

UNIVERZITA HRADEC KRÁLOVÉ  
FAKULTA INFORMATIKY A MANAGEMENTU  
KATEDRA INFORMATIKY A KVANTITATIVNÍCH METOD

## Analýza protokolu OpenFlow v LAN

### Bakalářská práce

**Autor:** Radek Vašátko

**Studijní obor:** Informační management

**Vedoucí práce:** Ing. Jakub Pavlík, MSc.

Hradec Králové

duben 2015

## **Prohlášení**

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 27. dubna 2015

Radek Vašátko

## **Poděkování**

Rád bych zde poděkoval Ing. Jakubovi Pavlíkovi, MSc. za odborné vedení práce, podnětné rady a čas, který mi věnoval.

## **Anotace**

Cílem této bakalářské práce je analyzovat protokol OpenFlow, typy zpráv vyměňované mezi controllerem a OpenFlow switchem a porovnat funkce dvou vybraných SDN controllerů. Práce je zaměřena na novou síťovou technologii softwarově definovaných sítí a protokol OpenFlow. Porovnává tradiční řešení počítačových sítí a SDN. Dále popisuje základní principy této nové technologie, architekturu, možnosti nasazení a controller, který je centrálním bodem sítě. Následně je rozebrán protokol OpenFlow, jeho podstata, architektura a typy zpráv vyměňované mezi controllerem a switchem podporující protokol OpenFlow. V praktické části je analyzován OpenFlow protokol při prvotní komunikaci controlleru se switchem a otestovány funkce a možnosti dvou zvolených controllerů (OpenDaylight a HP VAN SDN).

## **Annotation**

### **Analysis of OpenFlow Protocol in Local Area Networks**

The aim of this bachelor thesis is to analyze the protocol OpenFlow, types of messages exchanged between the controller and OpenFlow switch and also compare the features of two selected SDN controllers. The work is focused on new networking technology Software defined networking and OpenFlow protocol. It compares traditional solutions of computer networking and SDN. It describes the basic principles of the new technology, architecture, deployment options and controller, which is the central point of the network. Subsequently, OpenFlow protocol is analyzed and also its essence, architecture and kind of messages which are exchanged between the controller and OpenFlow switch. In practical part, there is protocol OpenFlow analyzed for the initial communication the switch with the controller and after that the features of two selected controllers (OpenDaylight and HP VAN SDN) are tested.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Principy softwarově definovaných sítí</b>	<b>2</b>
2.1	Úvod . . . . .	2
2.2	Tradiční řešení počítačových sítí . . . . .	3
2.2.1	Switch . . . . .	4
2.2.2	Router . . . . .	4
2.3	Základy softwarově definovaných sítí . . . . .	4
2.3.1	Control Plane . . . . .	5
2.3.2	Data Plane . . . . .	5
2.4	Architektura SDN . . . . .	6
2.5	Možnosti nasazení SDN . . . . .	7
2.5.1	Centralizovaný model SDN . . . . .	8
2.5.2	Distribuovaný model SDN . . . . .	8
2.5.3	Hybridní model SDN . . . . .	9
2.6	SDN Controller . . . . .	10
<b>3</b>	<b>OpenFlow</b>	<b>11</b>
3.1	Úvod . . . . .	11
3.2	Architektura protokolu OpenFlow . . . . .	12
3.3	OpenFlow switch . . . . .	12
3.4	Typy zpráv protokolu OpenFlow . . . . .	13
3.4.1	Controller-to-switch zprávy . . . . .	13
3.4.2	Asynchronní zprávy . . . . .	14
3.4.3	Symetrické zprávy . . . . .	15
3.5	Navázání spojení mezi switchem a controllerem . . . . .	15
<b>4</b>	<b>OpenFlow controllery pro SDN</b>	<b>17</b>

---

4.1	Seznam OpenFlow controllerů . . . . .	17
4.2	OpenDaylight controller . . . . .	19
4.2.1	Hydrogen . . . . .	19
4.2.2	Grafické prostředí . . . . .	21
4.3	HP VAN SDN controller . . . . .	21
4.3.1	Grafické prostředí . . . . .	23
<b>5</b>	<b>Test controllerů</b>	<b>25</b>
5.1	Porovnání OpenDaylight a HP VAN SDN controllerů . . . . .	25
5.1.1	Identické funkce a vlastnosti . . . . .	25
5.1.2	Specifické funkce OpenDaylight controlleru . . . . .	26
5.1.3	Specifické funkce HP VAN SDN controlleru . . . . .	26
5.2	Mininet . . . . .	26
5.3	Instalace . . . . .	28
5.3.1	OpenDaylight controller . . . . .	28
5.3.2	HP VAN SDN controller . . . . .	28
5.3.3	Mininet . . . . .	29
5.4	Test 1 . . . . .	31
5.4.1	Vytvoření topologie v Mininet . . . . .	31
5.4.2	Připojení ke controlleru . . . . .	32
5.4.3	Výsledky . . . . .	34
5.5	Test 2 . . . . .	34
5.5.1	Vlastní topologie . . . . .	35
5.5.2	Testování . . . . .	36
5.5.3	Výsledky . . . . .	42
<b>6</b>	<b>Závěr</b>	<b>43</b>
	<b>Literatura</b>	<b>47</b>

# Seznam obrázků

2.1	Architektura softwarově definovaných sítí (převzato z [6]) . . . . .	7
2.2	Centralizovaný model SDN (převzato z [7]) . . . . .	8
2.3	Distribuovaný model SDN (převzato z [7]) . . . . .	9
2.4	Hybridní model SDN (převzato z [7]) . . . . .	9
3.1	Architektura protokolu OpenFlow (převzato z [1]) . . . . .	12
3.2	Komunikace mezi controllerem a OpenFlow switchem (převzato z [9]) . . . .	13
3.3	Navázání spojení mezi OpenFlow controllerem a OpenFlow switchem (převzato z [4]) . . . . .	16
4.1	OpenDaylight Base Edition - architektura [15] . . . . .	20
4.2	OpenDaylight - grafické uživatelské prostředí . . . . .	21
4.3	Architektura HP VAN SDN controlleru[14] . . . . .	22
4.4	Grafické uživatelské prostředí HP VAN SDN controlleru . . . . .	24
5.1	Příklad sítě vytvořené v prostředí Mininet (převzato z [1]). . . . .	27
5.2	Ověření instalace HP VAN SDN controlleru . . . . .	29
5.3	Mininet - zobrazení všech dostupných verzí . . . . .	30
5.4	Mininet - ověření funkčnosti . . . . .	30
5.5	Mininet - definované topologie vyobrazené v OpenDaylight controlleru . . . .	32
5.6	Síťová topologie v Mininet pro test 1 . . . . .	32
5.7	Zahájení komunikace mezi OpenDaylight controllerem a OpenFlow switchem	33
5.8	Zahájení komunikace mezi HP VAN SDN controllerem a OpenFlow switchem	34
5.9	MiniEdit - grafické uživatelské rozhraní . . . . .	35
5.10	MiniEdit - nastavení controlleru . . . . .	36
5.11	MiniEdit - všeobecné nastavení SDN sítě . . . . .	36
5.12	Mesh topologie vytvořená v MiniEdit . . . . .	37
5.13	Příkaz pingall v Mininet ověřující dostupnost všech koncových stanic . . . . .	37

5.14	OpenDaylight - editace nastavení switche . . . . .	38
5.15	OpenDaylight - upravená mesh topologie . . . . .	38
5.16	Akce, které umožňuje OpenDaylight controller provádět na portu switche . .	39
5.17	OpenDaylight - ověření nedostupnosti počítače h1 v síti . . . . .	39
5.18	HP VAN SDN - nalezení nejkratší cesty v síti . . . . .	40
5.19	HP VAN SDN - nalezení nejkratší cesty v síti, po odpojení jedné z linek . . . .	41
5.20	OpenFlow Trace - nástroj na zachytávání OpenFlow komunikace . . . . .	41



# 1 Úvod

V tradičních sítích se nacházejí jednotlivé síťové prvky, jako jsou routery a switche a různé protokoly určující optimální cestu od zdroje k cíli na základě IP adresy obsažené v hlavičce paketu. Ale s nárůstem velikostí sítí je již současné řešení nedostačující a neúčinné. Roste tedy poptávka po efektivních, snadno rozšiřitelných a konfigurovatelných počítačových sítích. Právě tyto požadavky přesně splňuje nová technologie nazvaná softwarově definované sítě – SDN.

SDN je nový přístup k budování počítačových sítí, který se vyznačuje především programovatelností celé sítě, a to díky oddělení řídicí (control plane) a datové (data plane) úrovně, což poskytuje nezávislou a centralizovanou správu dané sítě.

Důležitou součástí SDN je OpenFlow, což je první standardizovaný protokol umožňující programovatelné řízení toků a definování síťových cest bez ohledu na topologii sítě. Ve své podstatě, OpenFlow slouží ke komunikaci mezi controllerem a switchem nebo routerem. V poslední době získal protokol OpenFlow značný zájem mezi výrobci a vývojáři síťových prvků a serverů.

Bakalářská práce popisuje tradiční sítě, principy nové architektury SDN a protokol OpenFlow. Část o SDN je zaměřena na architekturu, možnosti nasazení a na centrální bod této technologie, kterým je controller. V další kapitole je vysvětlena podstata OpenFlow protokolu, jeho architektura a typy zpráv, které putují mezi OpenFlow switchem a controllerem. Cílem této bakalářské práce je analýza OpenFlow paketů, které putují mezi controllerem a OpenFlow switchem a porovnání funkcí dvou zvolených SDN controllerů. Praktická část práce je zaměřena na instalaci vybraných controllerů a síťového emulátoru Mininet, který umožňuje vytvořit SDN topologii sítě. Dále následuje analýza OpenFlow zpráv při začátku komunikace mezi switchem a controllerem. Nakonec jsou otestovány vybrané controllery a funkce, které nabízejí.

# 2 Principy softwarově definovaných sítí

V následující kapitole jsou popsány tradiční síťové technologie a základní principy SDN. Je zde rozebrána architektura SDN, způsoby nasazení a centrální bod SDN, kterým je controller.

## 2.1 Úvod

Nárůst mobilních zařízení, virtualizace serverů a rozšíření cloudových služeb donutil vedoucí firmy v oblasti počítačových sítí znovu prozkoumat, jak učinit sítě flexibilnější a lépe ovladatelné. Tradiční sítě zkonstruované s pevně danými zařízeními, nejčastěji složené do stromových struktur, již nemohou udržet zvyšující se nápor provozní zátěže. Statická struktura nevyhovuje potřebám dnešních sítí, které mají velmi vysoké nároky na velikost paměťového prostoru a především na výpočetní výkon.

V softwarově definovaných sítích je nový přístup k vytváření sítí, ve kterém jsou hardwarové síťové prvky odděleny od řídicích funkcí. To slouží k zjednodušení síťového managementu, inovacím v oblasti tvorby sítí a k dalšímu rozvoji síťových technologií. Základní myšlenka SDN je taková, aby vývojáři softwaru mohli používat stejný přístup k síťovým zdrojům, jako již uplatňují k počítačovým nebo paměťovým zdrojům. V SDN je inteligence sítě soustředěna do softwarově orientovaných řídicích jednotek, zatímco z hardwarových síťových prvků se stávají zařízení, která mají za úkol jen přijímat datové pakety a odesílat je správným směrem, což je naprogramováno přes otevřené rozhraní.

Vývojem SDN se zabývá nezisková organizace ONF (Open Networking Foundation), která byla vytvořena skupinou operátorů a provozovatelů služeb počítačových sítí. ONF je organizace orientovaná na uživatele, zaměřuje se na propagaci a přijetí SDN prostřednictvím rozvoje otevřených standardů. Organizace byla založena v roce 2011 firmami jako je například Facebook, Google, Microsoft, Verizon nebo Yahoo!. Řídí ji zástupci z akademické

i podnikové sféry vyspělých zemí. Jejich hlavním cílem je podpora, prosazování a praktické rozšiřování SDN, a také standardizace základního protokolu OpenFlow. [1]

## 2.2 Tradiční řešení počítačových sítí

Počítačové sítě se rozdělují podle velikosti a funkce na dvě hlavní skupiny – LAN (Local Area Networks) a WAN (Wide Area Networks).

Místní síť LAN je základem každé počítačové sítě. Architektura sítě LAN může být jednoduchá (spojení dvou počítačů) nebo složitá (stovky propojených počítačů a periferních zařízení v jedné oblasti). Rozlišující vlastností sítě LAN je její omezenost jen na určitou geografickou oblast, jako je jedna budova nebo oddělení.

Síť WAN nemá žádné geografické omezení. Skládá se z více vzájemně propojených sítí LAN. Jejich spojování se provádí speciálními linkami či bezdrátově. Rozlehlost sítí může být různá, od sítí městských či firemních (firma s pobočkami ve více městech, zemích či kontinentech) až po nejznámější celosvětovou síť – Internet. [2]

Hlavním účelem sítí je jejich vzájemné propojení, a tak se musela stanovit pravidla pro přenos dat v sítích a mezi nimi. Mezinárodní ústav pro normalizaci ISO (International Standards Organization) vypracoval tzv. referenční model OSI (Open Systems Interconnection), který rozdělil práci v síti do sedmi vzájemně spolupracujících vrstev.

Model ISO/OSI rozděluje síťovou práci do jednotlivých vrstev. Princip spočívá v tom, že vyšší vrstva převezme úkol od podřízené vrstvy, zpracuje jej a předá vrstvě nadřazené. Model ISO/OSI určuje, jak mají vrstvy spolupracovat horizontálně – stejné vrstvy modelu mezi různými sítěmi musejí spolupracovat.

Dále model ISO/OSI utváří představu o tom, co vše je nutné zajistit pro úspěšnou komunikaci v síti. Bezprostřední komunikaci zajišťují první tři vrstvy modelu ISO/OSI – fyzická, linková a síťová, které vykonávají poměrně složité úkoly. Část úkolů je integrovaná v elektronice síťové karty a data následně přenesou kabel. Výběr vyhovující trasy, kontrola správnosti paketů, rozhodnutí, do které sítě může paket projít a kam ne či mnoho dalších úkolů provádějí aktivní prvky v síti. Mezi nejznámější prvky, které aktivně ovlivňují dění v síti, patří switch a router. [3]

V tradičním řešení počítačových sítí, proprietární firmware routerů a switchů říká síťovým zařízením, kam a jak směřovat a přeposílat pakety. Každé zařízení v síti má své vlastní využití, například router používá routovací protokol (OSPF, EIGRP, IS-IS a další).

## 2.2.1 Switch

Switch neboli přepínač je aktivní prvek – to znamená, že obnovuje a znovu odesílá pakety. Přepínač pracuje na druhé vrstvě (datová vrstva) referenčního modelu ISO/OSI a předává data z jednoho uzlu do jiného uzlu, který je připojený ke stejnému nebo jinému přepínači v dané síti. V podstatě vytváří virtuální okruh mezi aktuálně komunikujícími stanicemi, neboli odděluje komunikující stanice od zbytku sítě. Spojení jednotlivých přepínačů probíhá na základě MAC adres. MAC adresa je hardwarová adresa zařízení, které je jedinečné a je nastavena výrobcem daného zařízení. Každý switch má svoji MAC tabulku, kde jsou uloženy MAC adresy přímo připojených zařízení a na jejich základě se switch rozhoduje, kam paket pošle.

Vzhledem k tomu, že switch pracuje na druhé vrstvě modelu ISO/OSI, tak pakety určené pro jinou síť nemohou být zpracovány a jsou zahozeny. [2]

## 2.2.2 Router

Router neboli směrovač pracuje na třetí vrstvě (síťová vrstva) modelu ISO/OSI a spojuje síť LAN a WAN, aby mezi nimi mohla probíhat komunikace. Provádí výpočetní úkoly, které zahrnují zjištění optimální cesty od zdroje k cíli a její řízení.

Každý router si udržuje svou vlastní směrovací tabulku, která se obvykle skládá z cílových adres. Pro stanovení cílové adresy pro příchozí data obsahuje směrovací tabulka všechny známé síťové adresy, logické instrukce pro připojení k jiným sítím, cesty mezi routery a také náklady na posílání dat prostřednictvím dané cesty. Router použije svou směrovací tabulku k výběru nejlepší trasy pro přenos dat na základě nákladů a dostupných cest.

Rozhodnutí o nejvýhodnější cestě je provedeno pomocí směrovacích protokolů, které jsou statické a dynamické. Při použití statického směrování musí být všechny směrovače nakonfigurovány ručně správcem sítě. Směrovací tabulky jsou pevně dané, tudíž je vždy použita stejná cesta, ale není zaručeno, zda se jedná o nejkratší cestu. Naproti tomu dynamické směrování musí být nejprve nakonfigurováno, avšak umožňuje přizpůsobení se měnícím podmínkám v síti a nalezení alternativních tras, pokud například jeden směrovač nefunguje, data mohou být odeslána jinou cestou. [2]

## 2.3 Základy softwarově definovaných sítí

S prudkým růstem datového provozu v posledních několika letech se tradiční síťové architektury staly nedostatečně pružnými, pro splnění požadavků uživatelů a především

firem. Počítačové sítě se typicky skládají z koncových zařízení různého druhu a počítačů připojených k uzlům síťové infrastruktury, které umožňují jejich vzájemné propojení. Tradiční routery a switche mají uzavřené systémy s omezenými, v závislosti na výrobci specifickými rozhraními, to znamená, že rozhodnutí o průchodu dat v síti se musí vykonávat v modulech jednotlivých síťových prvků. V takovém prostředí není jednoduché nasazení nové síťové aplikace nebo nového prvku, a je nutné, aby změna byla realizována v celé infrastruktuře. [4]

Přeměnu síťové architektury nabízejí softwarově definované sítě - SDN. Jejich smyslem je usnadnit inovace a programově jednoduché řízení cest v síti.

Jedním z principů stojících na počátku vzniku SDN je potenciální výhoda v oddělení řídicí (Control Plane) a datové (Data Plane) úrovně. Softwarově definované sítě odstraňují control plane ze síťového hardwaru a zavádí jej do softwaru controlleru SDN, což umožňuje programovatelnost a flexibilnější správu sítě. Pokud bude nutné v síti udělat změny, je možné tak učinit z jednoho centrálního místa bez nutnosti aktualizovat každý prvek sítě zvlášť. [1]

### 2.3.1 Control Plane

Udržuje lokální informace sloužící k vytváření položek v tabulce pro přeposílání dat (forwarding table), které používá data plane na přeposílání informací mezi vstupními a výstupními porty na zařízeních. Informace k vytváření a ukládání topologie sítě se nazývá směrovací informační báze (routing information base – RIB). RIB zůstává konzistentní a to díky časté výměně informací mezi ostatními instancemi control plane v rámci sítě.

Záznamy v tabulce přeposílání se nazývají přeposílající informační báze (forwarding information base – FIB) a jsou často zrcadlem mezi control plane a data plane v klasickém zařízení. FIB je naprogramován, jakmile je RIB konzistentní a stabilní. K provedení tohoto úkolu musí řídicí jednotka nebo program rozvíjet pohled na topologii sítě, která splňuje určitá omezení. Tento pohled na síť lze naprogramovat ručně, naučit pozorováním nebo postavit z kousků informací získaných od ostatních instancí v control plane. [1]

### 2.3.2 Data Plane

Zpracovává příchozí datagramy (ze síťového kabelu, optického vlákna nebo bezdrátové technologie) prostřednictvím řady operací, které shromažďují datagramy a vykonávají základní kontrolu proti chybám.

Správné, dobře formulované datagramy jsou zpracovány v data plane a provádějí vyhledávání v tabulce FIB, která byla naprogramována dříve v control plane. Tento způsob se někdy označuje jako rychlé cesty pro zpracování paketů, protože se jen identifikuje cíl

paketu pomocí naprogramované FIB a další dotazování není nutné. Jedinou výjimkou tohoto zpracování je, když paket nemůže najít shodu s pravidly, například detekování neznámého cíle. Tyto pakety jsou poslány do routeru, kde je může control plane dále zpracovat pomocí RIB. [1]

## 2.4 Architektura SDN

SDN je síťová architektura, ve které je řízení sítě odděleno od přeposílání a je přímo programovatelná. Síťová inteligence je logicky centralizována a vlastní síťová architektura je více obecná. Oddělením řídicí složky je možné značné snížení nákladů. Jestliže se objeví nový protokol, není nutné měnit celou infrastrukturu, ale pouze učinit příslušné změny v řídicí složce.

SDN architektura je:

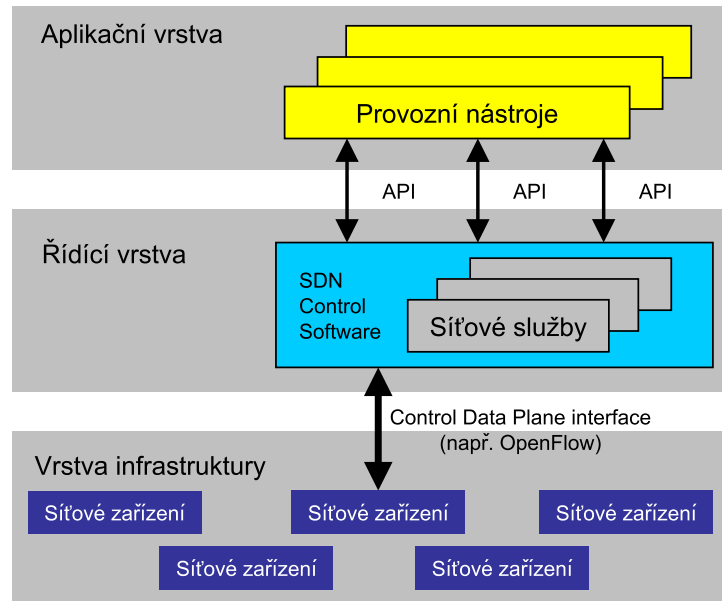
- Přímě programovatelná: kontrola sítě je přímě programovatelná, protože je oddělena od funkce přeposílání
- Inteligentní: umožňuje snadné změny v síti
- Centrálně řízená: síťová inteligence je logicky soustředěna do SDN controlleru, který udržuje globální pohled na síť
- Programově konfigurovatelná: SDN umožňuje správcům sítě konfigurovat, spravovat, zabezpečit a optimalizovat síťové prostředky velmi rychle pomocí automatizovaných programů SDN, které mohou psát samy, jelikož nejsou závislé na proprietárním softwaru [5]

Architektura SDN se skládá ze tří vrstev: infrastruktury, řídicí a aplikační. Vrstva infrastruktury a řídicí vrstva jsou spojeny přes řídicí rozhraní, jako je například OpenFlow, zatímco aplikační vrstva je připojena na řídicí vrstvu pomocí několika rozhraní pro programování aplikací (API). [6]

- Vrstva infrastruktury je základem každé sítě, zahrnuje veškeré vybavení (hardware), které umožňuje přeposílání (forwarding) dat.
- Řídicí vrstva je centrální část SDN. Je klíčovým prvkem, který odlišuje SDN od tradičních síťových technologií. Ústřední řídicí bod poskytuje širší pohled na celou síť a sledování detailů v síti. Umožňuje zabezpečení, správu, flexibilitu konfigurace a optima-

lizaci síťových zdrojů pomocí dynamických a automatizovaných programů pro celou síť z jednoho centrálního místa.

- Aplikační vrstva je nejinteligentnější vrstvou SDN sítě. Skládá se z různých provozních nástrojů sítě a uživatelských rozhraní, které umožňují skutečné řízení sítě prostřednictvím řídicí vrstvy. [6]



Obrázek 2.1: Architektura softwarově definovaných sítí (převzato z [6])

## 2.5 Možnosti nasazení SDN

Hlavní důvody pro nasazení technologie SDN jsou:

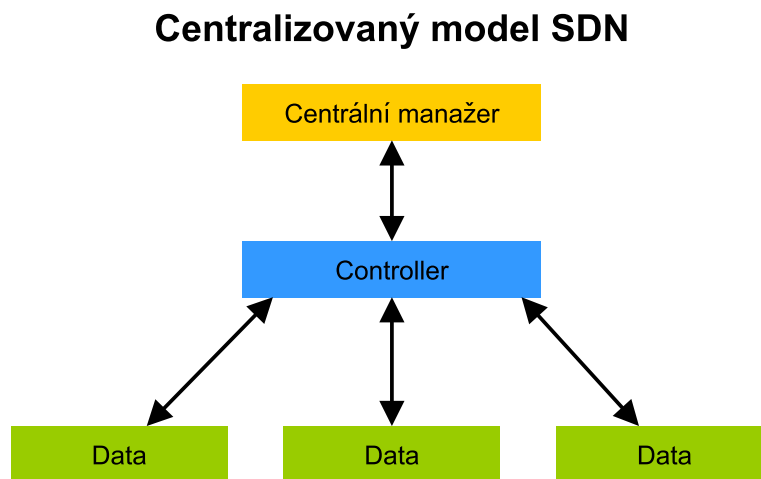
- Zjednodušené řízení sítě
- Snížení provozních nákladů
- Rychlé a efektivní zavádění nových služeb

Existují tři přístupy k nasazení SDN - centralizovaný, distribuovaný a hybridní model. První dva jmenované se liší především v ceně a složitosti implementace, zatímco třetí je jejich vzájemnou kombinací. [7]

## 2.5.1 Centralizovaný model SDN

V centralizovaném modelu je centrální manažer s jedním controllerem, který komunikuje s distribuovanou data plane. Cesty a toky dat jsou definované v rámci centrálního manažera. Centralizovaný přístup zjednodušuje řízení složitých toků související s konkrétní aplikací.

Nevýhodou tohoto přístupu je škálování, zejména v internetovém nebo cloudovém prostředí. Centralizované řízení vyžaduje větší a dražší systémy s výkonnějšími procesory, větší datové úložiště a odolnou infrastrukturu s nízkou tolerancí poruchy. Centralizovaný model tedy může být nepraktický, velmi nákladný a je méně odolný proti poruchám. [7]



Obrázek 2.2: Centralizovaný model SDN (převzato z [7])

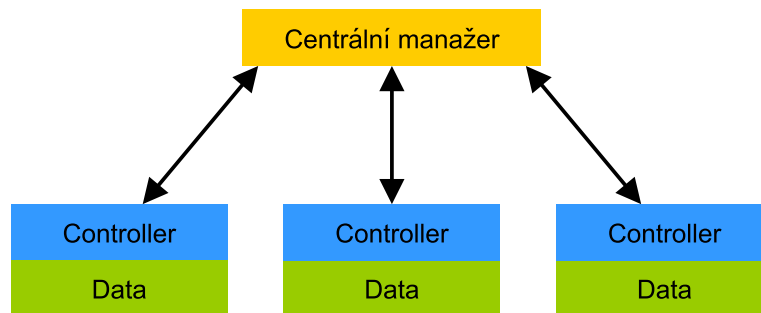
## 2.5.2 Distribuovaný model SDN

V distribuovaném modelu, centrální manažer komunikuje s kombinovaným zařízením zahrnujícím distribuovaný controller a data plane.

Z hlediska implementace je nasazení distribuované sítě mnohem složitější. Nelze se dostat do všech uzlů v síti a vytvořit tak stejnou cestu s totožnými prioritami, tudíž jsou zde synchronizační a časové problémy. Výhodou je, že distribuované SDN sítě jsou snadno škálovatelné a rozšiřitelné, než tradiční sítě. [7]



## Distribuovaný model SDN



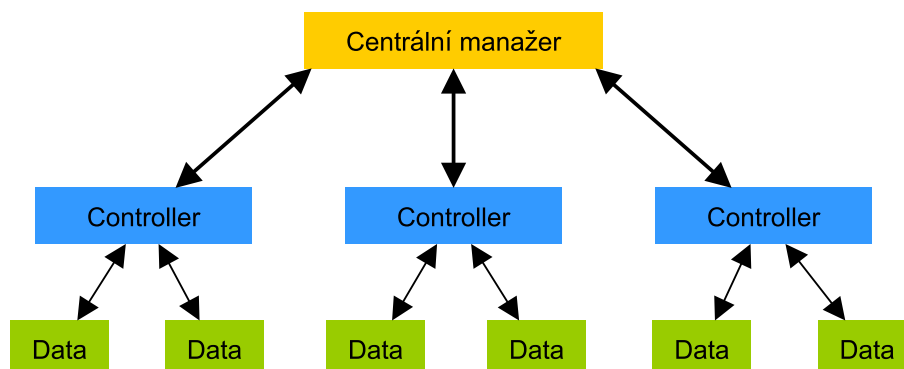
Obrázek 2.3: Distribuovaný model SDN (převzato z [7])

### 2.5.3 Hybridní model SDN

V hybridním modelu, centrální manažer komunikuje s distribuovanými controllery a ty komunikují s distribuovanou data plane. Je tedy kombinací centralizovaného a distribuovaného modelu SDN, proto nabízí to nejlepší z obou.

Model využívá výhody jednoduchého ovládání a řízení konkrétních datových toků definovaného v centralizovaném modelu s využitím škálovatelnosti a odolnosti z modelu distribuovaného. [7]

## Hybridní model SDN



Obrázek 2.4: Hybridní model SDN (převzato z [7])

## 2.6 SDN Controller

Controller v softwarově definované síti je „mozkem“ celé sítě. Jedná se o strategický řídicí bod, který předává informace routerům a switchům (přes southbound API) a aplikace a logiku (přes northbound API). Veškerá komunikace mezi zařízením a aplikací musí projít přes controller. Dále také využívá protokol jako je OpenFlow na konfiguraci síťových zařízení a na výběr optimálních cest v síťovém provozu. Controller tedy funguje jako určitý druh operačního systému pro počítačovou síť. Tím, že odděluje control plane od hardwaru sítě a funguje jako software, umožňuje automatickou správu sítě a také usnadňuje integraci a správu aplikací.

SDN controller obvykle obsahuje kolekci zásuvných modulů, které mohou provádět různé úkoly v síti. Některé z úkolů jsou například - jaká zařízení jsou v síti a jaké poskytují funkce, shromažďování statistik sítě a další. Zásuvné moduly mohou být rozšířeny o nové, které zvýší funkčnost a umožní jiné pokročilé funkce. [1]

Controller je tedy softwarový systém nebo soubor systémů, který dohromady poskytuje:

- Management stavu sítě
- Moderní rozhraní pro programování aplikací (API)
- Bezpečné a centralizované ovládání
- Standardizovaný protokol
- Náhled na topologii sítě a systém výběru nejkratší cesty
- A další informační služby o síti [1]

## 3 OpenFlow

V této kapitole je popsán protokol OpenFlow a jeho architektura. Dále je vysvětleno, co je to OpenFlow switch a jaké druhy zpráv jsou zasílány při komunikaci mezi controllerem a OpenFlow switchem.

### 3.1 Úvod

OpenFlow je první, nejrozšířenější a v současné době i jediný standardizovaný SDN protokol, který umožňuje oddělení control plane a data plane. Původně byl představen a realizován jako část výzkumnické sítě na Stanford University, nyní je standardizován organizací ONF. Prvotním záměrem bylo vytvoření experimentálního protokolu v rámci univerzitní sítě, který by mohl být použit pro výzkum a různé testy.

Otevřený protokol definuje, jakým způsobem může být control plane nakonfigurován a řízen z centrálního místa (controller). OpenFlow umožňuje controlleru řídit pakety, které se přeposílají v rámci sítě. [1]

OpenFlow je sada protokolů a API, které potřebuje OpenFlow controller a OpenFlow switch k správné činnosti v SDN síti. Jinak řečeno, controller nemá žádné funkce bez aplikačního programu, který mu dává instrukce, jaké toky posílat dál a na které prvky v síti.

Pomocí protokolu OpenFlow může controller přidávat, aktualizovat a mazat záznamy v tabulkách toků. [1]

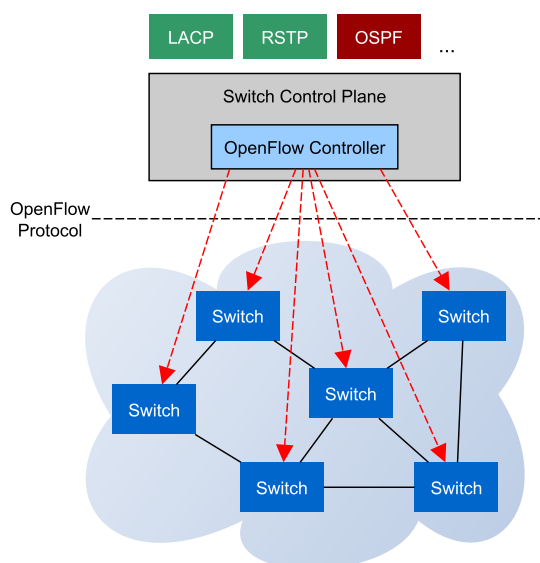
Klíčové komponenty modelu OpenFlow jsou poměrně stejné jako u společné definice SDN, a to zejména:

- Oddělení control a data plane. Control plane je řízena logicky centralizovaným systémem.
- Použití standardizovaného protokolu mezi controllerem a agentem na síťovém prvku.
- Zajištění programovatelnosti sítě z centralizovaného místa pomocí moderní a rozšiřitelné API. [1]

## 3.2 Architektura protokolu OpenFlow

Síťová architektura protokolu OpenFlow se skládá ze tří základních pojmů:

- Síť tvoří jeden nebo více virtuálních či fyzických OpenFlow switchů, které představují data plane.
- Control plane se skládá z jednoho, dvou nebo v některých případech i více OpenFlow controllerů.
- Switche jsou spojeny s control plane pomocí bezpečného ovládacího kanálu. [8]



Obrázek 3.1: Architektura protokolu OpenFlow (převzato z [1])

## 3.3 OpenFlow switch

OpenFlow switch je hardwarové zařízení nebo softwarový program, který přeposílá pakety v prostředí softwarově definovaných sítích pomocí protokolu OpenFlow.

OpenFlow switche dělíme na dvě kategorie:

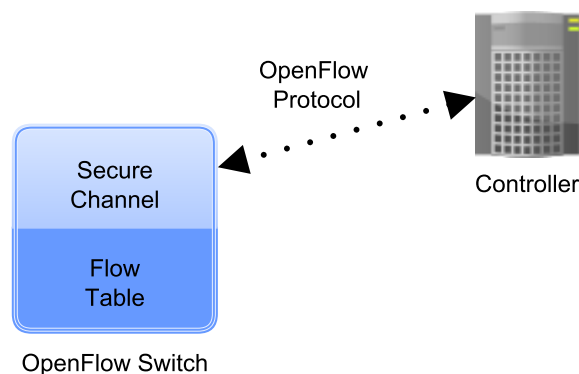
1. Čistý OpenFlow switch podporující pouze OpenFlow protokol.
  2. Hybridní OpenFlow switch, který podporuje tradiční ethernetové a síťové protokoly.
- [4]

Switch podporující protokol OpenFlow se skládá z jedné nebo více tabulek toků a skupinové tabulky, které vykonávají vyhledávání a přeposílání. Switch komunikuje s controllerem

a ten řídí switch pomocí protokolu OpenFlow. Připojení je provedeno přes zabezpečený kanál, pomocí kterého je možné vyměňovat OpenFlow zprávy mezi switchem a controllerem k realizování úkolů konfigurace a řízení. [9]

OpenFlow switch může být naprogramován tak, aby:

- identifikoval a kategorizoval pakety přicházející na port na základě několika různých hlaviček paketu,
- zpracoval paket různými způsoby, včetně úpravy hlavičky,
- zahodil nebo posunul paket na určitý výstupní port nebo na OpenFlow controller. [10]



**Obrázek 3.2:** Komunikace mezi controllerem a OpenFlow switchem (převzato z [9])

## 3.4 Typy zpráv protokolu OpenFlow

OpenFlow podporuje tři různé typy zpráv, a to controller-to-switch, asynchronní a symetrické. Zprávy controller-to-switch se používají pro řízení a kontrolu stavu switche. Asynchronní zprávy zahajuje switch a slouží k aktualizaci síťových událostí v controlleru a změnám stavu dalších switchů v síti. A poslední, symetrické zprávy mohou být inicializovaný jak controllerem, tak i switchem a jsou zasílány bez nároku na jejich odeslání. [10]

### 3.4.1 Controller-to-switch zprávy

Komunikaci začíná controller, který může nebo nemusí žádat odpověď od switche. Controller může požádat switch o zaslání jeho schopností. Switch musí na tento požadavek reagovat a odeslat zprávu, která určuje identitu a základní schopnosti. Toto obvykle probíhá při vytváření OpenFlow kanálu. [10]

Druhy controller-to-switch zpráv:

- **Modify-State:** zprávy jsou zasílány controllerem a umožňují spravovat stav switche. Hlavním účelem je přidávání, upravování a mazání záznamů v OpenFlow tabulkách a nastavení vlastností portů switche.
- **Read-State:** controller používá tyto zprávy ke sbírání informací ze switche jako jsou aktuální konfigurace, statistiky a schopnosti.
- **Packet-out:** využívá controller k posílání paketů směrem ven, ze specifického portu na switchi. Směr určuje zpráva Packet-in. Packet-out zpráva musí obsahovat úplný paket nebo vyrovnávací ID, které odkazuje na paket uložený ve switchi. Zpráva také musí zahrnovat seznam akcí, které budou použity v pořadí, v jakém jsou specifikovány. Jestliže je seznam akcí prázdný, paket bude zahozen.
- **Barrier:** zprávy mohou mít formu žádosti nebo odpovědi. Jsou důležité pro přijímání oznámení o dokončených operacích.
- **Role-Request:** controller pomocí těchto zpráv nastavuje role svého OpenFlow kanálu nebo se dotazuje na roli. Toto je velmi užitečné při připojení více controllerů.
- **Asynchronous-configuration:** umožňuje controller nastavit další filtr na asynchronní zprávy, které chce přijímat na svém OpenFlow kanálu, nebo se na filtr dotázat. [10]

### 3.4.2 Asynchronní zprávy

Asynchronní zprávy odesílá switch do controlleru a označují změnu stavu switche nebo přicházející pakety. Rozlišujeme tři hlavní typy zpráv:

- **Packet-in:** přenáší kontrolu nad paketem controlleru. Pro všechny pakety, které jsou přeoslány controlleru je rezervován port v záznamech toků. Ostatní informace, jako je například kontrola TTL (Time To Live), může také vytvořit packet-in zpráva a odeslat je do controlleru.
- **Flow-Removed:** informuje controller o odstranění záznamu z tabulky toků. Tyto zprávy jsou vytvořeny jako výsledek smazání záznamu toku controllerem nebo pokud na switchi vyprší platnost časového limitu nějakého toku.
- **Port-status:** oznamuje controlleru, že proběhla změna na portu. Tato událost zahrnuje změny v konfiguraci portů, například když byl port změněn do stavu „Down“ přímo uživatelem, anebo sama linka převede svůj stav na „Down“, změní se stav portu. [10]

### 3.4.3 Symetrické zprávy

Symetrické zprávy jsou posílány bez žádosti na jejich odeslání v obou směrech. Existují čtyři typy symetrických zpráv:

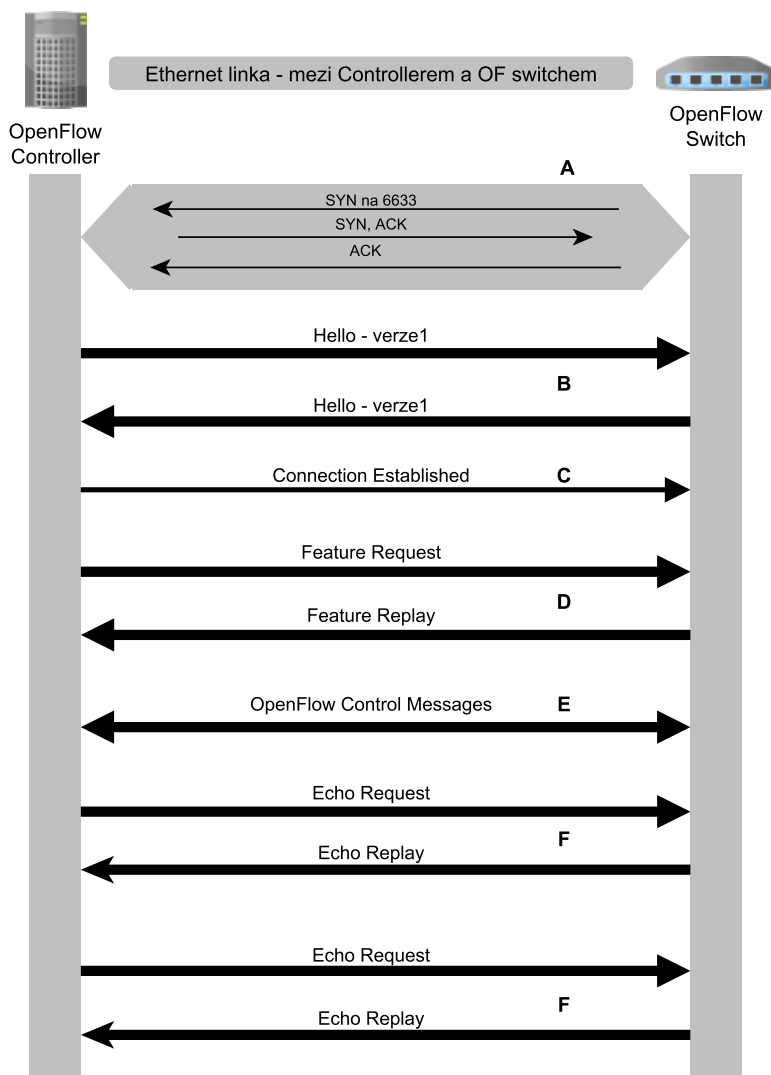
- Hello: zprávy jsou vyměňovány mezi controllerem a switchem při vzniku jejich spojení.
- Echo: zprávy mají formu žádosti nebo odpovědi a mohou být posílány controllerem i switchem. Musí vrátit odpověď „Echo replay“. Používají se zejména k ověření funkčnosti spojení controller – switch a mohou být využity pro měření šířky pásma a zpoždění (latency).
- Error: chybové zprávy jsou užívány pro oznámení výskytu problému. Většinou je používá switch k indikaci selhání žádosti, kterou požadoval controller.
- Experimenter: poskytuje standardní způsob pro OpenFlow switche, kterým nabízí možnost dalších OpenFlow zpráv. Představuje prostor pro nové funkce v budoucích verzích protokolu OpenFlow. [10]

## 3.5 Navázání spojení mezi switchem a controllerem

Protokol OpenFlow používá pro výměnu zpráv mezi switchem a controllerem speciální OpenFlow kanál. Typický OpenFlow controller spravuje více zabezpečených kanálů, kde každý switch má svůj vlastní kanál. Spojení mezi switchem a controllerem probíhá právě pomocí OpenFlow kanálu a zahrnuje šest po sobě jdoucích kroků. [10]

1. Krok A - Jestliže, je do sítě připojen nový OpenFlow controller nebo switch, spojení je navázáno pomocí TCP three-way handshake, který využívá TCP port číslo 6633.
2. Krok B - Když je TCP spojení dokončeno, je odeslán hello paket, který neobsahuje žádný obsah. Hello paket je symetrický typ zprávy v OpenFlow protokolu, tudíž jej posílá switch i controller.
3. Krok C - Po výměně hello paketů je spojení mezi controllerem a switchem navázáno. Je to možné zkontrolovat prostřednictvím GUI (grafické uživatelské prostředí) controlleru nebo CLI (příkazový řádek) u switche.
4. Krok D - Zahrnuje dva typy zpráv:

- Feature request: controller požaduje od switche výčet možností a dalších specifických funkcí, které na něm běží.
  - Feature reply: switch odpovídá na požadavek controlleru. Tento paket upřesňuje detaily OpenFlow switche. Obsahem tohoto paketu je například – popis (typ) portu, schopnosti switche a další.
5. Krok E - Jakmile controller zná všechny schopnosti a funkce switche, mohou začít další akce specifické pro OpenFlow protokol.
6. Krok F - Zprávy echo request a echo reply jsou symetrické typy paketů protokolu OpenFlow. Používají se k udržení spojení mezi switchem a controllerem. Echo request a echo reply se liší především v hlavičce paketu, kde každý obsahuje jiný typ hodnoty a také uživatelské data se mohou lišit v závislosti na výrobci daného zařízení. [4]



**Obrázek 3.3:** Navázání spojení mezi OpenFlow controllerem a OpenFlow switchem (převzato z [4])



# 4 OpenFlow controllery pro SDN

V kapitole jsou analyzovány a popsány existující OpenFlow controllery pro prostředí Softwarově definovaných sítí, z nichž budou vybrány dva pro bližší představení a následné otestování.

## 4.1 Seznam OpenFlow controllerů

V současnosti existuje již přes 30 controllerů, které podporují protokol OpenFlow. Tyto OpenFlow controllery od mnoha výrobců, vysokých škol a výzkumných skupin jsou napsány v různých programovacích jazycích a převážná většina je open-source projekt, tudíž podpora vývoje a implementace není až tak velká. [11]

Vybrané OpenFlow controllery pro prostředí SDN:

- NOX – První a nejstarší OpenFlow controller, původně vyvinut společností Nicira Networks, ale později byl projekt darován výzkumné skupině a od té doby byl základem pro mnoho různých výzkumných projektů na počátku zkoumání oblasti SDN.  
V podstatě je to síťový operační systém poskytující kontrolu a přehled nad SDN sítí skládající se z OpenFlow switchů. Podporuje souběžné aplikace napsané v programovacím jazyce Python a C++ a navíc obsahuje celou řadu aplikací v controlleru. Nabízí webové grafické uživatelské rozhraní a je možné jej nainstalovat a provozovat pouze na operačním systému Linux. Nevýhodou je podpora protokolu OpenFlow jen ve verzi 1.0.
- POX – Je mladší sourozenec NOX controlleru, a tak je v mnoha ohledech podobný. Je platformou pro rychlý vývoj a testování ovládání sítě pomocí jazyka Python, jejímž hlavním cílem bylo vytvořit moderní SDN controller. Nainstalovat a spustit jde na všech operačních systémech (Linux, Windows a Mac OS) a obsahuje stejné GUI jako jeho předchůdce NOX. Nejčastěji je využíván v akademické a výzkumnické sféře.
- Beacon – Je open-source projekt vyvíjející rozšiřitelný OpenFlow controller založený

na programovacím jazyku JAVA. Je postaven na rámci OSGi, což je dynamický modulární systém pro jazyk JAVA, který umožňuje nainstalování, spuštění a běh OpenFlow aplikací, a to bez odpojení switchu. Díky JAVA je spustitelný na operačních systémech Linux a mobilních zařízeních se systémem Android. Také zahrnuje webové uživatelské prostředí pro ovládání a nastavení controlleru.

- Floodlight – JAVA OpenFlow controller vytvořený Davidem Ericksonem na univerzitě ve Stanfordu. Spadá do podnikové třídy a pracuje s širokou škálou fyzických i virtuálních OpenFlow switchů. Běží na systémech Linux a vyznačuje se snadným uživatelským prostředím. Je podporován komunitou vývojářů a navíc i společností Big Switch Networks, která se velmi zajímá o SDN a aktivně tento controller vyvíjí a testuje.
- Ryu – Open-source projekt Ryu, vytvořený v jazyce Python, je síťový operační systém pro SDN s protokolem OpenFlow. Cílem je poskytnout logicky centralizované řízení a dobře definované rozhraní API, které usnadní vývojářům vytvářet nové síťové aplikace umožňující řízení a správu počítačových sítí. Lze jej nainstalovat na systému Linux a podporuje OpenFlow protokol ve verzích 1.0, 1.2, 1.3 a 1.4. [11] [12]
- OpenDaylight – Je open-source projekt s největší podporou velkých firem, vývojářů a výrobců na trhu. Členství je rozděleno do třech částí – platinové, zlaté a stříbrné, což vyjadřuje míru zapojení do vývoje projektu. Zatím byly vydány dvě verze controlleru OpenDaylight, a to Hydrogen a Helium, do dalších let je naplánován Lithium. Obě vydané verze byly vytvořeny na základě jazyka JAVA, proto je možné jej nainstalovat a spustit na operačních systémech Linux. Jednou z odlišností je jiné grafické uživatelské rozhraní u každé verze. [13]
- HP Virtual Application Networks (VAN) SDN – Je softwarový controller od společnosti HP, která je jednou z nejvíce angažovaných firem v oblasti SDN. HP VAN SDN poskytuje jednotný kontrolní bod do OpenFlow sítě, zjednodušuje řízení, zajišťování a orchestraci dané sítě. Controller vychází z programovacího jazyka JAVA a je poměrně hodně výpočetně náročný, proto jej lze nainstalovat a spustit pouze na výkonném počítači s operačním systémem Linux. Jako i ostatní výše zmíněné controllery také obsahuje přehledné webové GUI. [11] [14]

Pro bližší popsání a následné otestování byly vybrány controllery OpenDaylight a HP VAN SDN.

Tyto controllery byly zvoleny především proto, že OpenDaylight je open-source s velkou podporou ve vývoji od významných firem a nabízí již stabilní a provozuschopnou verzi

s názvem Hydrogen. Zatímco HP VAN SDN controller byl vybrán jako zástupce komerční sféry vývoje a také kvůli vydatnému zájmu společnosti HP o prostředí SDN.

## 4.2 OpenDaylight controller

Projekt OpenDaylight byl odstartován v roce 2013 organizací Linux Foundation a jeho cílem je urychlit vývoj a přijetí SDN. OpenDaylight je otevřená platforma pro softwarově definované sítě, která si úspěšně získává přízeň a podporu klíčových firem ve světě IT a stává se tak průmyslovým standardem. Mezi hlavní členy projektu patří Cisco, HP, Juniper, Dell, IBM, Intel, Microsoft, Brocade, Red Hat, Citrix a další.

OpenDaylight může být jádrem, hlavní složkou v jakékoli SDN architektuře. V návaznosti na open-source SDN controller umožňuje uživatelům snížit provozní složitost, prodloužit životnost stávajícího hardwaru a poskytnout nové služby a funkce, které jsou k dispozici pouze v SDN technologii.

Spíše než vytvářet nové standardy si projekt klade za cíl vytvořit rozšiřitelnou, virtuální a otevřenou platformu již existujících standardů, jako je OpenFlow, poskytující univerzální rozhraní, přes které lze ovládat virtuální nebo fyzické switche pomocí softwaru.

Jedná se o modulární a flexibilní SDN controller běžící na JVM (JAVA Virtual Machine) a je tudíž nezávislý na platformě, na které běží. V současné době byly vydány dvě verze OpenDaylight controlleru, první s názvem Hydrogen a druhá nese označení Helium. [13]

V této bakalářské práci je použita verze Hydrogen, především pro její plnou podporu a již vydanou finální verzi. Helium je současně stále ve vývoji a obsahuje tak některá omezení a nedostatky.

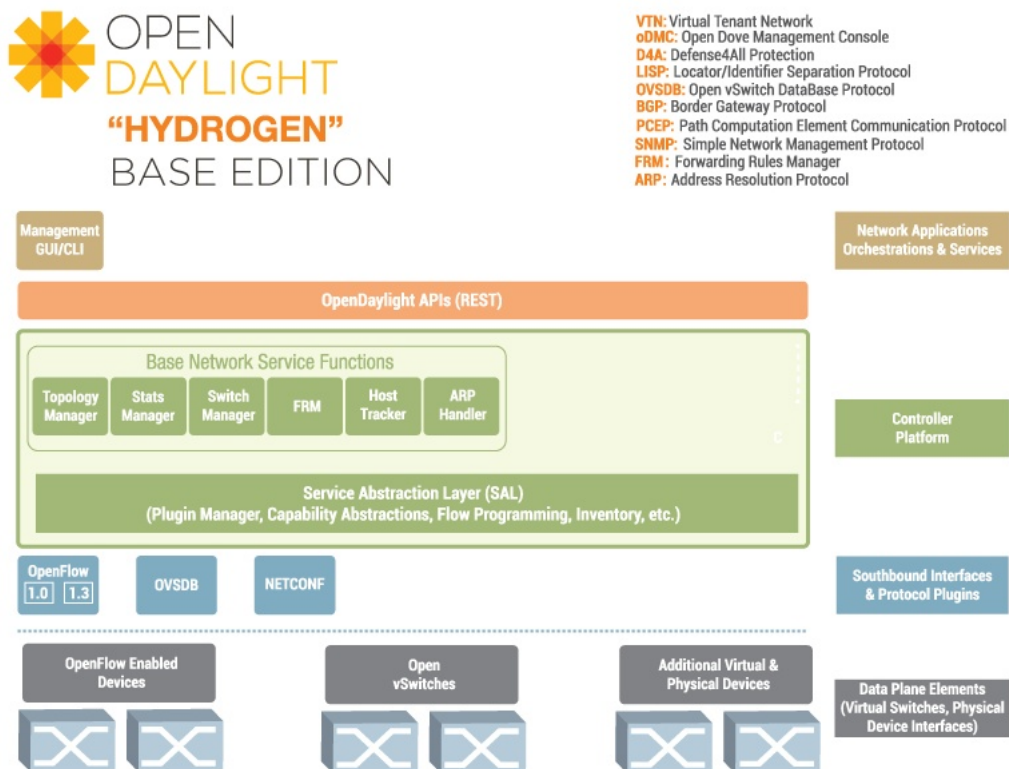
### 4.2.1 Hydrogen

Release Hydrogen se pro potřeby široké škály uživatelů dělí na tři různé edice:

- Base Edition
- Virtualization Edition
- Service Provider Edition

Base Edition je určeno pro ty, kteří chtějí zkoumat SDN a OpenFlow ve virtuálním či fyzickém prostředí, nejčastěji pro akademické účely. Virtualization Edition je především pro datové centra. Zahrnuje všechny komponenty, které jsou v Base Edition a navíc funkce pro vytváření

a správu virtuálních sítí, které zahrnují aplikace pro zabezpečení a správu sítě. A poslední verzi, kterou je Service Provider Edition, využijí poskytovatelé, kteří spravují již existující počítačové sítě a chtějí vyzkoušet nebo naplánovat přechod k SDN. Také obsahuje Base Edition a kromě toho podporu protokolů, které se běžně vyskytují v sítích poskytovatelů služeb. [15]



Obrázek 4.1: OpenDaylight Base Edition - architektura [15]

Klíčové vlastnosti Base Edition:

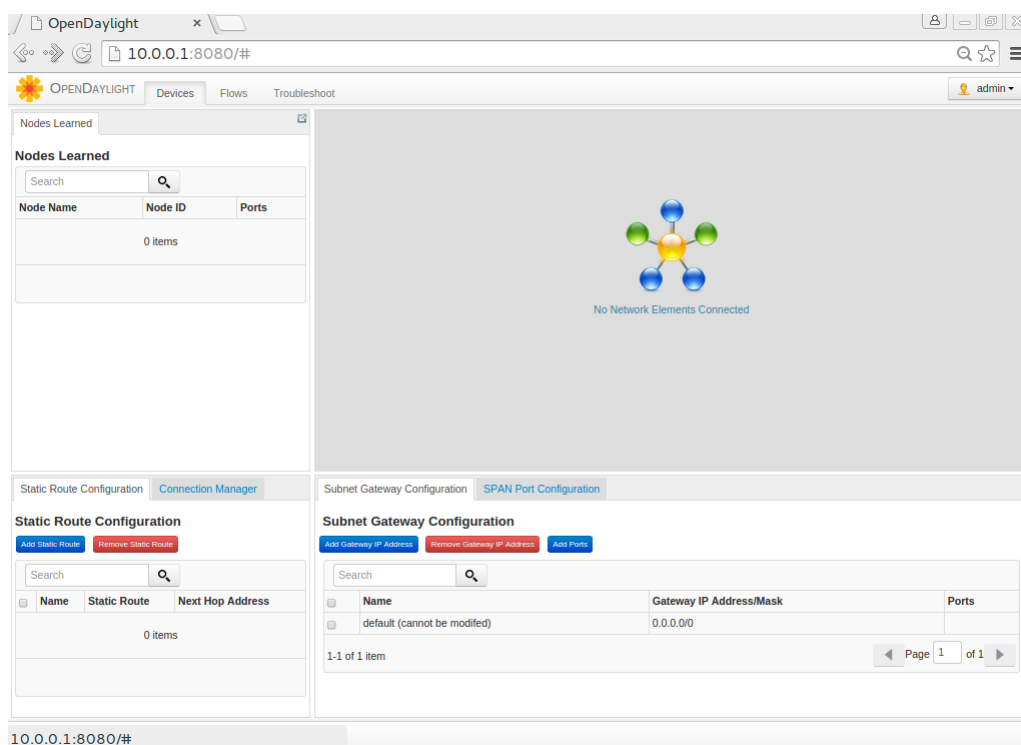
- Controller: modulární, rozšiřitelný, škálovatelný, multi-protokolový SDN controller založený na OSGi (programovací jazyk JAVA).
- OpenFlow plugin: integrace knihovny protokolu OpenFlow do controlleru.
- Protokol OpenFlow: podpora protokolu OpenFlow od verze 1.0 až do verze 1.3.x.
- OVSDB: konfigurační databáze a management protokolu pro podporu Open vSwitch.
- YANG: datový modelovací jazyk, poskytující potřebné nástroje a knihovny, které podporují NETCONF (Network Configuration Protocol) umožňující volání vzdálených procedur a oznámení. [15]

## 4.2.2 Grafické prostředí

OpenDaylight nabízí jednoduché, intuitivní a přehledné GUI.

Jsou zde tři hlavní karty:

- Devices – zobrazuje topologii sítě a připojené switche, kterým lze nastavit název, provozní režim a úroveň, kterou představují v rámci sítě.
- Flows – na této kartě je možné nastavit cesty (flows) v síti, případně co s paketem nastane po příchodu do switche a také detail těchto cest a akcí.
- Troubleshoot – slouží k hledání, zobrazení a řešení problémů v dané síti. [15]



Obrázek 4.2: OpenDaylight - grafické uživatelské prostředí

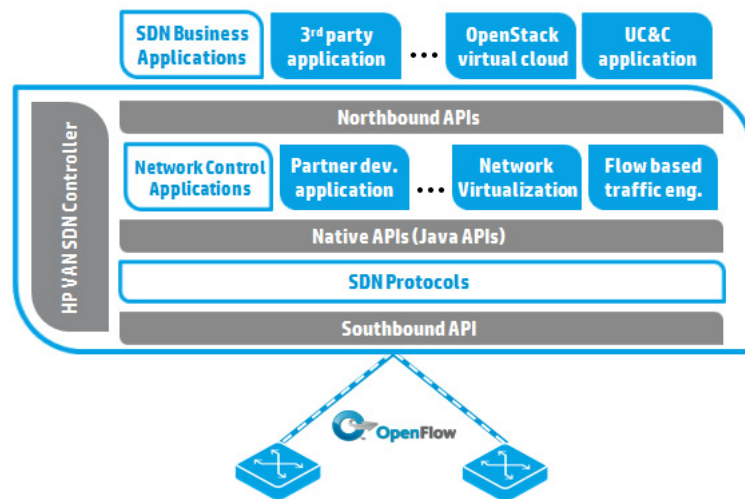
## 4.3 HP VAN SDN controller

Společnost Hewlett-Packard (HP) se zabývá technologií SDN již od roku 2007. Hlavní doménou jsou otevřenost a možnost nových funkcí, jako například schopnost zareagovat na požadavky aplikací, flexibilnější síť, nepotřebující ovládání pomocí příkazového řádku a přidání takových funkcí do sítě, pro které nebylo hardwarové zařízení vyrobeno. [16]

HP současně nabízí přes 10 routerů a 50 switchů podporující protokol OpenFlow, které již mají všechny předpoklady pro nasazení v prostředí SDN sítě.

HP Virtual Application Networks (VAN) SDN controller je software s jednotným kontrolním bodem a zjednodušeným řízením sítě, který podporuje protokol OpenFlow a využívá služby nové síťové generace. Poskytuje otevřené aplikační programové rozhraní (API), díky kterému mohou vývojáři třetích stran dodávat inovativní řešení pro dynamické obchodní požadavky síťové infrastruktury prostřednictvím vlastních programů v jazyce JAVA. Controller je navržen pro práci v prostředí datového centra, poskytovatele služeb nebo akademické sféry. [14]

Software HP VAN SDN controlleru přímo zajišťuje fyzické a virtuální switche pod svojí kontrolu prostřednictvím protokolu OpenFlow. Topologie sítě, porty a linky v síti jsou bezprostředně viditelné, což umožňuje centralizovanou správu pravidel a efektivní výběr cesty v síti, založený na dynamickém, globálním pohledu na síť. [14]



Obrázek 4.3: Architektura HP VAN SDN controlleru[14]

Klíčové vlastnosti HP VAN SDN controlleru:

- Otevřená programovatelná rozhraní: přináší těsnou integraci mezi počítačovou sítí a obchodním systémem s otevřeným, programovatelným rozhraním poskytující přípravu aplikací a automatizaci síťových funkcí.
- Centralizovaná kontrola: HP VAN SDN controller poskytuje centralizovanou kontrolu nad SDN sítí včetně funkcí, jako jsou například zjištění topologie sítě nebo výpočet nejkratší cesty skrz síť.
- Vysoká dostupnost a škálovatelnost: více controllerů lze seskupit, tudíž pokud některý controller v síti selže, jiný převeze jeho roli.

- Silné zabezpečení: je důležitým faktorem HP VAN SDN controlleru. Využívá robustní autentizační a autorizační metody tak, aby SDN aplikace mohli spolupracovat s controllerem a zároveň zabránit neoprávněným aplikacím získat přístup k síti. Spojení mezi OpenFlow switchem a controllerem je také šifrované a zabezpečené. [14]

Společnost HP vytvořila obchod s SDN aplikacemi (The HP SDN Application Store), ve kterém nabízí zpoplatněné aplikace od nezávislých výrobců rozšiřující možnosti a funkce HP VAN SDN controlleru. [17]

### 4.3.1 Grafické prostředí

Hlavní menu v GUI u HP VAN SDN controlleru nabízí více položek než v případě OpenDaylight, ale některé pouze informují o nastavení controlleru nebo komunikaci v síti a nelze je změnit. Položky v menu:

- Allerts – upozornění oznamují interní události, které mají vliv na činnost controlleru.
- Applications – umožňuje instalace, zapnutí, vypnutí a odinstalování spravovaných aplikací.
- Configurations – zobrazuje konfigurovatelné komponenty controlleru.
- Audit Log – zaznamenává události a operace související se změnami konfigurace.
- Licenses – aktivace nových a deaktivace již vložených licencí.
- Support Logs – automaticky shromažďuje vnitřní záznamy událostí pro potřeby řešení problémů.
- OpenFlow Monitor – výčet switchů připojených ke controlleru a jejich podrobnosti jako jsou porty, trasy a celkový přehled nastavení.
- OpenFlow Topology – zobrazuje právě připojenou topologii sítě.
- OpenFlow Trace – nástroj pro odstraňování problémů. Zachytává komunikaci mezi controllerem a sítí.
- OpenFlow Classes – controller udržuje přehled o tom, co která aplikace v síti dělá. A to tím, že dostává zprávy od prvků v síti, například může obdržet paket přímo nebo jen jeho kopii.

- Packet Listeners – znázorňuje „posluchače paketů“ (Packet Listeners), které právě běží na controlleru. [18]

The screenshot shows the HP VAN SDN Controller web interface. The browser address bar displays `https://10.0.0.2:8443/sdn/ui/app/index#globalAlerts`. The page title is "General / Global Alerts". The left sidebar contains a navigation menu with the following items: Alerts, Applications, Configurations, Audit Log, Licenses, Support Logs, OpenFlow Monitor, OpenFlow Topology, OpenFlow Trace, OpenFlow Classes, and Packet Listeners. The main content area displays a table of alerts with the following columns: Severity, Date/Time, Description, Origin, Topic, and Controller ID. The table contains 10 rows of alert data, all with a severity of "Warning" (indicated by a red 'W' icon) and a topic of "licensing". The first row has a severity of "Info" (indicated by a blue 'i' icon). The table also includes action buttons: Refresh, Acknowledge, and UnAcknowledge. At the bottom of the page, it shows "Showing page 1 of 2" and "Auto 10 20 100".

Severity	Date/Time	Description	Origin	Topic	Controller ID
i	today 11:59:17	OpenFlow Contro...	Core Controller	of_controller	61bb6741-c61e-...
W	today 11:59:07	HP VAN SDN Ctrl ...	compliance-man...	licensing	61bb6741-c61e-...
W	2015-04-10 16:0...	HP VAN SDN Ctrl ...	compliance-man...	licensing	61bb6741-c61e-...
W	2015-04-09 16:0...	HP VAN SDN Ctrl ...	compliance-man...	licensing	61bb6741-c61e-...
W	2015-04-08 16:0...	HP VAN SDN Ctrl ...	compliance-man...	licensing	61bb6741-c61e-...
W	2015-04-07 16:0...	HP VAN SDN Ctrl ...	compliance-man...	licensing	61bb6741-c61e-...
W	2015-04-06 16:0...	HP VAN SDN Ctrl ...	compliance-man...	licensing	61bb6741-c61e-...
W	2015-04-05 16:0...	HP VAN SDN Ctrl ...	compliance-man...	licensing	61bb6741-c61e-...
W	2015-04-04 16:0...	HP VAN SDN Ctrl ...	compliance-man...	licensing	61bb6741-c61e-...
W	2015-04-03 16:0...	HP VAN SDN Ctrl ...	compliance-man...	licensing	61bb6741-c61e-...

Obrázek 4.4: Grafické uživatelské prostředí HP VAN SDN controlleru



# 5 Test controllerů

V následující části této bakalářské práce budou porovnány a otestovány dva vybrané controllery (OpenDaylight a HP VAN SDN) a síťový emulátor Mininet, dále bude popsán postup a výsledky jednotlivých testů.

## 5.1 Porovnání OpenDaylight a HP VAN SDN controllerů

Ve světě SDN controllerů jsou možné dva modely vývoje, a to komerční nebo open-source. V té první je jeden dominantní výrobce, který vyvíjí a podporuje svůj vlastní controller a poskytuje jej ostatním za určitý poplatek. Kdežto open-source je vytvářen více vývojáři z různých firem a nabízí otevřený controller, ke kterému si každý může přidat své vlastní speciální funkce, které potřebuje. [1]

Největší rozdíl mezi controllery OpenDaylight a HP VAN SDN je tedy v jejich podpoře. Zatímco HP VAN SDN je komerční, tak OpenDaylight je open-source projekt, ale mnoho významných firem je jeho součástí včetně společnosti HP, která je platinovým členem.

Další rozdíl je v operačním systému, na kterém controller funguje. OpenDaylight lze nainstalovat na jakýkoliv operační systém, jedinou podmínkou je přítomnost modulu JAVA. Kdežto HP VAN SDN controller je možné nainstalovat jen na operačním systému Linux.

Mimo těchto odlišností mají i mnoho společného. Oba controllery běží na základě programovacího jazyku JAVA a v podstatě pracují na stejném principu. [13] [14]

### 5.1.1 Identické funkce a vlastnosti

- Přehledné ovládání a konfigurace controlleru přes grafické rozhraní v prohlížeči, dostupné přes IP adresu controlleru a specifický port.

- Zobrazení právě připojené topologie umožňující pružně pohybovat se všemi prvky sítě a upravit tak topologii podle potřeb.
- Výčet všech OpenFlow switchů připojených do sítě, jejich správa a nastavení.
- Podpora OpenFlow ve verzích 1.0 a 1.3. [13] [14]

### 5.1.2 Specifické funkce OpenDaylight controlleru

- Zvolení akce, která nastane po příchodu paketu do switche. Je možné nastavit smyčku (Loopback), zahození nebo přeoslání paketu na specifický port.
- Vytvoření a nastavení podsítí a statických cest v síti.
- Správa, editace a odstranění již existujících síťových tras.
- Zobrazení, analyzování a následné řešení problémů v síti - Troubleshooting. [13]

### 5.1.3 Specifické funkce HP VAN SDN controlleru

- Grafické zobrazení nejkratší cesty v síti pomocí výpočtů Dijkstrova algoritmu. Po vybrání a označení zdroje a cíle je okamžitě znázorněna nejkratší cesta mezi nimi.
- Správa a konfigurace aplikací běžících na controlleru, které lze aktivovat, deaktivovat nebo úplně odinstalovat.
- Zaznamenávání OpenFlow komunikace v síti přímo v controlleru, tudíž není potřeba další speciální software sloužící na odchyťování OpenFlow paketů.
- Propojení s HP SDN App Store umožňuje nakupovat, instalovat a spravovat nové aplikace, které rozšiřují a zlepšují funkce controlleru. [14]

## 5.2 Mininet

Open-source projekt Mininet je velmi důležitý pro komunitu SDN. Podporuje výzkum, vývoj, vzdělávání, testování, vychytávání chyb a veškeré další úkoly, které jsou spojeny s možnostmi nabízející kompletní experimentální síť běžící na notebooku nebo počítači.

Mininet je síťový emulátor, který vytváří a simuluje počítačovou síť složenou z virtuálních hostů (koncových počítačů), switchů, controllerů a linek mezi nimi. Virtuální switche

podporují OpenFlow pro vysoce flexibilní vlastní směrování a využití v SDN. V současné době jej lze spustit pouze na operačním systému Linux.

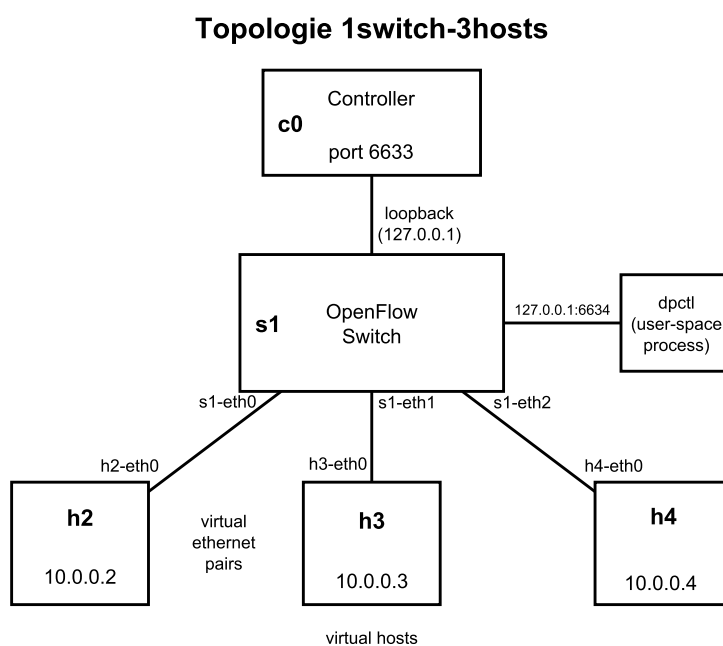
Virtuální hosti, switche, linky a controllery mají podobné chování jako fyzické prvky, jen jsou v tomto případě vytvořeny pomocí softwaru.

Kód, který se vyvíjí a testuje v prostředí Mininet, například pro OpenFlow controller nebo modifikovaný switch, je možné přesunout do reálného systému s minimálními změnami. To znamená, že návrh sítě, který funguje v Mininet lze obvykle rychle přemístit přímo na hardwarové zařízení. [19]

Klíčové vlastnosti:

- Poskytuje jednoduché a levné testování zařízení potřebné pro vývoj OpenFlow aplikací.
- Více vývojářů může pracovat souběžně a nezávisle na stejné topologii.
- Podporuje jednoduché a rozšiřitelné Python API pro vytváření sítí a experimentování.
- Obsahuje základní sadu parametrizovaných topologií sítě a navíc je zde možnost vytvoření libovolné vlastní topologie, pomocí programovacího jazyku Python.
- Zahrnuje CLI, které umožňuje testování celé virtuální sítě. [19]

Obrázek 5.1 ilustruje jednoduchou Mininet síť složenou ze tří počítačů, jednoho virtuálního OpenFlow switchu a OpenFlow controlleru. Všechny komponenty jsou připojeny přes virtuální Ethernet linky, ke kterým jsou přiřazeny privátní IP adresy (10.0.0.x). [1]



**Obrázek 5.1:** Příklad sítě vytvořené v prostředí Mininet (převzato z [1]).

## 5.3 Instalace

V následující podkapitole je popsána instalace jednotlivých controllerů a síťového emulátoru Mininet. Pro instalaci a následné testování je zvolen operační systém Ubuntu 14.04 x86\_64.

### 5.3.1 OpenDaylight controller

Instalace OpenDaylight controlleru ve verzi Hydrogen je poměrně jednoduchá:

1. Prvním krokem je stažení volně dostupného softwaru z webu OpenDaylight:  
`http://www.opendaylight.org/software/release-archives`
2. Po rozbalení (unzip) staženého souboru následuje již samotné spuštění OpenDaylight controlleru, které se provádí pomocí bashového skriptu s názvem `run.sh`. Ten se nachází v rozbaleném adresáři:  
`root@opendaylight:~/opendaylight# ./opendaylight/run.sh`
3. Grafické uživatelské rozhraní controlleru je přístupné přes webový prohlížeč na adrese: `http://ip_adresa_controlleru:8080`. Uživatelské jméno a heslo pro přihlášení je `admin`. [20]

### 5.3.2 HP VAN SDN controller

1. HP VAN SDN používá pro přihlášení a řízení přístupů ke controlleru službu Keystone z projektu OpenStack. Je tedy nutné tuto službu nejdříve nainstalovat lokálně pomocí příkazu:

```
root@hp-van-sdn:~# sudo apt-get install keystone -y
```

2. Dalším krokem v instalaci je vytvořit uživatele (Tenant) pro přístup ke controlleru. Tenant představuje logické uživatele v Keystone, kterým je možné přidělit různé typy definovaných rolí.

K založení uživatelského účtu pro získání přístupu ke controlleru je tedy nutné vytvořit tenanta, dva uživatele (`sdn-admin` a `sdn-user`) a k nim jejich uživatelské role. V první řadě se do proměnných prostředí přiřadí přihlašovací údaje:

```
root@hp-van-sdn:~# export OS_SERVICE_TOKEN=ADMIN
```

```
root@hp-van-sdn:~# export OS_SERVICE_ENDPOINT=http://127.0.0.1:35357/v2.0
```

3. Vytvoření uživatelů a jejich rolí:

```
root@hp-van-sdn:~# keystone tenant-create --name sdn
root@hp-van-sdn:~# keystone user-create --name hp --pass hp --tenant sdn
root@hp-van-sdn:~# keystone role-create --name sdn-admin
root@hp-van-sdn:~# keystone role-create --name sdn-user
root@hp-van-sdn:~# keystone user-role-add --user hp --role sdn-admin --tenant sdn
root@hp-van-sdn:~# keystone user-role-add --user hp --role sdn-user --tenant sdn
```

4. Po vytvoření uživatelského účtu přichází na řadu instalace controlleru, který lze stáhnout z webu HP: <http://bit.do/hpvansdn>

5. Poté následuje rozbalení staženého souboru a samotná instalace HP VAN SDN controlleru:

```
root@hp-van-sdn:~# sudo dpkg -i hp-sdn-ctl_2.4.5.0614_amd64.deb
```

6. A nakonec již jen kontrola, zda se instalace podařila:

```
root@hp-van-sdn:~# dpkg -l hp-sdn-ctl
root@hp-van-sdn:~# service sdnc status
```

```
root@hp-van-sdn:~# dpkg -l hp-sdn-ctl
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
||/ Name          Version          Architecture Description
+++-----+-----+-----+-----+
ii hp-sdn-ctl      2.4.5.0614      amd64          HP VAN SDN Controller
root@hp-van-sdn:~# service sdnc status
sdnc start/running, process 1172
root@hp-van-sdn:~#
```

Obrázek 5.2: Ověření instalace HP VAN SDN controlleru

7. GUI HP VAN SDN controlleru je také dostupné přes webový prohlížeč a adresu:

[http://ipadresa\\_controlleru:8443/sdn/ui](http://ipadresa_controlleru:8443/sdn/ui). Uživatelské jméno a heslo je takové, jaké jsme zvolili při instalaci, vytváření uživatelů a přidání jejich rolí do keystone. V našem případě je jméno a heslo totožné, a to - hp. [21]

### 5.3.3 Mininet

Mininet lze nainstalovat třemi postupy, pro účely této práce byl vybrán způsob nativní instalace ze zdrojových kódů:

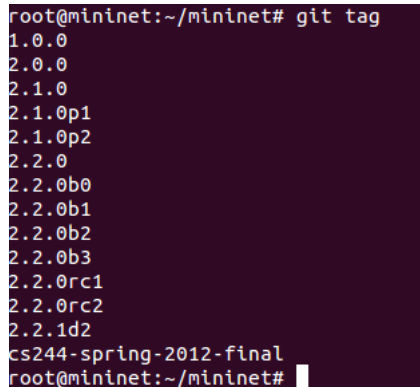
1. Nejdříve je nutné nainstalovat program Git, pomocí kterého je možné získat zdrojový kód:

```
root@mininet:~# sudo apt-get install git
```

```
root@mininet:~# git clone git://github.com/mininet/mininet
```

2. Dále lze vypsat všechny verze Mininet, které jsou k dispozici:

```
root@mininet:~/mininet# git tag
```



```
root@mininet:~/mininet# git tag
1.0.0
2.0.0
2.1.0
2.1.0p1
2.1.0p2
2.2.0
2.2.0b0
2.2.0b1
2.2.0b2
2.2.0b3
2.2.0rc1
2.2.0rc2
2.2.1d2
cs244-spring-2012-final
root@mininet:~/mininet#
```

Obrázek 5.3: Mininet - zobrazení všech dostupných verzí

3. Vybrání požadované verze k instalaci, v našem případě 2.2.0:

```
root@mininet: ~/mininet# git checkout -b 2.2.0 2.2.0
```

4. Následuje samotná instalace Mininet:

```
root@mininet:~# mininet/util/install.sh -a
```

5. A konečně ověření instalace:

```
root@mininet:~# sudo mn --test pingall [22]
```



```
root@mininet:~# mn --test pingall
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 1 switches
s1 ..
*** Stopping 2 links
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.475 seconds
root@mininet:~#
```

Obrázek 5.4: Mininet - ověření funkčnosti

## 5.4 Test 1

První test bude zaměřen na ověření základních operací protokolu OpenFlow při komunikaci mezi OpenFlow switchem v prostředí Mininet a controllerem.

### 5.4.1 Vytvoření topologie v Mininet

Jestliže je controller nainstalován na jiném počítači než Mininet, je nutné přidat do příkazu parametr, který říká, že je controller vzdálený (remote) a přístupný na určité IP adrese, případně i na specifikovaném portu. Příkaz pro připojení ke vzdálenému controlleru pak vypadá následovně:

```
root@mininet:~# mn -controller=remote,ip=ip_adresa_controlleru,port=číslo_portu
```

Dalším parametrem, který lze obměnit je topologie. Výchozí topologie v prostředí Mininet se skládá z jednoho switche a dvou koncových počítačů (hosts). Ale je možné ji změnit pomocí parametru `-topo`, díky kterému lze vybrat některou z již definovaných, přednastavených topologií. Ty jsou trojího typu – single, linear a tree. [23]

- Single: jak název napovídá, topologie obsahuje pouze jeden switch a zvolený počet počítačů.

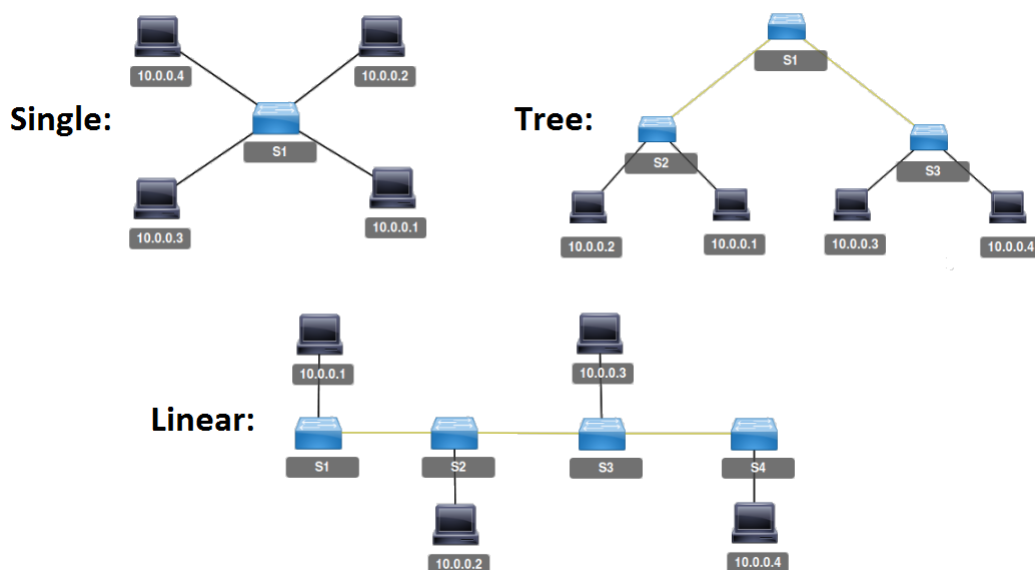
```
root@mininet:~# mn -controller=remote,ip=10.0.0.1 -topo=single,3
```

- Linear: switche jsou uspořádané v řadě za sebou v jedné linii. Je možné zvolit množství switchů v topologii. Počet počítačů je daný, každý switch má právě jeden.

```
root@mininet:~# mn -controller=remote,ip=10.0.0.1 -topo=linear,3
```

- Tree: propojení switchů do tvaru připomínající strom, přičemž každý switch má pod sebou další dva switche či počítače v případě, že se jedná o poslední vrstvu sítě. Je zde možnost zvolit počet úrovní hierarchie topologie.

```
root@mininet:~# mn -controller=remote,ip=10.0.0.1 -topo=tree,2
```



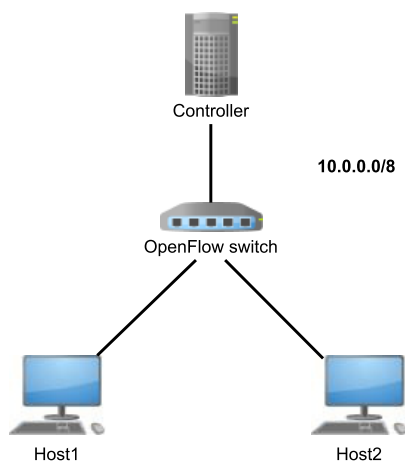
Obrázek 5.5: Mininet - definované topologie vyobrazené v OpenDaylight controlleru

## 5.4.2 Připojení ke controlleru

Pro zachytávání komunikace je použit program Wireshark nainstalovaný v Mininet, ve kterém je aplikován filtr „of“, který zobrazí jen komunikaci protokolu OpenFlow. Síťová topologie se skládá z jednoho OpenFlow switche a dvou počítačů, která bude postupně připojena ke dvěma námi vybraným controllerům, přičemž budeme porovnávat jakým způsobem controller komunikuje s OpenFlow switchem.

Příkaz pro vytvoření testované topologie:

```
root@mininet:~# mn -controller=remote,ip=ip_adresa_controlleru -topo=single,2
```



Obrázek 5.6: Síťová topologie v Mininet pro test 1



## OpenDaylight controller

Controlleru je přiřazena IP adresa 10.0.0.1 a Mininet je nainstalován na počítači s IP adresou 10.0.0.3. Jak je vidět z obrázku níže, komunikace začíná zasláním Hello OpenFlow paketu z controlleru do switchu, načez je opět odeslán hello packet ale v opačném směru. Také můžeme vyzorovat, že port controlleru je 6633. Poté jsou zasílány pakety určující vlastnosti a nastavení OpenFlow switchu, které následují výměny packet\_in a packet\_out zpráv.

The screenshot shows a network traffic capture interface with a filter set to 'of'. The main table displays the following data:

No.	Time	Source	Destination	Protocol	Length	Info
6	0.004247000	10.0.0.1	10.0.0.3	OF 1.0	74	of_hello
11	0.233429000	10.0.0.3	10.0.0.1	OF 1.0	74	of_hello
13	0.235959000	10.0.0.1	10.0.0.3	OF 1.0	74	of_hello
15	0.236077000	10.0.0.1	10.0.0.3	OF 1.0	74	of_features_request
17	0.236731000	10.0.0.3	10.0.0.1	OF 1.0	242	of_features_reply
18	0.237097000	10.0.0.1	10.0.0.3	OF 1.0	78	of_set_config
19	0.237112000	10.0.0.1	10.0.0.3	OF 1.0	74	of_get_config_request
21	0.237188000	10.0.0.3	10.0.0.1	OF 1.0	78	of_get_config_reply
22	0.241124000	10.0.0.1	10.0.0.3	OF 1.0	138	of_flow_delete
23	0.242219000	::	ff02::16	OF 1.0	174	of_packet_in
24	0.242796000	::	ff02::16	OF 1.0	180	of_packet_out
25	0.242814000	::	ff02::16	OF 1.0	180	of_packet_out
27	0.370464000	::	ff02::1:ff09:e39a	OF 1.0	162	of_packet_in
28	0.371276000	::	ff02::1:ff09:e39a	OF 1.0	168	of_packet_out
29	0.371317000	::	ff02::1:ff09:e39a	OF 1.0	168	of_packet_out
31	0.398146000	::	ff02::1:ff0f:2dff	OF 1.0	162	of_packet_in
32	0.398727000	::	ff02::1:ff0f:2dff	OF 1.0	168	of_packet_out
34	0.438375000	::	ff02::16	OF 1.0	174	of_packet_in

Below the table, the details for the selected frame (Frame 6) are shown:

- Frame 6: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
- Ethernet II, Src: IETF-VRRP-VRID\_00 (00:00:5e:00:01:00), Dst: 02:d2:69:e2:72:ab (02:d2:69:e2:72:ab)
- Internet Protocol Version 4, Src: 10.0.0.1 (10.0.0.1), Dst: 10.0.0.3 (10.0.0.3)
- Transmission Control Protocol, Src Port: 6633 (6633), Dst Port: 40917 (40917), Seq: 1, Ack: 2, Len: 8
- OpenFlow
  - version: 1
  - type: OFPT\_HELLO (0)
  - length: 8
  - xid: 1029634870

At the bottom, a summary bar indicates: OpenFlow (of), 8 bytes; Packets: 130 · Displayed: 70 (53.8%) · Dropped: ...; Profile: Default.

Obrázek 5.7: Zahájení komunikace mezi OpenDaylight controllerem a OpenFlow switchem

## HP VAN SDN controller

IP adresa u Mininet je stejná, tudíž 10.0.0.3, avšak IP adresa controlleru je 10.0.0.2. Komunikace je opět zahájena pomocí Hello OpenFlow paketu, ale v tomto případě spojení začíná switch a posléze stejný paket putuje opačnou cestou. Následně je odeslán požadavek controlleru na vlastnosti switchu (features\_request), který je obratem posílá zpět (features\_reply). Jsou zde také packet\_in a packet\_out zprávy ale nevyskytují se v takové míře jako u OpenDaylight controlleru.

The screenshot shows a Wireshark capture of network traffic. The filter is set to 'of'. The main pane displays a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The selected packet (No. 10) is an OpenFlow Hello message. The packet details pane shows the following structure:

- Frame 10: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
- Ethernet II, Src: 02:d2:69:e2:72:ab (02:d2:69:e2:72:ab), Dst: IETF-VRRP-VRID\_00 (00:00:5e:00:01:00)
- Internet Protocol Version 4, Src: 10.0.0.3 (10.0.0.3), Dst: 10.0.0.2 (10.0.0.2)
- Transmission Control Protocol, Src Port: 47528 (47528), Dst Port: 6633 (6633), Seq: 1, Ack: 1, Len: 8
- OpenFlow
  - version: 1
  - type: OFPT\_HELLO (0)
  - length: 8
  - xid: 32

The status bar at the bottom indicates: Frame (frame), 74 bytes; Packets: 69 · Displayed: 32 (46.4%) · Dropp...; Profile: Default.

Obrázek 5.8: Zahájení komunikace mezi HP VAN SDN controllerem a OpenFlow switchem

### 5.4.3 Výsledky

Jakmile byla spuštěna topologie sítě v Mininet, program Wireshark okamžitě zachytil OpenFlow pakety zasílané v síti, které ověřily připojení funkční SDN sítě.

Při analyzování OpenFlow protokolu jsme tak zjistili, že controllery používají při zahájení spojení stejný typ OpenFlow zpráv, jen v jiném pořadí. Další komunikace v síti je již rozdílná, například HP používá více paketů pro zjištění nastavení a vlastností sítě, kdežto OpenDaylight si vystačí pouze s feature\_request, feature\_reply a konfiguračními pakety. Po zjištění nastavení a bez nějaké další aktivity, sítí putují zprávy echo\_request a echo\_reply, které udržují spojení mezi controllerem a OpenFlow switchem.

## 5.5 Test 2

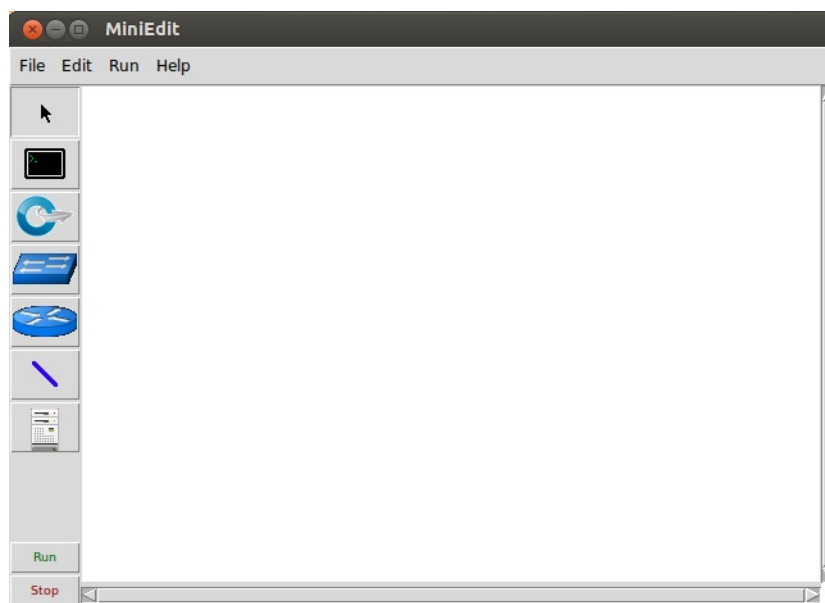
Druhý test je zaměřen na vytvoření vlastní topologie v Mininet a vyzkoušení funkcí, které námi vybrané controllery nabízejí.

## 5.5.1 Vlastní topologie

Vlastní topologii v prostředí Mininet lze vytvořit dvěma způsoby, a to buď napsáním vlastního kódu v programovacím jazyce Python nebo pomocí aplikace MiniEdit, která je součástí Mininet. V této práci je topologie vytvořena prostřednictvím grafického prostředí v programu MiniEdit.

MiniEdit je jednoduchý síťový editor pro Mininet, který umožňuje navrhovat a spravovat topologie. Skript `miniedit.py` se nachází v adresáři `example`. Chceme-li aplikaci spustit, stačí použít následující příkaz:

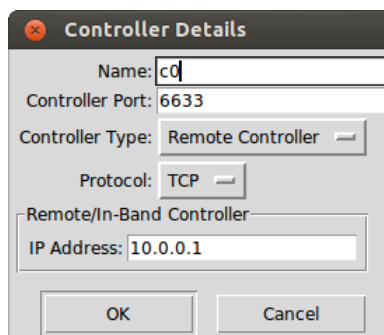
```
root@mininet:~# python /mininet/example/miniedit.py
```



**Obrázek 5.9:** MiniEdit - grafické uživatelské rozhraní

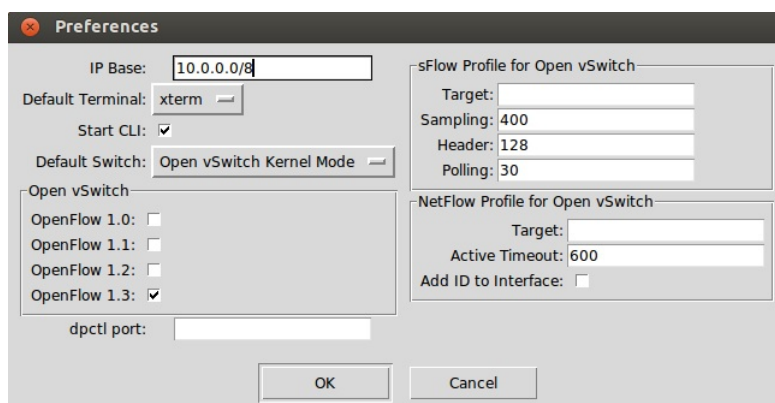
Snadné uživatelské rozhraní, které zahrnuje řadu ikon nástrojů na levé straně okna a bílé plátno, dovoluje rychlé a přehledné vytvoření topologie, při kterém je možné vybírat z několika prvku: controller, OpenFlow switch, koncový počítač, klasický switch, router a linky, které tyto prvky spojí. Přidání síťového prvku do návrhu topologie je velmi snadné, stačí pouze kliknout na zvolenou ikonu a přetáhnout jej na plátno. Stejným principem se i prvky propojí linkami. [24]

Kliknutím pravým tlačítkem myši na jakýkoli prvek v topologii je možné jej nastavit tak, jak potřebujeme. Především u controlleru se jedná o důležité nastavení, lze změnit jméno, port, typ controlleru (např. RemoteController), IP adresu a protokol, kterým bude komunikovat.



**Obrázek 5.10:** MiniEdit - nastavení controlleru

Dále můžeme změnit všeobecné nastavení sítě dostupné v hlavním menu MiniEdit přes tlačítko Edit a poté Preferences. V nově zobrazeném okně lze nastavit rozsah IP adres, který bude síti přidělen, zapnout nebo vypnout Mininet CLI, jaká verze OpenFlow protokolu bude použita a další.



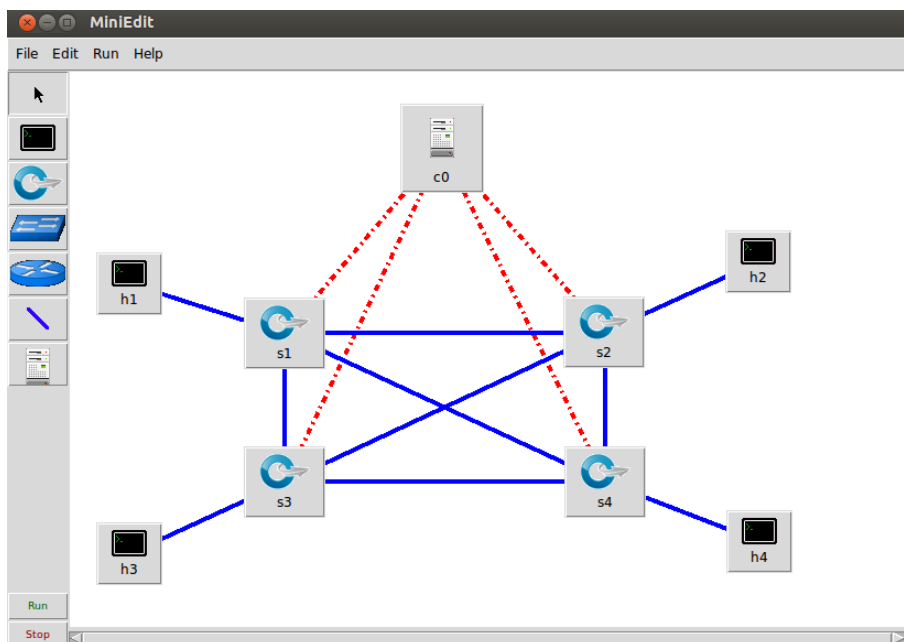
**Obrázek 5.11:** MiniEdit - všeobecné nastavení SDN sítě

Po dokončení lze vytvořenou topologii uložit, exportovat jako Mininet Python script nebo přímo spustit pomocí tlačítka start. Díky MiniEdit jsme tak schopni vytvořit vlastní složitou topologii sítě během několika minut, kdežto ruční psaní Python scriptu by trvalo mnohem déle.

## 5.5.2 Testování

Pro testování bude použita vlastní mesh topologie vytvořená v MiniEdit, zahrnující čtyři OpenFlow switche a stejný počet koncových počítačů, přičemž ke každému switchi je vždy připojen jeden počítač. Test bude zaměřen především na námi vybrané controllery a jejich funkce, které uživateli nabízejí.

Mesh topologie je takový druh topologie, kde každý prvek v síti (v tomto případě OpenFlow switch) je propojený se všemi ostatními.



Obrázek 5.12: Mesh topologie vytvořená v MiniEdit

Po připojení sítě ke controlleru jsou všechny OpenFlow switche a koncové počítače v síti dostupné a schopné komunikovat s ostatními. Veškeré cesty v síti tedy nastaveny a lze prověřit funkčnost spojení pomocí testu ping. V Mininet existuje pokyn, který dokáže kontaktovat všechny počítače jedním příkazem, a to: pingall. Jak je možné vidět z obrázku, počítače se v síti navzájem vidí a komunikují spolu.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
```

Obrázek 5.13: Příkaz pingall v Mininet ověřující dostupnost všech koncových stanic

## OpenDaylight controller

Prvním nastavením je editace switche, u kterého v OpenDaylight controlleru můžeme změnit název, úroveň, ve které bude pracovat a provozní režim.

V našem případě je jméno switche změněno na S1 a další switche mají stejný název, liší se jen jejich číslo. Úroveň není vybrána žádná protože všechny čtyři switche v síti jsou

na stejné vrstvě a v nabídce provozního režimu ponecháme výchozí hodnotu. Stejným způsobem nastavíme i další switche v síti.

**Update Node Information**

Node ID  
OF|00:00:00:00:00:00:01

Node Name  
S1

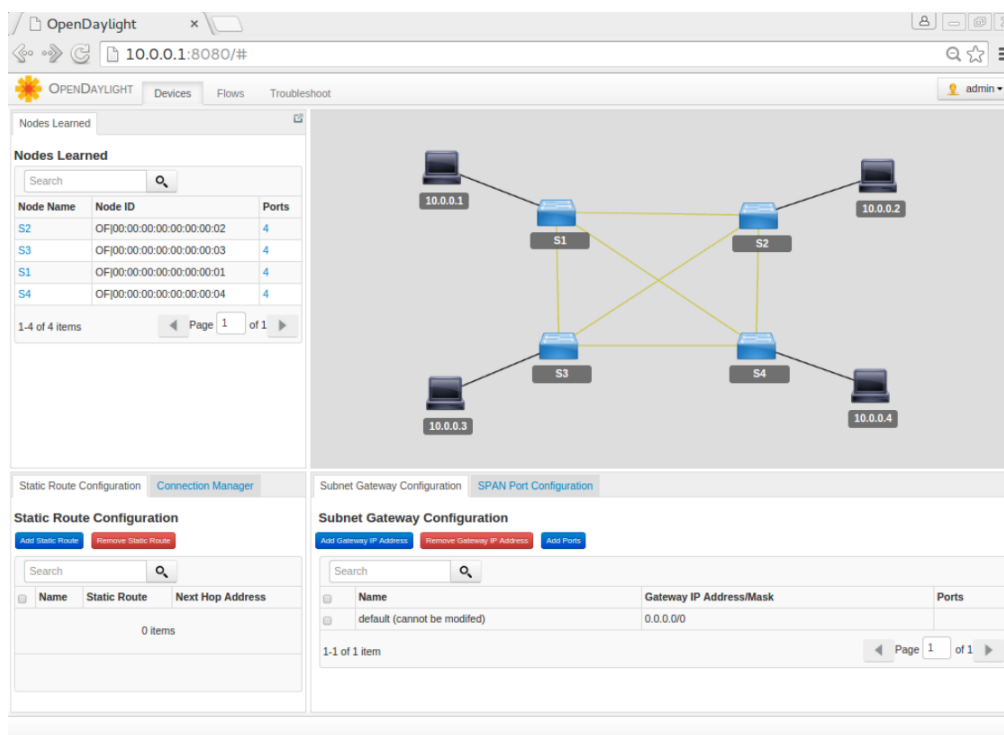
Tier  
Unknown

Operation Mode  
Allow reactive forwarding

Save

**Obrázek 5.14:** OpenDaylight - editace nastavení switche

Výsledná topologie po editaci switchů vypadá následovně:

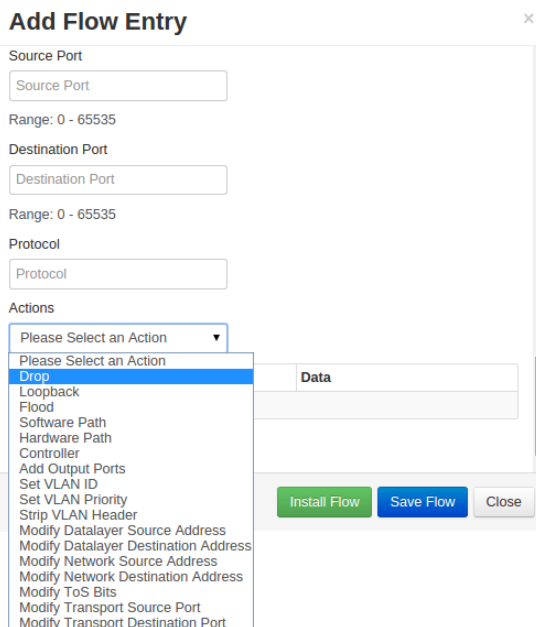


**Obrázek 5.15:** OpenDaylight - upravená mesh topologie

Dále controller umožňuje přidání záznamu toků a nabízí tak mnoho různých akcí a nastavení aplikovatelných na vybraný port switche jako jsou:

- přeposlat nebo zahodit pakety,
- nastavit odchozí port příchozích paketů,

- přidat port do určité VLAN,
- změnit zdrojovou a cílovou IP nebo MAC adresu,
- a další.



**Obrázek 5.16:** Akce, které umožňují OpenDaylight controller provádět na portu switche

Například, jestliže chceme zablokovat komunikaci hosta h1, dáme na specifický port switche, ke kterému je připojen počítač h1, pravidlo říkající všem paketům směřující na tento port, aby byli zahozeny (drop), a tak se nedostali do svého cíle.

Jak je možné pozorovat z následného testu pingall, spojení počítače h1 s ostatními je ve výpisu znázorněno písmenem X, což znamená, že komunikace s touto stanicí neprobíhá. Takto lze vyřadit kterýkoli prvek v síti a v případě hrozícího nebezpečí jej rychle a účinně eliminovat.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X X
h2 -> X h3 h4
h3 -> X h2 h4
h4 -> X h2 h3
*** Results: 50% dropped (6/12 received)
```

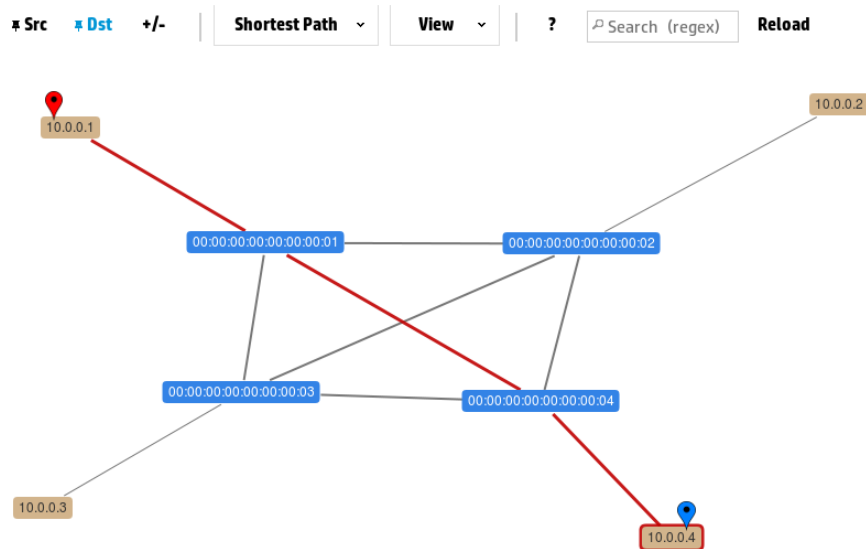
**Obrázek 5.17:** OpenDaylight - ověření nedostupnosti počítače h1 v síti

## HP VAN SDN controller

Poněvadž je HP VAN SDN controller komerční, mnoho funkcí v základní verzi nenabízí a veškeré doplňkové služby jsou tak zpoplatněné. Ve verzi zdarma jsou kromě různého nastavení využitelné v podstatě jen dvě funkce:

- Grafické nalezení nejkratší cesty v síti (Dijkstrův algoritmus)
- Zachytávání OpenFlow komunikace (OpenFlow zprávy)

Controller umí nalézt a vypočítat nejkratší cestu napříč sítí u všech prvků a poté ji graficky znázornit v topologii. V GUI controlleru stačí vybrat a označit prvky, které budou zastupovat zdroj (Src) a cíl (Dst) a následně bude mezi nimi nalezena nejkratší cesta v síti.



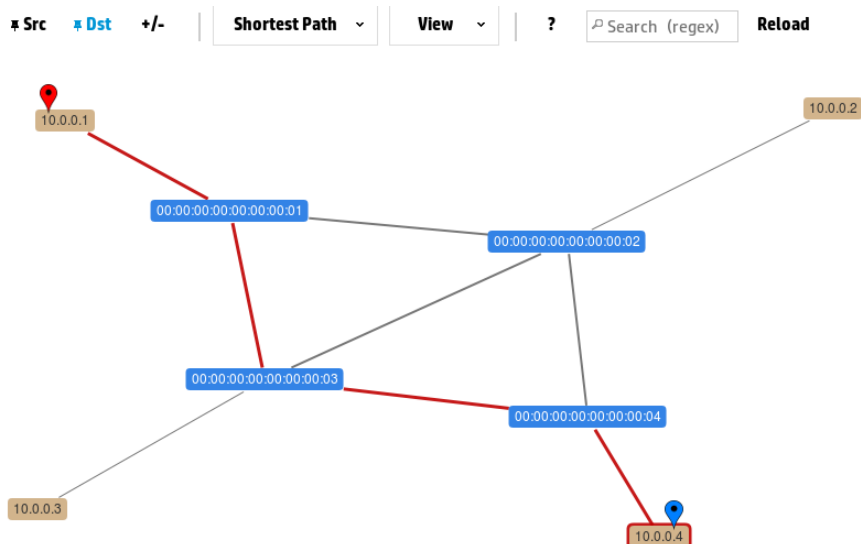
**Obrázek 5.18:** HP VAN SDN - nalezení nejkratší cesty v síti

Pokud se některá linka, která je označena jako součást nejkratší cesty, odpojí, okamžitě dojde k přepočítání a nalezní nové cesty k cíli, pokud to zapojení sítě dovoluje. V Mininet toto lze snadno simulovat příkazem určující odpojení linky mezi dvěma switchi, v našem případě jsou to switche č. 1 a č. 4:

```
mininet> link s1 s4 down
```

Poté dojde k výpočtu nové alternativní cesty a jejímu grafickému vyjádření v controlleru.





Obrázek 5.19: HP VAN SDN - nalezení nejkratší cesty v síti, po odpojení jedné z linek

Jestliže se závada odpojené linky vyřeší a ta se znovu připojí do sítě, opět se cesta přepočítá na nejkratší, tu původní. V Mininet vypadá příkaz pro přidání linky zpět takto: `mininet> link s1 s4 up`

Dále HP VAN SDN controller výrazně usnadňuje zaznamenávání OpenFlow komunikace v síti. Tato funkce je tedy obsažená přímo v GUI controlleru a pro analyzování spojení tedy není potřeba žádný další software jako například námi použitý Wireshark.

The screenshot shows the HP VAN SDN Controller GUI. The left sidebar contains navigation options like Alerts, Applications, Configurations, Audit Log, Licenses, Support Logs, OpenFlow Monitor, OpenFlow Topology, and OpenFlow Trace (selected). The main area displays the 'General / OpenFlow Trace' tool with a table of events.

Time	Event	Data Path ID	Message
13:45:05....	CkPt		Recording started [10s]
13:45:06....	Rx	00:00:00:00:00:0...	{ofm:[V_1_0,ECHO_REQUEST,8,0]}
13:45:06....	Tx	00:00:00:00:00:0...	{ofm:[V_1_0,ECHO_REPLY,8,0]}
13:45:08....	Rx	00:00:00:00:00:0...	{ofm:[V_1_0,ECHO_REQUEST,8,0]}
13:45:08....	Tx	00:00:00:00:00:0...	{ofm:[V_1_0,ECHO_REPLY,8,0]}
13:45:09....	Tx	00:00:00:00:00:0...	{ofm:[V_1_0,PACKET_OUT,91,2268],acts=[[Act:[OUTPUT,len=8],port=0x...
13:45:09....	Tx	00:00:00:00:00:0...	{ofm:[V_1_0,PACKET_OUT,91,2269],acts=[[Act:[OUTPUT,len=8],port=0x...
13:45:09....	Tx	00:00:00:00:00:0...	{ofm:[V_1_0,PACKET_OUT,91,2270],acts=[[Act:[OUTPUT,len=8],port=0x...
13:45:09....	Tx	00:00:00:00:00:0...	{ofm:[V_1_0,PACKET_OUT,91,2271],acts=[[Act:[OUTPUT,len=8],port=0x...
13:45:09....	Tx	00:00:00:00:00:0...	{ofm:[V_1_0,PACKET_OUT,91,2272],acts=[[Act:[OUTPUT,len=8],port=0x...
13:45:09....	Tx	00:00:00:00:00:0...	{ofm:[V_1_0,PACKET_OUT,91,2273],acts=[[Act:[OUTPUT,len=8],port=0x...
13:45:09....	Tx	00:00:00:00:00:0...	{ofm:[V_1_0,PACKET_OUT,91,2274],acts=[[Act:[OUTPUT,len=8],port=0x...
13:45:09....	Tx	00:00:00:00:00:0...	{ofm:[V_1_0,PACKET_OUT,91,2275],acts=[[Act:[OUTPUT,len=8],port=0x...
13:45:09....	Tx	00:00:00:00:00:0...	{ofm:[V_1_0,ECHO_REQUEST,8,2276]}
13:45:09....	Rx	00:00:00:00:00:0...	{ofm:[V_1_0,ECHO_REQUEST,8,2277]}
13:45:09....	Rx	00:00:00:00:00:0...	{ofm:[V_1_0,ECHO_REPLY,8,2277]}
13:45:10....	Rx	00:00:00:00:00:0...	{ofm:[V_1_0,ECHO_REPLY,8,2276]}
13:45:11....	Tx	00:00:00:00:00:0...	{ofm:[V_1_0,ECHO_REQUEST,8,2278]}
13:45:12....	Rx	00:00:00:00:00:0...	{ofm:[V_1_0,ECHO_REPLY,8,2278]}

Total Rows: 26

Obrázek 5.20: OpenFlow Trace - nástroj na zachytávání OpenFlow komunikace

### 5.5.3 Výsledky

Grafické prostředí obou controllerů poskytuje snadné a přehledné ovládání a ani vytvoření testované topologie v MiniEdit nebylo obtížné, proto práce v GUI těchto nástrojů byla velmi intuitivní.

OpenDaylight controller nabízí více funkcí, a tak jsme mohli lépe otestovat jeho chování v SDN. Komerční HP VAN SDN controller poskytuje v námi použité verzi „zdarma“ jen dvě funkce, které ale měli spíše informativní charakter.

Při testu bylo zjištěno i několik nedostatků SDN controllerů. Například, pokud jsme chtěli v MiniEdit vytvořit rozsáhlejší topologii (10 a více prvků v síti), po spuštění v Mininet s ní nebylo možné pracovat, program se zasekl a bylo potřeba restartovat počítač.

Další problém byl ve vyobrazení topologie v GUI OpenDaylight controlleru, který často znázorňoval jen část sítě a ani po delší době nedošlo k nápravě. Podobný nedostatek se objevoval i u HP VAN SDN controlleru, ale ne v takové míře. Spíše než chyba controlleru je toto vada síťového emulátoru Mininet, který neposkytl dostatek potřebných informací k vytvoření a zobrazení topologie.

## 6 Závěr

Technologie SDN je velkým přínosem do budoucna a může změnit celé chápání a řízení počítačových sítí. Do dalších let poskytuje ty nejlepší předpoklady k úspěchu, tedy širokému praktickému nasazení. Ale zatím zůstává otevřeno, v jaké konkrétní formě to bude. Obecný cíl SDN je ale pro všechny stejný – zjednodušení sítí a udržení kroku se zaváděním této nové technologie.

SDN je určeno především pro velké sítě a datová centra, nicméně již existují modely pro aplikování i v menších sítích jako je LAN. Reálné nasazení SDN zatím skýtá mnoho problému, z něhož největším je škálovatelnost celého řešení. Doposud jediný Google prakticky nasadil a používá technologii SDN, a to na své WAN síti, přičemž veškeré síťové prvky si vytvořily vývojáři v Googlu.

S pojmem SDN je často spojovaný protokol OpenFlow, avšak není to jedno a to samé. OpenFlow je prvním a nejrozšířenějším protokolem v prostředí SDN. Organizace ONF jej specifikovala jako způsob vzájemné komunikace controlleru a síťových prvků kompatibilní s OpenFlow, který umožňuje abstrahovat celou strukturu sítě, adresovat v ní a všeobecně spravovat všechny informace o topologii či stavu.

Pro testování byli vybrány dva softwarové controllery – open-source OpenDaylight a komerční HP VAN SDN. Oba nabízejí přívětivé uživatelské prostředí s několika funkcemi, přičemž OpendayLight nabízí více těch, které jsou užitečné pro management a správu sítě. Controllery jsou stále ve vývoji a v současné době není jejich praktické nasazení reálné, a tak jsou zatím určeny na testování a vyzkoušení všech funkcí, které nová technologie SDN nabízí.

Proto bych nedoporučoval investovat do současných řešení softwarově definovaných sítí, ale vyplatí se vyčkat na nové SDN prvky od velkých firem, které budou poskytovat více funkcí a možností.

# Zdroje

- [1] NADEAU, Thomas D. *SDN: software defined networks* [online]. Cambridge: O'Reilly, c2013, xxvii, 352 s. [cit. 2014-12-22]. ISBN 978-1-449-34230-2.
- [2] BIGELOW, Stephen J. *Mistrovství v počítačových sítích: správa, konfigurace, diagnostika a řešení problémů*. Vyd. 1. Překlad Petr Matějů. Brno: Computer Press, 2004, 990 s. [cit. 2014-12-22] ISBN 80-251-0178-9
- [3] HORÁK, Jaroslav a Milan KERŠLÁGER. *Počítačové sítě pro začínající správce*. 5., aktualiz. vyd. Brno: Computer Press, 2011, 303 s. [cit. 2014-12-22] ISBN 978-80-251-3176-3.
- [4] SHUKLA, Vishal. *Introduction to software defined networking: OpenFlow*. North Charleston: CreateSpace, c2013, ix, 103 s. [cit. 2014-12-22] ISBN 978-1-48267-813-0.
- [5] Software-Defined Networking (SDN) Definition. *Open Networking Foundation* [online]. 2012 [cit. 2014-12-27]. Dostupné z: <https://www.opennetworking.org/sdn-resources/sdn-definition>
- [6] Software-Defined Networking: The New Norm for Networks. *Open Networking Foundation* [online]. 2012 [cit. 2014-12-27]. Dostupné z: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>
- [7] Centralized vs. Distributed vs. Hybrid SDN – Which is the Best Approach for Today's Networks?. *Netsocket* [online]. 2013 [cit. 2015-02-03]. Dostupné z: <http://www.virtualnetwork.com/blog/2013/08/centralized-vs-distributed-vs-hybrid-sdn-which-is-the-best-approach-for-todays-networks>
- [8] Braun, W.; Menth, M. *Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices*. *Future Internet* 2014, 6, 302-336. [online]. [cit. 2015-02-20]. Dostupné z: <https://www.mdpi.com/1999-5903/6/2/302/pdf>

- 
- [9] OpenNetworking.org: OpenFlow Switch Specification 1.0.0. *Open Networking Foundation* [online]. 2009 [cit. 2015-02-12]. Dostupné z: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf>
- [10] OpenNetworking.org: OpenFlow Switch Specification 1.4.0. *Open Networking Foundation* [online]. 2013 [cit. 2015-02-12]. Dostupné z: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf>
- [11] Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers. [online]. [cit. 2015-02-20]. Dostupné z: <http://sit.sit.fraunhofer.de/mne/publications/download/SDNControllers.pdf>
- [12] List of OpenFlow Controllers for SDN. *The Tech* [online]. 2012 [cit. 2015-02-20]. Dostupné z: <http://www.thetech.in/2012/12/list-of-openflow-controllers-for-sdn.html>
- [13] OPENDAYLIGHT - AN OPEN SOURCE COMMUNITY AND MERITOCRACY FOR SOFTWARE-DEFINED NETWORKING. *OpendayLight* [online]. [cit. 2015-02-20]. Dostupné z: <http://www.opendaylight.org/publications/opendaylight-open-source-community-and-meritocracy-software-defined-networking>
- [14] HP Virtual Application Networks SDN Controller. *HP* [online]. 2013 [cit. 2015-02-20]. Dostupné z: <http://h17007.www1.hp.com/docs/networking/solutions/sdn/4AA4-8807ENW.PDF>
- [15] OPENDAYLIGHT DELIVERS OPEN SOURCE SOFTWARE TO ENABLE SOFTWARE-DEFINED NETWORKING. *OpendayLight* [online]. 2013 [cit. 2015-02-20]. Dostupné z: <http://www.opendaylight.org/announcements/2013/09/opendaylight-project-releases-new-architecture-details-its-software-defined>
- [16] Prepare for software-defined networking: Build the foundation for SDN with OpenFlow. [online]. 2013 [cit. 2015-02-20]. Dostupné z: <http://people.ucsc.edu/~warner/Bufs/hp2920-openflow.pdf>
- [17] HP Launches Industry's First SDN App Store, Unleashing New Wave of Networking

- 
- Innovations. *HP* [online]. 2014 [cit. 2015-02-20]. Dostupné z: <http://www8.hp.com/us/en/hp-news/press-release.html?id=1798074#.VSwfgFSsXEZ>
- [18] HP VAN SDN Controller 2.4 Administrator Guide. *HP* [online]. 2014 [cit. 2015-02-20]. Dostupné z: <http://h20564.www2.hp.com/hpsc/doc/public/display?docId=c04495133>
- [19] Mininet Overview. *Mininet* [online]. [cit. 2015-02-20]. Dostupné z: <http://mininet.org/overview/>
- [20] Release/Hydrogen/Base/Installation Guide. *Opendaylight* [online]. 2014 [cit. 2015-03-10]. Dostupné z: [http://wiki.opendaylight.org/view/Release/Hydrogen/Base/Installation\\_Guide](http://wiki.opendaylight.org/view/Release/Hydrogen/Base/Installation_Guide)
- [21] HP VAN SDN Controller Installation Guide. *HP* [online]. 2013 [cit. 2015-03-10]. Dostupné z: <http://h20564.www2.hp.com/hpsc/doc/public/display?docId=c03998700>
- [22] Download/Get Started With Mininet. *Mininet* [online]. [cit. 2015-03-10]. Dostupné z: <http://mininet.org/download/>
- [23] Mininet Walkthrough *Mininet* [online]. [cit. 2015-03-11]. Dostupné z: <http://mininet.org/walkthrough/>
- [24] Mini Edit App. [online]. [cit. 2015-03-11]. Dostupné z: <http://github.com/mininet/mininet/wiki/Mini-Edit-App/>



**UNIVERZITA HRADEC KRÁLOVÉ**

**Fakulta informatiky a managementu**

Rokitanského 62, 500 03 Hradec Králové, tel: 493 331 111, fax: 493 332 235

## **Zadání k závěrečné práci**

Jméno a příjmení studenta:

**Radek Vašátko**

Obor studia:

Informační management (3)

Jméno a příjmení vedoucího práce:

**Jakub Pavlík**

Název práce:

**Analýza protokolu OpenFlow v LAN**

Název práce v AJ:

Analysis of OpenFlow Protocol in Local Area Networks

Podtitul práce:

Podtitul práce v AJ:

Cíl práce: Cílem této bakalářské práce je seznámit čtenáře se základy SDN a protokolem OpenFlow. Porovnání současných řešení počítačových sítí a technologie SDN. Dále bude zvážena reálnost nasazení technologie SDN.

Osnova práce:

Úvod

Principy softwarově definovaných sítí

OpenFlow

Analýza prostředí pro nasazení SDN

Zavěr

Projednáno dne:

Podpis studenta

Podpis vedoucího práce