Foundation studuje otevřené API pro podporu řízení od různých výrobců, které otevírá dveře poptávkám na přidělování zdrojů, vlastní správu opravných položek a opravdu virtualizované a zabezpečené služby cloudu. [2]

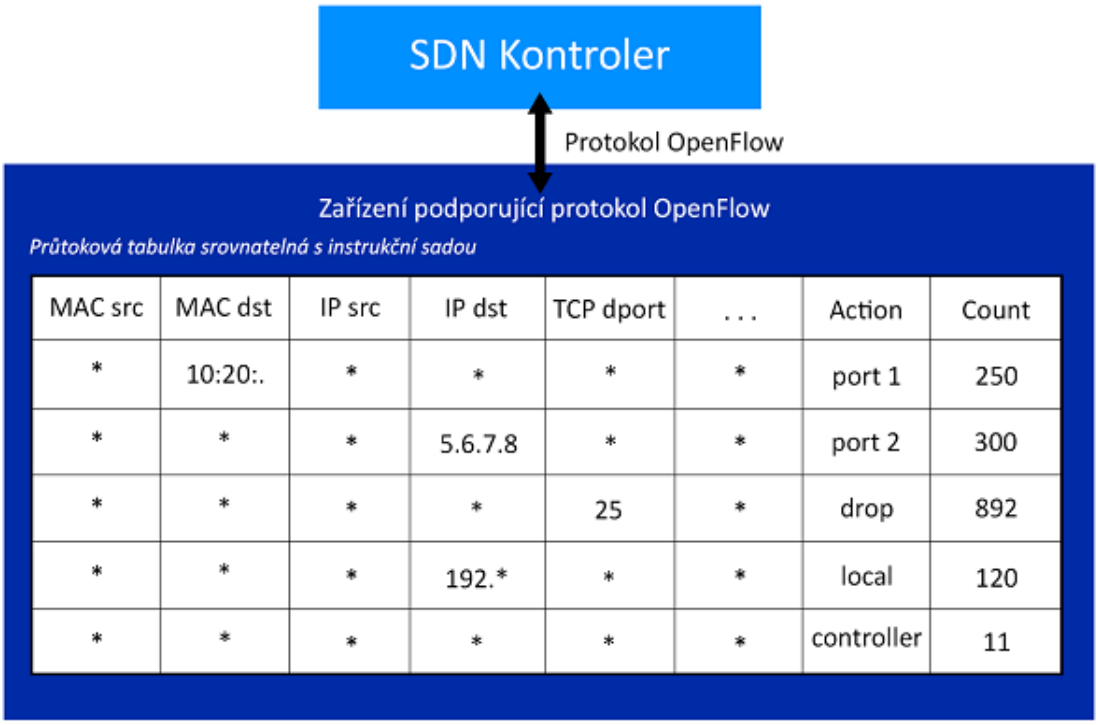
S otevřeným API mezi kontrolní a aplikační vrstvou mohou obchodní aplikace využívat síťové služby a jejich schopnosti, aniž by byly vázány na podrobnosti o jejich stavu. Je to síť, která není tolik „aplikačně-informovaná“ jako „aplikačně-přizpůsobená“. V důsledku toho, výpočetní, skladovací a síťové prostředky mohou být optimalizovány. [2]

2.4 Protokol OpenFlow

OpenFlow je první standard komunikačního rozhraní definovaného mezi kontrolní a provozní vrstvou architektury SDN. Protokol umožňuje přímý přístup a manipulaci z provozního plánu síťových prvků jako přepínačů a směrovačů, jak fyzických tak i virtuálních. Žádný jiný protokol nedělá to, co OpenFlow, že síť přepne do logicky centralizované, která je ovládána pomocí ovládacího softwaru. [2]

OpenFlow může být porovnáno s instrukční sadou procesoru, znázorněné na obr. 2.2. Protokol stanovuje základní primitiva, která mohou být použita podle vnější softwarové aplikace naprogramované k provozu síťových zařízení, stejně jako instrukční sady procesoru by naprogramovaly počítačový systém. [2]

Protokol je realizován na obou stranách rozhraní mezi zařízeními síťové infrastruktury a řidicím SDN softwarem (kontrolerem). OpenFlow využívá pojem toků k identifikování síťového provozu na základě předdefinovaných pravidel, která mohou být statická nebo dynamicky naprogramovatelná na ovládacím SDN softwaru.



MAC src	MAC dst	IP src	IP dst	TCP dport	...	Action	Count
*	10:20:.	*	*	*	*	port 1	250
*	*	*	5.6.7.8	*	*	port 2	300
*	*	*	*	25	*	drop	892
*	*	*	192.*	*	*	local	120
*	*	*	*	*	*	controller	11

Obr. 2.2: Příklad OpenFlow instrukční sady. [2]

To umožňuje, aby správce definoval provoz toků síťovými zařízeními na základě parametrů, jako je použití různých protokolů, portů, aplikací a cloudových zdrojů. Vzhledem k tomu, že OpenFlow umožňuje síti být naprogramovaná pro jednotlivé toky na základě SDN architektury, zajišťuje mimořádně detailní kontrolu a umožňuje síti reagovat na právě probíhající (real-time) změny v aplikačních, uživatelských a relačních úrovních. Aktuální směrování založené na IP neposkytuje takovou úroveň kontroly nad sítí, jelikož všechen provoz mezi dvěma koncovými body musí procházet přes stejnou cestu v síti bez ohledu na požadavky jednotlivých služeb. [2]

Protokol OpenFlow je klíčovým předpokladem pro softwarově definované sítě a v současné době i jediný standardizovaný SDN protokol, který umožňuje přímou manipulaci provozních plánů síťových zařízení. Zatímco zpočátku byl aplikován jen na Ethernetovou síť, OpenFlow rozšířil přepínání na mnohem širší okruh případů použití. SDN architektura tak může být integrována hladce v podniku ve stávající infrastruktuře a poskytnout také jednoduchou cestu k přechodu pro ty segmenty sítě, které potřebují SDN funkčnost nejvíce. [2]

Velký počet výrobců, vyrábějících směrovače a přepínače, oznámilo svůj úmysl podporovat protokol OpenFlow, jako Brocade Communications, Arista Networks, Cisco, Force 10, Extreme Networks, IBM, Juniper Networks, Larch Networks, HP a NEC.

Protokol podporuje tři typy zpráv, *controller-to-switch*, *asynchronní* a *symetrické*. Controller-to-switch zprávy zahajuje kontroler a používají se přímo pro řízení nebo kontrolu stavu přepínače. Asynchronní zprávy jsou iniciovány přepínačem a využívají se k aktualizaci síťových událostí kontroleru a změn stavu dalších přepínačů. Symetrické zprávy jsou inicializovány přepínačem nebo kontrolerem a posílány bez jakékoliv žádosti na jejich odeslání. [5]

2.4.1 Controller-to-Switch zprávy

Zprávy zahajuje kontroler a může nebo nemusí požadovat odpověď od přepínače. Kontroler může požádat přepínač k zaslání jeho možností a přepínač musí na tuto otázku odpovědět, což se obvykle provádí při vytváření OpenFlow kanálu. Dále může nastavovat a dotazovat se na konfiguraci parametrů v přepínači. [5]

Vlastnosti zprávy:

- **Modify-State**: Tyto zprávy řídí stav přepínačů. Jejich primárním cílem je přidávat, mazat a upravovat provozní záznamy v OpenFlow tabulkách a nastavení vlastností portů přepínače.
- **Read-State**: Takto kontroler kontroluje běžící konfiguraci, statistiku a schopnosti přepínače.
- **Packet-out**: K posílání rámců směrem ven specifickým portem na přepínači, směrem určeným přes zprávu Packet-in. Zpráva Packet-out musí obsahovat úplný rámec nebo vyrovnávací ID odkazující na rámec uložený v přepínači. Dále musí obsahovat seznam akcí, které mají být aplikovány v pořadí v jakém jsou specifikovány. Prázdný seznam akcí způsobí zahození rámce.
- **Barrier**: Bariérové žádost/odpověď zprávy jsou používány pro přijímání oznámení o dokončených operacích.
- **Role-Request**: Používány pro nastavení role kontroleru jeho OpenFlow kanálu, nebo pro dotaz na roli. To je užitečné při připojení s více kontrolery.
- **Asynchronous-Configuration**: Slouží na nastavení dalšího filtru v asynchronních zprávách, které chce přijímat na vlastní OpenFlow kanál, nebo se dotázat na jeho filtr. Což je užitečné, když se přepínače připojí k více kontrolerům. [5]

2.4.2 Asynchronní zprávy

Asynchronní zprávy jsou odesílány jen z přepínačů. Posílají je kontrolerům na označení příchodu rámce, změnu stavu přepínače, nebo označení chyby. Čtyři hlavní typy zpráv:

- **Packet-in**: Přenese kontrolu rámce kontroleru. Pro všechny rámce poslané do kontroleru rezervuje port v záznamech toků. Ostatní záznamy jako TTL záznam může vygenerovat packet-in zpráva.
- **Flow-Removed**: Informuje kontroler o smazání záznamu z tabulky toků. Tyto zprávy jsou označeny v záznamech jako `OFPFF_SEND_FLOW_REM`. Jsou vytvořeny po smazání záznamů toků kontrolerem, nebo když je na přepínači překročen nějaký odpočet.
- **Port-Status**: Informuje kontroler o změně na portu. Tento záznam zahrnuje změny v konfiguraci záznamů portu, když je přiveden do stavu „Down“ uživatelem, a změnu stavu portu, když linka sama přejde do stavu „Down“.
- **Error**: Přepínač tak ohlašuje problémy kontroleru. [5]

2.4.3 Symetrické zprávy

Symetrické zprávy jsou odesílány bez výzvy na jejich odeslání a jsou posílány v obou směrech. Jsou to zprávy:

- **Hello**: Hello zprávy jsou vyměňovány mezi přepínačem a kontrolerem během zakládání jejich spojení.
- **Echo**: Tyto žádost/odpověď zprávy mohou být poslány od obou a musí přijmout odpověď „Echo reply“. Jsou hlavně určeny k ověření funkčnosti *kontroler – přepínač* spoje a mohou být dobře využity na měření jejich zpoždění a šířky pásma.
- **Experimenter**: Poskytují standardní způsob pro OpenFlow přepínače na nabídnutí dalších možností s OpenFlow zprávami. Toto je místo pro budoucí OpenFlow změny. Zatím se nepoužívá. [5]

2.4.4 OpenFlow kanálové spojení

OpenFlow kanál slouží k výměně OpenFlow zpráv mezi přepínačem a kontrolerem. Typický OpenFlow kontroler řídí více kanálů, každý na rozdílné přepínače. Přepínač s podporou OpenFlow může mít jeden kanál na jeden kontroler, nebo více kanálů, pro zajištění větší spolehlivosti, na více rozdílných kontrolerů. [5]

Kontroler typicky řídí přepínač vzdáleně přes jednu nebo více sítí. OpenFlow kanál je typicky popisovaný jako jedno síťové spojení využívající TLS[3] nebo čistě TCP protokol. Přepínač vždycky inicializuje spojení s kontrolerem. [5]

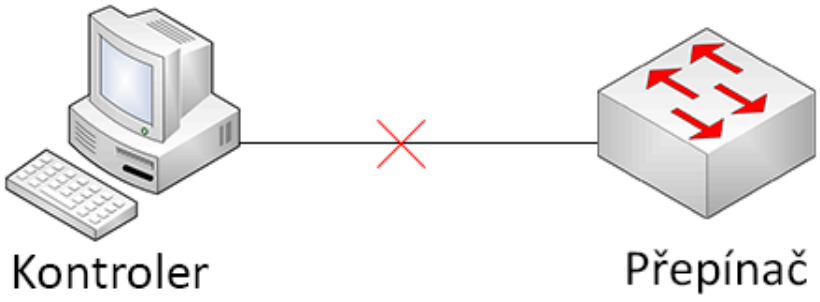
[3]Transport Layer Security

Nastavení spojení

Přepínač musí být schopný komunikovat s kontrolerem pomocí nastavené IP adresy používající specifický port. Když zná přepínač adresu kontroleru, naváže klasické TLS nebo TCP spojení. Tento provoz neprochází přes OpenFlow tabulku, proto přepínač musí identifikovat příchozí provoz jako lokální a zkontrolovat znovu provozní tabulky. [5]

Při prvním spojení obě strany musí ihned poslat `OFPT_HELLO` zprávu s nejvyšší verzí OpenFlow protokolu, kterou podporuje. Tato zpráva dále obsahuje pár elementů k pomoci při spojení. Během příjmu této zprávy musí příjemce určit verzi OpenFlow protokolu. Jestliže obě odeslané a přijímané Hello zprávy obsahují element `OFPHET_VERSIONBITMAP` a mají obě společně nastavené tyto bity, obě musí podporovat co nejvyšší verzi. Jestliže ne, je nastavena nižší verze protokolu a pošle se zpráva `OFPT_ERROR` s typem pole `OFPET_HELLO_FAILED`, kde je v kódovém poli nastaveno `OFPHFC_INCOMPATIBLE` a spojení je ukončeno. [5]

Přerušení spojení



Obr. 2.3: Přerušení spojení kontroleru s přepínačem.

V případě, že přepínač ztratí kontakt s kontrolerem, viz obr. 2.3, jako výsledek vypršení „echo request“ zprávy, vypršení TLS spojení, nebo jiného odpojení, přepínač musí ihned vložit „fail secure mode“ nebo „fail standalone mode“, závisející na implementaci a konfiguraci přepínače. V prvním případě se jen změní chování přepínače, že veškeré rámce a zprávy směřující ke kontroleru jsou zahozeny a existující záznamy toků zůstávají v přepínači do vypršení nastavené doby. Podle „fail standalone mode“ začne přepínač pracovat jako Ethernetový přepínač, toto je ale možné jen u hybridních přepínačů. Během opětovného kontaktování kontroleru jsou existující záznamy zachovány, kontroler poté má možnost záznamy vymazat, pokud tak požaduje. [5]

2.4.5 OpenFlow tabulky

OpenFlow podporuje přepínače i směrovače. OpenFlow přepínače jsou kompatibilní v dvou typech: *OpenFlow-only* a *OpenFlow-hybrid*. OpenFlow-only přepínače se řídí, při přeposílání rámců, čistě jen OpenFlow tabulkou a nemohou je zpracovávat jinak. OpenFlow-hybrid přepínače podporují OpenFlow operaci i normální Ethernetové přepínání. Mohou povolit rámec, který jde z softwarově definované sítě do normální Ethernetové sítě. [5]

OpenFlow pipeline z každého přepínače obsahuje několik tabulek toků, kde každá tabulka má několik různých záznamů. Přepínač potřebuje nejméně jednu tabulku, ale optimální je víc tabulek. Přepínač s jednou tabulkou je platný, ale v tomto případě pipeline pracuje značně zjednodušeně. [5]

Tabulky jsou popořadě číslovány a začínají na 0. Zpracování začíná vždy na první tabulce, tedy číslu 0, další tabulky jsou poté prohledávány v závislosti na předchozích tabulkách. Až se najde patřičná tabulka, tak se začne vyhledávat záznam toku. Po nalezení označí instrukční sada záznam toku jako provedený. Tato instrukce může rámec poslat do tabulky, kde se provede stejný proces znovu. Záznam vždy může poslat rámec jen do vyšší tabulky. Když vhodný záznam nepošle rámec do další tabulky, tak se proces zastaví v této tabulce a rámec je zpracován s jeho přiřazenou sadou a obvykle odeslán. Jestliže rámec nenalezne v tabulkách žádný záznam, rozhoduje jeho *table-miss* záznam toku, kde může být nastaveno: zahození rámce, přechod do další tabulky, nebo poslání jej ke kontroleru, pomocí *packet-in* zprávy. [5]

Tabulka se skládá ze záznamů.

Tab. 2.1: Hlavní komponenty záznamů toků v tabulce.

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie

Tab. 2.1 obsahuje:

- **match fields**: na porovnání dalších rámců. Skládá se z vstupního portu, hlavičky rámce a volitelné hodnoty specifikované předchozí tabulkou.
- **priority**: porovnávání priority záznamů toků.
- **counters**: aktualizované, když rámec nalezne záznam.
- **instructions**: na modifikování akční sady nebo porovnávacích procesů uvnitř tabulky.
- **timeouts**: maximální množství času nebo nečinného času před vypršením přeposílání od přepínače.
- **cookie**: vybraná hodnota kontrolerem k filtrování průtokové statistiky, její modifikace a vymazání. Není použito při zpracování rámce. [5]

Odstranění záznamů

Záznamy toků mohou být odstraněny z tabulky toků pomocí dvou způsobů, na žádost kontroleru, nebo po uplynutí nastavené doby v průtokového mechanismu na přepínači. Tento mechanismus je spuštěn přepínačem nezávisle na kontroleru a je založen na stavu a konfiguraci záznamu toku. Každý záznam má `idle_timeout` a `hard_timeout`. Jestliže `hard_timeout` nemá hodnotu nula, tak přepínač musí zaznamenat příchozí čas prvního rámce a způsobí, že záznam toku bude vymazán po uplynutí stanoveného času, bez ohledu na množství rámců, které jsou zpracovány. Pro `idle_timeout` platí pro nenulovou hodnotu, že se zaznamená příchozí čas posledního rámce patřícího k záznamu a záznam toku je odstraněn, když jeden z odpočtů překročí stanovenou hodnotu. Kontroler může odstranit záznam z tabulky toků posláním zprávy `OFPFC_DELETE` nebo `OFPFC_DELETE_STRICT`. Po odstranění musí přepínač zkontrolovat označení záznamu (`OFPFC_SEND_FLOW_REM`), pokud je toto označení nastaveno, tak musí poslat kontroleru zprávu o smazání záznamu. Tato zpráva obsahuje kompletní popis smazaného záznamu, důvod smazání (vypršení nebo smazání), dobu smazání a průtokovou statistiku v tento čas. [5]

Měřící tabulky

Tyto tabulky se skládají z měřících záznamů, definujících měření jednotlivých toků, což povoluje OpenFlow implementovat různé jednoduché QoS[4] operace, jako například DiffServ. Měřící opatření přiřazuje hodnotu k rámcům a to povoluje kontrolovat hodnotu těchto rámců. Měření je připojeno přímo k záznamu toku. Každý záznam může obsahovat specifické měření v jeho instrukční sadě, různé měření mohou být použita ve stejné tabulce.

Každý měřící záznam je identický jeho měřícím identifikátorem a obsahem:

- **měřící identifikátor**: je 32 bitové neoznačené celé číslo, které je unikátní,
- **měřící skupiny**: je neuspořádaný list měřících skupin, kde každá specifikuje hodnotu skupiny a způsob zpracování rámce,
- **čítače**: jsou aktualizovány, když je rámec zpracován. [5]

Čítače

Čítače jsou udržovány pro každé tabulky toků, záznamy toků, porty, fronty, skupiny, měřící tabulky a měřící skupiny. OpenFlow kompatibilní čítače mohou být implementovány v softwaru a udržovány dotazováním hardwarových čítačů s více

[4] Quality of service

omezenými rozsahy. Tabulka obsahuje sadu čítačů definovaných OpenFlow specifikací. Přepínač není povinen podporovat všechny čítače, jen ty označené v tabulce jako „důležité“. [5]

Instrukce

Každý záznam toku obsahuje sadu instrukcí, které jsou provedeny při porovnání rámce se záznamem. Tyto instrukce znamenají změny v rámci. Přepínač nemusí podporovat všechny instrukční typy, jenom ty označené „důležitá instrukce“, kontroler se také může zeptat, které „volitelné instrukce“ přepínač podporuje. Instrukční sady spojené se záznamem toku obsahují pouze jednu instrukci každého typu. Přepínač musí odmítnout záznam toku, jestliže je nemožné provést instrukci přiřazenou k záznamu. V tomto případě musí vrátit chybu o nepodporujícím záznamu. Tabulka toků nemusí podporovat všechny instrukce, nebo akce. [5]

2.4.6 Verze protokolu OpenFlow

První verze protokolu neměly žádné specifikace. 28. 3. 2008 vyšla první verze 0.2.0, kdy tentýž den byla upravená ještě na verzi 0.2.1 s označením „1“. Začátkem května 2008 byl protokol upraven na verzi 0.8.0 a označením 0x83. Tato verze přeskládala typy zpráv protokolu, přidala prioritu pro jednotlivé toky, chybové zprávy a byly například přidány zprávy Table Stats a Port Stats. [5]

Následovaly verze 0.8.1 (0x83), 0.8.2 (0x85) a 2. 12. 2008 verze 0.8.9 (0x97), která již má svoji OpenFlow Switch specifikaci verze 0.8.9. Zprostředkovala záznamům toků mít masku podsítě. Ovšem obsahuje inverzní masku, takže CIDR notace „24“ v inverzní podobě je „255.0.0.0“. Zpráva Port Stats byla rozšířena pro získání více informací. Dále se upravila zpráva In Port pro funkci u bezdrátové sítě, kde se zprávy posílají přes stejné rozhraní. Byla přidána informace o stavu rozhraní, zda není ve stavu „down“. Byly přidány zprávy Echo Request/Reply, zpráva pro přidávání rozšíření od výrobců, zacházení s IP fragmenty, podpora 802.1D protokolu STP, hard timeout pro záznamy, vytvořená zpráva Hello, podpora portů zvýšena na 65280 a řada dalších vylepšení... [6]

Verze 0.9 (označení 0x98) byla vydána 20. 7. 2009 a popsána v OpenFlow Switch specifikaci 0.9.0. Zde byl vytvořen mechanismus, který při poruše kontroleru předal kontrolu druhému, záložnímu, byla přidána bariérová zpráva a další. [7]

31. 12. 2009 byla zveřejněna verze 1.0 (0x01) se specifikací 1.0.0. Je zde obsažen popis cesty mezi kontrolerem a přepínačem, slouží k popsání přepínače. Podpora porovnávání IP ToS/DSCP bitů, průtoková doba a uplynutí zpráv je nyní popsáno v nanosekundách, dříve bylo v milisekundách. [8]

Dne 28. 2. 2011 byla vydána verze 1.1 (0x02) a popsána specifikací 1.1.0. Zde byla zavedena podpora více tabulek toků, což má výhodu, že hardware má uvnitř více tabulek, jako L2 tabulky, L3 a další... Podpora značkování VLAN a MPLS, předchozí verze měly značně omezenou podporu VLAN, ovšem nyní lze značky explicitně vkládat, upravovat a mazat. Je zde podpora virtuálních portů na prvcích a vyřešeno přerušení spojení kontroler – přepínač. [9]

Poté vznikla rada ONF (Open Networking Foundation), která dne 5. 12. 2011 vydala specifikaci 1.2, která popisuje protokol 1.2 (0x03). Byla přidána podpora IPv6, experimentální Error zpráva pro generování vlastních chybových zpráv, upraveny zprávy Packet In, Flow Mod, ... [10]

13. 4. 2012 byla vydána specifikace 1.3.0 popisující protokol 1.3 (0x04). Tato verze obsahuje flexibilnější rámce pro expresní možnosti. Tyto možnosti byly přesunuty pryč z tabulkových statistik do vlastních žádost/odpověď zpráv. Byla zde přidána možnost table-miss, měřící tabulky, tokové čítače na žádost, opraveny některé chyby a mnoho dalších funkcí zde bylo implementováno. [11]

V protokolu bylo ale pár drobných chyb, které v srpnu 2012 opravil protokol verze 1.3.1 se stejným označením 0x04. Toto je prozatím nejvyšší verze protokolu OpenFlow. [5]

2.5 Výhody softwarově definované sítě založené na protokolu OpenFlow

SDN umožňuje síti být konkurenčně výhodnou s nižšími náklady na datové centrum do budoucna. Umožňuje správci řešit velkou šířku pásma, dynamickou povahu dnešních aplikací, přizpůsobit síť měnícím se obchodním potřebám, výrazně snížit provoz a složitost řízení. [2]

Výhody této sítě:

- **Centralizované ovládání zařízení od různých dodavatelů**: SDN řídicí software může ovládat kterékoliv OpenFlow podporující síťové zařízení od libovolného výrobce, včetně přepínače, směrovače a přepínače virtuálního.
- **Možnost automatizace ke zmenšení složitosti**: OpenFlow založené na SDN nabízí flexibilní síťovou automatizaci, což dovoluje vytvořit nástroje, které automatizují mnoho úkolů správy, které se dnes provádějí ručně. Tyto automatizační nástroje sníží provozní režii, sníží nestabilitu sítě vzniklou chybnou obsluhou a podporuje nově vznikající IT-as-a-Service a samoobslužné zprostředkovací modely. Kromě toho s SDN mohou být aplikace založené na cloudu řízeny inteligentní orchestrací a provozními systémy, což dále snižuje provozní režie a zároveň zvyšuje obchodní agilitu.

- **Vyšší míra inovací**: SDN přijetí urychlují obchodní inovace tím, že se síť musí přeprogramovat v reálném čase ke splnění konkrétních potřeb a požadavků uživatelů, pokud se vyskytnou. Při virtualizaci síťové infrastruktury a oddělení od jednotlivých síťových služeb, například pomocí SDN a OpenFlow což dá správci a případně i uživateli schopnost přizpůsobit chování sítě a zavádět nové služby v řádu hodin.
- **Zvýšení spolehlivosti sítě a bezpečnost**: SDN umožňuje správci definovat vysokou úroveň konfigurace a celkový přehled, který je pak přeložen do infrastruktury přes OpenFlow. Eliminuje potřebu individuálně konfigurovat síťová zařízení pokaždé, když koncový bod, služba, nebo aplikace je přesunuta nebo přidána, což snižuje pravděpodobnost selhání sítě kvůli špatné konfiguraci nebo nějaké nesrovnalosti. SDN poskytuje kompletní přehled a kontrolu nad sítí.
- **Přesnější řízení sítě**: OpenFlow umožňuje správci aplikovat politiky na velmi různorodé úrovni, včetně zasedání, uživatele a zařízení, automatizovaným způsobem.
- **Lepší uživatelské rozhraní**: Centralizace sítě ani tvorby stavu informací nejsou k dispozici na vyšší úrovni aplikace, infrastruktura SDN se může lépe přizpůsobit dynamickým potřebám uživatelů. S OpenFlow bude schopna video aplikace detekovat aktuální propustnost v síti v reálném čase a automaticky nastavit rozlišení videa odpovídajícím způsobem. [2]

2.6 Nevýhody softwarově definované sítě založené na protokolu OpenFlow

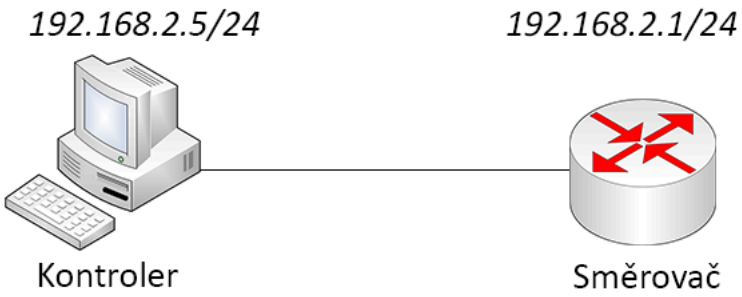
Hlavní nevýhodou této sítě je nyní její obrovská složitost. Doposud není vydán veřejnosti kontroler, který by obsahoval grafické rozhraní, ve kterém by šlo síť jednoduše ovládat. Jsou zde jen softwarové řešení, kde je zapotřebí znát programovací jazyky. Vytvoření logické topologie v těchto kontrolerech trvá nemalou dobu. Softwarových řešení zatím není mnoho, jen kolem pěti, některé jsou zatím jen v beta fázi testování.

Další nevýhodou je rada Open Networking Foundation, která spravuje protokol. Tato organizace je ale nová, a proto zatím nebudí příliš velké ohlasy v sítích. Je původně vedená a vlastněná velkými americkými firmami, jako jsou Google, Facebook, Verizon, Yahoo, ... Například firma Google v nedávné době nasadila na vlastní síť Softwarově definované sítě, na což ale vytvořila sama potřebný hardware a software, který sama drží v tajnosti před veřejností. Tento krok mohl značně pozastavit vývoj těchto sítí, nyní ale začínají síť prosazovat firmy jako Hewlett-Packard, Juniper Networks, IBM, Cisco a další.

2.7 Měření softwarově definovaných sítí

2.7.1 Inicializace spojení kontroler – směrovač

Pro měření jsem použil reálné prvky, kde kontroler je použit jako softwarové řešení od Stanfordské univerzity, a to kontroler Beacon ve verzi 1.0.2, který je naprogramovaný v programovacím jazyku Java. Toto softwarové řešení bylo spuštěno na přenosném počítači s operačním systémem Windows 7. Jako další prvek zde byl použit směrovač Mikrotik RB800, který po implementování balíčku dokáže podporovat protokol OpenFlow.



Obr. 2.4: Schéma zapojení kontroler – směrovač.

Prvkům jsem nastavil dané síťové adresy, viz obr. 2.4, a propojil je pomocí UTP kabelu. Dále jsem spustil program Wireshark na notebooku, pro odchytávání provozu na sítí, kam jsem implementoval OpenFlow dissector pro analyzování dat přes protokol OpenFlow. Nakonec jsem spustil kontroler a odchytil jsem následující komunikaci, viz obr. 2.5.

No.	Time	Source	Destination	Protocol	Length	Info
1442	129.686277	192.168.2.1	192.168.2.5	OFP	74	Hello (SM) (8B)
1443	129.697146	192.168.2.5	192.168.2.1	OFP	74	Hello (SM) (8B)
1445	129.702141	192.168.2.5	192.168.2.1	OFP	74	Features Request (CSM) (8B)
1447	129.702332	192.168.2.1	192.168.2.5	OFP	98	Features Reply (CSM) (32B)
1448	129.722176	192.168.2.5	192.168.2.1	OFP	78	Stats Request (CSM) (12B)
1449	129.722415	192.168.2.1	192.168.2.5	OFP	1134	Stats Reply (CSM) (1068B)
1450	129.724212	192.168.2.5	192.168.2.1	OFP	78	Set Config (CSM) (12B)
1451	129.724570	192.168.2.5	192.168.2.1	OFP	82	Get Config Request (CSM) (8B)
1452	129.724828	192.168.2.1	192.168.2.5	OFP	74	Barrier Reply (CSM) (8B)
1454	129.919162	192.168.2.1	192.168.2.5	OFP	78	Get Config Reply (CSM) (12B)
1455	129.922750	192.168.2.5	192.168.2.1	OFP	138	Flow Mod (CSM) (72B)
1456	129.923823	192.168.2.5	192.168.2.1	OFP	74	Barrier Request (CSM) (8B)
1458	129.924123	192.168.2.1	192.168.2.5	OFP	74	Barrier Reply (CSM) (8B)
1485	135.691097	192.168.2.5	192.168.2.1	OFP	74	Echo Request (SM) (8B)
1486	135.691540	192.168.2.1	192.168.2.5	OFP	74	Echo Reply (SM) (8B)
1528	140.698310	192.168.2.5	192.168.2.1	OFP	74	Echo Request (SM) (8B)
1529	140.698594	192.168.2.1	192.168.2.5	OFP	74	Echo Reply (SM) (8B)
1545	146.688765	192.168.2.5	192.168.2.1	OFP	74	Echo Request (SM) (8B)
1546	146.689189	192.168.2.1	192.168.2.5	OFP	74	Echo Reply (SM) (8B)
1555	149.702220	192.168.2.1	192.168.2.5	OFP	74	Echo Request (SM) (8B)
1556	149.702814	192.168.2.5	192.168.2.1	OFP	74	Echo Reply (SM) (8B)

Obr. 2.5: Prvotní komunikace mezi kontrolerem – směrovačem.

Jednotlivé zprávy v inicializaci spojení:

- **Zpráva typu Hello**: První, kdo započal komunikaci, je směrovač, který posílá kontroleru tuto zprávu na port 6633, na kterém kontroler defaultně naslouchá. V této zprávě je obsažena informace o nejvyšší verzi podporovaného OpenFlow protokolu směrovačem. Kontroler vyhodnotil podporu stejné verze protokolu, tudíž potvrzuje spojení znovu zprávou hello poslanou směrovači.

Header	
`Version: 0x01`	`Použitá verze protokolu OpenFlow, pro nynější specifikaci je verze 0x04.`
`Type: Hello (SM) (0)`	`Typ zprávy, SM = symetrická zpráva, 0 = zpráva typu Hello.`
`Length: 8`	`Délka zprávy.`
`Transaction ID: 0`	`ID transakce této zprávy. Odpovědi používají stejné ID.`

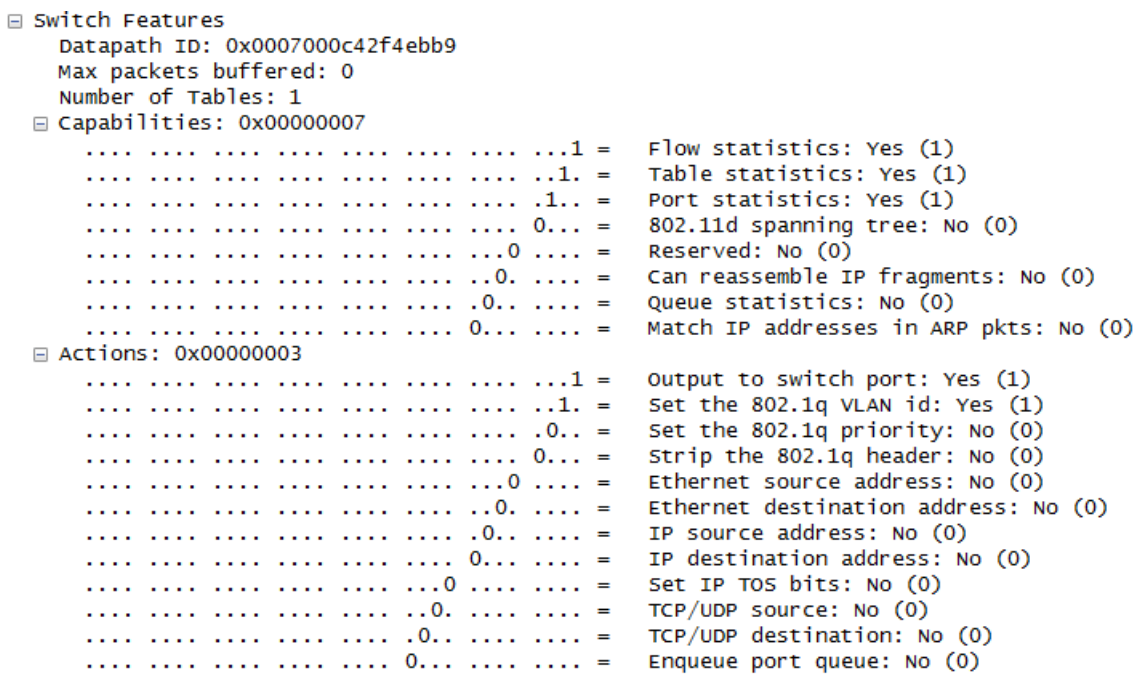
- **Zpráva typu Features Request**: Posílá kontroler a dotazuje se v ní na možnosti a akce směrovače, které může podporovat.

Header	
`Version: 0x01`	`Použitá verze protokolu.`
`Type: Feature Request (CSM) (5)`	`Typ zprávy, CSM = Controller-to-Switch zpráva, 5 = zpráva typu Feature Request.`
`Length: 8`	`Délka zprávy.`
`Transaction ID: 0`	`ID transakce této zprávy.`

- **Zpráva typu Features Reply**: Směrovač odpovídá kontroleru a u každé služby vloží na příslušné místo buď 1, že službu podporuje, nebo 0, službu nepodporuje. Podpora jednotlivých služeb tohoto směrovače obr. 2.6.

Header	
`Version: 0x01`	`Použitá verze protokolu.`
`Type: Feature Reply (CSM) (6)`	`Typ zprávy, CSM = Controller-to-Switch zpráva, 6 = zpráva typu Feature Reply.`
`Length: 32`	`Délka zprávy.`
`Transaction ID: 0`	`ID transakce této zprávy.`

- **Zpráva typu Stats Request**: Odesílá znovu kontroler a dotazuje se tak na název směrovače, sériové číslo a další označení.



Obr. 2.6: Podpora jednotlivých možností směrovače Mikrotik.

Header

Version: 0x01	Použitá verze protokolu.
Type: Stats Request (CSM) (16)	Typ zprávy, CSM = Controller-to-Switch zpráva, 16 = zpráva typu Stats Request.
Length: 12	Délka zprávy.
Transaction ID: 0	ID transakce této zprávy.

Stats Request

Type: Description of this OpenFlow switch (0x0000)

Flags: 0x0000

Body: <MISSING>

- **Zpráva typu Stats Reply**: Směrovač odpovídá na dotaz kontroleru.

Header

Version: 0x01	Použitá verze protokolu.
Type: Stats Reply (CSM) (17)	Typ zprávy, CSM = Controller-to-Switch zpráva, 17 = zpráva typu Stats Reply.
Length: 1068	Délka zprávy.
Transaction ID: 0	ID transakce této zprávy.

Stats Reply

```
Type: Description of this OpenFlow switch (0x0000)
Flags: 0
    Desc Stats Reply
    Mfr Desc: MikroTik          Popis výrobce.
    HW Desc:                    Popis hardwaru.
    SW Desc:                    Popis softwaru.
    Serial Num:                 Sériové číslo.
    DP Desc: beacon             Nastavitelný popis.
```

- **Zpráva typu Set Config**: Kontroler posílá konfiguraci k směrovači, jak zacházet s fragmenty a jak veliký soubor může být poslán ke kontroleru.

```
Header
Version: 0x01                 Použitá verze protokolu.
Type: Set Config (CSM) (9)    Typ zprávy, CSM = Controller-
                              to-Switch zpráva, 9 = zpráva
                              typu Set Config.
Length: 12                    Délka zprávy.
Transaction ID: 0             ID transakce této zprávy.
Switch Configuration
   Flags
   0 = Handling of Ip fragments:No special fragment handling
   (0).
   Max Bytes of New Flow to Send to Controller: 65535
```

- **Zpráva typu Get Config Request**: Zde se kontroler dotazuje na existující záznamy toků a na dobu jejich vypršení. Tato zpráva obsahuje i baiérovou zprávu pro zjištění o dokončení této operace, k ní následuje odpověď barrier reply.

```
Header
Version: 0x01                       Použitá verze protokolu.
Type: Get Config Request (CSM) (7)  Typ zprávy, CSM = Controller-
                                    to-Switch zpráva, 7 = zpráva
                                    typu Get Config Request.
Length: 8                           Délka zprávy.
Transaction ID: 0                   ID transakce této zprávy.
Header
Version: 0x01                       Použitá verze protokolu.
Type: Barrier Request (CSM) (18)    Typ zprávy, CSM = Controller-
```

```
                                    to-Switch zpráva, 18 = zpráva
                                    typu Barrier Request.
Length: 8                           Délka zprávy.
Transaction ID: 0                   ID transakce této zprávy.
```

- **Zpráva typu Get Config Reply**: Směrovač odpovídá na zprávu get config request.

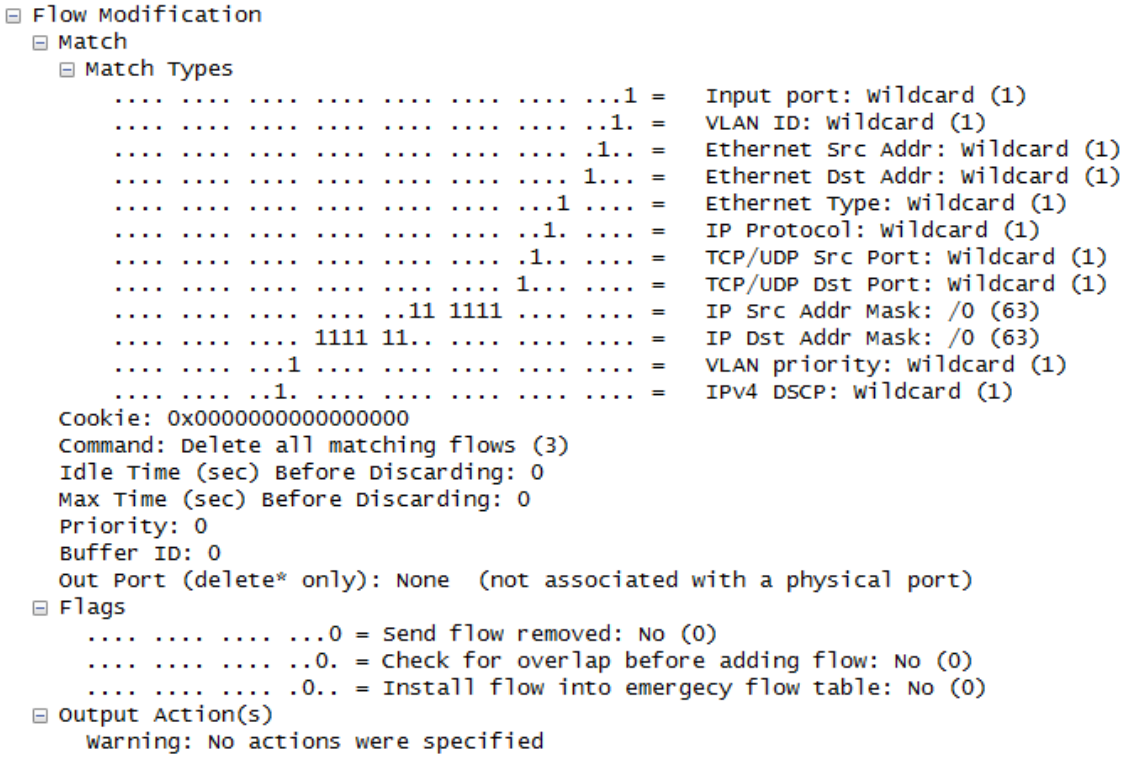
```
Header
Version: 0x01                       Použitá verze protokolu.
Type: Get Config Reply (CSM) (8)    Typ zprávy, CSM = Controller-
                                    to-Switch zpráva, 8 = zpráva
                                    typu Get Config Reply.
Length: 12                          Délka zprávy.
Transaction ID: 0                   ID transakce této zprávy.
Switch Configuration
   Flags
   0 = Handling of Ip fragments:No special fragment handling
   (0).
   Max Bytes of New Flow to Send to Controller: 65535
```

- **Zpráva typu Flow Mod**: Kontroler zde zasílá instrukce směrovači k přidání záznamů do jednotlivých tabulek toků, viz obr. 2.7. Což poté kontroler ověřuje bariérovou zprávou žádost/odpověď, zda toto směrovač provedl.

```
Header
Version: 0x01                   Použitá verze protokolu.
Type: Flow Mod (CSM) (14)       Typ zprávy, CSM = Controller-
                                to-Switch zpráva, 14 = zpráva
                                typu Flow Mod.
Length: 72                      Délka zprávy.
Transaction ID: 0               ID transakce této zprávy.
```

- **Zpráva typu Echo Request**: Zde posílá kontroler, ale může tomu být i naopak, a zjišťuje tak životnost spoje mezi těmito dvěma prvky. Zpráva byla vždy zaslána v intervalu přibližně 5 sekund a směrovač na ni odpovídá zprávou (Echo Reply). [8]

```
Header
Version: 0x01                    Použitá verze protokolu.
Type: Echo Request (SM) (2)      Typ zprávy, SM = symetrická
```



Obr. 2.7: Obsah zprávy typu Flow Mod.

	`zpráva, 2 = zpráva typu Echo Request.`
`Length: 8`	`Délka zprávy.`
`Transaction ID: 0`	`ID transakce této zprávy.`

Jako jedinou možnost k vkládání záznamů toků jsem objevil utilitu *dpctl*. Bohužel toto nejde prozatím implementovat do daného směrovače, tudíž jsem musel přejít k virtualizovanému řešení Mininet, což dokáže realizovat infrastrukturu s přepínači a jednotlivými hosty. Tato sada Mininet 2.0.0 pracuje pod operačním systémem Ubuntu 12.10 server 64-bit.

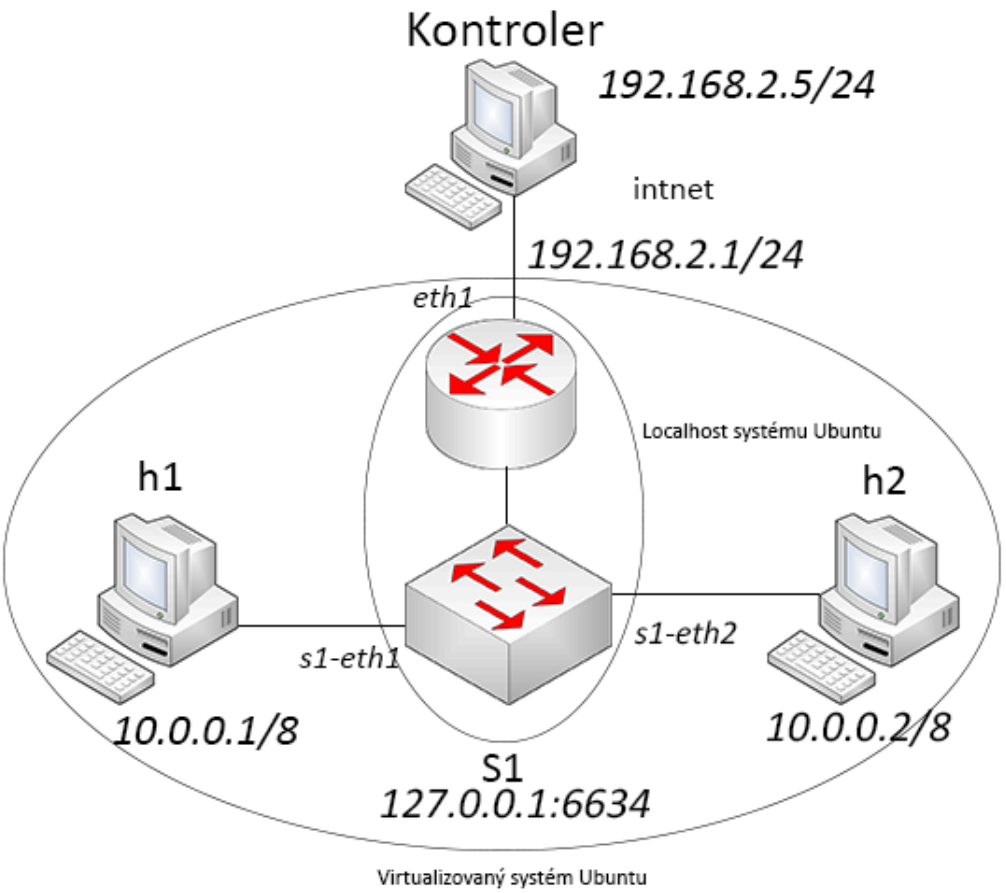
2.7.2 Vkládání záznamů toků

Na stránkách Mininetu jde přímo stáhnout .OVF soubor s operačním systémem Ubuntu a implementovanou utilitou Mininet. Jako virtualizační nástroj jsem zvolil VirtualBox od firmy Oracle, kam jsem importoval .OVF soubor. Dále jako kontroler jsem použil virtualizovaný systém Windows 8 Pro 64-bit. Tyto dva stroje jsem spojil pomocí vnitřní sítě „intnet“. Následně jsem spustil kontroler Beacon a vytvořil virtuální topologie následujícím příkazem:

```
sudo mn -topo single,2 -mac -switch ovsk -controller remote,
```

```
ip=192.168.2.5
```

kde se vytvoří jeden přepínač, 2 hosté se stejnou MAC adresou, jako je jejich název(h1 má 00:00:00:00:00:01), poté se upozorní přepínač, že kontroler naslouchá na ip adrese 192.168.2.5 a portu 6633, který je defaultně určen. Výsledná topologie s reálnými i virtuálními prvky je na obr. 2.8.

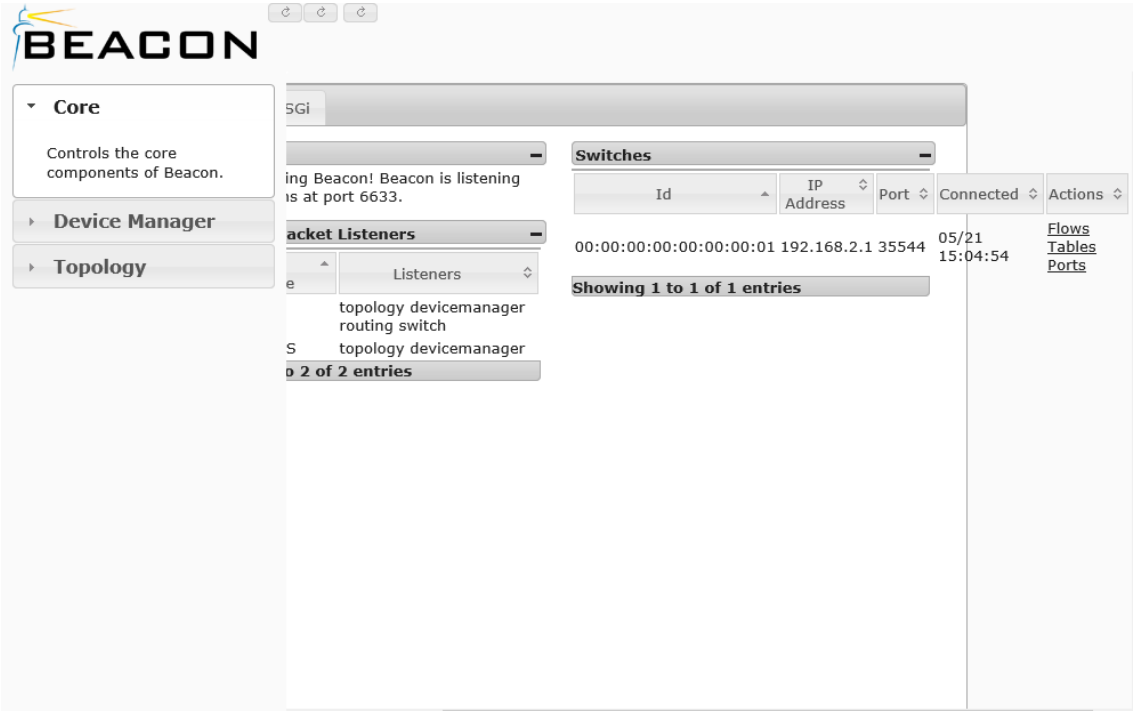


Obr. 2.8: Vytvořená topologie určená pro vkládání záznamů toků.

Po propojení je přepínač již přítomný v grafickém rozhraní kontroleru Beacon, viz obr. 2.9, nacházejícího se na adrese localhost:8080. V tomto uživatelském rozhraní jsou zaznamenané i klientské stanice s informací o připojení k jednotlivým přepínačům a označení rozhraní tohoto připojení, což znázorňuje obr. 2.10.

U těchto prvků je znovu podporovaný protokol OpenFlow verze 0x01, který vychází z OpenFlow Switch specifikace verze 1.0.0, nyní je již verze 0x04 popsaná ve specifikace 1.3.1. Při posílání zpráv Hello zde došlo od přepínače k přeposlání stejné zprávy, jelikož kontroler nestihl pružně odpovědět, a to přibližně po 0,2 sekundách. Dále ve zprávě Features Reply je zde nastaven počet tabulek toků na hodnotu 255, je zde podporována většina akcí přepínače, viz obr. 2.11, a jsou zde vypsány rozhraní

přepínače, kde je napsána MAC adresa rozhraní, jeho jméno, jestli rozhraní u protokolu STP[5] zahazuje rámce a další specifikace tohoto protokolu. Dále zda je rozhraní ve stavu „down“, jeho podporovaná rychlost: 10 Gb full-duplex rate a podporované médium: Copper medium (klasické měděné UTP vedení).

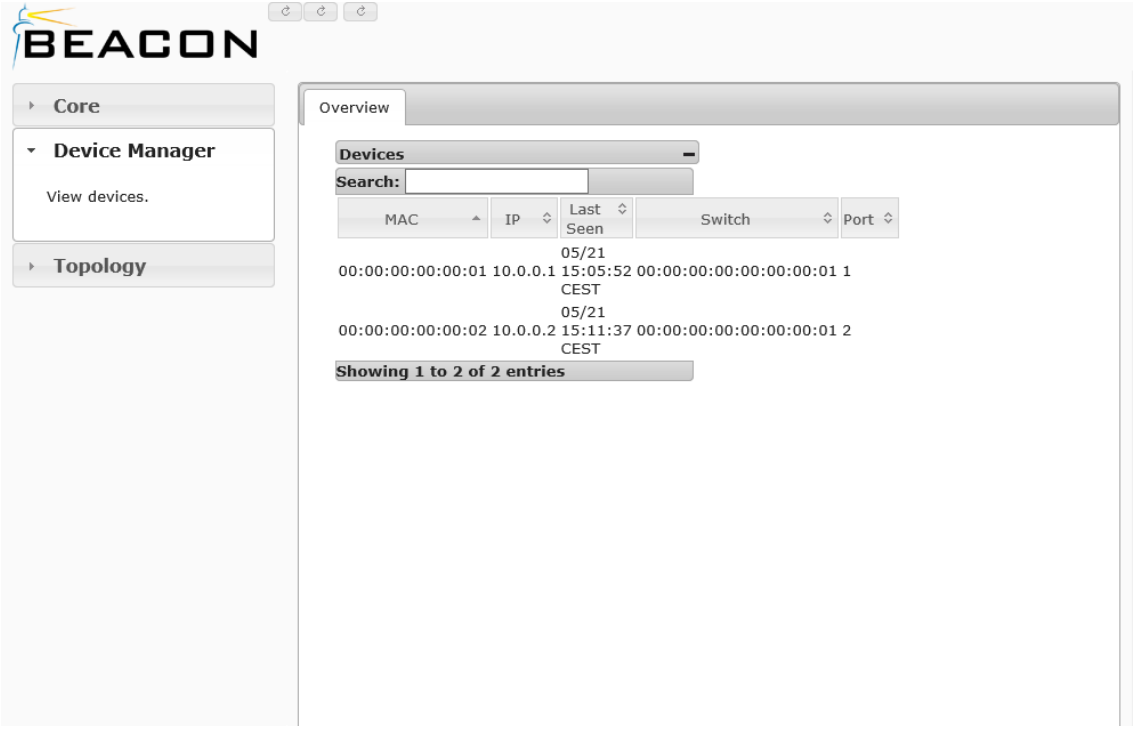


Obr. 2.9: Grafické rozhraní kontroleru Beacon.

Zpráva Stats Reply obsahuje následující informace:

```
Header
Version: 0x01                  Použitá verze protokolu.
Type: Stats Reply (CSM) (17)   Typ zprávy, CSM = Controller-
                               to-Switch zpráva, 17 = zpráva
                               typu Stats Reply.
Length: 1068                   Délka zprávy.
Transaction ID: 0              ID transakce této zprávy.
Stats Reply
Type: Description of this OpenFlow switch (0x0000)
Flags: 0
    Desc Stats Reply
    Mfr Desc: Nicira Networks, Inc.  Popis výrobce.
    HW Desc: Open vSwitch            Popis hardwaru.
```

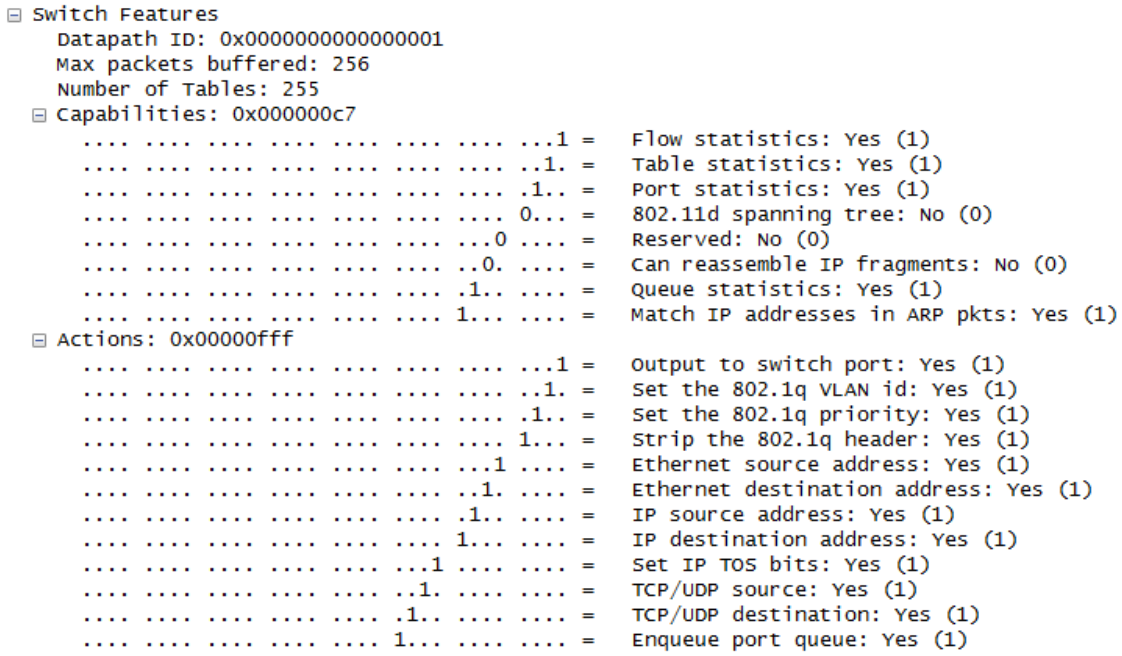
[5]Spanning tree protocol



Obr. 2.10: Grafické rozhraní kontroleru Beacon s ukázkou přípojených zařízení.

`SW Desc: 1.4.3`	`Popis softwaru.`
`Serial Num: None`	`Sériové číslo.`
`DP Desc: None`	`Nastavitelný popis.`

Po ověření funkce ping z h1 na h2, dojde k poslání ARP dotazu od h1 k přepínači. Ten porovná své záznamy toků se zadaným dotazem, kde nenajde odpověď. Přepošle tedy dotaz spolu se zprávou Packet In směrem ke kontroleru, ten pošle zprávu typu Packet out, díky které může přepínač poslat ARP dotaz na všechny rozhraní, kromě vstupního rozhraní a těch, které jsou zakázané STP protokolem. Přepínač zjistí, že ip adresa 10.0.0.2 má MAC adresu 00:00:00:00:00:02. Tato informace by byla normálně poslána k dotazujícímu se prvku, ale zde je odeslána se zprávou Packet In ke kontroleru. Zde kontroler vytvoří záznam toku a pošle ho přepínači zprávou Flow Mod s `idle_time` platným po dobu 5 sekund, kde je obsaženo, že veškeré ARP dotazy směřující od h2 k h1 jsou předány na příslušné rozhraní. V tutéž dobu pošle i zprávu Packet Out, kde uvádí, že tento rámec má být poslán na rozhraní s1-eth1. H1 tak tedy zjistilo, jakou MAC adresu má h2 a pošle mu ICMP zprávu, používanou u příkazu ping. Přepínač znovu nenajde záznam v žádné tabulce, tudíž pošle zprávy ICMP a Packet In kontroleru. Kontroler odpovídá v jednom rámci zprávou typu Flow Mod a zprávou Packet Out se zprávou ICMP. Při odpovědi Echo reply se toto znovu opakuje. Obsah Packet Out zprávy, kde je poslána s ICMP zprávou:



Obr. 2.11: Podpora jednotlivých možností přepínače Open vSwitch.

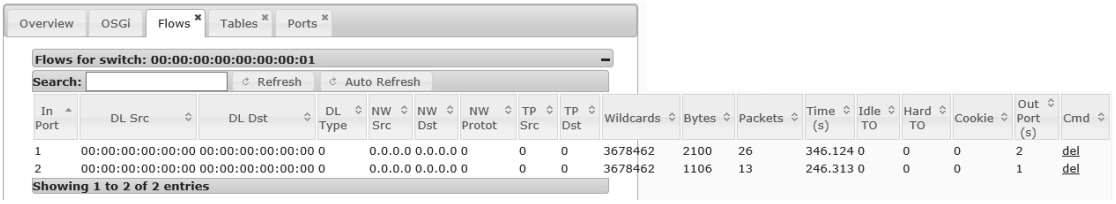
```
Header
Version: 0x01                  Použitá verze protokolu.
Type: Packet Out (CSM) (13)    Typ zprávy, CSM = Controller-
                               to-Switch zpráva, 13 = zpráva
                               typu Packet Out.
Length: 122                    Délka zprávy.
Transaction ID: 0              ID transakce této zprávy.
Packet Out
Buffer ID: None
Frame Recv Port: None (not associated with a physical port)
Size of action array in bytes:8
    Output Action(s)
    Type: Output to swtich port (0)
    Len: 8
    Output port: 2             ICMP zpráva je poslána na s1-eth2.
    Max Bytes to Send: 0
  # of Actions: 1
```

Pomocí příkazu `dpctl_show` na přepínači jsem si ověřil, že port značený jako 1 je s1-eth1 a 2 je s1-eth2. Nyní jsem mohl započít s vkládáním záznamů:

```
sudo dpctl add-flow tcp:127.0.0.1:6634 in_port=1,idle_timeout=0,
actions=output:2
```

```
sudo dpctl add-flow tcp:127.0.0.1:6634 in_port=2,idle_timeout=0,
actions=output:1
```

Ze záznamů je vidět, že příkaz dpctl mohu vzdáleně použít na jakékoliv prvky, zde jsem nastavil adresu s1, dále vstupní a výstupní porty a doba vypršení záznamu toku na nekonečno. Záznam se musí nastavit v obou směrech, jak již bylo patrné z předchozího příkladu. Po tomto kroku se již přepínač nemusí dotazovat kontroleru, jelikož má vše ve své tabulce toků. Z obr. 2.12 lze vidět jednotlivé záznamy a jejich vstupní, výstupní rozhraní a další záznamy. Jako hlavní vidím možnost pod záložkou Cmd, a to smazání záznamů, kterou jsem ověřil a skutečně funguje, což snad naznačuje i budoucí grafické přidávání záznamů, které by značně ulehčilo práci.



Obr. 2.12: Grafické rozhraní kontroleru Beacon s přidanými záznamy toků.

Pomocí příkazu dpctl, jde jednotlivé toky směřované stejnou cestou rozlišovat, i podle hodnoty jejich zdrojového, či cílového portu, což by oddělilo jednotlivé služby, dále podle MAC adres, IP adres, VLAN tagování a řadě dalších identifikátorů přenosu.

2.8 Dostupnost zařízení podporujících softwarově definované sítě a možnost jeho využití pro laboratorní úlohy Architektury sítí

V současné době je dostupnost pouze softwarových kontrolerů, které jsou ale příliž složité k používání, které by mělo zjednodušit konfiguraci sítě v této učebně. Podle mého názoru se vyplatí počkat až na dostupnost hardwarových kontrolerů, kde ale bude velkou otázku hrát i cena tohoto prvku. Nejblíže k oficiálnímu vydání kontroleru je společnost HP.

Jedná se o typ **Virtual Application Networks SDN Controller**. Je vybaven plnou podporou protokolu OpenFlow a otevřenou API k třetímu vývoji SDN aplikací. Zařízení je momentálně v uživatelském beta testování a bude uvolněno k prodeji koncem roku 2013, je znázorněno na obr. 2.13.

Protokol OpenFlow podporují i prvky MikroTik, které jsou již součástí laboratoře, ovšem tyto prvky podporují zatím jen verzi 1.0.0. Jak je napsáno na webové

Obr. 2.13: Hardwarový kontroler společnosti HP. [13]

prezentaci výrobce, je tato implementace pomocí balíčku čistě jen experimentální a není určena k provozu. [12]

Společnost HP podporuje protokol u těchto typů výrobků přepínačů: 8200, 5400, 3800, 3500 a 2920. Za zmínku stojí typ HP 3800 Series, který je již v prodeji. Je to Gigabit Ethernetový L3 přepínač. Tento typ přepínačů by zcela vyhovoval učebně. Bylo by potřeba několik těchto prvků, a to minimálně 3, s tím že prvky by musely být propojeny do plného polygonu a nastaveny s informací o přítomnosti kontroleru. Poté by se již veškerá konfigurace řešila skrz hardwarový kontroler. [13]

Tento návrh je ale spíše vizí do budoucna, jelikož zde do učebny byly nedávno pořízeny nové prvky a zatím není kontroler v prodeji.

3 ZÁVĚR

Tato nová síťová architektura, z teoretického hlediska, prozatím převažuje svými výhodami nad nevýhodami. Její funkčnost již prověřila i firma Google, která ji má nasazenou na své WAN síti. Veškeré prvky si sama vytvořila. Od jejich nasazení jim exponenciálně roste propustnost sítí a množství dat procházejících sítí, což zveřejnila ve své prezentaci. Toto ale ovšem není nekonečné, tento růst se zastaví na určitém bodě, kde se znovu může začít vyhledávat další síťová architektura.

Softwarově definované sítě mají jistě dobrou příležitost se uchytit v akademické sféře. Síť je použita na Leland Stanford Junior University a spojená s více univerzitami, které také využívají SDN sítě. Podle mého názoru tato síť má budoucnost i na naší univerzitě, kde by byla zapojená do úplného polygonu s větším počtem kontrolerů, které by řídily své kanály. Každý kontroler by měl být použit dvakrát pro případný výpadek prvního. Kontrolery by řídily vlastní logickou architekturu sítě, kde by různé aplikace mohly určovat různou cestu sítí, a tak například osamostatnit protokol RTP od ostatních. Dále by tato architektura mohla řešit problémy, které vznikají při registraci předmětů, či dalších, kde je síť zatížená obrovským počtem uživatelů.

Nejblíže k zpřístupnění hardwarového kontroleru veřejnosti má společnost HP, ovšem u těchto hardwarových kontrolerů bude hlavní roli hrát cena, která se zřejmě bude pohybovat ve vysokých sumách. Což by ale měly kompenzovat prvky sítě, které by nemusely být tak složité, ale jen uzpůsobeny pro protokol OpenFlow. Další výhodou by měla být centralizace, díky které odpadá potřeba většího množství administrátorů, ale potřebu jen zlomku s velkou znalostí převážně kontroleru.

Pro měření jsem využil softwarový kontroler Beacon od Stanfordské univerzity, který má náznak uživatelského rozhraní, ale hlavní funkce pro přidávání záznamů zde doposud chybí. Tento kontroler bych prozatím nepovažoval jako schopný reálnému provozu, je zde ještě plno záležitostí, které se musí doplnit. Z měření jsem zjistil řadu možností protokolu i kontroleru, jako průběh inicializace spojení *kontroler – přepínač*, tak i analyzování jednotlivých zpráv.

V současné době se ovšem vyplatí počkat, až firmy jako Cisco a Juniper Networks vstoupí na trh se svými prvky a představí vlastní výhody SDN kontroleru.

LITERATURA

[1] SEIFERT, R. a EDWARDS, J. *The all-new switch book: the complete guide to LAN switching technology*. 2nd ed. Indianapolis: Wiley Publishing, 2008, xxxi, 784 s. ISBN 978-0-470-28715-6.

[2] OpenNetworking.org: *Software defined networking: The New Norm for Networks*. Dostupné z URL: <https://www.opennetworking.org/images/stories/downloads/white-papers/wp-sdn-newnorm.pdf>.

[3] LEVIN, D., WUNDSAM, A., HELLER, B., HANDIGOL, N. a FELDMANN, A. *Logically Centralized? State Distribution Tradeoffs in Software Defined Newtorks*. Helsinki: 2012. Dostupné z URL: <http://conferences.sigcomm.org/sigcomm/2012/paper/hotsdn/p1.pdf>.

[4] Cariden: *Infrastrucuter SDN with Cariden Technologies*. Dostupné z URL: <http://www.sdncentral.com/wp-content/uploads/2012/08/Infrastructure_SDN.pdf>.

[5] OpenNetworking.org: *OpenFlow Switch Specification 1.3.1*. Dostupné z URL: <https://www.opennetworking.org/images/stories/downloads/specification/openflow-spec-v1.3.1.pdf>.

[6] OpenFlow.org: *OpenFlow Switch Specification 0.8.9*. Dostupné z URL: <http://www.openflow.org/documents/openflow-spec-v0.8.9.pdf>.

[7] OpenFlow.org: *OpenFlow Switch Specification 0.9.0*. Dostupné z URL: <http://www.openflow.org/documents/openflow-spec-v0.9.0.pdf>.

[8] OpenFlow.org: *OpenFlow Switch Specification 1.0.0*. Dostupné z URL: <http://www.openflow.org/documents/openflow-spec-v1.0.0.pdf>.

[9] OpenFlow.org: *OpenFlow Switch Specification 1.1.0*. Dostupné z URL: <http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf>.

[10] OpenNetworking.org: *OpenFlow Switch Specification 1.2*. Dostupné z URL: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.2.pdf>.

[11] OpenNetworking.org: *OpenFlow Switch Specification 1.3.0*. Dostupné z URL: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf>.

[12] MikroTik: *Manual:OpenFlow* [online]. [cit. 2013-05-28]. Dostupné z URL: <http://wiki.mikrotik.com/wiki/Manual:OpenFlow>.

[13] Hewlett-Packard: *Virtual Application Networks*. Dostupné z URL: <http://h17007.www1.hp.com/docs/interopny/4AA4-3881ENW.PDF>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ACL Acces Control List

API Application Programming Interface

ARP Address Resolution Protocol

CFI Canonical Format Identifier

CIDR Classless Inter-Domain Routing

CSM Controller-to-Switch Message

DiffServ Differentiated services

DSCP Differentiated services code point

FCS Frame Check Sequence

GNV Global Network View

HP Hewlett-Packard

HTTP Hypertext Transfer Protocol

I/O input/output

IBM International Business Machines Corporation

ICMP Internet Control Message Protocol

ID Identifier

IP Internet Protocol

IPv6 Internet Protocol version 6

ISO/OSI International Organization for Standardization/Open Systems Interconnection

IT Information Technology

LAN Local Area Network

L3 Layer 3

MAC Media Acces Control

MPLS Multiprotocol Label Switching

NS-OS Network Service-Operating system

ONF Open Networking Foundation

QoS Quality of Service

RTP Real-time Transport Protocol

SDN Software Defined Networking

SM Symmetric Message

STP Spanning Tree Protocol

TCP Transmission Control Protocol

TLS Transport Layer Security

ToS Terms of service

TTL Time to live

UDP User Datagram Protocol

URL Uniform Resource Locator

UTP Unshielded Twisted Pair

VLAN Virtual Local Area Network

VM Virtual machine

VPN Virtual private network

WAN Wide Area Network