OSHI – Open Source Hybrid IP/SDN sieťovanie

Bc. Juraj Volko
Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave
Bratislava, Slovensko
xvolko@stuba.sk

Bc. Marek Vlha
Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave
Bratislava, Slovensko
xvlha@stuba.sk

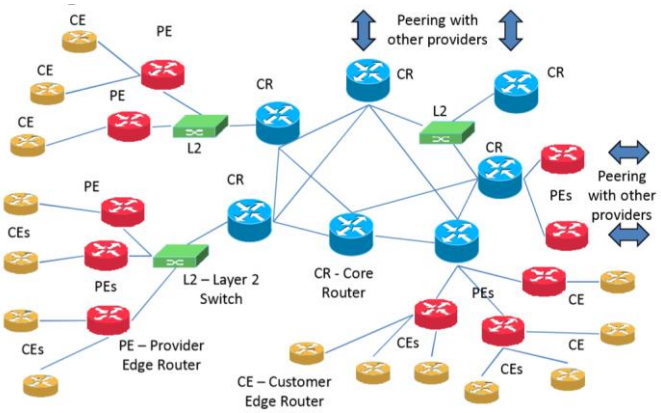
***Abstrakt* – Pre zapojenie SDN do internet backbone je nutná koexistencia IP smerovania a smerovania na základe SDN. V tejto práci je predstavená architektúra a použitie hybridného IP/SDN sieťovania. Ďalej je popísaný dizajn a open source implementácia hybridných IP/SDN (OSHI) uzlov. Tieto uzlty používajú Quagga-u pre OSPF smerovanie a OpenvSwitch pre OpenFlow smerovanie na Linuxe. Dostupnosť zariadení na validáciu a získanie výkonnosti SDN riešení je nutné pre ich vývoj. Výsledkom tejto práci vznikli open source nástroje, ktoré umožňujú vytvorenie hybridných IP/SDN sietí, ich nasadenie v Mininete alebo v distribuovaných SDN výskumných testovacích zariadeniach. Pre navrhnuté riešenie boli vyhodnotené kľúčové vlastnosti. Nasadenie OSHI je dostupné pre VirtualBox a bol vytvorený VM image. Riešenie z článku je testované použitím Mininetu s rozšírením o OSHI uzly a s použitím dostupnej VM image.**

Kľúčové slová – Software Defined Networking; Open Source; Emulácia

I. ÚVOD

Softvérovo definované siete (SDN) sú predstavované ako nový koncept, ktorý môže zmeniť fungovanie internetových sietí ako ich poznáme dnes. SDN sa používajú primárne v dátových centrách ale je mnoho pokusov o širšie použitie SDN v internetových sieťach. Základom SDN je oddelenie "control a data plane", ktoré umožní externému SDN kontroléru dynamicky aplikovať pravidlá pre SDN uzly. Na základe týchto pravidiel dokážu uzly vykonávať kontrolu vlastností siete, manipuláciu a smerovanie predovšetkým však dokážu kontrolovať a modifikovať hlavičky (header) paketu na rôznych úrovniach protokolov od druhej úrovne až po aplikačnú.

Na obrázku 1 je znázornený príklad poskytovateľa doménovej siete prepojenej s inými poskytovateľmi použitím BGP. Tento poskytovateľ umožňuje používateľom prístup na internet ako aj prístup k podobným transportným službám. Znázornená sieť sa skladá z niekoľkých hlavných smerovačov (CR – core router) a poskytovateľa (PE – provider edge), ktoré sú prepojene použitím point to point prepojenia. Koncový používateľ (CE – customer edge) sa pripája na PE, kde sú najčastejšie implementované IP a MPLS technológie. Pomocou MPLS sa vytvoria tunely medzi smerovačmi ktoré dokážu byť použité na vylepšenie smerovania regulárnych IP tokov použití ochrany pred chybou alebo snaha vyhnúť sa použitiu BGP smerovacej tabuľky na medzidoménovú komunikáciu. Nevýhodou MPLS je že používa tradičný architektúru "control plane", ktorá nie je prístupná novým riešeniam.



Obrázok 1 Distribuovaná sieť

Premena takejto siete na SDN by bola možná napríklad nahradením IP jadier a prístupových smerovačov SDN smerovačmi, čo by umožnilo realizáciu inovačných služieb ako aj optimalizáciu už existujúcich. Pri takomto prechode je nutné rátať so spoluprácou IP a SDN smerovania podobne ako pri IP a MPLS. Cieľom je zachovať výhody smerovacích zariadení IP ako aj SDN ako sa podarilo pri optimalizácií kontrolného a smerovacieho "plane" počas rokov a sprístupniť tak možnosť inovovania internetu.

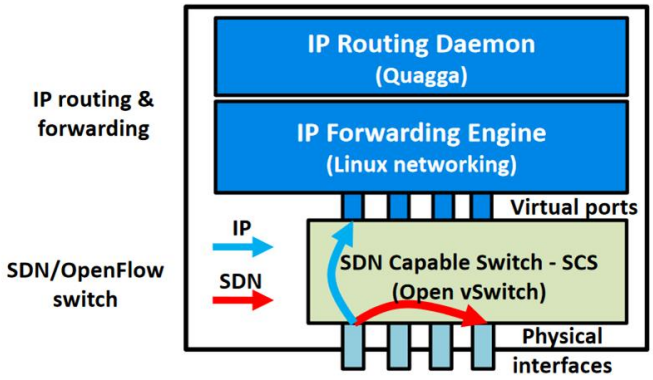
II. HYBRIDNÉ IP/SDN SIETE

Použitie IP/SDN má jasne dané vytváranie MPLS tunelov (LSP) zatiaľ čo SDN siete poskytujú možnosť vytvoriť tunely použitím širšieho spektra protokolov (TCP/UDP, IP, VLANs, Ethernet, MPLS). Hybridné IP/SDN uzly sú navrhnuté tak, aby služby poskytovali koexistenciu IP tokov a SBP (SDN based path) tokov a možnosť vytvárania tunelov. SBP predtavuje tunely vytvorené v SDN sieti. SDN poskytuje veľkú flexibilitu a dokáže klasifikovať pakety na viacerých úrovniach, takže je možné zadefinovať pravidlá aj pre IP úroveň avšak táto flexibilita môže spôsobiť prílišnú komplexnosť a teda je nutné brať do úvahy zvýšenú šancu na nesprávne nakonfigurovanie alebo chyby v smerovaní. Navrhnutá Hybridná IP/SDN sieť má v sebe implementovanú VLL (Ethernet Virtual Leased Line) službu, ktorá spôsobuje to, že sa dva koncové body správajú tak, ako by boli prepojené v jednej eternetovej sieti. Rozlíšenie,

či sa bude používať IP alebo SDN je rozhodnuté na základe VLAN tagu. Regulárny IP tok nenesie VLAN tag.

III. ARCHITEKTÚRA UZLA OSHI

OSHI uzol znázornený na obrázku 2 je navrhnutý kombináciou SDN smerovača (SCS – SDN Capable Switch), IP smerovacieho enginu a IP smerovacieho démona. SCS je prepojený s fyzickým rozhraním zatiaľ čo IP smerovací engine je prepojený s SCS cez virtuálne porty . SCS je implementovaný použitím Open vSwitch, IP engine používa Linux smerovanie a ako smerovací démon je použitá Quaga.



Obrázok 2 OSHI uzol

Každý vnútorný virtuálny port je prepojený s fyzickým portom hybridnej siete aby mohol IP smerovací engine pracovať nad virtuálnymi portmi a ignoruje tak fyzické porty. SCS slúži v prvom rade na to, aby rozoznal pakety. Regulárne IP pakety posiela z fyzických portov na virtuálne na spracovanie IP smerovacím enginom ktorý je kontrolovaný smerovacím démonom. Tento prístup zabraňuje tomu, aby musela byť IP smerovacia tabuľka duplicitne použitá aj v SDN pravidlách. Nevýhodou tohto riešenia je fakt, že regulárny IP paket bude v SCS spracovávaný dvakrát (raz pri vstupe z fyzického portu, druhý raz pri vstupe z virtuálneho portu). Vytvorenie VLL tunelu je implementované s SBP smeruje VLAN tagy medzi dvoma koncovými užívateľmi. Vytvorenie SBP zabezpečuje python skript, ktorý používa TOPOLOGY REST API kontrolera Floodlight. Pomocou tohto kontroleru sa získa cesta medzi dvoma koncovými používateľmi a následne vytvorí spojenie.

IV. OSHI EMULAČNÉ NÁSTROJE

Pre realizáciu OSHI uzlov a ich otestovanie bolo vytvorených niekoľko emulačných nástrojov. Implementovali svoje vlastné uzly do nástrojov pre možnosť testovania. Medzi tieto nástroje patria Virtual Box, Mininet a OFELIA. Naše úsilie sa sústredilo na Mininet Emulátor a pokusy na ňom.

A. *Mininet rozšírenie*

Rozšírením , ktorým sme sa zaoberali bolo teda rozšírenie pre nástroj Mininet. Tento nástroj v základe ponúka možnosť vytvárania switch uzly alebo host uzly. Toto bolo potrebné pre potreby projektu rozšíriť o pridávanie vlastných uzlov. Jedná sa o vyššie spomínané OSHI uzly. Bola pridaná uzly, v ktorých bolo umožnené rozbehnutie Quagga a OSPFD démonov. Potom k nim bola pridaná OVS funkcionalita, pre realizáciu OSHI uzlov. Uzol je možné pozrieť si na obrázku 2. Nástroj Mininet Deployer je schopný realizovať všetky aspekty experimentov, zahŕňajúc konfiguráciu IP adries A dynamického smerovania vo všetkých uzloch.

B. *VLL Pusher*

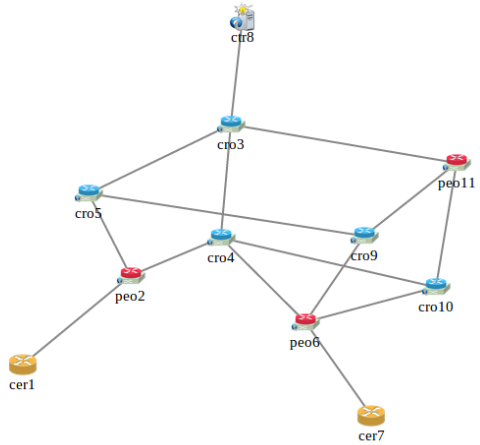
Tento nástroj dokáže do vytvorenej siete v emulátore Mininet nahrať respektíve nastaviť všetky zostávajúce prepojenia z konfiguračného súboru. Tento súbor sa vytvára pri zapnutí emulácie v nástroji Mininet. Je vytvorený z JSON návrhu Topológie. Vytvára takzvané „virtuálne okruhy“ a taktiež VLL tunely medzi jednotlivými uzlami podľa konfigurácie.

V. TESTOVANIE

Testovanie bolo realizované na virtálnom stroji v nástroji Virtual Box. Najprv bol používaný normálny Mininet so stiahnutými doplnkami, konkrétne s Mininet extension a VLL Pusher. Tieto zdrojové kódy však neboli aktuálne a spolu kompatibilné preto nebolo možné zapojiť do behu „virtuálne okruhy“ a ani VLL tunely. Preto bola vybraná alternatívna možnosť a to použitie virtuálneho stroja priamo z [2], ktorý obsahuje aktuálne a nakonfigurované nástroje potrebné pre uskutočnenie experimentu.

A. *Topológia*

Prvým problémom, s ktorým sme sa stretli, bolo vytvorenie topológie. Pre tento účel bola využitá služba Topology Designer. V nej je možné vytvárať OSHI topológie a následne ich vyexportovať pre použitie v mininete. Bohužiaľ JSON formát nesedel úplne s požiadavkami rozšírenia Mininetu a preto bolo potrebné JSON súbor opraviť do podoby akceptovateľnej pre nástroj Mininet. Zostavená bola topológia, ktorú je možné vidieť na obrázku 3. Táto topológia je podobná s topológiou z pôvodného článku, avšak boli odstránené nepotrebné uzly a časti irelevantné pre projekt a testovanie.



Obrázok 3 Testovaná topológia.

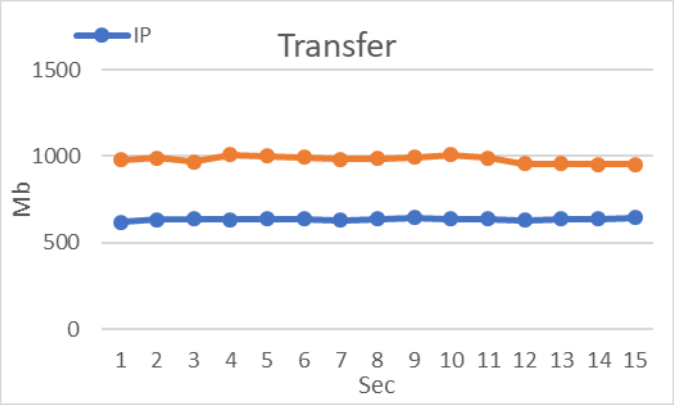
Testovaná topológia pozostáva z dvoch koncových uzlov, ďalej troch Prístupových OSHI uzlov a piatich tzv. Core OSHI uzlov. Samozrejme topológia obsahuje aj kontroler pre celkové riadenie.

B. *Emulácia v Mininete*

Rozšírenie Mininetu pre OSHI umožňuje vytvorenie siete navrhnutej topológie z JSON súboru. Toto rozšírenie automaticky vytvorí danú sieť podľa súboru, respektíve vytvorí uzly a fyzické pripojenia, teda pripojenia fyzických koncových používateľov do siete. Sieť ma teda vytvorené uzly no nie sú vytvorené niektoré prepojenia medzi nimi nazývané aj v "virtuálne okruhy". Tieto prepojenia je treba vytvoriť pomocou ďalšieho nástroja nazývaného VLL Pusher. Ide o nástroj, ktorý vďaka vytvorenému konfiguračnému súboru dokáže vytvoriť spojenia medzi uzlami, ktoré ešte neboli prepojené a sú definované v topológii. Tento konfiguračný súbor sa automaticky vytvorí pri vytváraní uzlov siete, teda pri spúšťaní Mininet emulátora. VLL Pusher vytvorí všetky predom definované „virtuálne okruhy" a takisto VLL tunely. Po tomto je sieť vytvorená , nakonfigurovaná a plne funkčná. Sieť je prípravná pre účely testovania.

C. *Testovací scenár*

Testovanie v tomto článku sa zaoberá myšlienkou a porovnaním z článku [1]. Testovací scenár je postavený na otestovaní a porovnaní OSHI siete. Jedná sa o funkčné vytvorenie OSHI siete a teda jej uzlov aj jej prepojení. Testovaná topológia je topológia na obrázku 3.



Graf 1 Transfer (Mb/sec)

Testovanie ako je spomenuté v [1] sa týkalo priemernému transferu a troughputu. Ide o porovnanie transferu v OSHI sieti za pomoci bežného smerovania a za pomoci VLL tunelu. Na grafe 1 môžeme vidieť porovnanie medzi týmito dvoma scenármi. Takisto sa môžeme pozrieť do tabuľky 1, kde vidíme, že nárast priemerného troughput za sekundu sa pri použití VLL tunela oproti IP smerovania pohybuje niekde okolo 36%.

	VLL (Mb/s)	IP (Mb/s)
Priemer	981.93333	636.4
STD Odchýlka	4.13	16.08

Tabuľka 1.

VI. Záver

V tomto článku sme si predstavili koncept OSHI, ako koncept novej siete používajúcej špeciálne OSHI uzly. Ďalej sme si predstavili tieto uzly a nástroje a rozšírenia, vďaka ktorým je možné robiť experimenty s konceptom tejto siete. Taktiež sme si zostavili a vyskúšali tieto nástroje vlastným experimentom a ukázali naše výsledky.

[1] Stefano Salsano, OSHI - Open Source Hybrid IP/SDN Networking (and its Emulation on Mininet and on Distributed SDN Testbeds) - IEEE Conference Publication Published in: Software Defined Networks (EWSDN), 3. Sep. 2014 [ISBN]: 978-1-4799-6919-7

[2] Networking group, University of Rome, [Online] : http://netgroup.uniroma2.it/twiki/bin/view/Oshi