Univerzita Hradec Králové
Fakulta informatiky a managementu
katedra informačních technologií

# Možnosti nasazení SDN ve firemním prosředí

## Bakalářská práce

**Autor:** Tomáš Hradecký

**Studijní obor:** Aplikovaná informatika

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

Hradec Králové duben 2015

## Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a uvedl jsem všechny použité prameny a literaturu.

V Hradci Králové dne 27. dubna 2015 Tomáš Hradecký

## Poděkování

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

## Anotace

Tato bakalářská práce je zaměřena na problematiku technologie SDN a její možnosti nasazení na sítě v oblasti firemního prostředí, které vyžaduje dobře fungující infrastrukturu.

V této práci je vysvětlen princip SDN zaměřený na oddělení síťových funkcí na jednotlivé komponenty. Práce vysvětluje klíčové pojmy a cíle spjaté s technologií, jako je control plane, data plane a probírá také spojení SDN s cloudovým prostředím. Byla také zvolena a popsána dvě řešení pro virtualizaci a automatizaci počítačových sítí spojené s platformou pro cloud OpenStack, mezi kterými bylo provedeno také funkční porovnání.

V praktické části je pak nasazeno jedno ze dvou zkoumaných řešení. Zvolené řešení je dále podrobeno sérii testů zkoumajících vlastnosti při různých konfiguracích daného řešení.

## Annotation

**Title: Implementation of SDN on enterprise networks**

This bachelor thesis is focused on the issue of SDN technology and its capability to be used in enterprise network environments, which requires a well-functioning infrastructure.

This work explains SDN principle aimed on separation of networks function into individual components. It also describes key concepts and objectives associated with SDN technology such as control plane, data plane and connection with cloud environments. Two SDN solution for virtualization and automatization of computer networks associated with cloud platform OpenStack were chosen to describe and for mutual comparison.

In the practical part, there is implementation of one of two examined solutions. The implemented solution is then subjected to series of tests investigating the properties of various configuration of the solution.

# Obsah

# Seznam obrázků

# Seznam tabulek

# 1 Úvod

Počítačové sítě již od svých počátků čelí několika problémům, které se spolu s nimi nesou až do dneška. Náročná správa, nutná obměna hardware pro možné nasazení nejnovějších technologií a v dnešní době datových center i problematické nasazení pro cloudová prostředí. Všechny tyto problémy vedly k myšlence jak zajistit programovatelnost počítačových sítí, která by zároveň zjednodušila správu, ale také zvýšila flexibilitu a škálovatelnost.

Myšlenka se začala měnit ve skutečnost s příchodem SDN, neboli Software Defined Networking. Tato technologie se zaměřuje na oddělení softwarové části a hardware díky čemuž by měla přinést do oblasti počítačových sítí hledanou programovatelnost a možnou automatizaci.

Význam SDN je značně spjat i s řešením virtualizace a automatizace počítačových sítí v prostředí cloudu, čímž výrazně posílil potenciál technologie pro nasazení na IaaS.

Tato práce se zaměřuje na prozkoumání problematiky SDN. Cílem práce je prozkoumání dvou zvolených řešení a následně jejich funkční porovnání. Pro implementace a testování bude na základě porovnání zvoleno jedno řešení.

Obsah bakalářské práce je kromě úvodu členěn do 6 kapitol. V následující kapitole je popsán princip SDN. V kapitole 3 jsou popsána dvě vybraná SDN řešení z pohledu struktury i podporovaných funkcí. Kapitola 4 se zabývá srovnáním nabízených funkcí obou zvolených řešení. Implementace s testováním jsou popsány v kapitole 5. V kapitole 6 jsou zhodnoceny výsledky dosažené při testování nasazeného řešení. V závěrečné kapitole 7 jsou poté uvedeny závěry a doporučení získané v průběhu práce.

V obsahu bakalářské práce jsou používány některé názvy v původním anglickém znění. K tomuto rozhodnutí mě vedla především skutečnost, že v technické literatuře i praxi jsou tyto pojmy běžně zmiňovány v původním znění, což zvyšuje jednoznačnost.

# 2 Principy SDN

V následující kapitole jsou popsány základní principy používané v technologii SDN. Je rozebrána architektura technologie, hlavní cíle i způsob jakým se jich snaží dosáhnout. Dále jsou zmíněny důvody proč se zajímat o použití a stálý rozvoj technologie SDN.

## 2.1 Proč právě SDN

Aby bylo možné odpovědět na otázku proč se zajímat o SDN, je nejdříve nutné, shrnout nejvýznamnější problémy se kterými bojují počítačové sítě při tradičním řešení. Mezi tyto problémy patří následující:

- složitá správa
- složitý vývoj
- vysoká komplexnost
- téměř nulová programovatelnost
- nepřehledná standardizace (okolo 5400 RFC)
- složité nasazení v cloudovém prostředí
- jediné co se mění je přenosová rychlost
- nekompatibilita mezi různými výrobci
- rozvoj BigData(vyžaduje vysokou síťovou propustnost)
- složité techniky pro multicastové vysílání

Tyto faktory zapříčinily, že v roce 2004 byl započat výzkum jak vylepšit způsob jakým spravovat počítačové sítě a zjednodušit vývoj nových technologií. Následuje je krátký přehled hlavních historických událostí:

- 2004 - začátek výzkumu jak spravovat počítačové sítě
- 2008 - představení SDN a OpenFlow (Nicira/Stanford)
- 2011 - založení ONF (Open Networking Foundation) - organizace společností z oblasti SDN
- 2011 - ONF přebírá OpenFlow
- 2011 - ONF standardizuje OpenFlow
- 2011 - ONF pořádá první ONS (Open Networking Summit) - představení novinek za uplynulý rok
  [6] [1]

Při návrhu SDN byla využívána vysoká úroveň abstrakce, která umožnila stanovení jasného a jednoduchého úkolu každé části architektury. [3] [5]

Pro řešení problémů se správou počítačových sítí, přináší SDN myšlenku centralizace, jejíž možnosti jsou dnes velmi omezené. Je tedy nutné konfigurovat každý prvek jednotlivě, což je jednak značně časově náročné a navíc se tím poskytuje prostor pro zanášení chyb lidského faktoru. [5]

Další cílem, na který se SDN zaměřuje je programovatelnost sítí, která by umožnila jednodušší nasazení nejnovějších technologií. K dosažení tohoto cíle bylo využito oddělení software a hardware a následné oddělení data plane a control plane složek v software. Tento přístup slibuje umožnit nasazení nové technologie pomocí software bez nutnosti měnit veškerý síťový hardware v topologii, jak tomu bylo doposud. Přínos je dále spojen zároveň s minimalizací rozhodování, které je nutné provést na síťových prvcích, což má vest vést ke zrychlení celkové síťové komunikace. [1] [2]

Hlavní cíle SDN dají shrnout do následujících bodů:

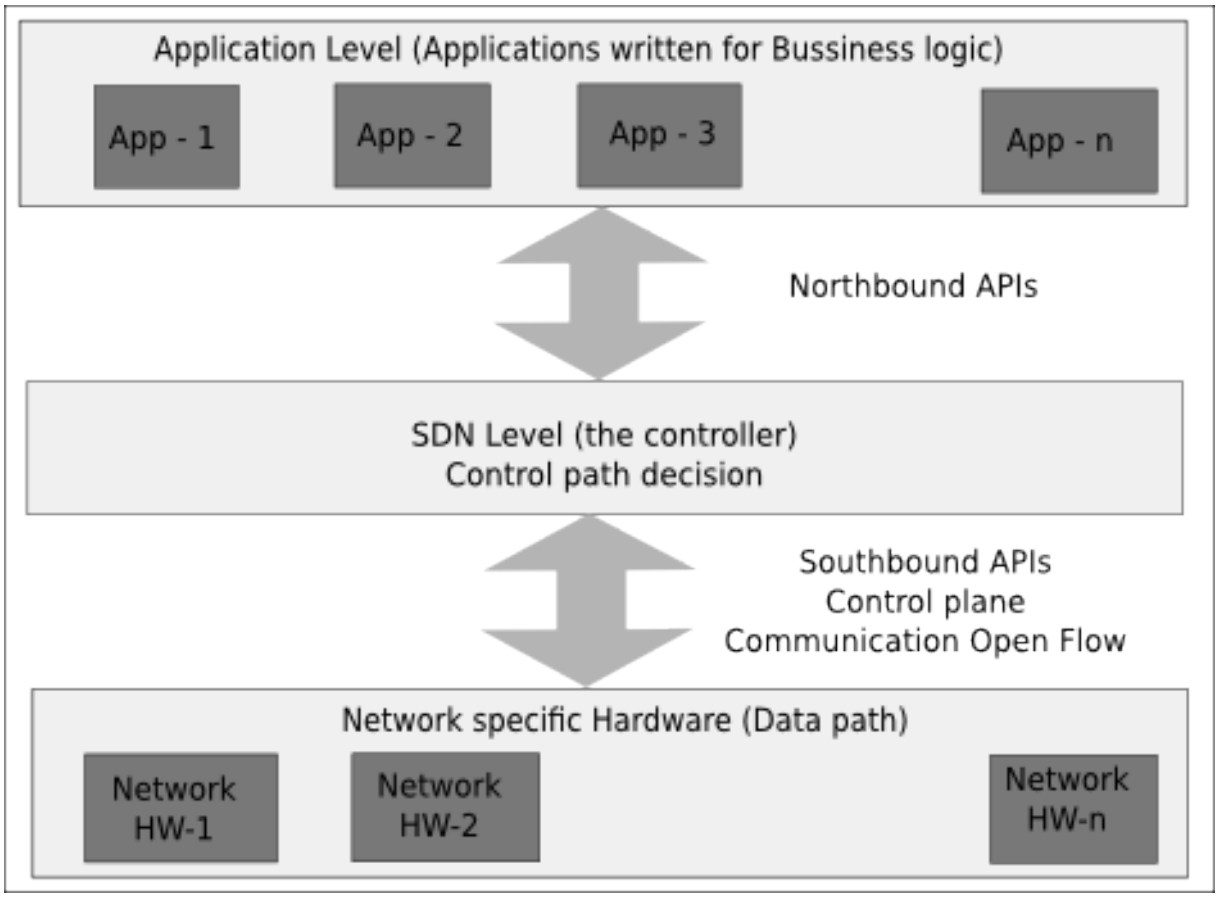
- oddělení Data plane a Control plane
- centralizace Control plane

- programovatelnost Control plane
- automatizace správy

SDN technologie se také zaměřuje na zjednoduší postupů potřebných pro nasazení multicastových přenosů. Momentálně patří multicastová komunikace mezi velmi složité problémy. Při využití SDN by měla být technika nasazení výrazně zjednodušena. Následná konfigurace má přejít do stavu, kdy na jednotlivých síťových prvcích bude aplikováno jednoduché pravidlo definující odeslání datagramu s danými parametry všem cílovým uživatelům v multicastové doméně. [7]

## 2.2 Architektura

Architektura SDN je rozdělena do několika vrstev popsaných s vhodnou úrovní abstrakce, díky níž jsou jasně definovány cíle každé vrstvy, které mezi sebou spolupracují, aby vytvořili fungující celek. SDN se vyznačuje umístěním jednotlivých částí architektury na rozdílné prvky, což vychází z jednoho z hlavních bodů uvedených v myšlence a to oddělení control plane a data plane. [2]




**Obrázek 2.1:** Logické vrstvy architektury SDN, převzato a upraveno dle: [2]

### 2.2.1 Logické vrstvy SDN

Jak je podle obrázku 2.1 patrné, architekturu SDN lze rozdělit do několika logických vrstev. Tyto části jsou následující:

- Application level (aplikační úroveň)
- Northbound APIs
- SDN level (the controller)

  Control plane
- Southbound APIs
- Network specific Hardware (konkrétní síťový hardware)

  Data plane

### 2.2.2 Application level

Na úplném vrcholu se nachází samotné aplikace, poskytující síti veškerou funkcionalitu jako jsou například DHCP, ACL, NAT, DNS a mnoho dalších. Jejich vytváření by mělo být poskytováno prostřednictvím nižší vrstvy, nazývané northbound API. Zároveň by však mělo být možné nasadit aplikace od třetích stran pro rozšíření funkcionality. [2] [1]

### 2.2.3 Northbound APIs

Mezi nejvyšší vrstvou (Application layer) a střední vrstvou(Controller layer) se nachází prostor pro Northbound API, která bude uživateli poskytovat prostor pro definici aplikací a zároveň těmto aplikacím umožní komunikaci s SDN controlerem. SDN controller pak může tyto aplikace za pomoci Southbound API přeložit do formy, které porozumí i nižší vrstvy (Network specific Hardware) a následně aplikovat potřebná nastavení pro zajištění funkcionality daných aplikací. Momentálně je nabídka northbound API dosti prázdná, což je jednou z nevýhod SDN. Počítá se však, že s pokračujícím rozvojem SDN technologie bude počet dostupných Northbound API růst. Mezi momentálně dostupné patří Quantum API(dnes známé jako modul Neutron v cloudové platformě OpenStack), které je otevřené a je integrováno ve většině OpenFlow/SDN controllerů. [2]

### 2.2.4 SDN level - Controller layer

Do této vrstvy spadají SDN controllery (v případě že komunikují pomocí Southbound API OpenFlow, je někdy používané označení OpenFlow controller), což jsou softwarové entity, které disponují globálním pohledem na celou síťovou strukturu. Jelikož disponují kompletním pohledem na topologii mohou nejlépe rozhodovat o cestách, které jsou zvoleny pro doručování dat. Tato rozhodnutí mohou odeslat na jednotlivé síťové prvky, aby se podle nich naprogramovaly a nadále se mohli rozhodovat bez nutnosti komunikace s SDN controllerem.

Momentálně je většina SDN cotrollerů vybavena grafickým rozhraním, které poskytuje administrátorům grafické znázornění síťové topologie především za účelem snadnější správy. Software nacházející se na SDN controlleru je control plane. Jeho

oddělení od data plane je jednou z hlavních myšlenek, které SDN sleduje. [1] [6]

#### Control plane

Control plane je část software, ve které jsou uloženy informace pro vytvoření jednotlivých záznamů reprezentovaných uvnitř forwarding table (tabulky pro předávání dat). Tyto informace jsou poté prostřednictvím data plane použity k předávání datového provozu v síti.

K vytvoření síťové topologie slouží informace zvané RIB - Routing Information Base (směrovací informační báze), která je udržována konzistentní díky neustálé výměně informací mezi jednotlivými instancemi v control plane. Záznamy forwarding table jsou společně nazývány jako FIB - Forwarding Information Base (předávací informační báze). FIB je naprogramována poté co je RIB považována za stabilní a konzistentní. Právě z tohoto důvodu mají řídící entity náhled na celou topologii. Tento náhled může být naprogramován manuálně, naučen z pozorování datových toků, případně ze sledování výměny informací mezi jednotlivými řídícími entitami. [1]

### 2.2.5 Southbound APIs

Jedná se o skupinu API protokolů, které pracují mezi nejnižší vrstvou (Network specific Hardware nebo VM) a střední vrstvou (Controller layer). Jejím hlavním úkolem je komunikace, která povoluje SDN controlleru instalovat na samotné síťové prvky rozhodnutí zahrnutá v control plane z aplikací nejvyšší vrstvy . Zřejmě nejznámějším protokolem tohoto typu, který je zároveň považován za standard, je OpenFlow. [1] [2]

### 2.2.6 Network specific Hardware

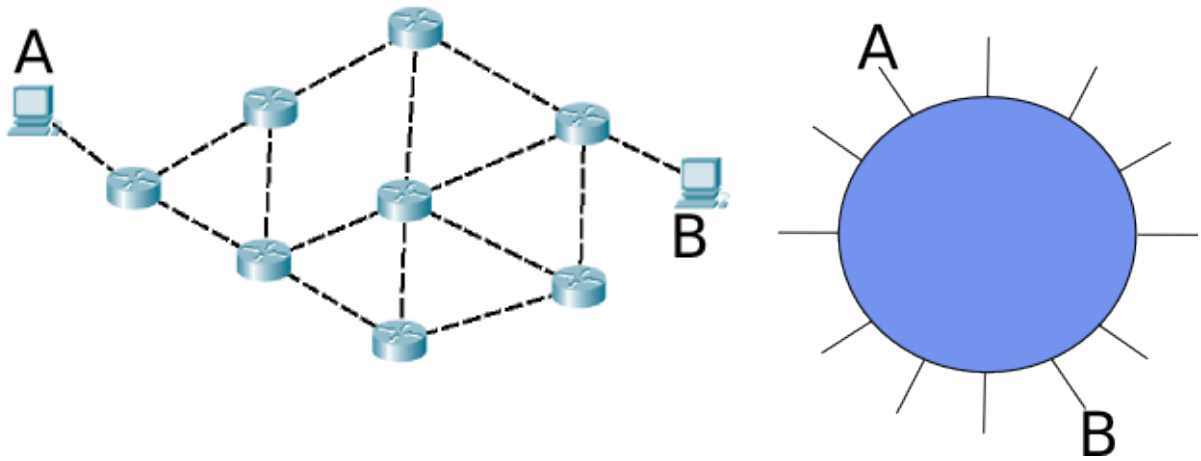
Nejnižší vrstvou je samotný hardware pro předávání datagramů na fyzické úrovni. Pro funkčnost celé architektury je nutné, aby zde byla nasazena zařízení, která umí přijímat pokyny od control plane skrze southbound API a podle nich sestavit vlastní data plane. Takováto zařízení, která umí přijímat pokyny od OpenFlow porotokolu, jsou nazývána OpenFlow switch. U těchto zařízení chybí softwarová inteligence, která je soustředěna v controlleru. Na těchto zařízeních reprezentující samotný síťový hardware tak zůstává pouze základní software v podobě data plane. Jednotlivé prvky mo-

hou být reprezentovány prostřednictvím entit VM - Virtual Machines (virtuální stroje) v prostředí pro virtualizaci. [2] [7]

### Data plane

Data plane slouží k manipulaci se samotnými datagramy příchozími do síťového zařízení přes přenosové médium. Po přijetí datagramu, proběhne reverzibilní pohled do FIB tabulky, která byla dříve naprogramována prostřednictvím southbound API aplikující funkcionalitu zahrnutou v control plane. V případě, že se jedná o datagram bez odpovídajícího záznamu v FIB, jsou informace z hlavičky datagramu zaslány na controller. Zde je podle veškerých pravidel zahrnutých v RIB rozhodnuto o následující akci. V případě, že je nalezen odpovídající záznam je datagram okamžitě předán dále. [1] [6]

Později se přidala ještě vrstva Virtual topology, která je umístěna mezi Controller layer a Northbound APIs. Tato vrstva přebírá data o vzhledu logické topologie, které předává v abstrahované podobě Northbound API pouze s nutnou úrovní podrobnosti, čímž zachovává delegování zodpovědností v architektonickém modelu SDN. Pro názornost přínosu virtuální topologie je uveden obrázek 2.2 kde je vlevo zachycena logická topologie a vpravo jak bude tato topologie popsána pro Northbound API.




**Obrázek 2.2:** Logická a virtuální topologie SDN, převzato a upraveno dle: [6]

Pokud by byl pro danou topologii definován požadavek, který zakazuje komunikaci hosta A s hostem B, bylo by nutné specifikovat jej na všech zařízeních, která by se mohla podílet na přenosu dat. Zatímco při použití abstrakce poskytované virtuální

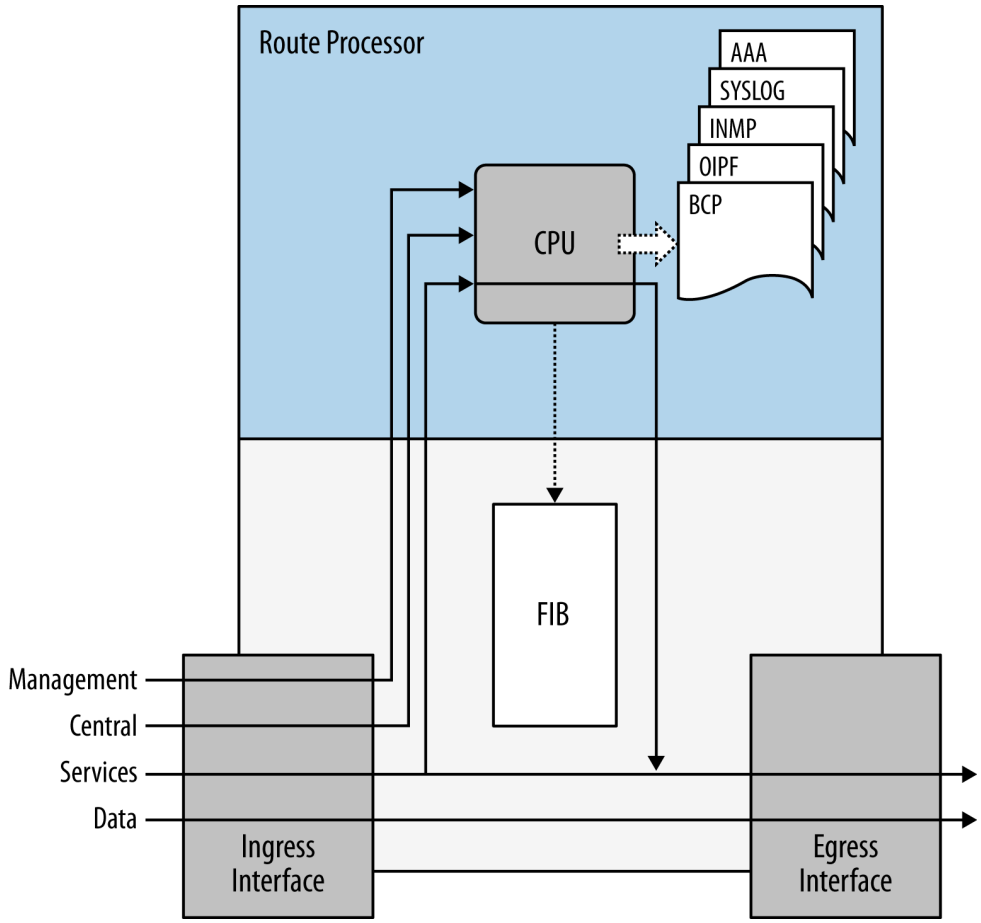
topologií je síť vnímána jako jediný velký switch, jak je vidět na obrázku 2.2 vpravo. Pro vyřešení požadavku tedy stačí pouze zadat příslušné pravidlo, v tomto případě host A nesmí komunikovat s hostem B, a o zbytek se postará nižší vrstva. Zde se opět potvrdila abstrakce jako velice silný nástroj při inovaci. [1] [6]

## 2.3 Oddělení control plane od data plane

Proč se vlastně zabývat oddělením control plane od data plane? Co to přinese a také jaká jsou rizika? Toto jsou jen některé otázky, se kterými je možno se setkat.

Ačkoli by se mohlo zdát, že se jedná o něco nového, opak je pravdou. S odděleným control plane a data plane se lze setkat v podstatě u všech víceslotových switchů a routerů. Zde má každá karta s porty vlastní procesorovou jednotku, jejíž úkolem je rychlé rozhodování kam bude příchozí datagram předán podle informací uložených v FIB. Dále je zde centrální procesorová jednotka, která má u sebe kompletní RIB. Tato jednotka se stará o šíření příslušných FIB informací do jednotlivých karet. Zároveň je využívána v případě, že na libovolnou kartu dorazí datagram, pro jehož hlavičku nemá ve své FIB odpovídající záznam. V takovém případě je hlavička neznámého datagram odeslána na centrální procesorovou jednotku, kde se rozhodne o další vykonané akci. Zároveň je toto rozhodnutí zapsáno do FIB příslušných karet. Tento přístup je naprosto shodný s jakým se lze setkat u SDN.
Výsledně pak jedno fyzické zařízení, logicky funguje jako skupina kooperujících. U SDN je pouze větší vzdálenost mezi control plane a data plane.

Z obrázku 2.3 je patrné, jaké služby jsou kde uplatňovány. Control plane je zde značen modrou částí jako Route Processor, zatímco data plane vyznačen bílou části, které dominují převážně vstupní a výstupní porty. [1]




**Obrázek 2.3:** Oddělení control plane a data plane [3]

## 2.4 Umístění control plane

Jak již bylo řečeno, control plane je část software, která slouží k vytvoření FIB. Velice důležitou otázkou je však umístění control plane. Ve standardní síťové topologii (bez nasazení SDN) je control plane v každém síťovém prvku, kde reprezentuje inteligentní část pro rozhodování ohledně síťového provozu. Na rozdíl od standardního přístupu, u technologie SDN je cílem tuto inteligentní část software oddělit. Důvodů proč se zajímat o umístění control plane je hned několik, těmi nejzávažnějšími jsou správa a výkon sítě. [2] [1]

## 2.5 Distribuovaný control plane

Distribuovaný control plane je způsob, jaký je používán u síťových topologií bez technologie SDN. Každý prvek má svou vlastní rozhodovací logiku. Podle níž se řídí

při práci s datovými toky. Aby síť fungovala správně, musí být plně konvergována. Konvergencí se rozumí stav, kdy všechny síťové prvky v celé síťové topologii mají kompletní informace pro řízení datových toků, v bez smyčkové síti. Konvergence je u topologií s distribuovaným control plane řízena prostřednictvím routovacích protokolů jako jsou například (OSPF, EIGRP, ISIS).

Nevýhodou je hlavně náročná správa. V případě změny topologie může správci trvat i několik hodin něž jej plně začlení do existující síťové topologie. Naproti tomu distribuovaný control plane přináší odolnost celé sítě. Topologie jsou vytvářeny s redundantními cestami nebo i prvky. V případě, že některý prvek selže topologie může po spuštění algoritmů na přepočet cest a následné konvergenci fungovat dále. [1]

## 2.6 Centralizovaný control plane

Centralizací control plane je možno dosáhnout jednodušší správy počítačových sítí společně se zvýšením výkonu. V SDN je bodem s centralizovaným control plane SDN controller. SDN controller se stará o konvergenci všech podřízených síťových prvků, které mají u sebe pouze data plane.

V případech, kdy přes prvky putují datagramy jejichž hlavičky odpovídají záznamům v FIB je vše velmi rychlé. Především díky minimální účasti software v rozhodovacím procesu, je možné maximálně využít rychlého ASIC procesoru, i když dnes se hlavně využívají procesory na architektuře x86 a to především díky síťové virtualizaci.

Pokud na některý prvek dorazí datagram, kterému neodpovídá v FIB žádný záznam musí být od SDN controleru získány informace co s ním. Tím vzniká problém, kdy se k jedinému controlleru může najednou dotazovat více prvků. SDN controller má pouze omezenou kapacitu a při určitém počtu dotazů již nebude schopen reagovat dostatečně rychle, což může vést ke značnému zpomalení síťové komunikace.

Z těchto důvodů je dobré uvést rozdíly mezi plně centralizovaným a logicky centralizovaným control plane. [1]

### 2.6.1 Úplná a logická centralizace control plane

Plnou centralizací rozumíme skutečnou existenci jediného prvku, na němž je uložen control plane. Logickou centralizací rozumíme i skupinu kooperujících SDN

controllerů s control plane vystupujících jako jeden logický celek. Při rozhodování zda v dané topologii využít úplnou, nebo pouze logicky centralizovanou variantu control plane je vhodné zvážit následující:

- velikost topologie

  Jediný SDN controller pro celou topologii musí udržovat spojení s každým z řízených prvků. Čím je tento počet větší, tím větší počet požadavků musí SDN controller zpracovávat. Jsou-li vyžadovány doplňkové informace, například analytická data o stavu a výkonu sítě, zátěž vyvíjená na SDN controller dále stoupá. Při určitém množství požadavků SDN controller již není schopen efektivně reagovat, což má za následek zpomalení celé sítě.[1]

- dostupnost

  I přes fakt, že celou topologii je možno řídit jediným SDN controllerem, je nutné vzít v potaz také skutečnost, že jediný SDN controller je zároveň jediný krizový bod. Při neexistujících redundantních prvcích, které by zastaly funkčnost, je tak síť v případě výpadku jediného SDN controlleru předurčena ke kolapsu. Jednotlivé prvky jsou sice schopny pracovat bez nutnosti přístupu k SDN controlleru, za předpokladu již nastavených FIB, avšak pouze po omezenou dobu. Přesněji do doby, než bude nutné rozhodnout co s datagramy z neznámého datového toku.[1]

- poloha

  Poloha vstupuje do rozhodování o centralizaci control plane především z důvodů, kdy jsou jednotlivé části topologie odděleny velkými vzdálenostmi. Data musí cestovat nejen přes velkou vzdálenost, ale také přes jiné sítě, kde mohou být podrobeny sběru analytických dat, bezpečnostním politikám nebo nízkým přenosovým rychlostem, které mohou být způsobeny mnoha důvody od malých propustností, až po různé útoky. Pokud je toto zpomalení příliš vysoké, mohou nastávat problémy s konzistenci FIB a efektivním řízením sítové topologie podobně, jako když je v síti pouze jeden přetížený SDN controller.[1]

Není možné říci přesně pro jaký počet prvků použít jaký počet SDN controllerů, mimo jiné proto, že každý SDN controller může dosahovat rozdílné výkonnosti. Toto rozhodnutí je ovlivněné příliš mnoha faktory vyplývajících z individuality požadavků v dané

síťové topologii. V takovýchto situacích je nutné se spolehnout na zkušenosti společně s uvážením skutečnosti, zda pořizovací a provozní náklady spojené s dalším SDN controllerem nejsou vyšší než přínos jím získaný.

## 2.7 SDN Domény

Jak již bylo zmíněno, jediný SDN cotroller v celé topologii má různé nevýhody, avšak v případě, že nasadíme více SDN controllerů, je nutné vyřešit jejich vzájemnou komunikaci. Každý controller musí mít vždy aktuální informace o celé spravované topologii včetně záznamů, které jsou určeny pro prvky spravované jiným SDN controllerem.

U podnikových sítí je využívané rozdělení sítě do několika nepřekrývajících se logických částí. Části jsou nazývány SDN domény. Tato varianta přináší:

- škálovatelnost

  Při velkém počtu prvků spravovaných jedním SDN controllerem je schopnost efektivního řízení omezená, což je hlavním důvodem pro rozdělení do několika SDN domén.

- bezpečnost

  SDN domény nabídnou podobně jako VLAN různé skupiny uživatelů. Díky tomu je možné aplikovat různé bezpečnostní politiky v různých doménách. [3]

- přírůstkové nasazování technologií

  Rozdělením topologie do více nezávislých domén umožní nasazení nových technologií pouze v její oddělené části. Díky tomu je možné se vyhnout rizikům nefunkční síťové topologie z důvodu nasazení neodzkoušené technologie. Poté co proběhne testování je možné na nasadit technologie do celé sítě. [3]

K tomuto účelu je určen protokol SDNi. Tento protokol zprostředkuje komunikaci mezi jednotlivými SDN controllery v různých SDN doménách, které si tak vyměňují informace zahrnuté v RIB. SDNi zahrnuje funkce:

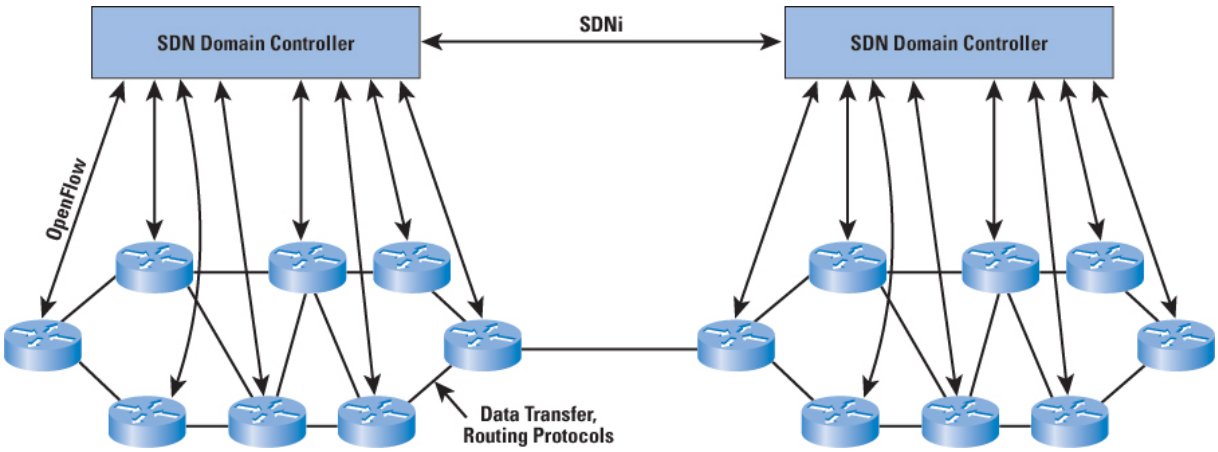
- Koordinované toky nastavení od aplikací, které obsahují informace jako QoS (Quality of Services - kvalita poskytovaných služeb kde dochází k upřednostňování

služeb s vyšší prioritou), požadavky na cestu a úroveň služeb přes více SDN domén.

- Výměna informací o dostupnosti pro správné fungování inter-SDN datových toků. Tyto informaci umožní každému SDN controlleru z dané domény vybrat ty nejvhodnější cesty do zbylých domén v celé topologii. [3]

Typy zpráv u SDNi by měli obsahovat následující:

- updaty o dosažitelnosti síťových lokací
- požadavky na update, nastavení, vypnutí toků (včetně aplikačních požadavků jako QoS, přenosové rychlosti, latence, ...)




**Obrázek 2.4:** Komunikace mezi SDN doménami [3]

## 2.8 Přístupy konvergence SDN

Při vývoji technologie SDN bylo řešeno jaký přístup zvolit pro naplnění síťových prvků řídícími instrukcemi potřebnými pro správnou funkci sítě. Řídícími instrukcemi rozumíme pravidla ovlivňující akce prováděné s datagramy prostřednictvím pravidel v FIB. V tomto ohledu existují dva možné přístupy, proaktivní a reaktivní.

### 2.8.1 Proaktivní přístup

Proaktivní přístup je založen na distribuci veškerých FIB pravidel na všechny prvky v síti, před započetím předávání samotných datagramů. Tato distribuce je oz-

načována jako konvergenční proces síťové topologie. Během konvergence není daná síť schopna plné funkčnosti. V případě rozlehlých sítí s mnoha prvky podílejících se na datových přenosech existuje riziko, že čas potřebný pro konvergenci bude příliš dlouhý a zapříčiní dlouhou nefunkčnost v případě prvotního spuštění, či při výpadku některého síťového prvku. Zároveň tento přístup přináší jistotu, že síťové prvky mají vždy veškeré informace nutné k datovému provozu, čímž nedojde k situaci, kde bude nutné dotazovat se SDN controlleru na informace pro rozhodnutí ohledně akce provedené s příchozím datagramem.

Lze tedy říci, že proaktivní přístup nabízí nízkou latenci právě díky konvergenci sítě. Nutností je počítat s časovým intervalem potřebným pro konvergenci sítě a zvážit zda rozsah dané síťové topologie není příliš velký. Příliš rozlehlá topologie by velkým obsahem záznamů vedla ke zpomalení při vyhledávání konkrétního záznamu. [1] [6]

### 2.8.2 Reaktivní přístup

Reaktivní přístup naopak od zmíněného proaktivního neprovádí konvergenci jako takovou. V tomto případě jsou informace pro předávání datagramů od SDN controlleru vyžadovány během procesu předávání vždy, když přijde neznámý datagram.

Síť s reaktivním přístupem nemusí být nikdy plně konvergována. Je tomu tak proto, že síť je považována za plně konvergovanou v případě, kdy mají všechny prvky informace pro akce se všemi datagramy, které se mohou v síti vyskytnou. Tento stav nemusí u reaktivního přístupu nikdy nastat, jelikož některé datagramy se mohou v datových tocích vyskytovat jen velice vzácně. Avšak u reaktivního přístupu není úplná konvergence nutná k funkci dané sítě. Reaktivní síť započne datagramovou komunikaci hned po spuštění, přičemž na začátku bude SDN controller dotazován na velký počet pravidel týkajících se každého neznámého datagramu. Tato situace může u sítí s velkým počtem hostů vést k dlouhé latenci, trvající do doby, než si jednotlivé síťové prvky naplní své FIB do přijatelné znalostní úrovně. Dopad dané situace by bylo možné zmírnit velkým počtem SDN controllerů připravených na velkou zátěž, je ale nutné si uvědomit, že v pozdější fázi síťové funkce již nebude velký počet SDN controllerů potřebný. U reaktivního přístupu je značně snížena pravděpodobnost přehlcení FIB. [1]

Jak je zřejmé, oba přístupy mají své výhody i nevýhody. Skloubení obou přístupu

by bylo možné dosáhnout ideálního výsledku. Momentálně je zastávána myšlenka, jejíž základem je proaktivní přístup, který však v rámci konvergenčního procesu přivede síť pouze do částečné informovanosti. Pravidla zahrnutá v této části by měla korespondovat s nejočekávanějšími datagramovými toky v síti. Zbylá pravidla by pak byla doplňována podle reaktivního přístupu.

## 2.9 SDN a Cloud

Ačkoli se může zdát že největším přínosem SDN je centralizovaná správa, není tomu tak. SDN je velice účinný nástroj pro implementaci sítí podle zákaznických potřeb v prostředí cloudu. SDN je proto velice často zmiňován společně s cloudovými technologiemi, které jsou v dnešní době pro svou žádanost velice aktuální. Díky cloudu dostává SDN mimo hromadné správy, programovatelnosti a dalších dříve zmíněných cílů využití i pro snadnější IaaS. [8] [12]

### 2.9.1 SDN jako IaaS

IaaS - Infastructure as a Service (Infrastruktura jako služba) je jedna ze služeb nabízená v cloudovém prostředí. Princip služby spočívá v poskytnutí síťové infrastruktury podle přání zákazníka virtualizované v cloudu. Tato služba neumožňuje zákazníkovi postrádat veškerou vlastní síťovou infrastrukturu. Umožňuje ji ale značně zjednodušit. Přístupy pro řešení jsou často takové, které abstrahují od fyzické infrastruktury na níž vše funguje. V takové situaci se mluví o overlay síti, kde packety putují zapouzdřené v tunelovacím protokolu do jehož komunikace fyzická síť nevidí. Mezi nejvyužívanější protokoly pro tuto komunikaci patří NVGRE a VXLAN jejichž popis je podrobněji rozebrán v kapitole 2.9.2. [8] [12]

### 2.9.2 Zapouzdření síťové komunikace

V případě řešení sítě přes virtualizaci dochází k zapouzdřování komunikačních dat v jiných formátech než je běžné mimo cloudová prostředí. Tato zapouzdření jsou spojena s overlay protokoly pro tunely jak bylo řečeno v 2.9.1. Momentálně nejvyužívanější jsou protokoly VxLAN a NVGRE.

- VxLAN

  VxLAN - Virtual Exstensible LAN je virtualizační síťová technologie vyvinuta společnostmi CISCO a VMware zaměřující se na zlepšení škálovatelnosti cloudových prostředí s využitím existující technologie VLAN. VxLan využívá obdobné zapouzdření jako VLAN pro zapouzdření Ethernet framů založených na MAC s využitím UDP packetů 3. vrstvy, čímž poskytuje VM abstrakci 2. vrstvy. Obrázek 2.5 zobrazuje VxLAN zapouzdření.

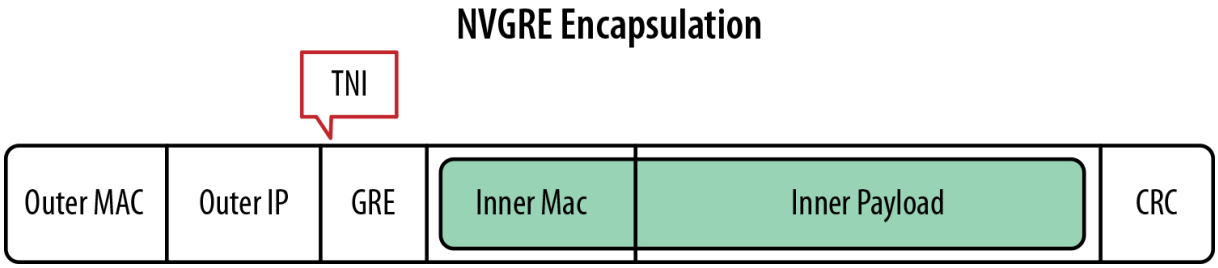

**Obrázek 2.5:** VxLAN zapouzdření, převzato z: [1]

  VxLAN také řeší problém s nedostatečným počtem poskytovaných vlan. Původní 802.1Q VLAN disponuje místem pro identifikátor o délce 12 bitů, což poskytuje okolo 4000 záznamů. Technologie VxLAN disponuje identifikačním prostorem o délce 24 bitů, čímž poskytuje okolo 16 milionů záznamů. VxLAN odpojuje VM z jejich fyzické sítě a povoluje komunikaci s ostatními použitím transparentní overlay sítě, která je umístěna nad fyzickou sítí. Další výhodou VxLAN je kompletní zbavení VM vědomostí o fyzické síti. Jelikož vidí pouze virtuální 2. vrstvu sítě poskytuje VxLAN i snadnou migraci VM. [1]

- NVGRE

  NVGRE - Netwrok Virtualization Generic Routing Encapsulation protokol je síťová virtualizační technologie vytvořená s důrazem na řešení problémů se škálovatelností spojených prostředím velkých datových center zmíněných dříve u VxLAN. Podobně jako VxLAN využívá zapouzdření informací 2. vrstvy do packetů vrstvy 3. NVGRE protokol využívá GRE tunelů pro přenos framů 2. vrstvy přes síť 3. vrstvy. NVGRE umožňuje komunikaci mezi dvěma i více sítěmi 3. vrstvy a poskytuje VM zdání, že sdílí společný prostor 2. vrstvy. Obrázek 2.6 zobrazuje zapouzdření používané protokolem NVGRE. Pro jednoznačnou identifikaci je využíváno pole Tenant Network Identifier(TNI) o délce 24 bitů, které je přidané k Ethernetovému framu 2. vrstvy. TNI poskytuje shodně s VXLAN přes 16 milionů záznamů. TNI je dále využíváno jako identifikátor jednotlivých GRE tunelů

využívaných ke komunikaci. [1]




**Obrázek 2.6:** NVGRE zapouzdření, převzato z: [1]

### 2.9.3 OpenStack

OpenStack je open-source platforma pro cloudové prostředí disponující komponentami pro řešení širokého spektra cloudových služeb. Komponentami jsou Nova, Swift, Cinder, Storage, Neutron, Horizon, Keystone, Glance, Ceilometer, Heat, Trove, Itonic, Zaqar a Sahara, kde každá komponenta má své specifické určení. Ve spojitosti s touto prací stojí za zmínku především komponenty Nova a Neutron. [9]

Komponenta Nova slouží pro cloud computing a jedná se o jednu z hlavních částí z IaaS systému. Jako taková byla navržena k poskytnutí virtuálních výpočetních stanic spolupracujících pod hypervisor technologii jako je KVM a VMware. Jednotlivé stanice mohou využívat obrazy systémových disků nahrané v modulu Glance.

Hypervisor je technologie starající se o běh a vytváření virtuálních strojů (VM). Stroj provozující hypervisor je nazýván jako Host machine(Hostící stroj) zatímco VM jako Guest machine(Hostovaný stroj). KVM je hypervisor, který je oproti VMware ESXi open-source. Je určen pro linuxový kernel, ze kterého vytváří hypervisor. Naproti tomu VMware ESXI je komerční hypervisor s vlastním kernelem uvnitř pracujícím přímo na fyzickém stroji. Mezi další funkce modulu Nova patří oddělení svých vnitřních částí zvaných Tenant neboli Project, což jsou oblasti pro jednotlivé zákazníky. [9] [10]

Komponenta Neutron (původně známá jako Quantum) je určena pro síťové služby. Poskytuje moduly pro různé síťové funkce napříč druhé až sedmé vrstvy ISO/OSI modelu jako jsou VLAN, DHCP, NAT, DNS. Neutron vytváří jednotlivé virtuální sítě, které se na fyzické části mapují do jednotlivých VLAN. [9] [11] [12]

## 2.10 Otázky a možné problémy SDN

Stejně jako každá technologie, ani SDN není bez negativ a jelikož je stále v rané fázi své existence, zůstávají některé otázky pro řešení dané situace nezodpovězené.

- centralizace (logická - úplná)

  Jak již bylo řečeno, úplná centralizace zjednodušuje správu na úkor odolnosti a výkonnosti, kterou nabízí logická centralizace potýkající se naopak s nutností řešit komunikaci mezi jednotlivými doménami. [1]

- počet SDN controllerů

  Velmi důležitá otázka s neexistující jednoznačnou odpovědí, což je pro SDN nevýhodou, zvláště pro klíčovou roli SDN controlleru. Vhodný počet controllerů je individuální pro každou síť. Počet síťových prvků, zvolený konvergenční přístup, požadovaná rychlost a redundance. Všechny tyto faktory ovlivňují potřebný počet SDN controllerů v síti. Je také známo, že vytížení SDN controlleru je proměnlivé, a v určitých situacích může dosahovat téměř nulového využití. Vynaložené náklady za velký počet SDN controlerů jsou poté zbytečné. Zde se jako možné řešení ukazuje virtualizovaný SDN controller, jehož výpočetní výkon, případně počet, by byl závislý na vytížení momentálně dostupných SDN controllerů v daný časový okamžik. [4]

- nový failure model(model selhání)

  S nasazením SDN přichází, stejně jako s každou novou technologií, nový failure model. U současných technologií je takový model již dobře známý a disponuje zpracovanými postupy pro řešení vzniklých selhání. Takovéto postupy pro SDN zatím neexistují, což může vést k velkým problémům spojeným s řešením daných selhání. Tento faktor může odradit společnosti od nasazení SDN, jelikož riskují situaci, ve které se jejich IT oddělení nedokáže vypořádat se vzniklým selháním v únosném čase. Zároveň je však nutno říci, že s rozvojem SDN budou rozvíjeny také postupy pro řešení failure modelu. [6]

- velikost tabulek FIB

  Switche mají pro ukládání FIB vyhrazenou paměť jejíž velikost samozřejmě není neomezená a je tedy na místě zvážit situaci, kdy switch nebude mít dostatečnou

paměťovou kapacitu. V případě plně proaktivního přístupu by tak mohlo dojít k nekonečné konvergenci sítě. Řešením tohoto problému bude správný přístup konvergence na pomezí reaktivního a proaktivního přístupu, jež byl zmíněn dříve. [6] [1]

- nutná aplikace záznamů na jednotlivé switche

  Switche jsou zbaveny jakékoli vyšší funkcionality, což je výhodou přinášející zrychlení, nicméně tyto záznamy je nutné nahrát do FIB. Nahrávání je proces zabírající určitý čas, který dříve switche nijak neovlivňoval. K tomuto účelu bylo nutné navrhnout a vytvořit příslušné protokoly jako například OpenFlow. [6]

- switche nejsou navrženy pro SDN architekturu

  I přes fakt, že byly vyvinuty potřebné technologie a SDN již funguje v reálných prostředí, je stále nutné mít na paměti původní účel switchů. Ten totiž není totožný s účelem v SDN. Může nastat situace, kdy bude objeven nevhodný návrh switche pro určitou funkcionalitu v SDN. [6]

V dalších kapitole této bakalářské práce jsou popsány dvě konkrétní řešení pro SDN v platformě OpenStack. K výběru dvou řešení vedla skutečnost, že trh SDN není momentálně příliš bohatý a zvolená řešení patří mezi aktuálně nejvýznamnější.

# 3 SDN řešení

V této kapitole jsou popsána zvolená SDN řešení z hlediska struktury i nabízených funkcí společně s důvody, které vedly k jejich výběru.

## 3.1 PLUMgrid ONS 2.0

Jedná se o SDN technologii postavenou na klíčových principech jako jsou programovatelný data plane, škálovatelný control plane, layer-less networking (bezvrstvá síť), důkladné zabezpečení, otevřená platforma a SDK(Software Development Kit).[14] [13]

PLUMgrid ONS (Open Networking Suite) je založen na API a jedná se o řešení virtuální síťové infrastruktury. Uživatelům umožňuje vytvoření a správu víceuživatelských virtuálních sítí v prostředí OpenStack, do nějž je PLUMgrid implementován v podobě pluginu.

Vlastností tohoto řešení je především možnost plné virtualizace, což má značně zjednodušit nasazení. PLUMgrid byl pro zkoumání vybrán z následujících důvodů:

- plugin do OpenStack
- otevřená platforma
- grid jako výpočetní jednotka

  jedinečný přístup mezi všemi SDN řešeními
- zapouzdření VXLAN nebo NVGRE

  VXLAN poskytuje nezávislost na 2. a 3. síťové vrstvě
- poskytuje end-to-end šifrování v rámci Virtual Domain
- udávaný výkon až 40Gbps na jeden server

- možnost výkonové agregace až na úroveň Tb.
- možnost importu produktů třetích stran pro služby 4.-7. síťové vrstvy

[14]

### 3.1.1 PLUMgrid koncept

PLUMgrid ONS 2.0 je založeno na několika klíčových komponentách, které vzájemnou spoluprací poskytují funkční celek. Obrázek 3.1 zobrazuje celkový koncept. Jednotlivé komponenty jsou následující:

- Zone (Zóna)

  Reprezentuje fyzické rozložení PLUMgrid řešení. Zone je hlavní kolekcí Edge a Gateway prvků, které jsou závislé na stejném úseku Director prvku v jedné fyzické lokaci. Datová centra mohou být spojením z více zone. [14]

- PLUMgrid Director

  Zastává roli SDN controllleru, což z něj činí mozek celé PLUMgrid platformy. Prostřednictvím Director prvku je pro zákazníky realizována virtuální síťová infrastruktura. Konfigurace může být provedena přes OpenStack (prostřednictvím PLUMgrid Neutron pluginu), PLUMgrid API nebo konzoli pro správu(GUI). Director je zodpovědný za koordinaci a správu všech ostatních platforem. [14]

- PLUMgrid Edge

  Jedná se o prvek nazývaný v OpenStack prostředí jako Compute node (výpočetní uzel), zpracovávající PLUMgrid software jako kernel modul. Poskytuje síťovou konektivitu pro zákaznické VM (VirtualMachines - virtualizované stanice). Komunikace mezi jednotlivými VM je umožněna za využití overlay tunelů používajících zapouzdření VXLAN. [14]

- PLUMgrid Gateway

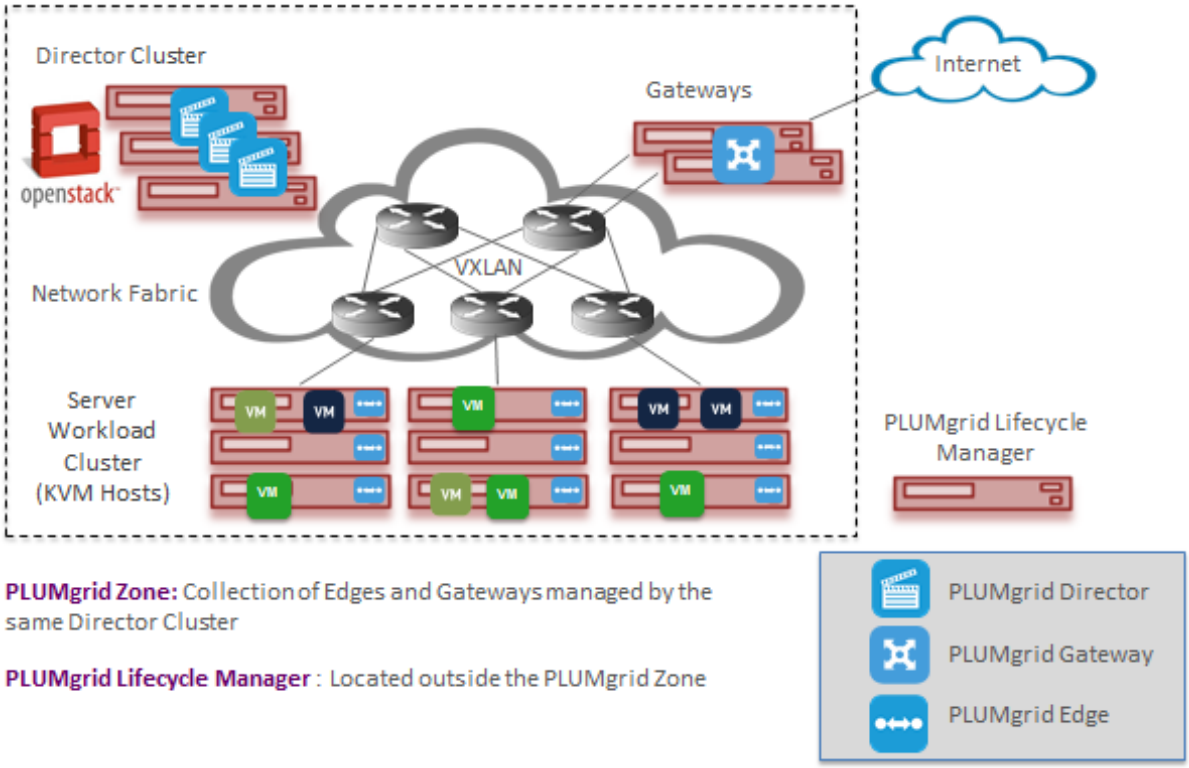
  Poskytuje podobnou funkcionalitu jako PLUMgridEdge, přičemž disponuje navíc rozhraním nebo rozhraními pro připojení k externí síti, čímž poskytne konektivitu z vnitřních prvků(založených na VXLAN) do IP sítě.

- Virtual Domains

  Virtual domain je označení pro logické datové centrum, které může být vytvořeno na požadavek s poskytnutím všech síťových služeb (routery, switche, DHCP, NAT, atd.) potřebných k vystavění virtuální sítě pro zákazníka v OpenStack. [14]

- LCM

  LCM je VM využívaná během instalačního procesu PLUMgrid ONS 2.0. Tato VM poskytuje všechny potřebné PLUMgrid balíčky pro instalaci nebo update PLUMgrid zone. LCM VM je dostupné mimo OpenStack cluster. [14]




**Obrázek 3.1:** Koncept PLUMgrid ONS 2.0, převzato z: [15]

### 3.1.2 Podporované funkce

PLUMgrid ONS 2.0 disponuje funkcemi pro vybudování síťové infrastruktury ve virtualizovaném prostředí. Mezi funkce dostupné pro budování samotné síťové infrastruktury patří:

- bridge

- router

  Umožňuje routing na základě statických pravidel.

- DHCP
- DNS
- NAT

Jak je z výčtu funkcí patrné, není nabízena žádná možnost řešení firewall. PLUMgrid ONS 2.0 uvádí řešení zabezpečení pomocí oddělení jednotlivých zákazníků do samostatných Virtual Domains poskytující soustavu bezpečnostních pravidel a politik pro vlastní cloudové prostředí. Další technikou pro zabezpečení je end-to-end šifrovaná komunikace v rámci Virtual Domain.[14]

Chybějící funkce je možné implementovat díky open SDK (Software Develpment Kit), které umožňuje jak samotnou implementaci prostřednictvím specifického doménového jazyka, knihoven a objektových modelů, tak i v podobě programů třetích stran.[14]

### 3.1.3 Podporované hypervisory

PLUMgrid ONS 2.0 musí jakožto virtualizační technologie pracovat společně s hypervisorem, který interpretuje fyzickou infrastrukturu v abstrahované podobě. Umožňuje tedy její logické členění na jednotlivé části. Podporované hypervisory jsou KVM a VMware ESXi, přičemž poskytuje i funkci multi-hypervisor, která umožňuje využít kombinaci obou hypervisorů. [14]

## 3.2 HP SDN

Řešení HP SDN je založeno na spolupráci více komponent. Hlavní částí je HP Helion OpenStack a implementace síťového prostředí HP Virtual Cloud Networking. Výstupy těchto součástí sbírá HP VAN SDN controller, který tak může ovlivňovat fyzickou i virtuální síťovou infrastrukturu.

### 3.2.1 Co je HP Helion OpenStack

HP Helion OpenStack je produkt vystavěný na základě standardní OpenStack platformy, který byl obohacen o některé funkce a připraven pro snadnější implementaci SDN produktů od HP. Klíčovou vlastností je otevřenost celého řešení, jehož výhody jsou shrnuty v následujících bodech:

- dobře zdokumentované otevřené programovací rozhraní (API)
- kód řešení je otevřený, je možné ho volně stáhnout a pokračovat ve vlastním vývoji
- jednotlivé implementované moduly je možné nahradit

HP Helion OpenStack byl pro zkoumání vybrán z následujících důvodů:

- jedná se o nadstavbu rozšířené platformy OpenStack
- otevřený přístup HP při vývoji
- nové funkce (distribuovaný router)
- množství dalších produktů HP možných využít společně s HP Helion OpenStack

[19][20]

### 3.2.2 HP Helion OpenStack struktura

Struktura HP Helion OpenStack se stejně jako samotný OpenStack skládá z několika modulů, kde každý má svou klíčovou funkci pro nabídnutí účinného nástroje pro virtualizaci.Celková struktura je znázorněna obrázkem 3.2. Jednotlivé moduly jsou:

- OpenStack core

  Využívá jádro OpenStack, momentálně ve verzi Juno. Pro správu VM je tu modul Nova shodně s OpenStack.

- storage s HP StoreVirtual VSA, 3PAR a HP Sirius

  V licenci pro HP Helion OpenStack je zahrnuta i licence na HP StoreVirtual VSA, což je pokročilý nástroj pro softwarově definované úložiště mimo jiné s funkcí RAID pro vysokou dostupnost a výkon. HP Helion OpenStack také v rámci otevřenosti řešení umožňuje použít ovladače pro úložiště od třetích stran. Součástí je i HP Sirius, což je komponenta zjednodušující převzetí úložiště pod IaaS.

- instalátor cloudu postavený na TripleO, Ironic a HP Sherpa

  Ironic je program umožňující ovládat z OpenStack nejen VM, ale také fyzické stroje. TripleO někdy též nazývané jako OpenStack on OpenStack je nástroj pro instalaci OpenStacku OpenStackem. Sherpa je modul umožňující stahování aktualizací cloudu.

- správa a monitoring - Ceilometer, Icinga, Kibana

  Pro shromažďování informací o infrastruktuře je využíván modul Ceilometer. O sběr logů a se stará implementace Lohstash a ElasticSearch , což jsou open-source projekty HP. Monitoring dostupnosti komponent infrastruktury je poskytovaný prostřednictvím projektu Icinga.

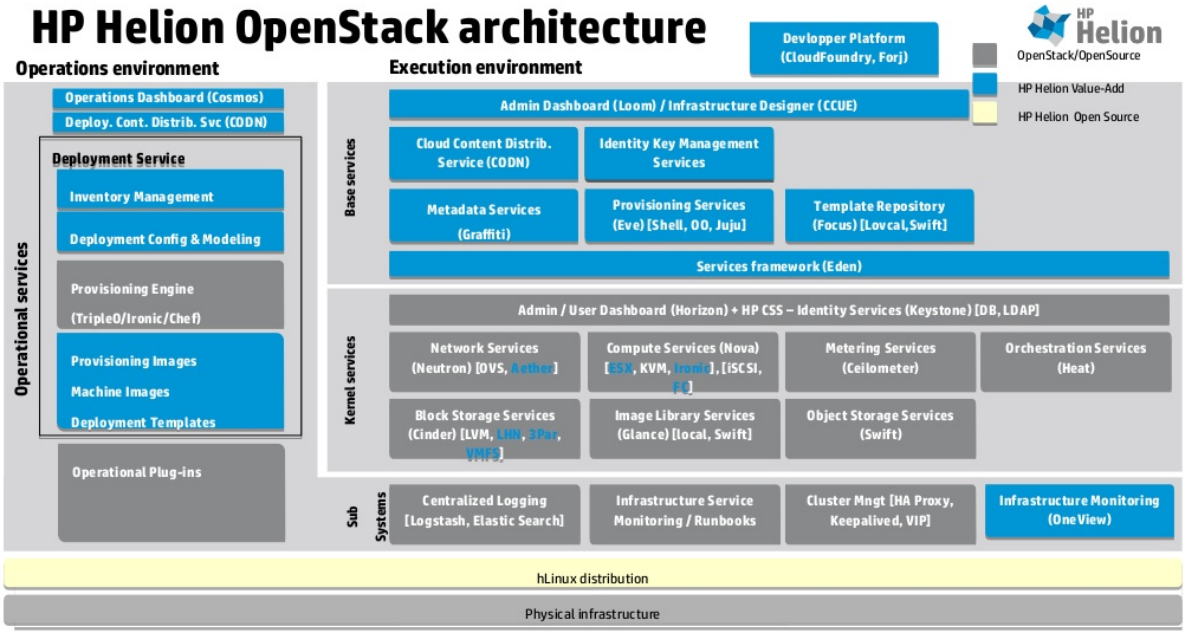
- HP Virtual Cloud Networking (Neutron)

  Na standardní komponentě Neutron má HP Helion OpenStack vystavěnou implementaci pro virtualizaci a automatizaci sítě zvanou HP Virtual Cloud Networking (HP VCN). Dále bude HP VCN věnována kapitola 3.2.5.

- podpora VMware ESX

  HP Helion OpenStack obsahuje jako svou přímou součást open-source hypervisor KVM, čímž dotváří kompletní balíček hypervisoru, virtualizace sítě i cloudu. Kromě hypervisoru KVM je také podporován VMware ESXi.

[19][20]




**Obrázek 3.2:** Architektura HP Helion OpenStack, převzato z: [19]

### 3.2.3 Podporované funkce

Kromě standardních funkcí nabízených v OpenStack přináší Helion několik inovací. Nejvýznamnější je podpora virtuálního distribuovaného routeru (DVR) místo standardního switche. Tyto routery přináší podporu firewall (FWaaS), ale také VPN (VPNaaS) a rozložení zátěže (LBaaS). Možné je implementovat aplikace HP nebo třetích stran. [19][20]

### 3.2.4 Podporované hypervisory

Podpora hypervisorů je u HP Helion OpenStack připravena pro nasazení KVM i ESXi. Není však nutné využívat pouze jednu ze zmíněných variant. HP Helion OpenStack je totiž schopný využívat funkci multi-hypervisor díky níž, je možné aby oba dva hypervisory spolupracovali najednou.

### 3.2.5 HP Virtual Cloud Networking (HP VCN)

Jedná se o implementaci pro virtualizaci a automatizaci sítí v HP Helion OpenStack, kde jsou oproti standardnímu OpenStack přidané funkce od HP. Nové funkce

nezůstanou proprietární, ale v rámci otevřenosti HP Helion OpenStack budou zakomponovány do dalších verzí samotného OpenStack. S výstupy HP VCN pracuje controller HP VAN SDN, který slouží pro virtuální sítě, ale díky své podpoře protokolu OpenFow také pro ty fyzické. HP VCN podporuje funkci multi-hypervisor díky níž lze využívat najednou hypervysoru KVM i VMware ESXi.[21]

HP VCN je momentálně součástí HP Helion OpenStack kam přináší přidané funkce (VxLAN, distribuovaný routing, FWaaS, DNSaaS, ...).

Využití HP VCN:

- automatizace datového centra s klasickými aplikacemi
- správa veřejného cloudu s per-tenant přístupem pro síťové funkce
- řešení nad otevřeným systémem (HP Helion OpenStack)

[22]

# 4 Srovnání SDN řešení

V následující kapitole této bakalářské práce je provedeno funkční porovnání obou dříve zmíněných řešení.

## 4.1 Funkční provnání

Z pohledu funkcí nabízených v jednotlivých řešení nabízejí HP i PLUMgrid v mnohých směrech shodné možnosti. Z těch rozdílných stojí za zmínění především přidaná funkcionalita u HP Helion OpenStack. Oproti řešení PLUMgrid, nabízí HP možnosti firewall, VPN a rozložení zátěže. Přidnou funkcionalitu má HP díky implementaci distribuovaného routeru. Tato skutečnost značně přispívá ve prospěch řešení HP. PLUMgrid řešení naopak oproti HP poskytuje propracovanější model zabezpečení komunikace a to i v rámci samotného řešení, což je funkce, kterou HP nijak nezmiňuje.

Obě řešení disponují možností implementovat vlastní funkce prostřednictvím otevřeného API, nebo instalací od třetích stran. Tato vlastnost je velice důležitá především z důvodu zachování otevřenosti řešení, což je ostatně jedním z cílů samotné SDN technologie.

Shodné jsou obě řešení i v podpoře hypervisorů, kde je podporovaný KVM i VMware ESXi a to včetně možnosti kooperace obou hypervisorů při tzv. multi-hypervisor. Nicméně stojí za zmínku, že PLUMgrid ONS 2.0 je orientován spíše pro virtuální prostředí, kde umožňuje virtuální zastoupení všech svých prvků. V tomto ohledu HP nabízí integraci s HP VAN SDN pro správu i fyzických strojů.

Ohledně datové komunikace nabízí řešení PLUMgrid výběr zapouzdření mezi VxLAN a NVGRE. HP Helion OpenStack nabízí pouze VxLAN.

Řešení HP Helion OpenStack je velice silně provázáno s mnoha dalšími produkty HP nabízenými v oblasti SDN. Tento fakt činí z HP možného dodavatele SDN řešení pro

velice specifické požadavky, zatímco PLUMgrid sází spíše na univerzálnost jediného řešení. HP má také již v současné době v provozu portál pro nakupování jednotlivých aplikací na doplnění požadované funkcionality.

Poněkud rozdílně je pojetí architektury obou řešení. Zatímco PLUMgrid ONS 2.0 je vystavěn jakožto rozšiřující plugin do OpenStack, kde je jeho úkolem v podstatě nahradit síťový modul Neutron, řešení HP Helion OpenStack je nahrazení standardního OpenStack vlastní implementací disponující moduly standardního OpenStack rozšířené o přidané funkce.

Maximální hodnoty virtualizovaných prvků jsou obsaženy v tabulce 4.1 pro PLUMgrid ONS 2.0. HP Helion OpenStack poskytuje pouze informaci o maximálním počtu 40 VM na prvek při maximálním počtu 100 prvků. V tomto ohledu tedy PLUMgrid jasně předčí možnosti řešení HP Helion OpenStack.

**Tabulka 4.1:** Limity PLUMgrid ONS 2.0, převzato a upraveno dle: [15]

| **Komponenta** | **Maximální počet prvků** |
|---|---|
| Edge | 150 |
| Virtuální doména | 1000 |
| VM na jeden Edge | 50 |
| VM na jednu Virtuální doménu | 500 |
| virtualizované síťové funkce v jedné doméně | 7 |
| Portů na jednom Bridge prvku ve Virtuální doméně | 256 |
| Maximální tok na NAT, Edge | 500 |

Celkové porovnání podle jednotlivých parametrů porovnání společně s jejich hodnotami jsou uvedeny v tabulce 4.2.

**Tabulka 4.2:** Shrnutí porovnání SDN řešení

| | **HP Helion OpenStack** | **PLUMgrid ONS 2.0** |
|---|---|---|
| **Firewall** | Ano | Ne |
| **VPN** | Ano | Ne |
| **Rozložení zátěže** | Ano | Ne |
| **Otevřené API** | Ano | Ano |
| **Podpora hypervisoru** | KVM, VMware ESXi | KVM, VMware ESXi |
| **Zapoudření** | VxLAN | VxLAN, NVGRE |
| **Max. VM na prvek** | 40 | 100 |

Na základě provedeného prozkoumání a funkčního porovnání bylo po konzultaci se zástupci firmy tcp cloud a.s. rozhodnuto o praktické implementaci řešení PLUMgrid ONS 2.0. Klíčové byly při rozhodování skutečnosti ohledně podpory dvou různých zapouzdření (VxLAN a NVGRE), propracovanějšího modulu zabezpečení komunikace, plné podpoře virtualizace a možnost nasadit vysoké množství prvků v rámci řešení.

# 5 Implementace SDN řešení

V této kapitole bakalářské práce je popsán experiment, který byl zaměřen na implementaci SDN řešení PLUMgrid ONS 2.0. Součástí je také postup instalace a dosažené výsledky.

## 5.1 Parametry experimentu

Experiment byl navržen s ohledem na reálnost samotné implementace a zaměřením především na zjištění výkonnostní stránku řešení PLUMgrid ONS 2.0. Z těchto důvodů byl kladen důraz na využití co největšího podílu fyzických prvků v testovacím prostředí.

### 5.1.1 Laboratorní prostředí

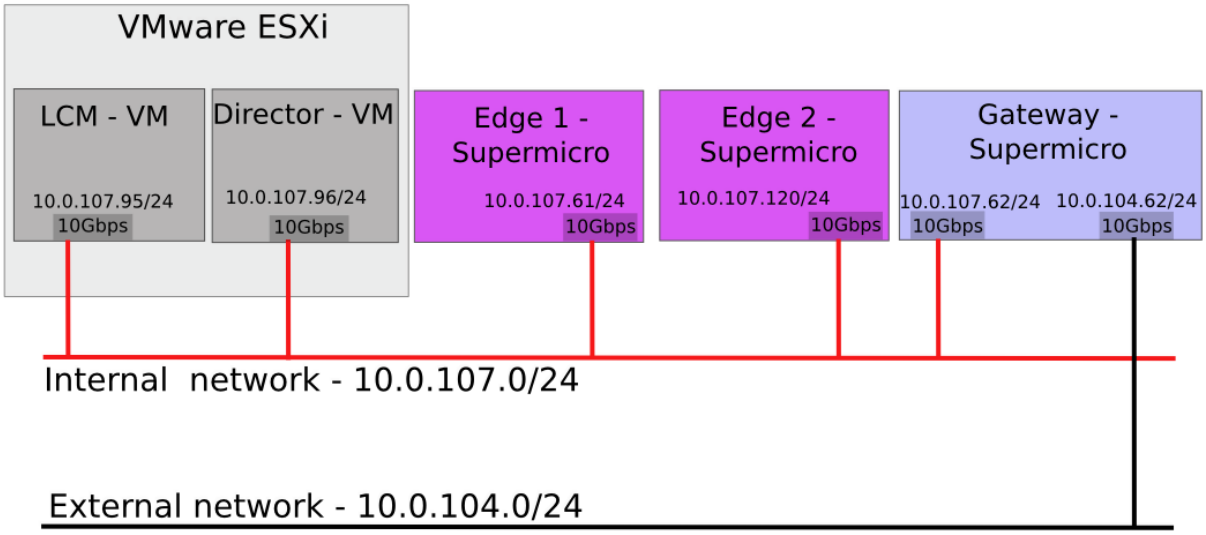
Prostředí pro implementaci bylo ve spolupráci s firmou tcp cloud a.s. prováděno na infrastruktuře datového centra v Písku. Prostředí datového centra v Písku nabízí jedno z nejmodernějších vybavení v České republice a umožnilo provést aplikaci testů v produkčním prostředí.

### 5.1.2 Výkonnostní metriky

V experimentu byla zkoumána především výkonová stránka zvolených řešení v rámci lokálních stanic i s externím serverem. Pro zkoumání byly využity nástroje iperf a iperf3, kterými byla zjištěna přenosová rychlost, jakých je dané řešení schopno dosáhnout. Měření bylo prováděno se zapouzdřením VxLAN i NVGRE ve variantách east/west, north/south v režimech half-duplex i full-duplex

### 5.1.3 PLUMgrid topologie

Pro nasazení PLUMgrid ONS 2.0 byla navržena topologie zobrazená na obrázku 3.2. V topologii jsou zastoupeny všechny prvky nutné pro správnou funkci. Pro možnost testovat reálné přenosové rychlosti v rámci PLUMgrid domény(tzv. East/West komunikace) byla topologie vybavena dvěma Edge prvky zprostředkované servery Supermicro, který byl použit rovněž pro Gateway o konfiguraci uvedené v tabulce 5.1. Zbylé prvky, tedy LCM a Director, byly realizovány prostřednictvím virtuálních strojů v prostředí VMware ESXi s konfigurací odpovídající hodnotám uvedeným v tabulce 5.2.




**Obrázek 5.1:** Testovací topologie pro PLUMgrid ONS 2.0

**Tabulka 5.1:** HW konfigurace Supermicro serverů

| | |
|---|---|
| **Základní deska** | X9DR7-LN4F: 6x SATA, 8x SAS2, 4X LAN |
| **CPU** | 2x Intel Xeon 6-Core E5-2630v2 2.6GHz 15MB 7.20GT/s |
| **RAM** | 128GB(8x 16GB) ECC Reg DDR3 1600MHz |
| **HDD** | 2600GV SAS II HGST 2.5" 10K |
| **Síťová karta** | Intel 10 Gigabit X520-DA2 SFP+ Dual Port |
| **OS** | CentOS 7.0 Minimal |

V následující tabulce jsou obsaženy hodnoty požadované pro jednotlivé prvky účinkující v PLUMgrid ONS 2.0 topologii. Tyto hodnoty mají roli minimálních poža-

davků a v experimentu byly použity pro prvky reprezentované virtuálními stroji. Prvky fyzické splňovaly minimální uvedené požadavky, avšak jejich konkrétní konfigurace je uvedena v tabulce 5.2.

**Tabulka 5.2:** HW požadavky pro komponenty PLUMgrid ONS 2.0, převzato a upraveno dle: [15]

| | **LCM** | **Director** | **Edge** | **Gateway** |
|---|---|---|---|---|
| **Procesor** | 4 x CPU | 4 x CPU | 4 x CPU | 4 x CPU |
| **RAM** | 8 GB | 16GB | 8GB | 8GB |
| **HDD** | 20 GB | 40 GB | 40 GB | 40 GB |
| **Síťová karta** | 1 x 10Gb/s | 1 x 10Gb/s | 1 x 10Gb/s | 2 x 10Gb/s |
| **Operační systém** | CentOS 6.5 Minimal | CentOS 7.0 Minimal | CentOS 7.0 Minimal | CentOS 7.0 Minimal |

## 5.2 Instalace PLUMgrid

Při instalaci PLUMgrid ONS 2.0 byl následován postup popsaný v uživatelské příručce poskytnutý dodavatelem řešení PLUMgrid.

### 5.2.1 Instalace LCM

Prvním krokem je instalace LCM VM. V našem případě se jednalo o VM ve virtualizačním prostředí VMware ESXi. Konfigurace VM LCM byla zvolena shodně s hodnotami v tabulce 5.2. Jak již bylo řečeno, hlavní rolí LCM VM je koordinace instalace celého PLUMgrid ONS 2.0. Postup instalace a konfigurace LCM VM prvku:

1. instalace operačního systému CentOS 6.6 minimal
2. nastavení síťového rozhraní v příslušném konfiguračním souboru, u CentOS `/etc/sysconfig/netwrok-scripts/jmeno_rozhrani`
   Klíčové je nastavení statické IP adresy a správné nastavení výchozí brány společně s DNS serverem pro zajištění fungujícího připojení do internetu pro následné stahování instalačních balíčků.

3. nastavení hostname a FQDN (Full Qualified Domain Name - plně specifikované doménové jméno)

   Důležité pro správně fungující komunikaci mezi jednotlivými komponenty PLUMgrid ONS 2.0.

4. spuštění instalačního scriptu z veřejného repositáře PLUMgrid.

   ```
   bash <(curl -s https://plumgrid:gridplum@pubrepo1.
    plumgrid.com/files/lvm-installer.sh)
   ```

   Během provádění tohoto scriptu je nutno vyplnit několik údajů, ze kterých je vhodné zmínit především Zone name specifikující jméno a dále nastavení podpory pro OpenStack, který bude zprostředkovávat orchestraci.

   Po úspěšném dokončení tohoto scriptu je instalace LCM VM hotova a je možno pokračovat instalací Director prvku.

### 5.2.2 Instalace Director

Dalším krokem je instalace Director, který byl rovněž realizován VM v prostředí VMware ESXi s konfigurací odpovídající hodnotám v tabulce 5.2. Postup instalace a konfigurace Director prvku:

1. instalace operačního systému CentOS 7.0 minimal

2. nastavení síťového rozhraní v příslušném konfiguračním souboru, u CentOS `/etc/sysconfig/netwrok-scripts/jmeno_rozhrani`

   Opět je nutné nakonfigurovat statikou IP adresu společně s nastavením výchozí brány a DNS serveru pro zajištění konektivity do internetu.

3. nastavení hostname a FQDN

4. přidání záznamu do souboru `/etc/hosts` ve tvaru

   ```
   <Director IP> <Director VM FQDN> <Director hostname>
   \<LCM IP> <LCM FQDN> <LCM Hostname>
   ```

   Tento záznam lokálně přiřadí IP adresu k danému doménovému jménu LCM VM.

5. instalace balíčků za pomoci příkazů

```
yum install -y http://repos.fedorapeople.org/repos/
 openstack/openstack-icehouse/rdo-release-icehouse-
 4.noarch.rpmn
yum install -y http://dl.fedoraproject.org/pub/epel/
  7/x86_64/e/epel-release-7-5.noarch.rpm
```

6. do souboru `/etc/yum.repos.d/epel.repo` přidat parametr `exclude=python-sqlalchemy,python-migrate`

7. vytvoření souboru `/etc/yum.repos.d/plumgrid.repo` v roli repositáře odkazující na LCM VM pod adresou `http://<LCM IP>:81/yum/everest/el7/x86_64/`

8. import GPG key pro LCM VM

9. v souboru `/etc/selinux/config` nastavení `SELINUX = permissive`

   SELinux neboli NSA Security-Enhanced Linux je nadstavbová implementace přístupových práv. Využívá se hlavně z důvodů, že klasická UNIX omezení umožňují rozlišovat pouze vlastníka, skupinu a ostatní, což je pro některé případy příliš obecné. Editací tohoto souboru dojde k udělení bezpečnostní výjimky.

10. deaktivace firewall

11. reboot systému

12. instalace puppet augeas

    Jedná se o nástroj typu klient/server pro centralizovanou údržbu operačních systémů. Princip funkce je založen na vygenerování manifestu na stroji vystupujícím v roli server, který popisuje kompletní specifikace systému pro klientskou stanici. Podle tohoto manifestu se následně stanice v roli klienta za pomocí puppet agent uvede do požadovaného stavu, což může zahrnovat jak instalaci balíčků, tak konfiguraci systémových souborů. [18] V prostředí PLUMgrid ONS 2.0 spolupracuje s aplikací Foreman.

13. nastavení puppet augeas

```
augtool -s <<EOA
set /files/etc/puppet/puppet.conf/main/server <LCM-FQDN>
set /files/etc/puppet/puppet.conf/main/pluginsync true
EOA
```

14. spuštění puppet agenta příkazem `puppet agent -t -w 30`
    Tento příkaz spustí registraci Director prvku do dané zone, na kterou je nutné reagovat zadáním příkazu pro autorizaci certifikátu
    `pupet cert sign <director-FQDN>`

15. připojení do Foreman

    Foreman je open-source aplikace sloužící k automatizované správě infrastruktury. Nabízí instalaci operačních systémů a jejich následnou konfiguraci podle nastavených parametrů za pomoci služby Puppet, dále také nabízí monitoring konfigurace. [16] [17]

    Tento nástroj je využíván pro konfiguraci jednotlivých komponent PLUMgrid ONS 2.0 Pro přihlášení je používaná IP adresa LCM VM vložena do internetového prohlížeče. Pomocí nástroje Foreman lze ověřit, zda došlo k registraci Director prvku a zároveň nastavit několik parametrů nutných ke správné funkci.

    Director je nutné přidělit do Enviroment shodného s názvem zone a do Host Group na hodnotu Zone123-controller, kde část Zone123 je zvoleným zone jménem.

    Následující tabulky shrnují konfigurační parametry v jednotlivé oblasti.

**Tabulka 5.3:** konfigurační parametry PLUMgrid Director prvku

| **Parametr** | **Význam** | **Hodnota** |
|---|---|---|
| plumgrid_ip | IP všech Director | 10.0.107.96 |

**Tabulka 5.4:** Konfigurační parametry sal Director prvku

| Parametr | Význam | Hodnota |
|---|---|---|
| plumgrid_ip | IP všech Director | 10.0.107.96 |
| virtual_ip | virtuální IP Director | 10.0.107.97 |

**Tabulka 5.5:** Konfigurační parametry OpenStack Director prvku

| Parametr | Význam | Hodnota |
|---|---|---|
| controller_address_api | IP všech Director | 10.0.107.96 |
| controller_address_management | IP všech Director | 10.0.107.96 |
| mysql_allowed_hosts | povolení mysql hosti | "10.0.107.%" |
| network_api | IP uživatelské API sítě | 10.0.107.0/24 |
| network_data | IP sítě pro data | 10.0.107.0/24 |
| network_management | IP sítě pro management | 10.0.107.0/24 |
| pg_director_server | virtuální IP Director | 10.0.107.97 |
| pg_password | heslo pro PLUMgrid | — |
| pg_username | login pro PLUMgrid | — |
| storage_address_api | IP všech Director | 10.0.107.96 |
| storage_address_management | IP všech Director | 10.0.107.96 |

16. spuštění `puppet agent -vt` způsobí konfiguraci podle parametrů nastavených v předchozím kroku.

### 5.2.3 Instalace Edge

Následuje instalace prvku v roli Edge jehož úkolem je lokální komunikace. Tento prvek byl reprezentován prostřednictvím fyzického serveru Supermicro s konfigurací uvedenou v tabulce 5.1. Postup instalace a konfigurace Edge prvku:

1. instalace operačního systému CentOS 7.0 minimal item nastavení síťového rozhraní v příslušném konfiguračním souboru, u CentOS

`/etc/sysconfig/netwrok-scripts/jmeno_rozhrani`

Opět je nutné nakonfigurovat statikou IP adresu společně s nastavením výchozí brány a DNS serveru pro zajištění konektivity do internetu.

2. nastavení hostname a FQDN
3. přidání záznamu do souboru `/etc/hosts` ve tvaru
   `<Edge IP> <Edge FQDN> <Director hostname>\<LCM IP>`
   `<LCM FQDN> <LCM>Hostname>`
   Tento záznam lokálně přiřadí IP adresu k danému doménovému jménu LCM VM.
4. instalace balíčků za pomoci příkazů

```
yum install -y http://repos.fedorapeople.org/repos/
 openstack/openstack-icehouse/rdo-release-icehouse-
 4.noarch.rpmn
yum install -y http://dl.fedoraproject.org/pub/epel/
  7/x86_64/e/epel-release-7-5.noarch.rpm
```

5. do souboru `/etc/yum.repos.d/epel.repo` přidat parametr
   `exclude=python-sqlalchemy,python-migrate`
6. vytvoření souboru `/etc/yum.repos.d/plumgrid.repo` v roli repositáře odkazující na LCM VM pod adresou

   `http://<LCM IP>:81/yum/everest/el7/x86_64/`

7. import GPG key pro LCM VM
8. v souboru `/etc/selinux/config` nastavení `SELINUX = permissive`
9. deaktivace firewall
10. reboot systému
11. instalace puppet augeas
12. nastavení puppet augeas

```
augtool -s <<EOA
set /files/etc/puppet/puppet.conf/main/server <LCM-FQDN>
set /files/etc/puppet/puppet.conf/main/pluginsync true
EOA
```

13. spuštění puppet agenta příkazem `puppet agent -t -w 30`
    Tento příkaz spustí registraci Edge do dané zone, na kterou je nutné reagovat zadáním příkazu pro autorizaci certifikátu
    `pupet cert sign <edge-FQDN>`

14. připojení do Foreman

    Pomocí nástroje Foreman lze ověřit, zda došlo k registraci Edge pvrku a zároveň nastavit několik parametrů nutných ke správné funkci.

    Edge je nutné přidělit do Enviroment shodného s názvem zone a do Host Group na hodnotu Zone123-compute, kde část Zone123 je zvoleným zone jménem.

    Následující tabulky shrnují konfigurační parametry v jednotlivé oblasti.

**Tabulka 5.6:** Konfigurační parametry PLUMgrid Edge prvku

| **Parametr** | **Význam** | **Hodnota** |
|---|---|---|
| plumgrid_ip | IP všech Director | 10.0.107.96 |

**Tabulka 5.7:** Konfigurační parametry OpenStack Edge prvku

| **Parametr** | **Význam** | **Hodnota** |
|---|---|---|
| controller_address_api | IP všech Director | 10.0.107.96 |
| controller_address_management | IP všech Director | 10.0.107.96 |
| mysql_allowed_hosts | povolení mysql hosti | "10.0.107.%" |
| network_api | IP uživatelské API sítě | 10.0.107.0/24 |
| network_data | IP sítě pro data | 10.0.107.0/24 |
| network_management | IP sítě pro management | 10.0.107.0/24 |
| nova_libvirt_type | nastavení knihoven | kvm |
| storage_address_api | IP všech Director | 10.0.107.96 |
| storage_address_management | IP všech Director | 10.0.107.96 |

15. spuštění `puppet agent -vt` čímž dojde ke konfiguraci podle parametrů nastavených v předchozím kroku.

### 5.2.4 Instalace Gateway

Posledním prvkem nutným pro funkci PLUMgrid ONS2.0 je Gateway zajišťující konektivitu s okolními sítěmi. Tento prvek byl shodně jako Edge reprezentován prostřednictvím fyzického serveru Supermicro s konfigurací uvedenou v tabulce 5.1. Postup instalace
a konfigurace Gateway prvku:

1. instalace operačního systému CentOS 7.0 minimal item nastavení síťového rozhraní v příslušném konfiguračním souboru, u CentOS
   `/etc/sysconfig/netwrok-scripts/jmeno_rozhrani`
   Opět je nutné nakonfigurovat statikou IP adresu společně s nastavením výchozí brány a DNS serveru pro zajištění konektivity do internetu.

2. nastavení hostname a FQDN

3. přidání záznamu do souboru `/etc/hosts` ve tvaru
   `<Gateway IP> <Gateway FQDN> <Director hostname>\<LCM IP>`

`<LCM FQDN> <LCM>Hostname>`

Tento záznam lokálně přiřadí IP adresu k danému doménovému jménu LCM VM.

4. instalace balíčků za pomoci příkazů

```
yum install -y http://repos.fedorapeople.org/repos/
 openstack/openstack-icehouse/rdo-release-icehouse-
 4.noarch.rpmn
yum install -y http://dl.fedoraproject.org/pub/epel/
  7/x86_64/e/epel-release-7-5.noarch.rpm
```

5. do souboru `/etc/yum.repos.d/epel.repo` přidat parametr `exclude=python-sqlalchemy,python-migrate`

6. vytvoření souboru `/etc/yum.repos.d/plumgrid.repo` v roli repositáře odkazující na LCM VM pod adresou

```
http://<LCM IP>:81/yum/everest/el7/x86_64/
```

7. import GPG key pro LCM VM

8. v souboru `/etc/selinux/config` nastavení `SELINUX = permissive`

9. deaktivace firewall

10. reboot systému

11. instalace puppet augeas

12. nastavení puppet augeas

```
augtool -s <<EOA
set /files/etc/puppet/puppet.conf/main/server <LCM-FQDN>
set /files/etc/puppet/puppet.conf/main/pluginsync true
EOA
```

13. spuštění puppet agenta příkazem `puppet agent -t -w 30`
    Tento příkaz spustí registraci Gateway do dané zone, na kterou je nutné reagovat zadáním příkazu pro autorizaci certifikátu
    `pupet cert sign <Gateway-FQDN>`

14. připojení do Foreman

    Pomocí nástroje Foreman lze ověřit, zda došlo k registraci Gateway a zároveň nastavit několik parametrů nutných ke správné funkci.

    Gateway je nutné přidělit do Host Group na hodnotu Zone123-compute, kde část Zone13 je zvoleným jménem zone.

Následující tabulky shrnují konfigurační parametry v jednotlivé oblasti.

**Tabulka 5.8:** Konfigurační parametry PLUMgrid Gateway prvku

| **Parametr** | **Význam** | **Hodnota** |
|---|---|---|
| plumgrid_ip | IP všech Director | 10.0.107.96 |
| gateway_devs | název rozhraní v roli gateway | ["enp129s0f1"] |

**Tabulka 5.9:** Konfigurační parametry OpenStack Gateway prvku

| **Parametr** | **Význam** | **Hodnota** |
|---|---|---|
| controller_address_api | IP všech Director | 10.0.107.96 |
| controller_address_management | IP všech Director | 10.0.107.96 |
| mysql_allowed_hosts | povolení mysql hosti | "10.0.107.%" |
| network_api | IP uživatelské API sítě | 10.0.107.0/24 |
| network_data | IP sítě pro data | 10.0.107.0/24 |
| network_management | IP sítě pro management | 10.0.107.0/24 |
| nova_libvirt_type | nastavení knihoven | kvm |
| storage_address_api | IP všech Director | 10.0.107.96 |
| storage_address_management | IP všech Director | 10.0.107.96 |

15. spuštění `puppet agent -vt` čímž dojde ke konfiguraci podle parametrů nastavených v předchozím kroku.

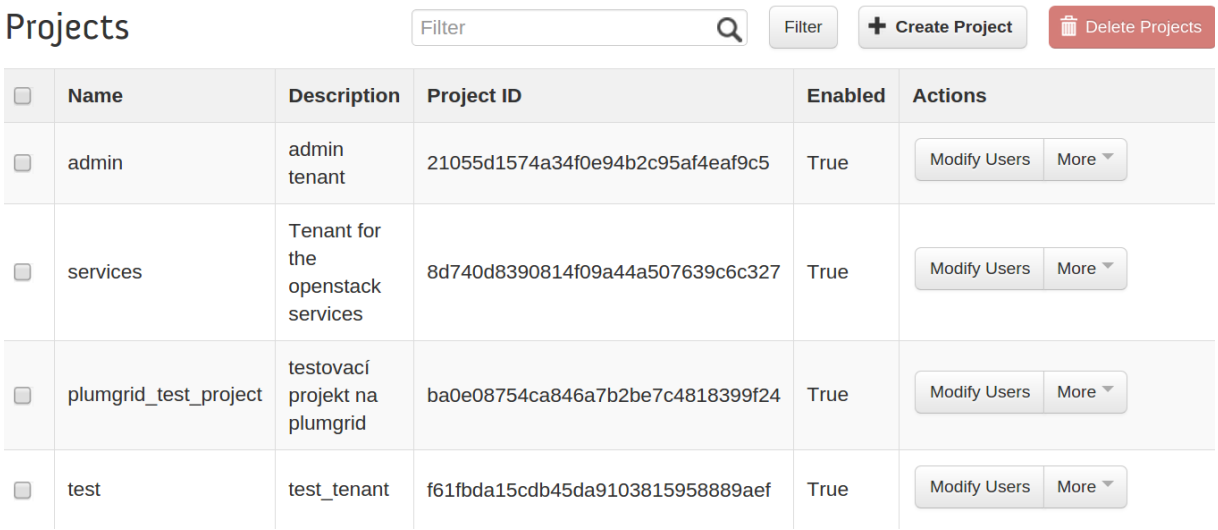
Po dokončení instalace těchto součástí je připravena infrastruktura pro PLUMgrid ONS 2.0. Je ještě nutné připravit prostředí v OpenStack. Následující tabulka shrnuje jednotlivá prostředí s přístupovými informacemi ve výchozím nastavení.

**Tabulka 5.10:** Přístupové informace ke konfiguračním prostředím

| **Prostředí** | **Přístup** | **Login** | **Heslo** |
|---|---|---|---|
| Foreman | IP LCM: 10.0.107.95 | admin | changeme |
| OpenStack | IP Director: 10.0.107.96 | admin | changeme |
| PLUMgrid | virtuální IP: 10.0.107.97 | dle parametrů při instalaci Director | |

### 5.2.5 Konfigurace OpenStack

PLUMgrid ONS 2.0 funguje jako plugin do OpenStack, proto je nutné některá nastavení provést v samotném prostředí OpenStack. Následující část tyto kroky popisuje. Prvním krokem je Tenant neboli project poskytující oddělené prostředí, který se vytváří v Admin -> Identity Panel -> Projects kliknutím na Create Project. Tenant je vhodný pro reprezentaci prostředí jednotlivých členů cloud prostředí.




**Obrázek 5.2:** Vytvoření Tenantu neboli Projectu v OpenStack prostředí

Pro nový tenant je nutné vytvořit sítě, a to externí a interní. Přestože byla na obrázku 5.1 interní síť 10.0.107.0/24, v kontextu OpenStack vytvoříme ještě novou interní síť, která bude poskytovat IP adresy jednotlivým prvkům. Tyto adresy budou následně pomocí statického NAT překladu překládány na adresy ze sítě externí. Síť se vytváří v Projects -> Network -> Networks kliknutím na Create Network.

| Project | Network Name | Subnets Associated | Shared | Status | Admin State | Actions |
|---|---|---|---|---|---|---|
| admin | external_network | **external_subnet** 10.0.104.0/24 | No | ACTIVE | UP | Edit Network \| More |
| plumgrid_test_project | net1 | 192.168.24.0/24 | No | ACTIVE | UP | Edit Network \| More |

**Obrázek 5.3:** Vytvoření sítě v OpenStack prostředí

Pro funkční komunikaci mezi vytvořenými sítěmi je nutné vytvořit router. Router se vytváří v Projects -> Network -> Routers kliknutím na Create Router.

Routers

+ Create Router | Delete Routers

| Name | Status | External Network | Actions |
|---|---|---|---|
| router1 | Active | external_network | Clear Gateway \| More |

**Obrázek 5.4:** Vytvoření routeru v OpenStack prostředí

Router je nutné nakonfigurovat přidáním rozhraní pomocí Add Interface pro nastavení interní a externí gateway (brány). Konkrétní konfigurace routeru je zobrazena na obrázku 5.5.

Interfaces

+ Add Interface | Delete Interfaces

| Name | Fixed IPs | Status | Type | Admin State | Actions |
|---|---|---|---|---|---|
| (293ce2da) | 10.0.104.2 | ACTIVE | External Gateway | UP | |
| (3ba8a535) | 192.168.24.254 | ACTIVE | Internal Interface | UP | Delete Interface |

**Obrázek 5.5:** Konfigurace routeru v OpenStack prostředí

Nakonec již zbývá pouze vytvoření VM (Virtual Machines - virtuální stroje). VM se vytváří v Projects -> Compute -> Instances kliknutím na Launch Instance.

Instances

Filter | Filter | + Launch Instance | Soft Reboot Instances | Terminate Instances

| Instance Name | Image Name | IP Address | Size | Key Pair | Status | Availability Zone | Task | Power State | Uptime | Actions |
|---|---|---|---|---|---|---|---|---|---|---|
| Ubuntu-GW | Ubuntu_14_04 | 192.168.24.12<br>10.0.104.5 | m1.small \| 2GB RAM \| 1 VCPU \| 20.0GB Disk | hradecky | Active | nova | None | Running | 1 week, 1 day | Create Snapshot<br>More |
| ubuntu-Edge2 | Ubuntu_14_04 | 192.168.24.9<br>10.0.104.7 | m1.small \| 2GB RAM \| 1 VCPU \| 20.0GB Disk | hradecky | Active | nova | None | Running | 1 week, 1 day | Create Snapshot<br>More |
| ubuntu-Edge1 | Ubuntu_14_04 | 192.168.24.8<br>10.0.104.8 | m1.small \| 2GB RAM \| 1 VCPU \| 20.0GB Disk | hradecky | Active | nova | None | Running | 1 week, 1 day | Create Snapshot<br>More |

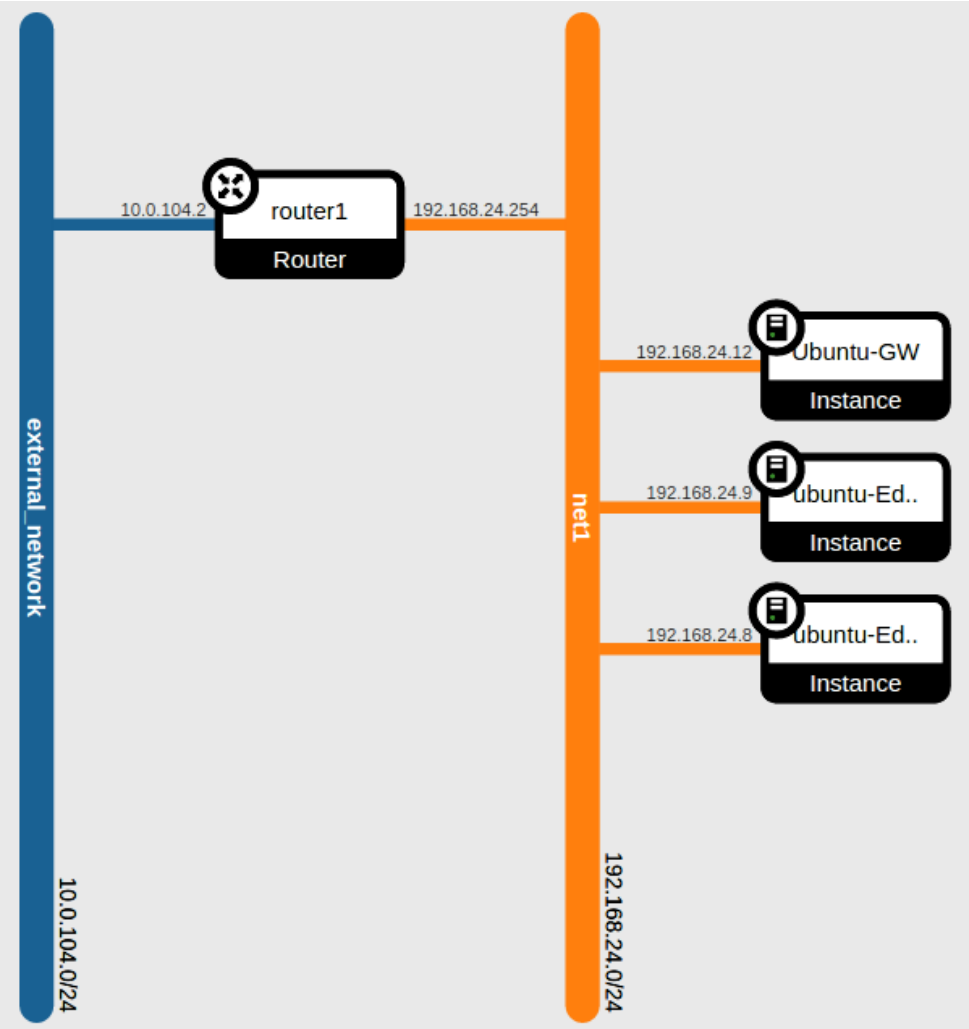
**Obrázek 5.6:** Instance v OpenStack prostředí

## 5.3 Výkonové testy PLUMgrid ONS 2.0

Pro zjišťování výkonu byly provedeny testy propustnosti se zapouzdřením VxLAN i NVGRE a to jak v rámci PLUMgrid prostředí(East/West), tak s externím serverem (North/South). Testovací instance byl Ubuntu 14.04 (2GB RAM, 1 VCPU, 20GB Disk) běžící z QCOW2 v prostředí OpenStack a externí server Supermicor pod označením cpt156 s konfigurací obsaženou v tabulce 5.1. Seznam všech testovacích instancí shrnuje obrázek 5.7. Jako testovací nástroj byl zvolen iperf a iperf3, kde byla měřena vždy half-duplex a full-duplex komunikace s packety protokolu TCP při délce testu 10 vteřin. Síťová topologie v OpenStack je znázorněna obrázkem 5.8.

| Host | Name | Image Name | IP Address | Size | Status | Task | Power State |
|---|---|---|---|---|---|---|---|
| cpt02.dev.tcp.cloudlab.cz | ubuntu-Edge2 | Ubuntu_14_04 | 192.168.24.9 | m1.small \| 2GB RAM \| 1 VCPU \| 20.0GB Disk | Active | None | Running |
| cpt01.maas.test.tcpcloud.cz | ubuntu-Edge1 | Ubuntu_14_04 | 192.168.24.8 | m1.small \| 2GB RAM \| 1 VCPU \| 20.0GB Disk | Active | None | Running |

**Obrázek 5.7:** Testovací instance v OpenStack prostředí




**Obrázek 5.8:** Síťová topologie v OpenStack prostředí

### 5.3.1 East/West test

Výkonový test mezi instancemi Ubuntu VM v OpenStack prostředí běžících na rozdílných Edge prvcích. Instance Ubuntu-Edge2 v roli serveru a Ubuntu-Edge1 v roli klienta.

Hodnoty a role v testu jsou obsaženy v tabulce 5.11. V případě testů v režimu duplex jsou navazována dvě spojení v navzájem opačných směrech, což je důvodem dvou hodnot uvedených ve výsledcích.

**Tabulka 5.11:** East/West TCP test mezi Ubuntu-Edge2 a Ubuntu-Edge1

| **Server** | **Klient** | **Zapouzdření** | **Duplex** | **Přenosová rychlost** |
|---|---|---|---|---|
| Ubuntu-Edge2 | Ubuntu-Edge1 | VxLAN | Ne | 2.70 Gb/s |
| Ubuntu-Edge2 | Ubuntu-Edge1 | VxLAN | Ano | 1.1 Gb/s, 1.53 Gb/s |
| Ubuntu-Edge2 | Ubuntu-Edge1 | NVGRE | Ne | 2.75 Gb/s |
| Ubuntu-Edge2 | Ubuntu-Edge1 | NVGRE | Ano | 1.7 Gb/s, 1.07 Gb/s |

### 5.3.2 North/South test

Výkonový test mezi instancí Ubnutu VM v OpenStack prostředí a externím fyzickým serverem pod označením cpt156. Externí server se shodnou konfigurací jako servery plnící funkci Edge či Gateway. Tento typ testu prověří reálnou schopnost řešení komunikovat s okolní sítí.

Hodnoty a role v testu jsou obsaženy v tabulce 5.12. V případě testů v režimu duplex jsou navazována dvě spojení v navzájem opačných směrech, což je důvodem dvou hodnot uvedených ve výsledcích.

**Tabulka 5.12:** Nort/South TCP test mezi Ubuntu-Edge1 a cpt156

| **Server** | **Klient** | **Zapouzdření** | **Duplex** | **Přenosová rychlost** |
|---|---|---|---|---|
| Ubuntu-Edge1 | cpt156 | VxLAN | Ne | 17.6 Mb/s |
| Ubuntu-Edge1 | cpt156 | VxLAN | Ano | 6.08 Mb/s, 2.12 Mb/s |
| Ubuntu-Edge1 | cpt156 | NVGRE | Ne | 19.9 Mb/s |
| Ubuntu-Edge1 | cpt156 | NVGRE | Ano | 9.11 Mb/s, 2.85 Mb/s |
| cpt156 | Ubuntu-Edge1 | VxLAN | Ne | 8.19 Mb/s |
| cpt156 | Ubuntu-Edge1 | VxLAN | Ano | 6.57 Mb/s, 2.26 Mb/s |
| cpt156 | Ubuntu-Edge1 | NVGRE | Ne | 8.45 Mb/s |
| cpt156 | Ubuntu-Edge1 | NVGRE | Ano | 2.68 Mb/s, 12 Mb/s |

Zhodnocení naměřených hodnot je věnována následující kapitola.

# 6 Shrnutí výsledků

Veškerá provedená konfigurace i samotný návrh experimentu byl konzultován s technickým oddělením PLUMgrid a zástupci firmy tcp cloud a.s., nicméně jak je z obrázků v kapitole 3.4 patrné, výsledky dosažné při testování PLUMgrid ONS 2.0 nedosahují příznivých hodnot.

Při komunikaci v rámci řešení bylo dosaženo rychlosti 2,7 Gb/s a při komunikaci mimo pouze 9 Mb/s, což je na infrastruktuře s podporou 10 Gb/s velice neuspokojivé. Při testech ve směru z externí sítě do PLUMgrid topologie bylo sice dosahováno téměř dvojnásobné rychlosti než ve směru opačném, ale jelikož jsou tyto rychlosti stále v řádu Mb/s, nelze mluvit o uspokojivých hodnotách. Při testovací komunikaci v režimu duplex byly dosažené výsledky obdobné. Jelikož těchto výsledků bylo dosaženo při obou testovaných protokolech zapouzdření (VxLAN, NVGRE) a navýšení se nepovedlo dosáhnout ani po konzultacích s technickým oddělením PLUMgrid, je nutné předpokládat že důvodem pomalého přenosu je samotné řešení.

Během závěrečného ladění testovaného řešení byla na doporučení technického oddělení kontrolovaná především nastavení firewall fyzických prvků, na kterých probíhalo samotné testování. Dále bylo ověřováno i nastavení ohledně offload, které může přinášet problémy spojené se zapouzdřením VxLAN, avšak nízké rychlosti dosahované při zapouzdření NVGRE popíraly možnost, že se jedná o příčinu problému. Možným důvodem nízkých přenosových rychlostí může také být fakt, že v experimentu byl kladen důraz na využití fyzických prvků, přičemž PLUMgrid ONS 2.0 uvádí jako jednu z největších výhod svého řešení možnost plné virtualizace. Tato skutečnost by však výrazně omezila možnosti reálného využití s hypervisorem KVM.

Dosažené výsledky tedy z důvodu nízkých přenosových rychlostí nemluví ve prospěch implementovaného řešení a to i přes nabízenou flexibilitu a automatizaci, kterou přináší do oblasti sítí v prostředí cloudu.

# 7 Závěry a doporučení

Téma SDN slibuje značné usnadnění pro správu a automatizaci sítí. Na základě rozdělení jednotlivých entit a přidělení klíčových cílů se snaží díky abstrakci přiblížit práci v oblasti počítačových sítí spíše k programování. V mnoha ohledech však technologie SDN naráží na problémy. Skutečnost, že se jedná o novou technologii s sebou nese určité riziko při jejím nasazení. Každá nová technologie se potýká s nižším rozsahem dokumentace dané problematiky, což může značně prodloužit řešení rozličných situací. Toto riziko si spousta společností může vyložit za příliš vysoké a upustit tak od jejího nasazení.

Cílem této páce bylo prozkoumat oblast SDN a implementovat jedno vybrané řešení na základě funkčního porovnání dvou zvolených zástupců. Dále pak na základě teoretického zkoumání a výsledků implementace posoudit reálnost nasazení.

Obecně bylo zjištěno, že SDN technologie nabízí nikoli revoluční pohled na danou problematiku, ale spíše evoluční postup správným směrem, ke zjednodušení. Stále zde však zůstává velký prostor pro pokrok, který přivede SDN do stavu, kdy její nasazení nebude znamenat riziko spojené s novou technologií se spoustou neznámých.

Ze zmíněných důvodů a výsledků získaných v této práci, lze SDN považovat za správné řešení s velikým potenciálem změnit oblast počítačových sítí, avšak stále se nacházející ve stádiu vývoje. Je dobré, že se SDN dostalo vysoké popularity, jelikož to značně urychlí jeho vývoj. SDN již v této době nachází využití ve firemních prostředí, nicméně vyžaduje tým pracovníků schopných pohotově pracovat i v neznámé oblasti.

Závěrem bych tedy doporučil s nasazením SDN do firemního prostředí ještě nějakou dobu vyčkat. Další možností, která však bude vyžadovat spojené náklady, je ponechání současného řešení a souběžné testování SDN pro možnosti vlastního nasazení.

# Literatura


[1] NADEAU, Thomas D. *SDN: Software Defined Networks*. Cambridge: O'Reilly, c2013, xxvii, 352 s. ISBN 978-1-449-34230-2

[2] SHUKLA, Vishal. *Introduction to software defined networking: OpenFlow*. North Charleston: CreateSpace, c2013, ix, 103 s. ISBN 978-1-48267-813-0

[3] STALLINGS, William. *Software-Defined Networks and OpenFlow*[online]. [cit. 2015-01-30]. Dostupné z `http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_16-1/161_sdn.html`

[4] KRISHNAMURTHY, Anand, S.P. Chandrabose, and A. Gember-Jacobson. *Pratyaastha: an efficient elastic distributed SDN control plane*. Proceedings of the third workshop on Hot topics in software defined networking. 2014, ACM: Chicago, Illinois, USA. p. 133-138.

[5] OLZAK, Tom. *Software Defined Networking Enhances Security Management*. Toolbox [online]. 2013 [cit. 2015-04-04]. Dostupné z: `http://it.toolbox.com/blogs/adventuresinsecurity/software-defined-networking-enhances-security-management-57741`

[6] GŐRANSSON, Paul a Chuck BLACK. *Software defined networks: a comprehensive approach*. Morgan Kaufmann, 2014, xxv, 325 pages. ISBN 978-012-4166-752.

[7] PASCUAL, Inaki. *OpenStack Neutron & Software Defined Networks (SDN)*. Slideshare [online]. 2014 [cit. 2015-04-04]. Dostupné z: `http://www.slideshare.net/inakipascual/openstack-neutron-and-sdn`

[8] AMIES, Alex, Sluiman, Harm; Tong, Qiang Guo; Liu, Guo Ning *Developing and hosting applications on the cloud*. Upper Saddle River, NJ: IBM Press/Pearson, 2012. ISBN 9780133066845

[9] *OpenStack Open Source Cloud Computing Software*. [online]. 2014 [cit. 2015-04-10]. Dostupné z: https://www.openstack.org/

[10] KUBICA, Tomáš. *OpenStack (3) - letmý pohled na kataklyzmaticky vybuchnuvší hvězdu*. NetSvět [online]. 2014 [cit. 2015-04-10]. Dostupné z: http://goo.gl/GepeMz

[11] KUBICA, Tomáš. *OpenStack (4) - síťová neutronová hvězda*. NetSvět [online]. 2014 [cit. 2015-04-10]. Dostupné z: http://www.netsvet.cz/clanky.html/7_28-openstack-4-sitova-neutronova-hvezda

[12] DENTON, James. *Learning OpenStack Networking (Neutron)*. Packt Publishing, 2014, 300 s. ISBN 978-1783983308.

[13] *PLUMgrid | The Wolrdwide leader of Secure Cloud Networks* [online]. [cit. 2015-02-24]. Dostupné z: http://www.plumgrid.com/technology/overview/

[14] *PLUMgrid | The Wolrdwide leader of Secure Cloud Networks* [online]. [cit. 2015-03-20]. Dostupné z: http://www.plumgrid.com/

[15] RAFAY, Zafar. *PLUMgrid ONS: v2.0.5 RDO Manual Install Deployment Guide*. 2015

[16] *The Foreman* [online]. 2013 [cit. 2015-03-20]. Dostupné z: http://theforeman.org/

[17] *Foreman – hromadná instalace a konfigurace serverů (1. díl)*. ZAPLETAL, Lukáš. Fedora: Česká komunita linuxové distribuce Fedora [online]. 2013 [cit. 2015-03-20]. Dostupné z: http://fedora.cz/foreman-hromadna-instalace-a-konfigurace-serveru-1-dil/

[18] *Puppet Labs*. Puppet Labs Documentation [online]. 2015. [cit. 2015-03-20]. Dostupné z: http://docs.puppetlabs.com/

[19] CORNEC, Bruno. *HP Helion OpenStack - Definition, Architecture and Status*. Slideshare [online]. 2014 [cit. 2015-04-16]. Dustupné z: http://www.slideshare.net/eurolinux/helion-meetup2014

[20] KUBICA, Tomáš. *Helion OpenStack (2) - budoucnost je v otevřenosti*. NetSvět [online]. 2014 [cit. 2015-04-16]. Dostupné z: http://www.netsvet.cz/clanky.html/7_180-helion-openstack-2-budoucnost-je-v-otevrenosti

[21] KUBICA, Tomáš. *HP Virtual Cloud Networking (1) – bez SDN není cloud*. NetSvět [online]. 2014 [cit. 2015-04-16]. Dostupné z: `http://goo.gl/CsSt5u`

[22] KUBICA, Tomáš. *HP Distributed Cloud Networking (1) - SDN overlay pro náročné*. NetSvět [online]. 2014 [cit. 2015-04-16]. Dostupné z: `http://goo.gl/oc9xir`

FIM UHK
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: **Tomáš Hradecký**

Obor studia: Aplikovaná informatika
Jméno a příjmení vedoucího práce: **Jakub Pavlík**

Název práce:
**Možnosti nasazení SDN ve firemním prostředí**

Název práce v AJ:
Implementation of SDN on enterprise networks

Podtitul práce:

Podtitul práce v AJ:

Cíl práce: Cílem této bakalářské práce je seznámit čtenáře s teoretickými základy SDN společně s prozkoumáním SW controllerů PlumGrid a HP SDN. Dále je práce zaměřena na porovnání zmíněných controllerů.

Osnova práce:
Úvod
Principy SDN
PlumGrid
HP SDN
Porovnání PlumGrid x HP SDN
Závěr

Projednáno dne: 8.12.2015

Podpis studenta

Podpis vedoucího práce