ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

DIPLOMOVÁ PRÁCA

ZDENKO HOLEŠA

Technológia softvérovo-definovaných sietí a výučba KIS

Vedúci práce: doc. Ing. Pavel Segeč, PhD.

Registračné číslo: 355/2015

Žilina, 2016

ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

DIPLOMOVÁ PRÁCA

APLIKOVANÉ SIEŤOVÉ INŽINIERSTVO

ZDENKO HOLEŠA

Technológia softvérovo-definovaných sietí a výučba KIS

Žilinská univerzita v Žiline

Fakulta riadenia a informatiky

Katedra informačných sietí

Žilina, 2016

ŽILINSKÁ UNIVERZITA V ŽILINE, FAKULTA RIADENIA A INFORMATIKY.

ZADANIE TÉMY DIPLOMOVEJ PRÁCE.

Študijný program : Aplikované sieťové inžinierstvo

Zameranie: Sieťová infraštruktúra

Meno a priezvisko	Osobné číslo
Zdenko Holeša	555436

Názov práce v slovenskom aj anglickom jazyku

Technológia softvérovo-definovaných sieti a výučba KIS

Technology of software-defined networks and the department teaching

Zadanie úlohy, ciele, pokyny pre vypracovanie (Ak je málo miesta, použite opačnú stranu)

Cieľ diplomovej práce:
Katedra identifikuje tému SDN ako jednu z progresívnych technológií IP sietí, ktorá sa začína objavovať v zozname požadovaných kompetencií pre absolventov odboru ASI. Katedra preto do nového programu ASI navrhla aktualizáciu predmetu Integrácia sietí, v ktorom by sa mali absolventi danej téme venovať. Hlavným cieľom práce je teoreticky a prakticky sa oboznámiť s technológiou a v úzkej spolupráci s vedúcim vypracovať materiál implementácie SDN do vyučovania. Na základe oboznámenia sa s SDN je nevyhnutné rozpracovať metodiku ako a v akom rozsahu implementovať SDN do štúdia predmetu IS ako aj navrhnúť, implementovať a realizovať praktické témy pre potreby vedenia cvičení.

Obsah:
Pri riešení sa zamerajte:

- Analýza súčasnej technológie SDN, princípov, architektúry a protokolov s ohľadom na aktuálny stav. Technológia OpenFlow protokol a jeho fungovanie.
- Analýza SDN kurzov ponúkaných vo svete z pohľadu rozsahu, tém a cvičení. Analýza teoretických a praktických aspektov zvládnutia učenia sa SDN.
- Stanovenie cieľov práce.
- Analýza a výber tém vhodných pre predmet. Identifikácia a návrh postupu vhodného pre vedenie prednášok a cvičení.
- Analýza a výber vhodných komponentov (softvérových/hardvérových) ako demonštrácia riešenia SDN vhodného pre vyučovanie a laboratória KIS. Zahŕňa výber vhodnej distribúcie OS Linux, výber kontroléra (zvážte použitie riešenia OpenDaylight) a openFlow prepínača. Zvážte aj návrh a výber vhodného HW pre potreby vedenia cvičení. Ako súčasť riešenia zvážte aj využitie, resp. integráciu samostatných prvkov so sieťovým emulátorom mininet. Pri riešení analyzujte a zapracujte aj možnosti tvorby služieb prostredníctvom programovania kontroléra. Po dohode s vedúcim realizujte príklady programového rozšírenia.

Témy z predmetov študijného zamerania 5SI041: 2, 5, 6
Meno a pracovisko vedúceho DP: doc. Ing. Pavel Segeč, PhD., KIS, ŽU
Meno a pracovisko tútora DP:

08-02-2016

vedúci katedry
(dátum a podpis

Zadanie zaregistrované dňa 09.11.2015 pod číslom 355/2015 podpis _____

- Vypracovanie teórie a praktických demonštrácii tém do vhodnej hĺbky. Navrhované riešenia overte, zdokumentuje a vyhodnoťte.
- V závere navrhnite možnosti ďalšieho skúmania SDN/OpenFlow na KIS s využitím vo vyučovaní.

Poďakovanie

Chcel by som sa poďakovať vedúcemu diplomovej práce doc. Ing. Pavlovi Segečovi, PhD. za pripomienky, odbornú pomoc a usmerňovanie pri tvorbe tejto práce.

ABSTRAKT V ŠTÁTNOM JAZYKU

HOLEŠA, Zdenko: *Technológia softvérovo-definovaných sietí a výučba KIS*. [Diplomová práca]. – Žilinská univerzita. Fakulta riadenia a informatiky; Katedra informačných sietí. – Vedúci: doc. Ing. Pavel Segeč, PhD. – Stupeň odbornej kvalifikácie: Inžinier v študijnom programe Aplikované sieťové inžinierstvo. – Žilina: FRI ŽU, 2016. - 62 s.

Cieľom diplomovej práce bolo na základe oboznámenia sa s SDN sieťami navrhnúť materiál implementácie SDN do vyučovania. V teoretickej časti práce je popísaná analýza súčasného stavu vyučovania SDN vo svete, oboznámenie sa s technológiou SDN a prieskum dostupných SDN riešení. Praktická časť práce pozostáva z návrhu implementácie SDN do vyučovania. To zahŕňa výber komponentov SDN riešenia, výber a zariadenie laboratória KIS a návrh osnovy pre predmet Integrácia sietí.

Kľúčové slová: Softvérovo-definované siete (SDN).

ABSTRAKT V CUDZOM JAZYKU

HOLEŠA, Zdenko: *Technology of software-defined networks and department teaching*. [Master's thesis]. – The Univesity of Žilina. Faculty of Management Science and Informatics; Department of InfoComm Networks. – Tutor: doc. Ing. Pavel Segeč, PhD. – Qualification level: Master in study program Applied Network Engineering. – Žilina: FRI ŽU, 2016. - 62 p.

The goal of master's thesis was, based on the familiarization with SDN networks, to design material of SDN implementation to teaching. In theoretical part of thesis analysis of actual situation of SDN teaching in world, familiarization with SDN and exploration of accessible SDN solutions are described. Practical part of thesis consists of design SDN implementation to teaching. It involves component selection of SDN solution, selection and equipment department laboratory and design program for subject Integration of Networks.

Key words: Software-defined networks (SDN).

Obsah

Zoznam obrázkov **9**
Zoznam tabuliek **10**
Zoznam skratiek **11**
Úvod **13**
1 Analýza súčasného stavu vyučovania SDN vo svete **14**
1.1 Výučba čiastočne zameraná na SDN 14
1.2 Výučba primárne zameraná na SDN 15
1.2.1 Výučba SDN v rámci prednášok 16
1.2.2 Výučba SDN v rámci cvičení 17
2 Ciele práce **19**
3 Oboznámenie sa s technológiou SDN **20**
3.1 Potreba SDN 20
3.2 Koncept technológie SDN 21
3.3 Architektúra SDN 23
3.4 OpenFlow 26
4 Prieskum dostupných SDN riešení **28**
4.1 Softvérové SDN prepínače 28
4.1.1 Open vSwitch 28
4.1.2 Indigo Virtual Switch 29
4.1.3 Cisco Virtual Topology Forwarder 29
4.2 SDN kontroléry 29
4.2.1 POX 29
4.2.2 Floodlight 30
4.2.3 OpenDayLight 30
4.2.4 OpenMUL 30
4.2.5 Open Network Operating System 31
4.3 Hardvérové SDN produkty 32
4.3.1 Pica8 32
4.3.2 Brocade 32
4.3.3 HP 32
4.3.4 Cisco 32
4.3.5 Juniper 33
4.3.6 Mikrotik 33
4.3.7 Jednodoskové vstavané počítače 33
4.4 Komplexné SDN riešenia 33

4.4.1 Cisco ONE 33
4.4.2 Juniper Contrail 34
4.5 SDN emulátory 35
4.5.1 Mininet 35
4.5.2 EstiNet 36
5 Návrh implementácie SDN do vyučovania 37
5.1 Výber komponentov SDN riešenia 37
5.1.1 Výber SDN prepínača 37
5.1.2 Výber kontroléra 37
5.1.3 Výber hardvérového vybavenia 38
5.1.4 Výber operačného systému 39
5.1.5 Výber hypervizora 39
5.2 Výber a zariadenie laboratória na KIS 39
5.2.1 Hardvérové vybavenie laboratória 39
5.2.2 Softvérové vybavenie laboratória 40
5.3 Návrh osnovy pre predmet Integrácia sietí 40
5.3.1 Verzia pre výučbu SDN na celý semester 41
5.3.2 Verzia pre výučbu SDN na polovicu semestra 52
5.4 Ďalšie možnosti skúmania SDN na KIS 57
6 Implementácia protokolu OpenFlow 1.3 do Mikrotiku 59
6.1 Problémové riešenie 59
6.2 Výsledné riešenie 61
Záver 62
Zoznam použitej literatúry 63
Zoznam príloh 66
Prílohy 67
Príloha A: Vypracovanie praktických úloh pre cvičenia 68
Príloha B: Obsah DVD 86

Zoznam obrázkov

Obr. 1 Koncept technológie SDN 22
Obr. 2 Architektúra SDN 26
Obr. 3 OpenFlow architektúra 27
Obr. 4 Cisco ONE architektúra 34
Obr. 5 Juniper Contrail architektúra 35
Obr. 6 Topológia 1 46
Obr. 7 Topológia 2 47
Obr. 8 Topológia 3 48
Obr. 9 Topológia 4 50

Zoznam tabuliek

Tabuľka 1 Porovnanie parametrov vybraných kontrolérov 38

Zoznam skratiek

ACI	Application Centric Infrastructure
ACL	Access Control List
API	Application Programming Interface
ASI	Aplikované sieťové inžinierstvo
ASIC	Application Specific Integrated Circuit
BGP-LS	Border Gateway Protocol Link State
BSD	Berkley Software Distribution
CAPEX	Capital Expenditure
Cisco IOS	Cisco Internetwork Operating System
CLI	Command Line Interface
CPU	Central Processing Unit
ERSPAN	Encapsulated Remote Switched Port Analyzer
GRE	Generic Routing Encapsulation
GUI	Graphic User Interface
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPC	Interprocess Communication
KIS	Katedra informačných sietí
LISP	Locator/Identifier Separation Protocol
LLDP	Link Layer Discovery Protocol
MAC	Medium Access Control
MPLS	Multiprotocol Label Switching
NBAPI	Northbound Application Programming Interface
NETCONF	Network Configuration Protocol
NFV	Network Function Virtualization
ONE	Open Network Enviroment
ONF	Open Networking Foundation
ONOS	Open Networking Operating System
OPEX	Operational Expenditure

OVSDB	Open vSwitch Database
QOS	Quality of Service
REST	Representational State Transfer
RSPAN	Remote Switched Port Analyzer
SDN	Software-defined networking
SNMP	Simple Network Management Protocol
SPAN	Switched Port Analyzer
SSL	Secure Sockets Layer
STT	Stateless Transport Tunneling
TE	Traffic Engineering
VLAN	Virtual Local Area Network
VM	Virtual Machine
VNT	Virtual Network Topology
VTF	Virtual Topology Forwarder
VTS	Virtual Topology System
VXLAN	Virtual eXtensible Local Area Network
WAN	Wide Area Network
XMPP	Extensible Messaging and Presence Protocol

Úvod

Softvérovo-definované siete (SDN) sú relatívne nový prístup, ktorý mení spôsob ako dizajnovať, budovať a prevádzkovať siete, čo prináša významné technologické a finančné benefity. S technológiou SDN už siete nie sú uzatvorené, proprietárne a náročné na programovanie. Prostredníctvom SDN sú siete transforomované do otvorených a programovateľných komponentov. SDN umožňuje sieťovým operátorom väčšiu kontrolu nad ich infraštruktúrou, optimalizáciu siete, prispôsobenie siete potrebám zákazníkov a zníženie kapitálových a operačných nákladov.

Katedra informačných sietí identifikuje tému SDN ako jednu z progresívnych technológií IP (Internet Protocol) sietí, ktorá sa začína objavovať v zozname požadovaných kompetencií pre absolventov oboru Aplikovanie sieťové inžinierstvo. Katedra preto do nového programu ASI navrhla aktualizáciu predmetu Integrácia sietí, v ktorom by sa mali absolventi danej téme venovať. Cieľom práce je teoreticky a prakticky sa oboznámiť s technológiou a v úzkej spolupráci s vedúcim vypracovať materiál implementácie SDN do vyučovania.

1 Analýza súčasného stavu vyučovania SDN vo svete

Technológia SDN je horúcou témou v sieťovej oblasti, čomu nasvedčuje masívny záujem poskytovateľov služieb vo svete ponúkať a vyvíjať vlastné SDN riešenia. Otázka je, ako na tento trend reagujú školy vyučujúce technické obory (napr. počítačové siete podľa tradičnej architektúry IP sietí). Poznanie stavu a spôsobu je inšpiratívne a mohlo by ovplyvniť spôsob zavádzania predmetov s danou problematikou. Preto prvou úlohou je snaha zistiť, kde a do akej miery sa vyučuje SDN vo svete, resp. ako vypadá koncept jednotlivých predmetov zameraných na SDN. Táto analýza bude slúžiť ako podklad pre vyhotovenie materiálov implementácie SDN do vyučovania na Katedre informačných sietí (KIS) v rámci študijného programu Aplikované sieťové inžinierstvo.

Podľa zistených informácií viaceré univerzity vo svete zastrešujú obory, ktoré ponúkajú predmety a výučbu SDN. Niektoré z univerzít sa priamo zapájajú do vývoja SDN. Jednou z takých univerzít je Stanford university (USA), ktorá stojí zároveň aj za zrodením SDN a protokolu OpenFlow. Študenti a profesori zo Stanford University vedú v súčasnosti výskumy SDN vo výskumnom centre ONRC (Open Networking Research Center - http://onrc.stanford.edu/projects.html). Medzi ďalšie univerzity, ktoré sa venujú vývoju SDN, patrí Princeton University a Indiana University. Na druhej strane viaceré univerzity považujú vyučovanie SDN za komplikované, pretože je náročné naplánovať výučbu novej technológie, ktorá sa neustále vyvíja. Pokiaľ sa na škole rozbehne výučba konkrétneho SDN riešenia, je dosť možné, že v tom čase bude už dané riešenie zastarané.

Zo získaných informácii som rozdelil vyučované predmety na:

- čiastočne zamerané na softvérovo-definované siete
- primárne zamerané na softvérovo-definované siete.

V nasledujúcich podkapitolách sa pokúsim popísať metodiku výučby jednotlivých škôl vo svete s čiastočným, ako aj s primárnym zameraním na SDN technológiu.

1.1 Výučba čiastočne zameraná na SDN

Drvivá väčšina univerzít integruje SDN do existujúcich predmetov, ktoré sú zamerané na architektúru IP sietí v podobne, ako sú na našej katedre vyučované predmety Počítačové siete 1-3. Téme SDN sú venované zhruba 2-3 prednášky. Tieto prednášky

pozostávajú z vysvetlenia základných princípov SDN, ako sú SDN architektúra, dôležitosť oddelenia riadiacej časti od dátovej časti, OpenFlow protokol a pod. V rámci cvičení sa v osnovách predmetov uvádza práca so sieťovým emulátorom Mininet, ktorý slúži ako vhodná pomôcka pri výučbe SDN. Študenti majú za úlohu vypracovať rôzne zadania v Mininete. Takto orientované predmety sa vyučujú napríklad na Northumbria University (UK) alebo na Stanford University (USA).

V súčasnosti vznikajú aj nové predmety zamerané na kombináciu technológií Cloud Computing , SDN a NFV (Network Function Virtualization), ktoré so sebou úzko súvisia. Tieto typy predmetov sa snažia pokryť zložitejšie koncepty virtualizácie v dátových centrách. Laboratórne cvičenia sú zamerané na implementáciu cloudových platforiem (Openstack, Amazon EC2) s použitím rôznych hypervízorov (KVM, XEN, VMware) a OpenFlow kontrolérov (Floodlight, OpenDayLight). Takto konštruované predmety sa vyučujú na Northwestern University (UK) a University of Colorado (USA).

Na Santa Clara University (USA) je vyučovaný predmet Network Management zameraný na konfiguračné nástroje a protokoly využívané v SDN prostrediach (NETCONF, NetFlow, YANG, SNMP). Na Clemson university v Južnej Karolíne (USA) majú zase predmet zameraný na sieťovú bezpečnosť, v ktorom riešia bezpečnosť SDN a NFV technológií.

Na AGH University of Science and Technology v Krakowe sa vyučuje SDN v rámci budúcich trendov v IP sieťach. Na stránkach predmetu uvádzajú, že študent, ktorý absolvuje predmet, musí byť schopný nakonfigurovať virtuálne prepínače, použiť kontroléry na riadenie virtualizovanej siete a musí vedieť použiť SDN technológiu na reguláciu počítačovej siete podľa potrieb aplikácie. Použitie konkrétnych riešení neuvádzajú.

1.2 Výučba primárne zameraná na SDN

V tejto časti som sa zameral na získanie informácií o stave vyučovania ponúkajúceho predmety primárne zamerané na SDN. Takýmto predmetom je venovaných viac ako 6 prednášok o problematike SDN.

Predmety primárne zamerané na SDN sa vyučujú na Princeton university (USA), Columbia University (USA), Stony Brook University (USA), Nation Chi Nan University (Taiwan), University of Crete (Grécko), University of Southern Carolina (USA), Carnegie

Mellon University (USA), National Cheng Kung University (Taiwan), National Chiao Tung University (Taiwan), DUKE University (USA), University of Colorado (USA), University of Waterloo (Kanada), The University of Texas (USA), Charles Sturt University (Austrália), KTH Royal Institute of Technology (Švedsko).

V nasledujúcej časti uvádzam, ako a do akej hĺbky tieto univerzity vyučujú SDN v rámci prednášok a v rámci cvičení.

1.2.1 Výučba SDN v rámci prednášok

Väčšina univerzít (Princeton university, Stony Brook University, Nation Chi Nan University, University of Crete, University of Southern Carolina, Carnegie Mellon University, National Chiao Tung University, University of Waterloo) začína svoje prednášky úvodom do SDN technológie. Tento úvod zahŕňa historickú evolúciu sietí, nedostatky súčasných sietí, SDN koncepty, SDN princípy a históriu SDN.

Univerzity University of Waterloo, Carnegie Mellon University a University of Southern Carolina rozoberajú riadiacu a dátovú rovinu zariadení. Univerzity Princeton university, Columbia University a Stony Brook University zas rozoberajú abstrakciu SDN.

Univerzity Nation Chi Nan University, University of Crete, National Chiao Tung University, Columbia University venujú časť prednášok protokolu OpenFlow a jeho konceptu. Dbajú pritom na rozdiely vo verziách OpenFlow 1.0 až 1.5.

Univerzity University of Crete, Nation Chi Nan University, Columbia University vo svojich prednáškach uvádzajú prehľad o rôznych SDN riešeniach v podobe kontrolérov a softvérových prepínačov. Prepínače, ktorým sa venujú, sú zväčša riešenia Open vSwitch a Indigo. Medzi najčastejšie spomínané kontroléry patria NOX, POX, RYU, OpenDayLight, Floodlight, ONOS.

Univerzity Princeton university, Columbia University, Stony Brook University, DUKE University, National Chiao Tung University, University of Southern Carolina, University of Colorado uvádzajú príklady využitia SDN v rôznych prostrediach, ako sú dátové centrá, cloudové prostredia, WAN (Wide Area Network) prostredia, bezdrôtové prostredia a mobilné prostredia. V rámci týchto prednášok sa venujú technológiám riadenia toku (traffic enginnering) a monitoringu SDN.

Univerzity Columbia University, Nation Chi Nan University, National Cheng Kung University, University of Crete, National Chiao Tung University, Carnegie Mellon University, University of Waterloo spomínajú vo svojich prednáškach aj technológiu NFV (Network Function Virtualization) v úzkom spojení s SDN.

Univerzity Princeton university, Columbia University, Stony Brook University, National Cheng Kung University, University of Crete, National Chiao Tung University, University of Southern, Carolina University of Colorado, Carnegie Mellon University, University of Waterloo riešia otázku bezpečnosti, odolnosti voči chybám, testovanie a odstraňovanie chýb SDN sietí.

Univerzity University of Waterloo, Carnegie Mellon University, University of Crete, University of Colorado spomínajú programovacie jazyky pre SDN. Zvyčajne ide o jazyky Frenetic a Pyretic.

Univerzity Princeton university, Columbia University predstavujú svoju víziu o budúcnosti SDN.

Univerzita National Chiao Tung University uvádza prednášku o SDN migrácii v hybridnej sieti s tradičnými prepínačmi.

Univerzita Nation Chi Nan University oboznamuje v prednáškach s organizáciou Open Networing Foundation (ONF).

1.2.2 Výučba SDN v rámci cvičení

Drvivá väčšina univerzít (Princeton university, Stony Brook University, National Cheng Kung University, National Chiao Tung University, University of Southern Carolina, University of Colorado, Carnegie Mellon University) využíva v rámci cvičení prostredie Mininet. Univerzita National Cheng Kung University na rozdiel od ostatných univerzít využíva aj alternatívne simulačné prostredie EstiNet.

Na univerzitách Carnegie Mellon University, University of Crete, Stony Brook University sa pracuje v rámci cvičení s kontrolérom POX, na univerzitách DUKE University, University of Colorado sa pracuje s kontrolérom Floodlight, na univerzite University of Southern Carolina sa pracuje s kontrolérom ONOS a na univerzite National Cheng Kung University sa pracuje s kombináciou kontrolérov NOX a RYU.

Univerzity Stony Brook University, University of Crete, University of Southern Carolina, Carnegie Mellon University vyučujú programovanie v jazyku Pytetic, univerzita Columbia University vyučuje programovanie v jazyku Frenetic.

Univerzita National Cheng Kung University využíva na cvičeniach implementáciu OpenFlow prepínača v operačnom systéme OpenWrt.

Na univerzite University of Waterloo sa zadávajú v rámci cvičenia semestrálne projekty. Zadania vypadajú nasledovne:

- *Application Aware Networking* – monitorovanie prevádzky pomocou SDN kontroléra, nastavenie kvality služieb
- *Conducting experiments with DOT*– testovanie škálovateľnosti emulačného nástroja DOT (Distributed OpenFlow Testbed)
- *Managing the Software in Software Defined Networks* – preskúmanie problémov spojených s manažovaním rozličných SDN komponentov a ich porovnanie
- *Inter-Controller Communication* – analýza rozličných mechanizmov a návrh kritérií pre scenár nasadenia viacerých kontrolérov do produkčnej siete

2 Ciele práce

Hlavným cieľom práce je vyhotoviť koncept vyučovania SDN, ktorý zahŕňa metodiku čo, ako a v akom rozsahu by sa mohlo implementovať do štúdia v rámci študijného programu Aplikované sieťové inžinierstvo v predmete Integrácia sietí.

K naplneniu hlavného cieľa je potrebné realizovať parciálne ciele. Prvým parciálnym cieľom je analýza súčasného stavu vyučovania SDN vo svete. Druhým parciálnym cieľom je teoreticky a prakticky sa oboznámiť s technológiou SDN, čo pozostáva z analýzy jej princípov, architektúry a protokolov. Posledným parciálnym cieľom je vytvorenie odporúčania pre vyučovanie SDN.

V rámci tohto odporúčania je nutné vybrať vhodné témy pre prednášky a určiť ich rozsah. Taktiež je nutné navrhnúť a vybrať SDN komponenty pre cvičenia. Toto zahŕňa výber vhodnej distribúcie operačného systému Linux, SDN kontroléra, SDN prepínača a výber vhodného hardvéru. Posledným krokom je vypracovanie vybraných tém pre prednášky a praktických demonštrácií pre cvičenia.

3 Oboznámenie sa s technológiou SDN

Predtým ako začnem rozoberať konkrétne SDN implementácie, uvediem čitateľa do problematiky SDN.

3.1 Potreba SDN

Potreba SDN je v súčasnosti odôvodnená viacerými faktormi. Explózia mobilných zariadení [1], nástup serverovej virtualizácie a príchod cloudových služieb patria k trendom, ktoré predstavujú výzvu pre tradičné sieťové architektúry. Mnoho dnešných sietí je hierarchických, stavaných na ethernetových prepínačoch zapojených do stromovej štruktúry. Tento dizajn mal zmysel, keď bola dominantná klient-server sieťová architektúra. Avšak takáto statická architektúra je nežiaduca pre dynamické výpočtové a úložiskové potreby v dnešných dátových centrách, v campus alebo carrier prostrediach.

Súčasné sieťové technológie [2] pozostávajú z veľkého množstva protokolov navrhnutých na prepojenie koncových staníc spoľahlivo cez ľubovoľné vzdialenosti, linky rôznych rýchlostí a rozličné topológie. Aby sa vyhovelo technickým a biznis podmienkam za posledné desaťročia, odvetvie vyvinulo sieťové protokoly pre zvýšenie výkonu, spoľahlivosti, konektivity, bezpečnosti, pričom sa upustilo od nejakej základnej abstrakcie. To vyústilo v základný problém dnešných sietí, ktorý je **komplexnosť**. Napríklad v situácii, keď chceme pridať alebo premiestniť zariadenie v sieti, musíme zasiahnuť aj do samotných sieťových prvkov ako sú prepínače, smerovače, firewally a upraviť ich konfiguráciu (ako napríklad nastavenie funkcií Access Control List, Virtual Local Area Network, Quality of Service). Kvôli tejto komplexnosti a následne spojenej problematickej správe sú dnešné siete pomerne statické.

Statická povaha tradičných sietí [1] je v kontraste s dynamickou povahou v serverových prostrediach, ako sú dátové centrá. S nástupom serverovej virtualizácie sa výrazne zvýšil počet koncových staníc v dátových centrách vyžadujúcich sieťovú konektivitu. Jedno veľké dátové centrum [2] môže pozostávať až zo 120 000 fyzických serverov (napríklad dátové centrum Microsoftu). Zoberme do úvahy fakt, že na každom fyzickom serveri v takomto dátovom centre je prevádzkovaných v priemere 20 virtuálnych serverov. To znamená, že interná sieť vo veľkom dátovom centre môže prepájať až 2 400 000 koncových serverov. Prevádzka v takejto sieti je nepredvídateľná a dynamicky

sa meniaca. Dátové centrá veľkých spoločností ako sú Facebook, Google, Amazon sa musia vyrovnávať s potrebou škálovateľnosti ich činnosti a prevádzky, čo prináša obrovské nároky na konfiguráciu a manažment.

Moderné L2 a L3 prepínače [2], ktoré typicky tvoria aktívnu sieťovú infraštruktúru týchto dátových centier, využívajú pri svojej činnosti pomerne veľké množstvo rôznych sieťových protokolov (Label Discovery Protocol, Multiprotocol Label Switching, Internet Group Management Protocol, Multicast Source Discovery Protocol, Protcol Independent Multicast a pod.), ktoré zaťažujú ich riadiacu rovinu. Zmena prevádzkovanej topológie v takejto sieti sa zvyčajne prejaví na rastúcom čase konvergencie, kedy riadiaca rovina zariadení musí spraviť nové výpočty, poprípade ich rozdistribuovať, aby pracovné tabuľky s L2 alebo L3 informáciami (ovplyvňujúcimi činnosť na dátovej rovine) na každom zariadení ostali v stave odrážajúcom vykonanú zmenu. Tento fakt sa stáva nežiaducim v prípade topologickej zmeny v dátových centrách, kedy riadiaca rovina zariadení môže spôsobiť neakceptovateľne dlhý čas konvergencie. Dôležité je si uvedomiť, že fyzická sieťová infraštruktúra v dátovom centre je statická, dopredu známa, prevažne stabilná a centrálne kontrolovaná. Preto sa zdá byť výhodné navrhnúť nový jednoduchší prístup, v ktorom by sa oddelila riadiaca rovina zariadení od tej dátovej. Dosiahli by sme tak programovateľnosť dátových časti zariadení z iného miesta v sieti.

3.2 Koncept technológie SDN

Ako dôsledok spomenutých nedostatkov dnešných sietí vzniká nový prístup s názvom SDN (Software-Defined Networking). SDN [1] je v podstate sieťová architektúra, v ktorej je riadiaca rovina zariadení oddelená od dátovej roviny. Riadenie siete je tak presunuté zo zariadenia samotného do iného centralizovaného výpočtového zariadenia. Týmto je riadenie siete oddelené od samotného preposielania dát, čím sa dátová rovina zariadení stáva priamo programovateľná.

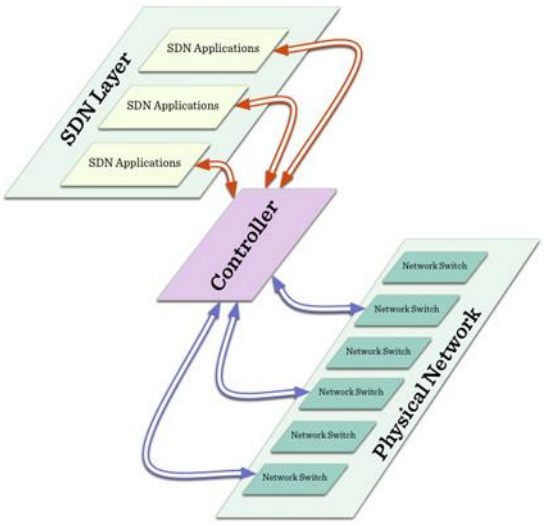
Táto migrácia riadenia [2] do iného výpočtového zariadenia umožňuje abstrakciu fyzickej sieťovej infraštruktúry pre aplikácie a sieťové služby, ktoré môžu zaobchádzať so sieťou ako s logickou alebo virtuálnou entitou (Obr. 1). Vďaka tejto abstrakcii sa môžu aplikácie a prostriedky na aplikovanie politík pozerať na sieť ako na jeden veľký logický prepínač.

Celá sieťová inteligencia [1] je centralizovane umiestnená do výpočtových softvérovo-založených zariadení, tzv. kontrolérov, ktoré sa starajú o globálny pohľad siete. S použitím kontrolérov nadobúdame kontrolu nad celou sieťou z jedného logického bodu, čím sa rapídne zjednoduší dizajn siete a sieťové operácie.

SDN taktiež zjednodušuje samotné sieťové zariadenia, pretože už nepotrebujú podporovať tisícky protokolových štandardov, ale stačí im prijímať inštrukcie od SDN kontroléra.

Sieťoví administrátori môžu vďaka SDN automatizovane konfigurovať celú sieť z jedného miesta v porovnaní s tým, ako by mali manuálne konfigurovať viacero nezávislých zariadení. Taktiež môžu modifikovať správanie siete v reálnom čase a nasadzovať nové aplikácie a sieťové služby v priebehu pár hodín v porovnaní s týždňami alebo mesiacmi, ako je to dnes bežné [1].

SDN [2] bolo vytvorené s myšlienkou podporiť otvorenosť sieťových prostredí. Špecifikácie SDN a SDN softvér boli distribuované voľne medzi univerzitnými skupinami bez komerčných dotácií. Hlavní podporovatelia SDN boli jednotlivci a inštitúcie, ktorých cieľom nebolo vyťažiť zisk z predaja technológie. Aj keď aktuálne vzrastá počet proprietárnych SDN technológií na trhu, štandardizačné SDN organizácie veria, že otvorenosť ostane charakteristikou SDN naďalej. Jednou z organizácií, ktorá podporuje otvorenosť SDN, je organizácia Open Networking Foundation (ONF), ktorá štandardizuje otvorené API rozhrania pre podporu zariadení od viacerých výrobcov.



Obr. 1 Koncept technológie SDN [3]

3.3 Architektúra SDN

Vývoj SDN prináša nové aspekty (napr. v oblasti programovateľnosti), čím dochádza k nejasnostiam v definíciách týkajúcich sa technológie SDN a jej architektúry. Niektoré definície architektúry SDN si dokonca navzájom odporujú [4]. Tieto nedostatky v definíciách sú skonsolidované v dokumente RFC 7426 [4], ktorý popisuje SDN architektúru komplexne vzhľadom na aktuálny stav. Tento dokument používa nasledovné pojmy v spojitosti s architektúrou SDN:

- zdroj – fyzický alebo virtuálny komponent v rámci systému, môže byť jednoduchý (port alebo front) alebo komplexný (sieťové zariadenie)
- sieťové zariadenie (fyzické alebo virtuálne) – vykonáva jednu alebo viacero sieťových operácií spojených s paketovým spracovaním a preposielaním
- rozhranie – bod interakcie medzi dvoma entitami
- aplikácia – softvér, ktorý používa služby na vykonanie funkcií
- služba – softvérové programy, ktoré poskytujú API rozhrania ostatným aplikáciám alebo službám
- roviny (preposielacia rovina, operačná rovina, riadiaca rovina, manažmentová rovina, aplikačná rovina)
- abstrakčné vrstvy (abstrakčná vrstva zariadení a zdrojov, riadiaca abstrakčná vrstva, manažmentová abstrakčná vrstva, abstrakčná vrstva sieťových služieb)

SDN architektúra podľa dokumentu RFC 7426 pozostáva z nasledujúcich rovín (Obr. 2):

- preposielacia rovina (forwarding plane)
  - zodpovedná za spracovanie paketov na dátovej ceste
  - založená na inštrukciách prijatých od riadiacej časti
  - akcie na preposielacej rovine pozostávajú (nie len) z preposielania, zahodenia a zmeny paketov
  - bod ukončenia pre služby a aplikácie riadiacej roviny
  - môže obsahovať preposielacie zdroje v podobe klasifikátorov
  - nazývaná aj dátová rovina alebo dátová cesta
- operačná rovina (operational plane)

  - zodpovedná za manažovanie operačných stavov sieťových zariadení, napríklad počet dostupných portov, stav každého portu
  - bod ukončenia pre služby a aplikácie manažmentovej roviny
  - týka sa zdrojov sieťových zariadení ako sú pamäť, porty a podobne
- riadiaca rovina (control plane)
  - zodpovedná za vykonávanie rozhodnutí, ako by pakety mali byť preposielané jedným alebo viacerými sieťovými zariadeniami
  - zodpovedná za aplikovanie rozhodnutí, ktoré majú byť vykonané sieťovými zariadeniami
  - zameriava sa viac na preposielaciu rovinu ako na operačnú rovinu zariadení
  - môže sa zaujímať o informácie operačnej roviny, ako je napríklad súčasný stav daného portu alebo jeho kapacít
  - hlavnou úlohou riadiacej roviny je doladenie preposielacích tabuliek uložených v dátovej rovine, ktoré sú založené na sieťovej topológií alebo požiadavkách externých služieb
- manažmentová rovina (management plane)
  - zodpovedná za monitoring, konfiguráciu a údržbu sieťových zariadení, napríklad vykonávanie rozhodnutí na základe stavu sieťového zariadenia
  - zameriava sa viac na operačnú rovinu ako na preposielaciu rovinu zariadení
  - môže byť použitá na konfiguráciu preposielacej roviny
  - môže nastaviť všetky alebo len časť preposielacích pravidiel naraz, ale takáto funkcia sa používa len výnimočne
- aplikačná rovina (application plane)
  - rovina, kde sídlia aplikácie a služby, ktoré definujú správanie siete
  - aplikácie, ktoré priamo podporujú operačnú alebo preposielaciu rovinu (ako napríklad smerovacie procesy v rámci riadiacej roviny), nie sú považované za súčasť aplikačnej roviny
  - aplikácie môžu byť implementované v modulárnej alebo distribuovanej forme a z toho dôvodu sa môžu pohybovať po viacerých rovinách SDN architektúry

Všetky vrstvy SDN architektúry sú prepojené rozhraniami. Rozhranie môže vystupovať vo viacerých roliach podľa pripojených rovín sídliacich na rovnakom fyzickom alebo virtuálnom zariadení. Ak príslušné roviny sú navrhnuté tak, aby sa nemuseli

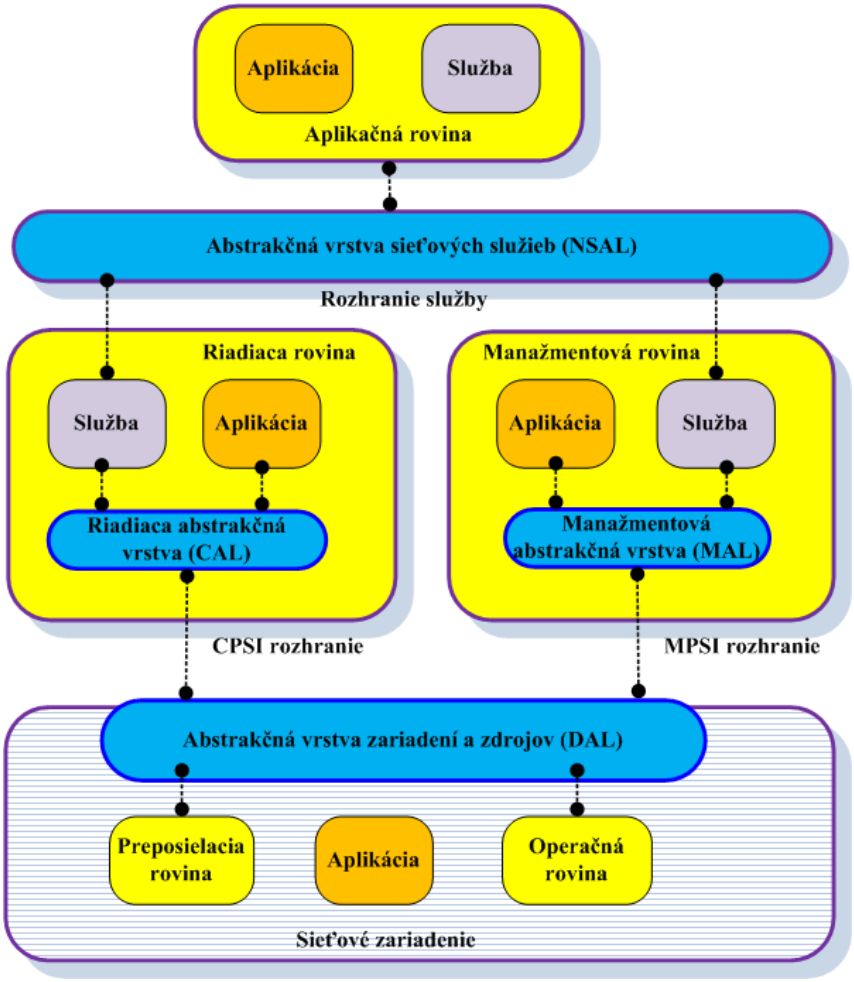
nachádzať na rovnakom zariadení, potom rozhrania môžu vystupovať vo forme protokolu. Ak sú roviny umiestnené na rovnakom zariadení, potom by rozhranie malo byť implementované cez otvorený/proprietárny protokol, otvorené/proprietárne softvérové medziprocesové komunikačné API rozhranie alebo cez systémové volania jadra operačného systému.

Aplikácie resp. softvérové programy vykonávajúce špecifické výpočty, ktoré využívajú služby bez poskytovania prístupu k iným aplikáciám, môžu byť implementované natívne vo vnútri roviny alebo môžu premosťovať viacero rovín.

Služby resp. softvérové programy, ktoré poskytujú API rozhrania ostatným aplikáciám alebo službám, môžu byť taktiež natívne implementované v špecifických rovinách.

Dokument RFC 7426 predpokladá 4 abstrakčné vrstvy (Obr. 2):

- abstrakčná vrstva zariadení a zdrojov (DAL) – abstrahuje zdroje preposielacej a operačnej roviny zariadení do riadiacej a manažmentovej roviny
- riadiaca abstrakčná vrstva (CAL) – abstrahuje rozhranie CPSI (Control-Plane Southbound Interface) a vrstvu DAL od aplikácií a služieb riadiacej roviny
- manažmentová abstrakčná vrstva (MAL) – abstrahuje rozhranie MPSI (Management-Plane Southbound Interface) a vrstvu DAL od aplikácií a služieb manažmentovej roviny
- abstrakčná vrstva sieťových služieb (NSAL) – poskytuje abstrakciu služieb pre použitie aplikácií a iných služieb



Obr. 2 Architektúra SDN

3.4 OpenFlow

OpenFlow [1] je prvé štandardizované komunikačné rozhranie definované medzi riadiacou a preposielacou rovinou SDN architektúry. Bol vytvorený v roku 2008 na Stanford university ako súčasť programu Clean Slate.

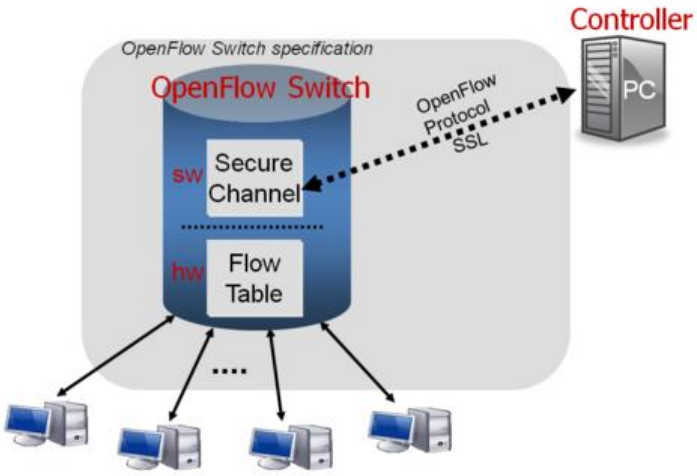
OpenFlow umožňuje manipulovať preposielaciu rovinu sieťových zariadení, ako sú smerovače a prepínače bez ohľadu, či sú fyzické alebo virtuálne. OpenFlow môže byť prirovnaný k inštrukčnej množine procesora, pretože špecifikuje základné správy, ktoré môžu byť použité externou softvérovou aplikáciou k naprogramovaniu preposielacej roviny sieťových zariadení presne v takom štýle, ako inštrukčná sada procesora môže naprogramovať počítačový systém.

OpenFlow protokol je implementovaný na oboch stranách rozhrania medzi sieťovými zariadeniami tvoriacimi infraštruktúru a SDN riadiacim softvérom. OpenFlow

používa koncept tokov na identifikovanie sieťovej prevádzky založenej na preddefinovaných pravidlách zhody, ktoré môžu byť staticky alebo dynamicky naprogramované SDN riadiacim softvérom. SDN architektúra založená na protokole OpenFlow, poskytuje kontrolu a možnosť siete reagovať na zmeny v reálnom čase. OpenFlow je štandardizovaný organizáciou ONF, ktorá zabezpečuje kompatibilitu medzi sieťovými zariadeniami a riadiacim softvérom rozličných výrobcov.

OpenFlow architektúra [5] pozostáva z 3 základných konceptov (Obr. 3):

1. Sieť je postavená z OpenFlow prepínačov, ktoré tvoria dátovú rovinu.
2. Riadiaca rovina pozostáva z jedného alebo viacerých kontrolérov.
3. Zabezpečený riadiaci kanál prepája prepínače s riadiacou rovinou (kontrolérom).



Obr. 3 OpenFlow architektúra [6]

4 Prieskum dostupných SDN riešení

V tejto časti sa budem snažiť popísať dostupné SDN riešenia. Toto zahŕňa popis softvérových SDN prepínačov, SDN kontrolérov, hardvérových SDN prepínačov, komplexných SDN riešení a emulačných SDN prostredí.

4.1 Softvérové SDN prepínače

4.1.1 Open vSwitch

Open vSwitch je softvérová implementácia virtuálneho viacvrstvového sieťového prepínača. Open vSwitch [7] bol vytvorený tímom spoločnosti Nicira, ktorú neskôr odkúpila spoločnosť VMware. Open v Switch bol primárne plánovaný pre potreby open source komunity v dobe, keď neexistoval žiadny funkciovo-bohatý virtuálny prepínač navrhnutý pre hypervizory bežiace v Linuxe, ako sú KVM a XEN. Open vSwitch sa stal rýchlo východiskovým riešením pre prostredia využívajúce hypervizora XEN. Dnes Open vSwitch zohráva veľmi dôležitú úlohu v iných open source projektoch, ako je napríklad cloudové riešenie OpenStack.

Open vSwitch [7] je rozhodujúci prvok pre mnoho SDN nasadení v dátových centrách, pretože spája dokopy všetky virtuálne servery v rámci hypervizora bežiaceho na serveri. Je to prvý vstupný bod pre všetky virtuálne servery posielajúce prevádzku do siete a je to aj vstupný bod do overlay sietí bežiacich nad fyzickou sieťou v dátovom centre. Ďalší dôvod pre použitie riešenia Open v Switch v dátových centrách je sieťová virtualizácia, kde Open vSwitch zohráva kľúčovú úlohu. Open vSwitch môže byť použitý taktiež pre smerovanie prevádzky cez sieťové funkcie (network functions).

Open vSwitch je zvyčajne riadený a manažovaný treťou stranou prostredníctvom kontrolérov. Avšak to neznamená, že SDN kontrolér je potrebný pre využívanie Open vSwitcha. Open vSwitch je možné nasadiť na server za cieľom vykonávania tradičnej L2 prepínacej funkcionality.

Open vSwitch podporuje:

- zber dát cez protokoly NetFlow, sFlow, IPFIX
- zrkadlenie prevádzky cez protokoly SPAN, RSPAN, ERSPAN
- protokol OpenFlow 1.x
- tunelovacie mechanizmy: GRE, Geneve, VXLAN, STT, LISP

4.1.2 Indigo Virtual Switch

Indigo Virtual Switch (IVS) [8] je open source virtuálny prepínač určený pre systémy LINUX s KVM hypervizorom. IVS využíva Open vSwitch modul jadra pre preposielanie paketov. IVS je súčasťou projektu Indigo Framework a využíva LoxiGen generovaný kód (loci) na spracovanie OpenFlow správ.

Projekt je distribuovaný pod EPL open source licenciou a je udržiavaný komunitou vývojárov a inžinierov zo spoločnosti Big Switch Networks.

IVS je odľahčený vysokovýkonný prepínač pre podporu protokolu OpenFlow. Je navrhnutý pre podporu aplikácií sieťovej virtualizácie a podporuje distribúciu cez fyzické servery použitím OpenFlow kontroléra.

4.1.3 Cisco Virtual Topology Forwarder

Cisco Virtual Topology Forwarder (VTF) [9] je odľahčený softvérový prepínač navrhnutý na vysokovýkonné paketové spracovanie na x86 serveroch. VTF je súčasťou otvoreného škálovateľného SDN riešenia Cisco VTS (Virtual Topology System) určeného pre virtuálny sieťový manažment v dátových centrách. VTF využíva inovatívnu technológiu od Cisca s názvom Vector Packet Processing (VPP) a Intel Data Path Development Kit (DPDK) pre L2, L3 a VXLAN paketové preposielanie umožňujúce priepustnosť až 10 Gbps na jednom procesorovom jadre. VTF je viacvláknový, čo umožňuje zákazníkom alokovať ďalšie procesorové jadrá na škálovanie výkonu.

4.2 SDN kontroléry

4.2.1 POX

POX [10] je open source platforma SDN kontroléra vyvíjaná Stanfordskou univerzitou založená na jazyku Python. POX je nástupca sesterského SDN kontroléra NOX, pričom POX ponúka jednoduchšie prostredie a dobre napísané API rozhranie s prehľadnou dokumentáciou. POX je navrhnutý pre rýchlejší vývoj a je veľmi užitočný pre programovanie vlastného sieťového softvéru. POX poskytuje taktiež webové grafické rozhranie. Použitím POX kontroléra [11] môžeme premeniť SDN zariadenia na hub, prepínač, load balancer alebo firewall.

4.2.2 Floodlight

Floodlight [12] je OpenFlow SDN kontrolér patriaci pod Apache licenciu a spoločnosť Big Switch Networks. Bol súčasťou projektu OpenDayLight, ale v júni 2013 spoločnosť Big Switch od tohto projektu odstúpila. Floodlight je založený na jazyku Java a je vyvíjaný otvorenou komunitou vývojárov. Floodlight zahŕňa v sebe:

- modulový načitávací systém, ktorý umožňuje rýchlejšie rozšírenie a vylepšenie
- ľahkú inštaláciu s minimálnou potrebou inštalovania závislostí
- podporuje široké spektrum virtuálnych a fyzických OpenFlow prepínačov
- ponúka vysoký výkon (je viacvláknový)
- podporuje cloudovú orchestračnú platformu OpenStack
- ponúka webové rozhranie
- podporuje komunikáciu cez REST (Representational state transfer) API rozhranie

4.2.3 OpenDayLight

OpenDayLight [13] je vysoko dostupný modulárny rozšíriteľný škálovateľný multiprotokolový kontrolér pod organizáciou Linux Foundation. Je určený pre nasadenie SDN do moderných heterogénnych sietí postavených na zariadeniach rôznych výrobcov. OpenDayLight poskytuje modelovo-orientovanú abstrakčnú platformu služieb, ktorá umožňuje používateľom programovať aplikácie, ktoré pracujú s rôznymi hardvérovými a southbound protokolmi. OpenDaylight je napísaný v Jave a pozostáva z rôznych modulov, ktoré môžu byť kombinované podľa potreby.

V súčasnosti existujú už 4 softvérové vydania projektu OpenDayLight. Prvé vydanie je nazvané Hydrogen, druhé vydanie Hellium, tretie vydanie Lithium a štvrté posledné aktuálne sa nazýva Beryllium.

OpenDayLight využíva v sebe nástroj Maven, OSGi (Open Services Gateway initiative) rozhrania a REST API rozhrania. OpenDayLight podporuje široké spektrum protokolov, ako sú OpenFlow, SNMP, LISP, OVSDB, BGP-LS, VNT a ďalšie.

4.2.4 OpenMUL

OpenMUL [14] je OpenFlow/SDN platforma kontroléra, ktorá je napísaná v jazyku C. Jadro kontroléra OpenMUL má viacvláknovú štruktúru. OpenMUL podporuje viacúrovňové northbound rozhranie pre hosťujúce aplikácie a zameriava sa na southbound

protokoly ako sú OpenFlow, OVSDB, NETCONF, OF-CONFIG. OpenMUL je navrhnutý pre výkon a spoľahlivosť. Je taktiež vysoko flexibilný, modulárny a ľahko naučiteľný. OpenMUL pozostáva z týchto hlavných častí:

- MUL jadro
- MUL služby infraštruktúry
- MUL systémové aplikácie

MUL služby infraštruktúry tvorí:

- Topology Discovery Service: Využíva LLDP pakety na objavenie sieťovej prepínacej topológie. Taktiež zabezpečuje detekciu a predchádzanie sieťových slučiek na žiadosť v interakcii s MUL jadrom.
- Path Finding Service: Využíva FFloyd-Warshallov algoritmus na výpočet najkratšej cesty medzi dvoma sieťovými uzlami.
- Path Connector Service: Poskytuje flexibilné rozhranie pre aplikácie na inštaláciu tokov pozdĺž cesty.

MUL systémové aplikácie tvorí:

- L2switch: Poskytuje L2 učiacu sa prepínanú logiku.
- CLI app: Poskytuje CLI nástroj pre všetky MUL komponenty.
- NBAPI webserver: Poskytuje RESTful API rozhranie pre MUL kontrolér.

4.2.5 Open Network Operating System

Open Network Operating System (ONOS) [15] je SDN operačný systém pre poskytovateľov služieb implementovaný v Jave, ktorý ponúka škálovateľnosť, vysokú dostupnosť, vysoký výkon a abstrakcie pre jednoduchšie vytváranie aplikácií a služieb. ONOS je navrhnutý ako operačný systém založený na klasteroch, ktorý je škálovateľný horizontálne s veľkosťou siete a potrebami aplikácie. ONOS umožňuje jednoduchšie programovanie aplikácii s bohatými northbound abstrakciami, ktoré zabezpečujú komplexný sieťový pohľad na aplikácie. Pripojiteľné southbound rozhranie dovoľuje riadenie klasických prepínačov a prepínačov založených na protokole OpenFlow.

4.3 Hardvérové SDN produkty

Aktuálne mnoho výrobcov sieťových zariadení implementuje podporu SDN do svojich produktov. Medzi najväčších hráčov na trhu patria Ciena, Cisco, Juniper, Brocade, BigSwitch, IBM, HP, NEC, Arista Networks a Pica8.

Vybral som si niekoľko kľúčových výrobcov sieťových zariadení a popísal ich ponuku hardvérových SDN produktov. Do tejto ponuky produktov pripájam aj jednodoskové vstavané počítače, ktoré nie sú primárne orientované na SDN, ale SDN funkcionalita sa do nich dá implementovať.

4.3.1 Pica8

Pica8 je prvá spoločnosť ponúkajúca otvorené prepínače nezávislé na hardvéri. Na fyzickom prepínacom hardvéri je prevádzkovaný PicaOS, otvorený sieťový operačný systém, ktorý podporuje štandardné L2/L3 protokoly spoločne s podporou protokolu OpenFlow cez integráciu prepínača Open vSwitch. Medzi hardvérové produkty Pica8 patria prepínače P-5401 (32x40G), P-5101 (40x10G, 8x40G), P-3930 (48x10G-T,4x40G), P-3922 (48x10G,4x40G), P-3297 (48 x 1G-T, 4 x 10G).

4.3.2 Brocade

Brocade zaviedol OpenFlow podporu v júni 2010. Prvé podporované zariadenia boli MLX smerovače. V súčasnosti OpenFlow 1.3 podporujú zariadenia: CES prepínače, CER smerovače, ICX prepínače (do campusov), VDX prepínače (do dátových centier).

4.3.3 HP

HP má širokú škálu OpenFlow kompatibilných produktov. Medzi ne patria prepínače rady 2920, 3500, 3800, 5400, 6200, 6600, 8200.

4.3.4 Cisco

Cisco podporuje OpenFlow vo verziách 1.0 a 1.3. Na niektoré zariadenia s kompatibilným operačným systémom (IOS-XE, NX-OS, IOS-XR) ponúka Cisco Plug-in pre Openflow (Nexus 3000, Nexus 6000, Catalyst 4500E). V súčasnosti implementuje OpenFlow podporu do nových prístupových prepínačov Catalyst 3850 a Catalyst 3650.

4.3.5 Juniper

Juniper ohlásil v júni 2012 svoju prvú SDN stratégiu zameranú na riešenia bezpečnosti pre dátové centrá. V súčasnosti Juniper do svojich zariadení pridáva podporu OpenFlow vo verzii 1.3. Medzi tieto zariadenia patria smerovače rady MX a prepínače rady EX.

4.3.6 Mikrotik

Mikrotik do svojich zariadení implementuje OpenFlow podporu vo verzii 1.0 prostredníctvom operačného systému RouterOS. Avšak súčasná implementácia je čisto experimentálna a nie je doporučené ju používať v produkčných prostrediach. OpenFlow podpora je k dispozícii len ako samostatný balíček.

Ďalšia možnosť ako implementovať podporu protokolu OpenFlow do Mikrotik zariadenia je nainštalovať operačný systém OpenWrt. OpenWrt je GNU/Linux distribúcia pre zabudované zariadenia rôznych výrobcov vrátane Mikrotiku. Pre OpenWrt existujú aplikácie Open vSwitch a Pantou, ktoré dokážu spraviť z klasického smerovača plnohodnotné SDN zariadenie. Pantou je aplikácia vyvíjaná univerzitou Stanford university a podporuje OpenFlow vo verzii 1.3.

4.3.7 Jednodoskové vstavané počítače

Vychádzam z analýzy jednodoskových vstavaných počítačov v diplomovej práci od M. Kršku [16], v ktorej sú spomenuté zariadenia Cubleboard1, Cubleboard2, Banana PI, Raspberry PI model B+. Na týchto zariadeniach je možné spustiť známe distribúcie Linuxu, ako napríklad Ubuntu, Debian, OpenSuse, Fedora. Vďaka tomu je možné do týchto zariadení implementovať ľubovoľný linuxový SDN prepínač, a tak z nich spraviť hardvérové SDN zariadenie.

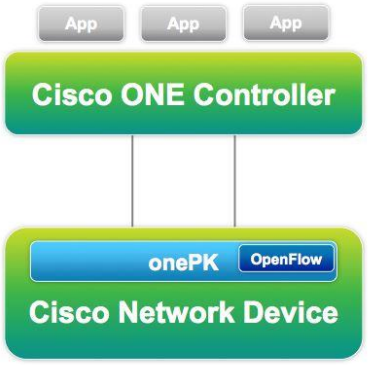
4.4 Komplexné SDN riešenia

4.4.1 Cisco ONE

Cisco [17] prišlo na trh v júni 2012 s oficiálnym SDN riešením Cisco Open Network Enviroment (ONE). Túto platformu tvoria agenti, API rozhrania, kontroléry, sieťové technológie, ktoré umožňujú programovateľnosť na rozličných vrstvách SDN architektúry. Cisco ONE prostredie zahŕňa (Obr. 4):

- **Cisco ONE Platform Kit (onePK) –** balík API rozhraní, ktorý umožňuje aplikáciam riadiť Cisco zariadenia bez použitia príkazového riadka. Cisco onePK je dostupné na platformách Cisco IOS, IOS-XE, IOS-XR a NX-OS.
- **Cisco ONE Controller –** riadiaca časť pre skupinu zariadení podporujúcich platformu ONE. Aktuálne Cisco ponúka komerčnú distribúciu kontroléra OpenDayLight s názvom Cisco Open SDN Controller vo verzii 1.2.
- **Overlay networks –** balík produktov, ktorý poskytuje overlay siete, virtuálizačné služby a orchestračné možnosti, založený na zariadeniach Cisco Nexus 1000V, CSR 1000V.

V súčasnosti Cisco tlačí do popredia namiesto riešenia ONE svoje nové riešenie Cisco ACI (Application Centric Infrastructure), ktoré sčasti spĺňa podmienky SDN riešenia a sčasti nie. Cisco ACI je architektúra s pevne zviazanou infraštruktúrou založenou na politikách.

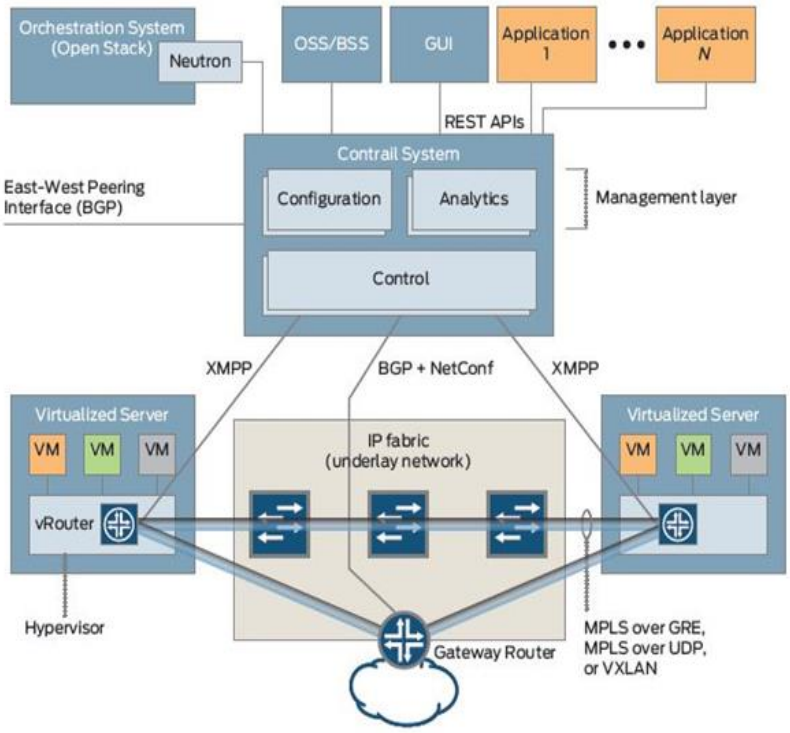


Obr. 4 Cisco ONE architektúra [18]

4.4.2 Juniper Contrail

V septembri 2013 Juniper vypustil prvú verziu svojho komplexného SDN riešenia s názvom Juniper Contrail, ktorý automatizuje vytváranie škálovateľných virtuálnych sietí. Toto riešenie [19] je navrhnuté na použitie v dátových centrách resp. cloudových prostriedach (Openstack, Cloudstack) a na orchestráciu resp. manažment sieťových funkcií (NFV) v sieti poskytovateľa služieb. Juniper Contrail pozostáva z dvoch hlavných komponentov – Contrail SDN Controller a Contrail SDN vRouter. Architektúru Contrailu tvorí viacero protokolov a technológií (Obr. 5). Obzvlášť je zaujímavá absencia protokolu

OpenFlow v Contrail riešení, ktorý je nahradený XMPP protokolom. Okrem komerčného riešenia Contrail Juniper taktiež ponúka jeho open source verziu s názvom OpenContrail (web: *http://www.opencontrail.org/)* pod licenciou Apache 2.0.



Obr. 5 Juniper Contrail architektúra [19]

4.5 SDN emulátory

4.5.1 Mininet

Mininet [20] je sieťový emulátor, ktorý emuluje kompletnú sieť z virtuálnych počítačov, prepínačov a liniek na jednom serveri. Mininet vytvára virtuálnu sieť použitím virtualizácie založenej na procesoch a tzv. sieťových oblastí mien (namespaces), ktoré sú implementované do súčasných linuxových jadier.

V Mininete sú virtuálne počítače emulované ako *bash* procesy bežiace v sieťovej oblasti mien, čo umožňuje spustiť ľubovoľný kód (napríklad web server alebo klientskú aplikáciu) vo vnútri virtuálneho počítača. Virtuálny počítač v Mininete ma svoje vlastné privátne sieťové rozhranie a môže vidieť len svoje vlastné procesy.

Prepínače v Mininete predstavujú softvérovo-založené prepínače podporujúce protokol OpenFlow (Open vSwitch, OpenFlow referenčný prepínač). Linky sú virtuálne ethernetové páry, ktoré sú prevádzkované v Linuxovom jadre a prepájajú emulované prepínače s emulovanými virtuálnymi počítačmi (procesmi).

Mininet je skvelý nástroj na vývoj a experimenty s protokolom OpenFlow a SDN systémami. Je aktívne vyvíjaný, podporovaný a vydaný pod BSD Open Source licenciou. Zdrojový kód Mininetu je takmer celý napísaný v jazyku Python.

4.5.2 EstiNet

EstiNet [21] je OpenFlow sieťový simulátor a emulátor. Podporuje dva módy – simulačný a emulačný mód. V simulačnom móde OpenFlow kontroléry ako NOX, POX, Floodlight, OpenDayLight a Ryu môžu byť priamo spustené na uzle kontroléra v simulovanej sieti. V emulačnom móde tieto kontroléry môžu byť spustené na externých serveroch oddelených od servera, na ktorom sú spustené simulované OpenFlow prepínače.

Hlavný rozdiel medzi Mininetom a EstiNetom je ten, že Mininet nedokáže garantovať preposielanie paketov rovnakou rýchlosťou. EstiNet v simulačnom móde simuluje presné vlastnosti liniek, ktoré prepájajú simulované OpenFlow prepínače. Tieto vlastnosti zahŕňajú šírku pásma linky, oneskorenie linky, časový prestoj linky a MAC (Medium Access Control) protokol použitý pozdĺž linky. EstiNet umožňuje zhromaždiť informácie o výsledkoch a vyhodnotiť výkon dátových tokov celej OpenFlow siete.

5 Návrh implementácie SDN do vyučovania

Návrh implementácie SDN do vyučovania pozostáva z výberu komponentov SDN riešenia, z výberu a zariadenia laboratória na KIS, z návrhu osnovy založenej na identifikácií kľúčových tém SDN pre predmet Integrácia sietí a nakoniec z návrhu ďalších možností skúmania SDN na KIS.

5.1 Výber komponentov SDN riešenia

5.1.1 Výber SDN prepínača

Vďaka širokému využitiu a podpory širokého spektra funkcií som sa rozhodol pre virtuálny prepínač Open vSwitch. Pri výbere som zohľadnil jednoduchosť inštalácie, podporu protokolu OpenFlow, jednoduchosť ovládania a podporu tunelovacích protokolov.

Open vSwitch je implementovaný do viacerých SDN riešení a je prevádzkovaný v mnohých veľkých produkčných prostrediach. Open vSwitch je implementovaný aj do emulačného prostredia Mininet, ktoré sa osvedčilo ako veľmi dobrý prostriedok na výučbu SDN.

5.1.2 Výber kontroléra

Keďže je k dispozícií veľké množstvo kontrolérov, rád by som do vyučovacieho procesu začlenil viac ako jeden kontrolér, aby si študenti mohli osvojiť rozličné prístupy a prostredia rozličných kontrolérov.

Na začiatku som pri výbere zvažoval použitie kontroléra OpenDayLight, keďže je to aktuálne najprogresívnejší a najvyvíjanejší kontrolér. Avšak konfigurovať toky prepínačom sa mi podarilo len cez jeho REST API rozhranie. Tento spôsob ovládania mi prišiel zdĺhavý a nepraktický pre potreby výučby. Hľadal som teda kontroléry s alternatívnejším a jednoduchším prístupom vzhľadom na konfiguráciu tokov.

Po vyskúšaní rôznych riešení som pre potreby cvičení vybral 3 kontroléry – Pox, Floodlight a OpenMUL. Pri výbere som zohľadňoval faktory zobrazené v tabuľke 1.

Kontrolér POX som vybral primárne pre jeho jednoduchosť inštalácie a ovládania. Kontrolér Floodlight som zas zvolil, pretože je stále aktívne vyvíjaný a má veľmi dobrý stav dokumentácie.

Za hlavný kontrolér pre výučbu SDN na KIS som vybral kontrolér OpenMUL. OpenMUL zahŕňa v sebe CLI rozhranie využívajúce syntax podobnú systému Cisco IOS. Naši študenti túto syntax poznajú a využívajú pri štúdiu. To je hlavný dôvod výberu tohto kontroléra. Avšak stále je otázny ďalší vývoj tohto riešenia. Posledná aktualizácia zdrojového kódu OpenMUL je datovaná na 8. septembra 2015.

	POX	Floodlight	OpenMUL
Jazyk	Python	Java	C
Podpora verzie OpenFlow	1.0	1.0/1.3	1.3
Stav dokumentácie	Dobrý	Veľmi dobrý	Veľmi dobrý
Webové GUI	Áno	Áno	Áno
Obtiažnosť inštalácie	Ľahká	Ľahká	Ľahká
Obtiažnosť ovládania	Ľahká	Ťažšia	Ľahká
REST API	Áno	Áno	Áno
Je naďalej vyvíjaný?	Nie	Áno	N/A

Tabuľka 1 Porovnanie parametrov vybraných kontrolérov

5.1.3 Výber hardvérového vybavenia

Čo sa týka výberu hardvérového vybavenia, zameral som sa na zariadenia v hodnote pod 100 eur. V tejto cenovej relácii som našiel smerovače od firmy Mikrotik s podporou protokolu OpenFlow vo verzii 1.0 a jednodoskové počítače ako Raspberry Pi alebo Banana Pi.

Nevýhoda jednodoskových počítačov je tá, že majú zväčša len jeden ethernetový port. Aj keď je možné do nich dokúpiť dva USB Ethernet adaptéry, ideálne by však bolo mať zariadenie s aspoň štyrmi ethernetovými portami. Tento nedostatok rieši smerovač Banana Pi BPI-R1, ktorý ma 5 ethernetových portov, a zároveň spĺňa cenovú podmienku.

Ďalšou možnosťou bol výber smerovača, na ktorý je možné nahrať operačný systém OpenWrt. Ako som spomenul v kapitole 4.3.6, jedným z kandidátov sú znova smerovače od firmy Mikrotik. Keďže Mikrotik smerovače podporujú OpenWrt systém a natívne aj OpenFlow 1.0, rozhodol som sa pre zariadenie od tejto firmy.

Zo širokej škály Mikrotik zariadení som si vybral produkt Routerboard RB951Ui-2HnD v hodnote 50 eur. RB951Ui-2HnD disponuje 600Mhz procesorom a dostatočne veľkou 128MB pamäťou. Postup a výsledky implementácie protokolu OpenFlow vo verzii 1.3 do zariadenia RB951Ui-2HnD sú spísané v kapitole 6.

5.1.4 Výber operačného systému

Výber operačného systému nebol zložitý. Išlo o výber vhodnej distribúcie operačného systému Linux, ktorý má podporu prepínača Open vSwitch. Taktiež bolo nutné preveriť, či tento operačný systém dokáže spustiť rôzne kontroléry ako OpenMul, Floodlight, OpenDayLight a pod. Tieto požiadavky úspešne spĺňa operačný systém Ubuntu.

5.1.5 Výber hypervizora

Na KIS je zaužívaná práca s hypervizorom Oracle VirtualBox, ktorý je nainštalovaný na počítačoch v učebniach KIS pod systémom Debian. Pre naše účely budeme potrebovať, aby bolo možné na počítačoch spustiť 3 virtuálne servery – Windows, Mininet a Ubuntu. Keďže sa nenašla závažnejšia prekážka pre prevádzkovanie týchto serverov vo VirtualBoxe, nie je nutná aktualizácia programového vybavenia samotných počítačov v učebniach KIS.

5.2 Výber a zariadenie laboratória na KIS

Po dohode s vedúcim práce bola pre účely vyučovania SDN vybraná učebňa B301 na KIS. Hlavná myšlienka SDN laboratória je mať k dispozícii softvérové aj hardvérové SDN riešenia na jednom mieste. Preto je nutné spraviť návrh ich nasadenia a použitia v laboratóriu.

5.2.1 Hardvérové vybavenie laboratória

Na hardvérové vybavenie laboratória som vybral Mikrotik smerovače s implementovaným Open vSwitch prepínačom (pozri kapitola 6). V učebni je 10 počítačov a pri práci vo dvojiciach by laboratórium mohlo byť vybavené približne 5-6timi Mikrotik smerovačmi (napríklad RB951Ui-2HnD) tak, aby na každú dvojicu vyšiel jeden smerovač. Tieto smerovače by bolo najlepšie umiestniť do racku v zadnej časti miestnosti.

Problémom je, že smerovače sú malé a nedajú sa upevniť do racku, čiže jedna možnosť je ich voľne položiť na existujúce zariadenia v racku. Druhou možnosťou je kúpiť špeciálny tzv. rack mount adapter (web: *http://www.balticnetworks.com/mikrotik-rackmount-adapter-for-routerboard-260-750-950-series.html*), ktorý umožní 3 takéto smerovače rady 260, 750, 950 pripevniť do racku.

Mikrotik smerovače budú použité na prepojenie virtuálnych servervov spustených na fyzických počítačoch v laboratóriu s SDN kontrolérmi. Prvý port každého Mikrotik smerovača bude použitý na komunikáciu s SDN kontrolérom prostredníctvom pripojenia do lokálnej siete katedry. Ostatné porty smerovača budú použité na pripojenie fyzických počítačov prostredníctvom patch panelu umiestneného v racku.

5.2.2 Softvérové vybavenie laboratória

Na počítačoch v miestnosti B301 je nainštalovaný hypervizor Oracle VirtualBox, pod ktorým je možné spustiť akékoľvek SDN riešenie vo virtuálnom serveri.

Pre potreby výučby budú potrebné 3 virtuálne servery. Prvý virtuálny server bude predstavovať emulátor Mininet. V prostredí Mininet budú študenti vytvárať vlastné virtuálne topológie. Druhý virtuálny server bude systém Ubuntu v úlohe SDN kontroléra. SDN kontrolér bude mať za úlohu spravovať Open vSwitch prepínače v Mininete, poprípade Open vSwitch prepínač implementovaný v Mikrotik smerovači. Tretí virtuálny server bude systém Windows, ktorý bude slúžiť ako hlavné pracovné prostredie. Zo systému Windows budú študenti ovládať prostredie Mininet.

Všetky tieto virtuálne servery budú pripojené do lokálnej siete katedry, čiže budú mať konektivitu medzi sebou, aj konektivitu s Mikrotik smerovačmi.

5.3 Návrh osnovy pre predmet Integrácia sietí

V nasledujúcej kapitole sa budem snažiť čo najpresnejšie popísať návrh tém pre novú osnovu predmetu Integrácia sietí.

S vedúcim práce sme sa dohodli na dvoch verziách. Prvá verzia bude obsahovať podklady venujúce sa technológií SDN na jeden celý semester, druhá verzia bude obsahovať podklady len na polovicu semestra. Osnova bude pozostávať z popisu tematických celkov pre náplň prednášok a z popisu praktických tém pre potreby vedenia cvičení.

Ku každému tematickému celku prikladám odporúčaný počet prednášok, ktoré by sa mali danej problematike venovať. Vypracovanie prednášok sa nachádza na DVD priloženom k diplomovej práci. Vypracovanie praktických úloh pre cvičenia sa nachádza v prílohe A.

5.3.1 Verzia pre výučbu SDN na celý semester

Na základe analýzy vykonanej v kapitole 1 navrhujem nasledujúci obsah prednášok a cvičení.

Prednášky

1. Úvod do technológie SDN (rozsah: 1 prednáška)

Prednáška má uviesť študenta do nedostatkov existujúcich sietí. Z týchto nedostatkov vyplýva potreba nového prístupu. Na základe tejto potreby má prednáška predstaviť technológiu SDN a jej historický vývoj.

Prednáška zahŕňa:

- aktuálny stav informačno-komunikačných technológií – popis klasickej architektúry sietí a jej nedostatkov, potreba nového prístupu, nástup virtualizácie
- úvod do SDN technológie – čo je to SDN, rozdelenie dátovej a riadiacej roviny siete, prínos centralizovaného prístupu, programovateľnosť, otvorenosť
- históriu SDN – technológie OpenSig, Active Networking, ForCES, Ethane

2. Architektúra SDN a jej terminológia (rozsah: 1 prednáška)

Vývojom SDN dochádza k nejasnostiam v definíciách týkajúcich sa architektúry SDN. Prednáška má presne zadefinovať architektúru SDN, ktorá je aplikovateľná na väčšinu existujúcich SDN riešení.

Prednáška zahŕňa:

- SDN entity – zdroj (port, front, sieťové zariadenie), sieťové zariadenie (fyzické alebo virtuálne), rozhranie (API, IPC, protokol), aplikácia
- roviny SDN – dátová rovina, operačná rovina, riadiaca rovina, manažmentová rovina, aplikačná rovina

- abstrakčné vrstvy SDN – riadiaca abstrakčná vrstva (CAL), manažmentová abstrakčná vrstva (MAL), abstrakčná vrstva sieťových služieb (NSAL), abstrakčná vrstva zariadení a zdrojov (DAL)

3. Protokol Openflow (rozsah: 1 prednáška)

OpenFlow je hlavný protokol, ktorého princípy je nutné ovládať pre potrebu cvičení. Prednáška ma študentovi sprostredkovať najdôležitejšie informácie o tomto protokole.

Prednáška zahŕňa:

- popis komunikácie – nadviazanie relácie medzi kontrolérom a prepínačom, typy správ
- popis tabuľky tokov – formát tabuľky tokov, porovnávacie kritéria, inštrukcie, akcie, sady akcií, zoznamy akcií
- odlišnosti vo verziách protokolu Openflow (1.0 až 1.5) – nové porovnávacie kritéria (Openflow Extensible Match), viacnásobný počet tabuliek, zreťazené spracovanie (pipeline processing), nové typy tabuliek (group tables a meter tables), podpora MPLS a VLAN tagov, podpora pripojenia viacerých kontrolérov

4. SDN riešenia (rozsah: 1 prednáška)

Aby študenti mali prehľad o technológii SDN, musia poznať existujúce SDN riešenia. Prednáška má popísať dostupné SDN riešenia na trhu.

Prednáška zahŕňa:

- popis virtuálnych SDN prepínačov – Open vSwitch, Indigo
- popis kontrolérov – POX, Floodlight, OpenDaylight, OpenMul, ONOS
- popis dostupných SDN HW prepínačov – Brocade, Pica8, HP, Juniper, Cisco
- popis komplexných SDN riešení – Cisco ONE, Juniper Contrail

5. SDN a virtualizačné technológie (rozsah: 2 prednášky)

Sieťová virtualizácia je kľúč k súčasnému a budúcemu úspechu cloudových technológií. Analýza vyučovania SDN vo svete ukázala, že drvivá väčšina univerzít spomína popri výučbe SDN technológiu NFV a sieťovú virtualizáciu. Táto prednáška poukazuje na dôvody pre použitie SDN v spojení s cloudovými technológiami a sieťovou virtualizáciou.

Prednáška zahŕňa:

- SDN a Cloud Computing (CC) – jednoduchá definícia CC, možnosti využitia SDN technológie v cloudovom prostredí, výhody kombinácie SDN a CC
- SDN a Network Function Virtualization (NFV) – jednoduchá definícia NFV, rozdiel NFV a SDN, možnosti využitia SDN technológie spolu s NFV
- SDN a Overlay siete – jednoduchý popis overlay technológii (VXLAN, NVGRE, STT), výhody použitia SDN s overlay sieťami

6. SDN aplikácie v rôznych prostrediach (rozsah: 2 prednášky)

SDN technológia nie je využívaná len v dátových centrách, ale je aplikovateľná do iných rôznych prostredí. Prednáška má uviesť prípady použitia SDN v rôznych prostrediach.

Prednáška zahŕňa:

- SDN aplikácie v Campus sieti – nasadenie bezpečnostných politík, zvýšenie bezpečnosti prístupu na internet
- SDN aplikácie v sieti SP – manažment šírky pásma, šetrenie CAPEX a OPEX nákladov, nastavenie politík na PE (Provider Edge) smerovačoch alebo NNI rozhraniach (Network-to-Network Interface), technológia MPLS-TE
- SDN aplikácie vo WAN – použitie technológií NetFlow, sFlow pre optimálne smerovacie rozhodnutia, použitie technológie MPLS-TE v spolupráci s kontrolérom pre optimálne cesty s prepínanými návestiami (LSP), technológia Path Computation Element (PCE) na výpočet optimálnej cesty medzi uzlami
- SDN aplikácie v mobilných sieťach – centralizované riadenie prostredí viacerých výrobcov, vyššie tempo inovácií, detailnejšie sieťové riadenie, zlepšená mobilita manažmentu

7. Bezpečnosť v SDN (rozsah: 1 prednáška)

Analýza vyučovania SDN vo svete ukázala, že viaceré univerzity skúmajú bezpečnostné aspekty SDN sietí. Prednáška má preto poukázať na bezpečnostné hrozby v SDN architektúre a spôsoby, ako sa im vyvarovať.

Prednáška zahŕňa:

- Bezpečnostné riziká architektúry SDN – vektory ohrozenia (threat vectors): falošné toky prevádzky, útoky na bezpečnostné diery SDN prepínačov, napadnutie riadiacej

časti architektúry SDN, útoky na bezpečnostné diery kontrolérov, nedostatočné zabezpečenie mechanizmov medzi kontrolérom a manažovacími aplikáciami, útoky na bezpečnostné diery v administratívnych staniciach, nedostatok dôveryhodných zdrojov v prípade výpadku SDN

- Bezpečnostné aplikácie založené na SDN – FRESCO framework pre zabezpečenie SDN aplikácií, CloudWatcher pre monitoring sieťovej bezpečnosti, CloudPolice pre riadenie prístupu v cloudovom prostredí, FlowVisor pre riadenie prístupu k viacerým kontrolérom

8. Organizácie v SDN ekosystéme a ich práca na SDN (rozsah: 1 prednáška)

Pre pochopenie vývoja SDN je vhodné vedieť, ktoré spoločnosti sa podieľajú na rozširovaní, štandardizácii, výrobe a výskume SDN riešení. Prednáška má uviesť organizácie tvoriace SDN ekosystém, a ako sa tieto organizácie podieľajú na vývoji SDN.

Medzi organizácie, ktoré treba v prednáške spomenúť, patria:

- Akademickí výskumníci – Stanford, UC Berkley, ONRC, Indiana (InCNTRE)
- Výskumné odvetvové laboratória – laboratória firiem HP, NTT, Microsoft, NEC, IBM, Fujitsu
- Výrobcovia sieťových zariadení – Cisco, Brocade, HP, NEC, Arista
- Softvéroví predajcovia – VMware, BigSwitch, Microsoft
- Predajcovia čipov – Broadcom, Intel
- Originálni výrobcovia zariadení – Accton, Quanta
- Enterprise spoločnosti – Google, Deutsche Telekom, Amazon, NTT, Verizon
- Štandardizačné spoločenstvá – ONF, IETF, IEEE

9. Alternatívne SDN technológie (rozsah: 1-2 prednášky)

OpenFlow nie je jediný SDN southbound protokol, o ktorom by študenti mali vedieť. Prednáška má uviesť iné alternatívne SDN technológie, ktoré sa využívajú v SDN prostrediach.

Prednáška zahŕňa popis protokolov:

- YANG – história, štruktúra, rozdelenie konfiguračných a stavových dát, modelovanie
- NETCONF – história, princíp, možnosti, operácie, použitie
- OF-CONFIG – motivácia, architektúra, použitie

- XMPP – alternatíva ku protokolu Openflow, motivácia

10. Budúcnosť SDN (rozsah: 1 prednáška)

OpenFlow má mnoho nedostatkov, preto je stále otázna jeho budúcnosť. Posledná prednáška by sa mohla venovať stručnej vízii o budúcnosti SDN.

Prednáška zahŕňa:

- OpenFlow 2.0 – riešenie nedostatkov Openflow 1.X, rekonfigurovateľný, protokolovo nezávislý, cieľovo nezávislý
- Huawei Protocol-Oblivious Forwarding (POF)
- Príchod novej generácie SDN prepínačov – náhrada hardvérových prepínacích čipov za procesory, Intel® DPDK vSwitch

Cvičenia

Zámerom pri budovaní obsahovej náplne cvičení bolo vybrať také témy, na ktorých si študenti prakticky overia problematiku preberanú na prednáškach. To všetko samozrejme so zámerom realizovateľnosti cvičenia. Študentov chceme previesť základnými princípmi, ktoré sa využívajú v SDN prostrediach, ako je napríklad riadenie toku dát prostredníctvom SDN kontroléra alebo tunelovacie mechanizmy.

1. Inštalácia Ubuntu a prepínača Open vSwitch

Cieľ cvičenia: inštalácia operačného systému Ubuntu vo VirtualBoxe, inštalácia prepínača Open vSwitch a ovládanie základných príkazov v tomto prepínači

Potrebné nástroje: Oracle VirtualBox, Ubuntu Server, Open vSwitch

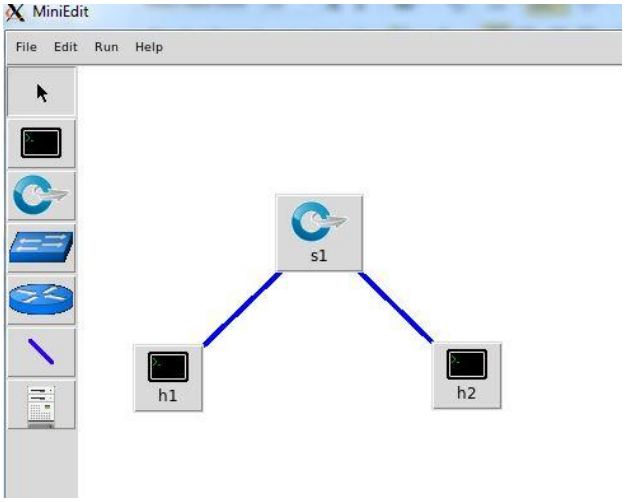
Popis: Študenti budú počas cvičení potrebovať vlastný virtuálny server so systémom Ubuntu. Buď si tento systém nainštalujú sami alebo im bude rozdistribuovaný už nainštalovaný Ubuntu systém, ktorý si importujú do prostredia VirtualBox. V systéme Ubuntu si študenti nainštalujú virtuálny prepínač Open vSwitch. V tomto prepínači si odtestujú základné príkazy, ako vytváranie bridge rozhraní, pridávanie a vymazávanie portov z bridge rozhraní, príkazy na odstraňovanie chýb, logovacie príkazy a pod. Tieto príkazy musia ovládať pre potrebu ďalších cvičení. Na toto cvičenie nie je potrebná žiadna topológia.

2. Práca s emulačným nástrojom Mininet

Cieľ cvičenia: vytvorenie topológie v Mininete a konfigurácia vlastných tokov prostredníctvom Open vSwitcha

Potrebné nástroje: Oracle VirtualBox, Mininet VM, Open vSwitch, Xming

Popis: Študenti si stiahnu predinštalovaný obraz Mininetu a importujú ho do prostredia VirtualBox. Spustia prostredie Mininet a pomocou zobrazovacieho servera Xming otvoria grafické rozhranie miniedit. V rozhraní miniedit si vytvoria triviálnu topológiu s jedným Open vSwitch prepínačom a dvomi virtuálnymi počítačmi (Obr. 6). Najprv sa naučia pracovať s príkazovým riadkom Mininetu a so všetkými jeho funkciami, ktoré budú testovať na spomínanej topológii. Ďalej študentom bude ukázané, ako môžu konfigurovať vlastné toky do prepínača Open vSwitch pomocou nástroja *ovs-ofctl*. Hlavnou úlohou cvičenia bude sfunkčniť cez tento nástroj konektivitu medzi virtuálnymi počítačmi.



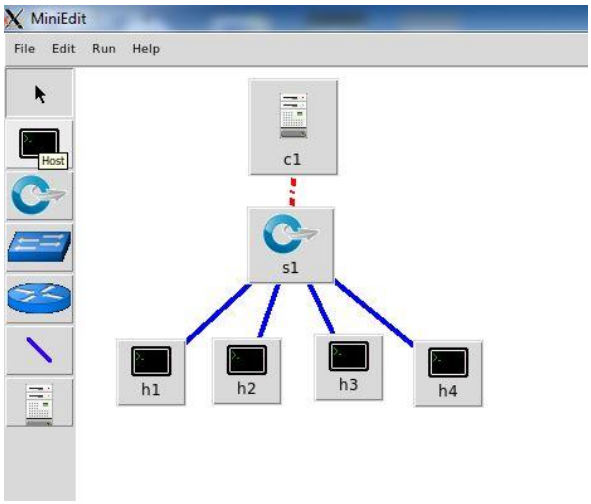
Obr. 6 Topológia 1

3. Napojenie topológie v Mininete na kontrolér POX

Cieľ cvičenia: napojenie Mininetu na kontrolér POX, odchytenie a analýza komunikácie medzi kontrolérom a prepínačom, nastavenie QoS nad portami prepínača

Potrebné nástroje: Oracle VirtualBox, Mininet VM, Open vSwitch, Xming, Pox, Wireshark, iperf

Popis: Študenti spustia prostredie Mininet vo VirtualBoxe a pomocou zobrazovacieho servera Xming otvoria grafické rozhranie miniedit. V rozhraní miniedit si vytvoria topológiu s jedným Open vSwitch prepínačom, s jedným kontrolérom a štyrmi virtuálnymi počítačmi (Obr. 7). Ďalej si študenti v rovnakom virtuálnom serveri spustia kontrolér Pox, ktorý bude plniť funkciu L2 prepínača a neskôr aj úlohu L3 smerovača. Študenti si odchytia a analyzujú OpenFlow 1.0 správy medzi Open vSwitch prepínačom a kontrolérom Pox pomocou Wiresharku. Študenti si vyskúšajú aj nastaviť QoS na konkrétny port Open vSwitch prepínača a pomocou aplikácie iperf overia, či sa QoS nastavenia správne aplikovali.



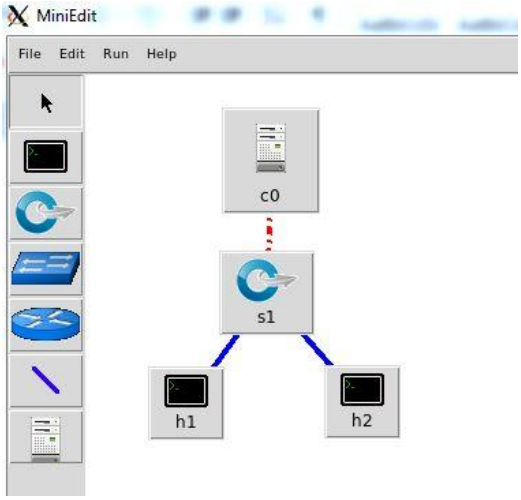
Obr. 7 Topológia 2

4. Napojenie topológie v Mininete na kontrolér Floodlight

Cieľ cvičenia: napojenie Mininetu na kontrolér Floodlight, odchytenie a analýza komunikácie medzi kontrolérom a prepínačom

Potrebné nástroje: Oracle VirtualBox, Mininet VM, Ubuntu VM, Open vSwitch, Xming, Floodlight, Wireshark, Avior, tcpdump

Popis: Študenti spustia prostredie Mininet vo VirtualBoxe a pomocou zobrazovacieho servera Xming otvoria grafické rozhranie miniedit. V rozhraní miniedit si vytvoria topológiu s jedným Open vSwitch prepínačom, s jedným kontrolérom a dvomi virtuálnymi počítačmi (Obr. 8). Študenti si spustia vo VirtualBoxe aj ďalší systém - Ubuntu, ktorý inštalovali na prvom cvičení. V prostredí Ubuntu stiahnu, skompilujú a spustia kontrolér Floodlight. Ďalej študenti odchytia a analyzujú OpenFlow 1.0 správy medzi Open vSwitch prepínačom a kontrolérom Floodlight pomocou Wiresharku a aplikácie tcpdump. Naučia sa aj čítať informácie z webového rozhrania kontroléra Floodlight. Študentom bude zadaná úloha zakázať komunikáciu medzi virtuálnymi počítačmi s použitím manažmentovej aplikácie Avior pre kontrolér Floodlight.



Obr. 8 Topológia 3

5. Napojenie topológie v Mininete na kontrolér OpenMUL

Cieľ cvičenia: napojenie Mininetu na kontrolér OpenMUL, odchytenie a analýza komunikácie medzi kontrolérom a prepínačom

Potrebné nástroje: Oracle VirtualBox, Mininet VM, Ubuntu VM, Open vSwitch, Xming, OpenMUL, Wireshark, tcpdump

Popis: Študenti spustia prostredie Mininet vo VirtualBoxe a pomocou zobrazovacieho servera Xming otvoria grafické rozhranie miniedit. V rozhraní miniedit si vytvoria topológiu s jedným Open vSwitch prepínačom, s jedným kontrolérom a dvomi virtuálnymi počítačmi (Obr. 9). Študenti si spustia vo VirtualBoxe aj ďalší systém - Ubuntu, ktorý inštalovali na prvom cvičení. V prostredí Ubuntu stiahnu, skompilujú a spustia kontrolér OpenMUL. Ďalej študenti odchytia a analyzujú OpenFlow 1.3 správy medzi Open vSwitch prepínačom a kontrolérom OpenMUL pomocou Wiresharku a aplikácie tcpdump. Študenti si naštudujú dokumentáciu k CLI rozhraniu OpenMUL kontroléra a prostredníctvom tohto rozhrania zakážu komunikáciu medzi virtuálnymi počítačmi.

6. Mikrotik zariadenie a kontrolér OpenMUL

Cieľ cvičenia: spustenie systému OpenWrt v Mikrotik smerovači, napojenie Open vSwitcha v OpenWrt na kontrolér OpenMUL, riadenie komunikácie medzi fyzickými počítačmi

Potrebné nástroje: Oracle VirtualBox, Ubuntu VM, Open vSwitch, OpenMUL, Mikrotik smerovač, winbox

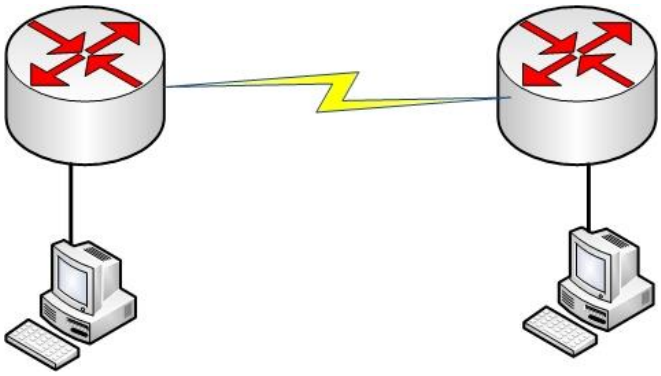
Popis: Študenti budú pracovať vo dvojiciach. Každá dvojica bude mať k dispozícii jeden Mikrotik smerovač. Študenti si zapoja svoje počítače do portu 2 a 3 na Mikrotik smerovači a port 1 pripoja do lokálnej siete katedry. Na Mikrotik smerovač sa napoja pomocou aplikácie winbox a cez funkciu Metarouter si spustia operačný systém OpenWRT, v ktorom bude implementovaný prepínač Open vSwitch. Študenti nakonfigurujú Open vSwitch podľa potreby. Ďalej si spustia vo VirtualBoxe systém Ubuntu a spustia OpenMUL kontrolér. Open vSwitch napoja na OpenMUL kontrolér. Študenti využijú poznatky z predošlých cvičení a budú manipulovať s komunikáciou medzi ich počítačmi prostredníctvom OpenMUL kontroléra.

7. Prepojenie dvoch virtuálnych prepínačov cez GRE tunel

Cieľ cvičenia: prepojenie virtuálnych prepínačov bežiacich na rozličných fyzických počítačoch cez GRE tunel, odchytenie a analýza komunikácie

Potrebné nástroje: Oracle VirtualBox, Open vSwitch, Mininet VM, Xming, Wireshark

Popis: Študenti budú pracovať vo dvojiciach. Vytvoria si fyzickú topológiu (Obr. 9) a nakonfigurujú smerovače tak, aby bola možná konektivita medzi počítačmi. Obaja študenti si spustia prostredie Mininet vo VirtualBoxe a pomocou zobrazovacieho servera Xming otvoria grafické rozhranie miniedit. V rozhraní miniedit si vytvoria virtuálnu topológiu s jedným Open vSwitch prepínačom a dvomi virtuálnymi počítačmi (Obr. 6). Študenti nastavia virtuálne počítače tak, aby dokopy všetky 4 virtuálne počítače mali rozdielne IP adresy z rovnakej siete. Následne obaja študenti vytvoria GRE tunel medzi svojimi virtuálnymi prepínačmi a odtestujú konektivitu medzi svojimi virtuálnymi počítačmi navzájom. Nakoniec študenti odchytia a analyzujú komunikáciu pomocou Wiresharku.



Obr. 9 Topológia 4

8. Prepojenie dvoch virtuálnych prepínačov cez VXLAN tunel

Cieľ cvičenia: prepojenie virtuálnych prepínačov bežiacich na rozličných fyzických počítačoch cez VXLAN tunel, odchytenie a analýza komunikácie

Potrebné nástroje: Oracle VirtualBox, Open vSwitch, Mininet VM, Xming, Wireshark

Popis: Študenti budú pracovať vo dvojiciach. Vytvoria si rovnakú fyzickú topológiu ako na prechádzajúcom cvičení (Obr. 9) a nakonfigurujú smerovače tak, aby bola možná konektivita medzi počítačmi. Obaja študenti si spustia prostredie Mininet vo VirtualBoxe a pomocou zobrazovacieho servera Xming otvoria grafické rozhranie miniedit. V rozhraní miniedit si vytvoria topológiu s jedným Open vSwitch prepínačom a dvomi virtuálnymi počítačmi (Obr. 6). Na rozdiel od predošlého cvičenia každý z dvojice nastaví svojím 2 virtuálnym počítačom rovnaké IP a MAC adresy. Následne obaja študenti vytvoria VXLAN tunel medzi svojimi virtuálnymi prepínačmi. Dodatočne musia nakonfigurovať toky, ktoré budú rozlišovať 2 rozdielne komunikácie medzi ich virtuálnymi počítačmi na základe VNI (VXLAN Network Identifier). Odtestujú konektivitu medzi svojimi virtuálnymi počítačmi navzájom. Nakoniec študenti odchytia a analyzujú VXLAN komunikáciu pomocou Wiresharku.

9. Zabezpečenie komunikácie prepínač - kontrolér

Cieľ cvičenia: zabezpečenie komunikácie medzi kontrolérom a prepínačom pomocou protokolu SSL

Potrebné nástroje: Oracle VirtualBox, Open vSwitch, Mininet VM, Xming, Wireshark

Popis: Študenti spustia prostredie Mininet vo VirtualBoxe. Vygenerujú si SSL certifikáty pre prepínač Open vSwitch a pre kontrolér ovs-controller, ktorý je súčasťou Open vSwitch riešenia. Následne aplikujú certifikáty a spustia ovs-controller. Pomocou zobrazovacieho servera Xming otvoria grafické rozhranie miniedit. V rozhraní miniedit si vytvoria topológiu s jedným Open vSwitch prepínačom, s jedným kontrolérom a dvomi virtuálnymi počítačmi (Obr. 8). Študenti overia konektivitu medzi virtuálnymi počítačmi. Nakoniec odchytia šifrovanú komunikáciu.

10. Doprogramovanie si vlastných častí do kontroléra podľa svojho výberu

Cieľ cvičenia: doprogramovanie vlastných častí do ľubovoľného kontroléra

Potrebné nástroje: ľubovoľné vývojové prostredie, ľubovoľný kontrolér

Popis: Študenti si môžu vybrať zo širokého spektra kontrolérov. Na stránke každého kontroléra je návod pre vývojárov určený na doprogramovanie aplikácii do daného kontroléra spolu s návrhmi na implementáciu. Formát tohto cvičenia môže slúžiť aj ako semestrálny projekt. Príklady programového rozšírenia jednotlivých kontrolérov sú dostupné na adresách:

POX

https://openflow.stanford.edu/display/ONL/Programming+in+POX

ONOS

https://wiki.onosproject.org/display/ONOS/Developer%27s+Guide

OpenDayLight

https://www.opendaylight.org/sites/opendaylight/files/bk-developers-guide-20150831.pdf

OpenMUL

http://www.openmul.org/devdoc-center.html

Floodlight

https://floodlight.atlassian.net/wiki/display/floodlightcontroller/For+Developers

5.3.2 Verzia pre výučbu SDN na polovicu semestra

Z verzie pre výučbu SDN na celý semester som vybral najdôležitejšie prednášky a cvičenia a poskladal som z nich verziu pre výučbu SDN na polovicu semestra. Zámer pre výber tém je, aby študenti získali len tie najzákladnejšie informácie o SDN, ktoré boli prezentované vo verzii na celý semester. Obsah jednotlivých prednášok a cvičení je rovnaký ako vo verzii na celý semester.

Prednášky

1. Úvod do technológie SDN (rozsah: 1 prednáška)

Prednáška má uviesť študenta do nedostatkov existujúcich sietí. Z týchto nedostatkov vyplýva potreba nového prístupu. Na základe tejto potreby má prednáška predstaviť technológiu SDN a jej historický vývoj.

Prednáška zahŕňa:

- aktuálny stav informačno-komunikačných technológií – popis klasickej architektúry sietí a jej nedostatkov, potreba nového prístupu, nástup virtualizácie
- úvod do SDN technológie – čo je to SDN, rozdelenie dátovej a riadiacej roviny siete, prínos centralizovaného prístupu, programovateľnosť, otvorenosť
- históriu SDN – technológie OpenSig, Active Networking, ForCES, Ethane

2. Architektúra SDN a jej terminológia (rozsah: 1 prednáška)

Vývojom SDN dochádza k nejasnostiam v definíciách týkajúcich sa architektúry SDN. Prednáška má presne zadefinovať architektúru SDN, ktorá je aplikovateľná na drvivú väčšinu existujúcich SDN riešení.

Prednáška zahŕňa:

- SDN entity – zdroj (port, front, sieťové zariadenie), sieťové zariadenie (fyzické alebo virtuálne), rozhranie (API, IPC, protokol), aplikácia
- roviny SDN – dátová rovina, operačná rovina, riadiaca rovina, manažmentová rovina, aplikačná rovina
- abstrakčné vrstvy SDN – riadiaca abstrakčná vrstva (CAL), manažmentová abstrakčná vrstva (MAL), abstrakčná vrstva sieťových služieb (NSAL), abstrakčná vrstva zariadení a zdrojov (DAL)

3. Protokol Openflow (rozsah: 1 prednáška)

OpenFlow je hlavný protokol, ktorého princípy je nutné ovládať pre potrebu cvičení. Prednáška ma študentovi sprostredkovať najdôležitejšie informácie o tomto protokole.

Prednáška zahŕňa:

- popis komunikácie – nadviazanie relácie medzi kontrolérom a prepínačom, typy správ
- popis tabuľky tokov – formát tabuľky tokov, porovnávacie kritéria, inštrukcie, akcie, sady akcií, zoznamy akcií
- odlišnosti vo verziách protokolu Openflow (1.0 až 1.5) – nové porovnávacie kritéria (Openflow Extensible Match), viacnásobný počet tabuliek, zreťazené spracovanie (pipeline processing), nové typy tabuliek (group tables a meter tables), podpora MPLS a VLAN tagov, podpora pripojenia viacerých kontrolérov

4. SDN riešenia (rozsah: 1 prednáška)

Aby študenti mali prehľad o technológii SDN, musia poznať existujúce SDN riešenia. Prednáška má popísať dostupné SDN riešenia na trhu.

Prednáška zahŕňa:

- popis virtuálnych SDN prepínačov – Open vSwitch, Indigo
- popis kontrolérov – POX, Floodlight, OpenDaylight, OpenMul, ONOS

- popis dostupných SDN HW prepínačov – Brocade, Pica8, HP, Juniper, Cisco
- popis komplexných riešení – Cisco ONE, Juniper Contrail

5. SDN aplikácie v rôznych prostrediach (rozsah: 1 prednáška)

SDN technológia nie je využívaná len v dátových centrách, ale je aplikovateľná do iných rôznych prostredí. Prednáška má uviesť prípady použitia SDN v rôznych prostrediach.

Prednáška zahŕňa:

- SDN aplikácie v Campus sieti – nasadenie bezpečnostných politík, zvýšenie bezpečnosti prístupu na internet
- SDN aplikácie v sieti SP – manažment šírky pásma, šetrenie CAPEX a OPEX nákladov, nastavenie politík na PE (Provider Edge) smerovačoch alebo NNI rozhraniach (Network-to-Network Interface), technológia MPLS-TE
- SDN aplikácie vo WAN – použitie technológií NetFlow, sFlow pre optimálne smerovacie rozhodnutia, použitie technológie MPLS-TE v spolupráci s kontrolérom pre optimálne cesty s prepínanými návestiami (LSP), technológia Path Computation Element (PCE) na výpočet optimálnej cesty medzi uzlami
- SDN aplikácie v mobilných sieťach – centralizované riadenie prostredí viacerých výrobcov, vyššie tempo inovácií, detailnejšie sieťové riadenie, zlepšená mobilita manažmentu

6. Alternatívne SDN technológie (rozsah: 1 prednáška)

OpenFlow nie je jediný SDN southbound protokol, o ktorom by študenti mali vedieť. Prednáška má uviesť iné alternatívne SDN technológie, ktoré sa využívajú v SDN prostrediach.

Prednáška zahŕňa popis protokolov:

- YANG – história, štruktúra, rozdelenie konfiguračných a stavových dát, modelovanie
- NETCONF – história, princíp, možnosti, operácie, použitie
- OF-CONF – motivácia, architektúra, použitie
- XMPP – alternatíva ku protokolu Openflow, motivácia

Cvičenia

1. Inštalácia Ubuntu a prepínača Open vSwitch

Cieľ cvičenia: inštalácia operačného systému Ubuntu vo VirtualBoxe, inštalácia prepínača Open vSwitch a ovládanie základných príkazov v tomto prepínači

Potrebné nástroje: Oracle VirtualBox, Ubuntu Server, Open vSwitch

Popis: Študenti budú počas cvičení potrebovať vlastný virtuálny server so systémom Ubuntu. Buď si tento systém nainštalujú sami alebo im bude rozdistribuovaný už nainštalovaný Ubuntu systém, ktorý si importujú do prostredia VirtualBox. V systéme Ubuntu si študenti nainštalujú virtuálny prepínač Open vSwitch. V tomto prepínači si odtestujú základné príkazy, ako vytváranie bridge rozhraní, pridávanie a vymazávanie portov z bridge rozhraní, príkazy na odstraňovanie chýb, logovacie príkazy a pod. Tieto príkazy musia ovládať pre potrebu ďalších cvičení. Na toto cvičenie nie je potrebná žiadna topológia.

2. Práca s emulačným nástrojom Mininet

Cieľ cvičenia: vytvorenie topológie v Mininete a konfigurácia vlastných tokov prostredníctvom Open vSwitcha

Potrebné nástroje: Oracle VirtualBox, Mininet VM, Open vSwitch, Xming

Popis: Študenti si stiahnu predinštalovaný obraz Mininetu a importujú ho do prostredia VirtualBox. Spustia prostredie Mininet a pomocou zobrazovacieho servera Xming otvoria grafické rozhranie miniedit. V rozhraní miniedit si vytvoria triviálnu topológiu s jedným Open vSwitch prepínačom a dvomi virtuálnymi počítačmi (Obr. 6). Najprv sa naučia pracovať s príkazovým riadkom Mininetu a so všetkými jeho funkciami, ktoré budú testovať na spomínanej topológií. Ďalej študentom bude ukázané, ako môžu konfigurovať vlastné toky do prepínača Open vSwitch pomocou nástroja *ovs-ofctl*. Hlavnou úlohou cvičenia bude sfunkčniť cez tento nástroj konektivitu medzi virtuálnymi počítačmi.

3. Napojenie topológie v Mininete na kontrolér POX

Cieľ cvičenia: napojenie Mininetu na kontrolér POX, odchytenie a analýza komunikácie medzi kontrolérom a prepínačom, nastavenie QoS nad portami prepínača

Potrebné nástroje: Oracle VirtualBox, Mininet VM, Open vSwitch, Xming, Pox, Wireshark, iperf

Popis: Študenti spustia prostredie Mininet vo VirtualBoxe a pomocou zobrazovacieho servera Xming otvoria grafické rozhranie miniedit. V rozhraní miniedit si vytvoria topológiu s jedným Open vSwitch prepínačom, s jedným kontrolérom a štyrmi virtuálnymi počítačmi (Obr. 7). Ďalej si študenti v rovnakom virtuálnom serveri spustia kontrolér Pox, ktorý bude plniť funkciu L2 prepínača a neskôr aj úlohu L3 smerovača. Študenti si odchytia a analyzujú OpenFlow 1.0 správy medzi Open vSwitch prepínačom a kontrolérom Pox pomocou Wiresharku. Študenti si vyskúšajú aj nastaviť QoS na konkrétny port Open vSwitch prepínača a pomocou aplikácie iperf overia, či sa QoS nastavenia správne aplikovali.

4. Napojenie topológie v Mininete na kontrolér OpenMUL

Cieľ cvičenia: napojenie Mininetu na kontrolér OpenMUL, odchytenie a analýza komunikácie medzi kontrolérom a prepínačom

Potrebné nástroje: Oracle VirtualBox, Mininet VM, Ubuntu VM, Open vSwitch, Xming, OpenMUL, Wireshark, tcpdump

Popis: Študenti spustia prostredie Mininet vo VirtualBoxe a pomocou zobrazovacieho servera Xming otvoria grafické rozhranie miniedit. V rozhraní miniedit si vytvoria topológiu s jedným Open vSwitch prepínačom, s jedným kontrolérom a dvomi virtuálnymi počítačmi (Obr. 8). Študenti si spustia vo VirtualBoxe aj ďalší systém - Ubuntu, ktorý inštalovali na prvom cvičení. V prostredí Ubuntu stiahnu, skompilujú a spustia kontrolér OpenMUL. Ďalej študenti odchytia a analyzujú OpenFlow 1.3 správy medzi Open vSwitch prepínačom a kontrolérom OpenMUL pomocou Wiresharku a aplikácie tcpdump. Študenti si naštudujú dokumentáciu k CLI rozhraniu OpenMUL kontroléra a prostredníctvom tohto rozhrania zakážu komunikáciu medzi virtuálnymi počítačmi.

5. Mikrotik zariadenie a kontrolér OpenMUL

Cieľ cvičenia: spustenie systému OpenWrt v Mikrotik smerovači, napojenie Open vSwitcha v OpenWrt na kontrolér OpenMUL, riadenie komunikácie medzi fyzickými počítačmi

Potrebné nástroje: Oracle VirtualBox, Ubuntu VM, Open vSwitch, OpenMUL, Mikrotik smerovač, winbox

Popis: Študenti budú pracovať vo dvojiciach. Každá dvojica bude mať k dispozícii jeden Mikrotik smerovač. Študenti si zapoja svoje počítače do portu 2 a 3 na Mikrotik smerovači a port 1 pripoja do lokálnej siete katedry. Na Mikrotik smerovač sa napoja pomocou aplikácie winbox a cez funkciu Metarouter si spustia operačný systém OpenWRT, v ktorom bude implementovaný prepínač Open vSwitch. Študenti nakonfigurujú Open vSwitch podľa potreby. Ďalej si spustia vo VirtualBoxe systém Ubuntu a spustia OpenMUL kontrolér. Open vSwitch napoja na OpenMUL kontrolér. Študenti využijú poznatky z predošlých cvičení a budú manipulovať s komunikáciou medzi ich počítačmi prostredníctvom OpenMUL kontroléra.

6. Prepojenie dvoch virtuálnych prepínačov cez GRE tunel

Cieľ cvičenia: prepojenie virtuálnych prepínačov bežiacich na rozličných fyzických počítačoch cez GRE tunel, odchytenie a analýza komunikácie

Potrebné nástroje: Oracle VirtualBox, Open vSwitch, Mininet VM, Xming, Wireshark

Popis: Študenti budú pracovať vo dvojiciach. Vytvoria si fyzickú topológiu (Obr. 9) a nakonfigurujú smerovače tak, aby bola možná konektivita medzi počítačmi. Obaja študenti si spustia prostredie Mininet vo VirtualBoxe a pomocou zobrazovacieho servera Xming otvoria grafické rozhranie miniedit. V rozhraní miniedit si vytvoria virtuálnu topológiu s jedným Open vSwitch prepínačom a dvomi virtuálnymi počítačmi (Obr. 6). Študenti nastavia virtuálne počítače tak, aby dokopy všetky 4 virtuálne počítače mali rozdielne IP adresy z rovnakej siete. Následne obaja študenti vytvoria GRE tunel medzi svojimi virtuálnymi prepínačmi a odtestujú konektivitu medzi svojimi virtuálnymi počítačmi navzájom. Nakoniec študenti odchytia a analyzujú komunikáciu pomocou Wiresharku.

5.4 Ďalšie možnosti skúmania SDN na KIS

SDN ponúka takmer neobmedzené možnosti pre obohatenie výučby na KIS. Pri riešení diplomovej práce som sa stretol s riešeniami, ktoré by mohli pomôcť rozšíriť výučbu SDN, resp. by mohli byť aplikované ako návrhy na záverečné práce.

Prvý postreh sa týka grafického rozhrania Avior vo verzii 2.0 (web: *http://sdn.marist.edu/avior).* Avior 2.0 podstatne zjednodušuje prácu s kontrolérom cez webové rozhranie, čo môže pomôcť vylepšiť výučbu SDN. Aktuálne Avior 2.0 podporuje kontroléry ako OpenDayLight, Floodligh, Ryu a OpenMUL. Keďže počas riešenia práce

mi nezostalo viac času na preštudovanie Avior 2.0 rozhrania, navrhujem, aby študenti počas projektovej výučby preskúmali jeho možnosti.

Na KIS je implementovaná cloudová platforma OpenStack. Ďalšia možnosť skúmania by mohla spočívať v integrácií platformy OpenContrail do OpenStacku na KIS. OpenContrail riešenie je spomenuté v kapitole 4.4.2 v rámci SDN riešenia Juniper Contrail. Táto kombinácia platforiem by mohla priniesť mnoho výhod do vyučovacieho procesu na KIS.

Ďalším zaujímavým návrhom na skúmanie je technológia POF (Protocol Obvious Forwarding) od spoločnosti Huawei. Toto open source riešenie je dostupné na adrese *http://www.poforwarding.org/*. POF sa zameriava na vylepšenie protokolu OpenFlow o flexibilnejší programovací model, v ktorom prepínače nie sú limitované preddefinovanými protokolmi alebo preposielacími pravidlami.

Na koniec by som chcel spomenúť kontrolér ONOS, ktorý je popísaný v kapitole 4.2.5. ONOS ponúka množstvo spôsobov použitia s rôznymi sieťovými technológiami a protokolmi. ONOS je veľmi pestrý kontrolér, ktorý odporúčam na podrobnejšie preskúmanie pre účely vyučovania SDN na KIS.

6 Implementácia protokolu OpenFlow 1.3 do Mikrotiku

Mojím cieľom bolo implementovať podporu protokolu OpenFlow vo verzii 1.3 do zariadenia RouterBoard RB951Ui-2HnD. Pre splnenie tejto požiadavky bolo nutnosťou na zariadenie nainštalovať operačný systém OpenWrt.

K dispozícii sú dve možnosti ako spustiť OpenWrt na Mikrotik zariadení. Prvá možnosť je nainštalovať systém na internú flash pamäť. Druhá možnosť je využiť novú funkciu systému RouterOS tzv. MetaRouter (od verzie 3.21), ktorý umožňuje spustiť OpenWrt ako virtuálnu inštanciu.

Ako východiskové riešenie pre podporu protokolu OpenFlow som si zvolil softvérový prepínač Open vSwitch, ktorý bolo potrebné integrovať do systému OpenWrt.

6.1 Problémové riešenie

Zvolil som si aktuálnu stabilnú verziu OpenWrt s názvom Chaos Calmer (15.05), balík Open vSwitch vo verzii 2.3.0 pre verziu Chaos Calmer a operačný systém Ubuntu 14.04, na ktorom som OpenWrt pre platformu Mikrotik skompiloval.

Po spustení systému OpenWrt na zariadení RB951Ui-2HnD som začal s testovaním. Prvý problém, ktorý som musel vyriešiť, sa týkal rozhraní smerovača. Systém rozpoznával len rozhranie eth0 (port 1 na zariadení) a rozhranie eth1 (porty 2-5 na zariadení), ktoré predstavovalo klasický prepínač. Pre naše účely bolo nevyhnutné mať 5 konfigurovateľných rozhraní odpovedajúcich piatim portom smerovača. Riešenie bolo rozhranie eth1 rozdeliť do viacerých VLAN sietí.

Do konfiguračného súboru */etc/config/network* som pridal nasledovné riadky.

```
config switch
        option name 'switch0'
        option reset '1'
        option enable_vlan '1'
        option enable_learning '0'

config interface 'lan1'
        option ifname 'eth1.1'
        option proto 'static'

config interface 'lan2'
        option ifname 'eth1.2'
```

```
        option proto 'static'

config interface 'lan3'
        option ifname 'eth1.3'
        option proto 'static'

config interface 'lan4'
        option ifname 'eth1.4'
        option proto 'static'

config switch_vlan
        option device 'switch0'
        option vlan '4'
        option vid '4'
        option ports '0t 1'

config switch_vlan
        option device 'switch0'
        option vlan '3'
        option vid '3'
        option ports '0t 2'

config switch_vlan
        option device 'switch0'
        option vlan '2'
        option vid '2'
        option ports '0t 3'

config switch_vlan
        option device 'switch0'
        option vlan '1'
        option vid '1'
        option ports '0t 4'
```

Vytvoril som tak 4 nové rozhrania (eth1.1 – VLAN 1, eth1.2 – VLAN 2, eth1.3 – VLAN 3, eth1.4 – VLAN 4), pričom každé rozhranie prislúchalo práve jednému portu na smerovači.

Všetko nasvedčovalo tomu, že mám k dispozícii 5 plnohodnotných, logicky oddelených rozhraní. Nastal však druhý problém, ktorý sa týkal aplikácie Open vSwitch a vytvárania bridge rozhraní v Linuxe všeobecne. Keď som vytvoril bridge rozhranie z viacerých „podrozhraní“ napr. spojením eth1.3 a eth1.4, tak komunikácia sa začala rozpadávať. Po odchytení komunikácie bolo možné vidieť, že niektoré pakety sa strácajú. Po preskúmaní všetkých možností som usúdil, že tento problém nevyriešim a môžem ho pokojne označiť za „bug“ samotného OpenWrt systému. Aj keď balík Open vSwitch, ktorý

som skompiloval pre verziu Chaos Calmer bol stabilný a plne funkčný, musel som sa vrátiť k pôvodnému systému zariadenia RouterOS.

6.2 Výsledné riešenie

Druhou možnosťou ako prevádzkovať systém OpenWrt na Mikrotik zariadení je spustiť ho ako virtuálnu inštanciu pomocou funkcie Metarouter v systéme RouterOS. Zvolil som si staršiu stabilnú verziu OpenWrt Attitude Adjustment (12.09), rovnaký balík Open vSwitch vo verzii 2.3.0 a operačný systém Ubuntu 14.04, na ktorom som OpenWrt pre platformu Metarouter skompiloval. Skompilovaný systém OpenWrt s Open vSwitchom pre platformu Metarouter je priložený na DVD k diplomovej práci.

Po spustení systému OpenWrt cez funkciu Metarouter bolo dôležité overiť stabilitu bežiaceho systému, pretože funkcia Metarouter je vysoko experimentálna a nie moc stabilná. OpenWrt verzia Attitude Adjustment pre platformu Metarouter sa mi osvedčila ako veľmi stabilná. Systém ani raz nespadol za 10 hodín prevádzky. Avšak balík Open vSwitch skompilovaný pre túto platformu je miestami nestabilný. Ak aplikácia nečakane spadne stačí ju reštarovať príkazom */etc/init.d/openvswitch restart*. Napriek tomu je toto riešenie plne funkčné a osvedčilo sa mi ako vyhovujúce pre potreby diplomovej práce.

Záver

Hlavným cieľom práce bolo vyhotoviť koncept vyučovania SDN v rámci študijného programu Aplikované sieťové inžinierstvo v predmete Integrácia sietí. K naplneniu tohto cieľa som na začiatku vypracoval analýzu súčasného stavu vyučovania SDN vo svete. Ďalej som sa teoreticky a prakticky oboznámil s technológiou SDN. Keď som získal dostatočne veľa informácií, vypracoval som samotný návrh vyučovania SDN.

Ako podklad pre návrh vyučovania SDN som vybral potrebné komponenty. To pozostávalo z výberu virtuálneho SDN prepínača Open vSwitch, kontroléra OpenMUL, operačného systému Ubuntu, hypervizora VirtualBox a hardvérového zariadenia značky Mikrotik.

Do Mikrotik zariadenia som implementoval prepínač Open vSwitch prostredníctvom operačného systému OpenWrt. OpenWrt systém je prevádzkovaný v Mikrotik zariadení ako virtuálna inštancia cez jeho funkciu Metarouter. Týmto som dosiahol v zariadení plnú podporu protokolu OpenFlow 1.3.

Ďalej som navrhol zariadenie KIS laboratória B301, v ktorom by mala prebiehať výučba SDN. Toto laboratórium by mohlo pozostávať z komponentov, ktoré som uviedol vyššie. Do návrhu som začlenil, kde budú komponenty môjho SDN riešenia umiestnené, ako a za akým účelom budú používané.

V poslednej fáze riešenia diplomovej práce som navrhol a vypracoval podklady pre prednášky a cvičenia v dvoch verziách – verzia pre výučbu SDN na celý semester a verzia pre výučbu SDN na polovicu semestra. Pri vypracovaní praktických úloh pre cvičenia som využil emulačné prostredie Mininet, v ktorom je možné vytvoriť ľubovoľnú SDN topológiu.

Na koniec som uviedol ďalšie možnosti skúmania SDN na KIS s využitím vo vyučovaní.

Zoznam použitej literatúry

[1] „Software-Defined Networking: The New Norm for Networks," [Online]. Dostupné na: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf. [Cit. 10. 4. 2016].

[2] P. Goransson a C. Black, Software Defined Networks A Comprehensive Approach, Elsevier Inc, 2014.

[3] „SDN Is Business, OpenFlow Is Technology," [Online]. Dostupné na: http://www.networkcomputing.com/networking/sdn-business-openflow-technology/53316220. [Cit. 10. 4. 2016].

[4] E. Haleplidis, K. Pentikousis, S. Denazis, J. Hadi Salim, D. Meyer a O. Koufopavlou, „Software-Defined Networking (SDN): Layers and Architecture Terminology," 2015. [Online]. Dostupné na: https://tools.ietf.org/html/rfc7426.

[5] W. Braun a M. Menth, „Software-Defined Networking Using OpenFlow: Protocols,Applications and Architectural Design Choices," 2014. [Online]. Dostupné na: http://www.mdpi.com/1999-5903/6/2/302/pdf. [Cit. 10. 4. 2016].

[6] „OpenFlow," [Online]. Dostupné na http://yuba.stanford.edu/cs244wiki/index.php/Overview. [Cit. 10. 4. 2016].

[7] „What is Open vSwitch (OVS)?," [Online]. Dostupné na: https://www.sdxcentral.com/resources/open-source/what-is-open-vswitch/. [Cit. 10. 4. 2016].

[8] „Indigo Virtual Switch," [Online]. Dostupné na: http://www.projectfloodlight.org/indigo-virtual-switch/. [Cit. 10. 4. 2016].

[9] „Cisco Virtual Topology System: Data Center Automation for Next-Generation Cloud Architectures White Paper," [Online]. Dostupné na: http://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/virtual-topology-system/white-paper-c11-734904.html. [Cit. 10. 4. 2016].

[10] „POX,“ [Online]. Dostupné na: http://searchsdn.techtarget.com/definition/POX. [Cit. 10. 4. 2016].

[11] S. Kaur, J. Singh a N. Ghumman Singh, „Network Programmability Using POX Controller,“ [Online]. Dostupné na: http://www.sbsstc.ac.in/icccs2014/Papers/Paper28.pdf. [Cit. 10. 4. 2016].

[12] „Floodlight,“ [Online]. Dostupné na: http://www.projectfloodlight.org/floodlight/. [Cit. 10. 4. 2016].

[13] „LITHIUM OVERVIEW,“ [Online]. Dostupné na: https://www.opendaylight.org/lithium. [Cit. 10. 4. 2016].

[14] „OPENMUL CONTROLLER,“ [Online]. Dostupné na: http://www.openmul.org/openmul-controller.html. [Cit. 10. 4. 2016].

[15] „Open Network Operating System (ONOS),“ [Online]. Dostupné na: https://www.sdxcentral.com/projects/on-lab-open-network-operating-system-onos/. [Cit. 10. 4. 2016].

[16] M. Krška, Softvérovo-definované siete, 2015.

[17] I. Turus, „OpenFlow Technology Investigation Vendors Review on OpenFlow implementation,“ 2012. [Online]. Dostupné na: https://www.terena.org/activities/netarch/ws1/slides/turus-openflow-211112-JRA1.pdf.

[18] „What is the Cisco ONE Controller?,“ [Online]. Dostupné na: https://www.sdxcentral.com/resources/cisco/cisco-one-controller/. [Cit. 10. 4. 2016].

[19] „Contrail Architecture,“ [Online]. Dostupné na: http://www.juniper.net/us/en/local/pdf/whitepapers/2000535-en.pdf. [Cit. 10. 4. 2016].

[20] „Mininet Overview,“ [Online]. Dostupné na: http://mininet.org/overview/. [Cit. 10. 4. 2016].

[21] „EstiNet 9.0 OpenFlow Network Simulator and Emulator,“ [Online]. Dostupné na: https://www.sdxcentral.com/products/estinet-8-1-openflow-network-simulator-and-

emulator/. [Cit. 10. 4. 2016].

Zoznam príloh

Príloha A Vypracovanie praktických úloh pre cvičenia

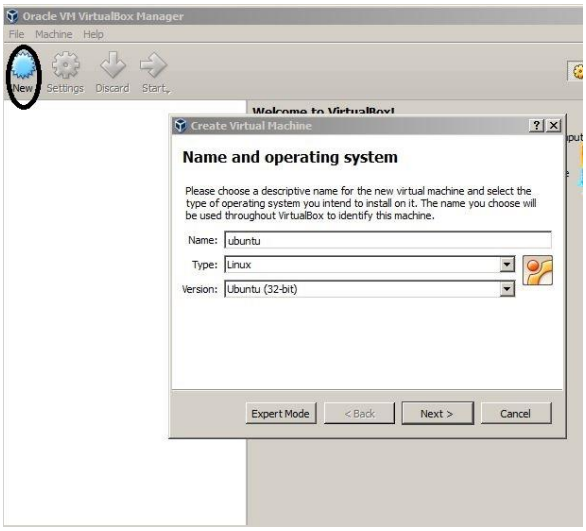
Príloha B Obsah DVD

Prílohy

Príloha A: Vypracovanie praktických úloh pre cvičenia

Cvičenie 1: Inštalácia Ubuntu a Open vSwitcha vo VirtualBoxe

1. Stiahneme obraz systému Ubuntu Server 14.04 LTS podľa architektúry procesora zo stránky *http://www.ubuntu.com/download/server* .
2. Vytvoríme nový virtuálny server (VM) vo VirtualBoxe, zadáme jeho meno, typ a verziu systému.



3. Zvolíme veľkosť RAM.
4. Vytvoríme nový virtuálny hard disk.
5. Zvolíme typ hard disku na VDI.
6. Zvolíme fixovanú veľkosť disku.
7. Zvolíme veľkosť disku na 5GB.
8. Vytvorí sa VM Ubuntu, ktorú nastavíme takto:

- do optickej mechaniky vložíme obraz Ubuntu, ktorý sme stiahli
- nastavíme sieťovú kartu, ktorou sa pripájame do internetu a zvolíme možnosť Bridged Adapter.

9. Spustíme Ubuntu VM.
10. Spustíme inštaláciu Ubuntu výberom anglického jazyka.
11. Vyberieme možnosť Install Ubuntu Server.
12. Vyberieme jazyk inštalácie výberom anglického jazyka (English).
13. Vyberieme lokáciu *other->Europe->Slovakia.*
14. Vyberieme lokalitu pre nastavenia klávesnice na United States – en_US.UTF-8.
15. Pri detekcii rozloženia klávesnice zvolíme možnosť No.

16. Nastavíme klávesnicu a rozloženie klávesnice na English(US).
17. Zadáme si ľubovoľný hostname, napríklad server.
18. Zadáme meno používateľa.
19. Zadáme prihlasovacie meno pre tohto používateľa.
20. Zvolíme heslo pre tohto používateľa a následne ho zopakujeme.
21. Povolíme zašifrovanie domovskej zložky zvolením Yes.
22. Potvrdíme správnosť časovej zóny Yes/No.
23. Zvolíme Guided – use entire disk.
24. Zvolíme vytvorený hard disk.
25. Zapíšeme zmeny na disk zvolením možnosti Yes.
26. Pri konfigurácii balíčkovacieho manažéra nevypĺňame nič.
27. Zvolíme možnosť No automatic updates.
28. Pri výbere softvéru na inštalovanie nemusíme zvoliť nič.
29. Nainštalujeme Grub Boot loader na hlavnú boot partíciu zvolením možnosti Yes.
30. Potvrdíme hotovú inštaláciu zvolením možnosti Continue.
31. Prihlásime sa pod menom a heslom, ktoré sme zvolili pri inštalácii.
32. Aktualizujeme systém zadaním príkazu sudo apt-get update.
33. Nainštalujeme Open vSwitch príkazom sudo apt-get install open vswitch-switch.
34. Skontrolujeme funkčnosť Open vSwitcha cez príkaz sudo ovs-vsctl show, výpis by mal zobraziť verziu Open vSwitcha.
35. Vytvoríme bridge rozhranie ovsbr príkazom sudo ovs-vsctl add-br ovsbr.
36. Skontrolujeme, či sa bridge rozhranie pridalo pomocou príkazu ifconfig.
37. Pridáme rozhranie napr. eth0 do bridge rozhrania príkazom sudo ovs-vsctl add-port ovsbr eth0.
38. Overíme či sa rozhranie pridalo do bridge rozhrania príkazom sudo ovs-vsctl show.
39. Odstránime rozhranie z bridge rozhrania príkazom sudo ovs-vsctl del-port ovsbr eth0.
40. Odstránime bridge rozhranie príkazom sudo ovs-vsctl del-br ovsbr eth0.

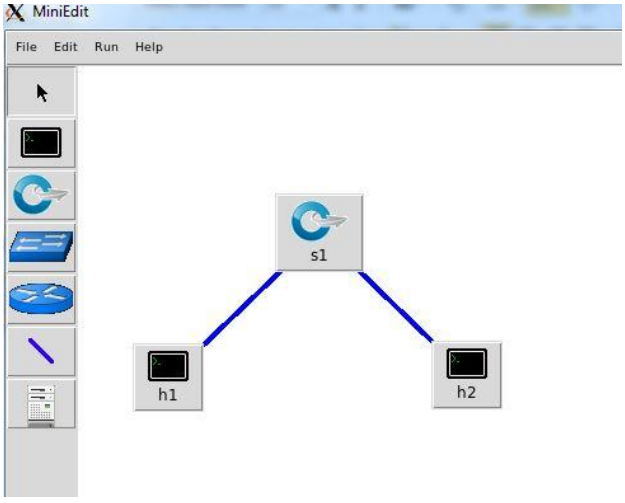
Cvičenie 2: Práca s emulačným nástrojom Mininet

1. Spustíme Windows VM.
2. Stiahneme, nainštalujeme a spustíme aplikáciu Xming zo stránky *https://sourceforge.net/projects/xming/* vo Windows VM.

3. Stiahneme si obraz Mininetu zo stránky *https://github.com/mininet/mininet/wiki/Mininet-VM-Images*.
4. Rozbalíme zip archív s oboma súbormi (jeden typ .vdmk a druhý .ovf) a pomocou súboru s koncovkou .ovf importujeme Mininet VM do VirtualBoxu. Stačí tento súbor spustiť a automaticky sa otvorí VirtualBox s ponukou importovania VM.
5. Vytvorí sa VM Mininetu, ktorú nastavíme takto:

- prestavíme systémovú pamäť (Settings ->System->Motherboard) na 2GB
- nastavíme sieťovú kartu , ktorou sa pripájame do internetu a zvolíme možnosť Bridged Adapter (Settings ->Network->Adapter 1).

6. Spustíme VM Mininetu.
7. Prihlasíme sa prihlasovacími údajmi, username: mininet, password: mininet.
8. Pomocou príkazu ifconfig zistíme, akú IP adresu nám priradil DHCP server.
9. Otvoríme aplikáciu putty vo Windows VM a v nastaveniach vyhľadáme položku (Connection->SSH->X11) a zaškrtneme Enable X11 forwarding.
10. Pripojíme sa prostredníctvom putty na IP adresu mininetu cez SSH, prihlasovacie údaje sú rovnaké ako v bode 7.
11. Vo vytvorenej relácii v putty zadáme príkaz sudo ~/mininet/examples/miniedit.py a otvorí sa nám grafické rozhranie miniedit.
12. Vytvoríme si topológiu podľa obrázka.

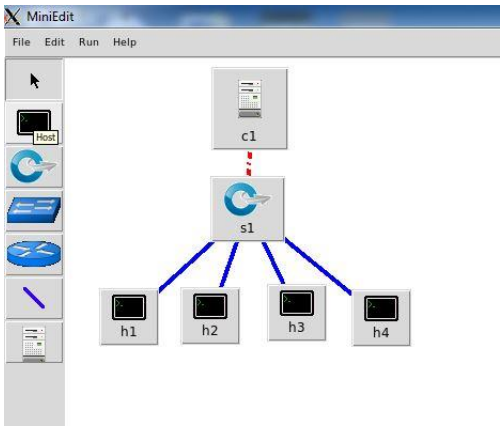


13. V Edit->Preferences zaškrtneme Start cli a IP Base nastavíme na 10.0.0.0/24.
14. Počítaču h1 nastavíme IP adresu 10.0.0.1/24 a počítaču h2 nastavíme IP adresu 10.0.0.2/24 (stlačíme pravé tlačítko myši na počítač a vyberieme properties).
15. V Run spustíme topológiu kliknutím na Run.

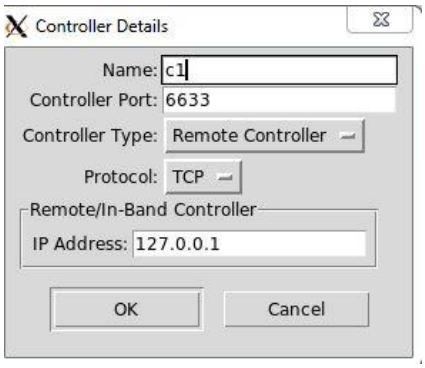
16. V putty okne sa spustí mininet CLI rozhranie. Študenti by si mali pozrieť a vyskúšať všetky možnosti tohto rozhrania cez príkaz help. V prípade nejasností si môžu pozrieť dokumentáciu na stránke *http://mininet.org/walkthrough/*.
17. Konektivita medzi virtuálnymi počítačmi h1 a h2 by nemala fungovať. Overiť konektivitu môžete cez Mininet CLI rozhranie príkazmi h1 ping h2 alebo h2 ping h1.
18. Nakonfigurované toky si môžeme pozrieť pomocou príkazu sh ovs-ofctl dump-flows s1. Výpis by mal byť prázdny.
19. Pre sfunkčnenie konektivity medzi virtuálnymi počítačmi h1 a h2 nakonfigurujeme dva toky príkazmi sh ovs-ofctl add-flow s1 "in_port=1,nw_dst=10.0.0.2,actions=output:2" a sh ovs-ofctl add-flow s1 "in_port=2,nw_dst=10.0.0.1,actions=output:1".

Cvičenie 3: Napojenie topológie v Mininete na kontrolér POX

1. Spustíme Windows VM.
2. Stiahneme, nainštalujeme a spustíme aplikáciu Xming zo stránky *https://sourceforge.net/projects/xming/* vo Windows VM.
3. Spustíme si Mininet a miniedit presne tak ako na cvičení 2.
4. Vytvoríme si topológiu podľa obrázka.



5. Kontrolér c1 nastavíme podľa obrázka. (stlačíme pravé tlačítko myši na kontrolér a vyberieme properties).



6. V Edit->Preferences zaškrtneme Start cli a IP Base nastavíme na 10.0.0.0/24.
7. Počítaču h1 nastavíme IP adresu 10.0.0.1/24, počítaču h2 10.0.1.2/24, počítaču h3 10.0.0.3/24, počítaču h4 10.0.1.4/24 (stlačíme pravé tlačítko myši na počítač a vyberieme properties).
8. V Run spustíme topológiu kliknutím na Run.
9. Otvoríme si ďalšiu ssh reláciu na mininet pomocou putty, prihlasíme sa a dostaneme sa do zložky kontroléra pox príkazom cd /home/mininet/pox.
10. Spustíme POX kontrolér, ktorý bude plniť úlohu L2 prepínača príkazom ./pox.py log.level --DEBUG forwarding.l2_learning
11. Overíme konektivitu medzi h1 a h3, h2 a h4 príkazmi h1 ping h3 a h2 ping h4.
12. Pozrieme si toky pomocou príkazu dpctl dump-flows.
13. Na odchytenie komunikácie môžeme použiť Wireshark. Pre použitie Wiresharku si potrebujeme otvoriť novú SSH reláciu v putty s Mininet VM. Ako obyčajne musíme povoliť X11 forwarding. Prihlásime sa a zadáme príkaz sudo wireshark, ktorý nám otvorí okno Wiresharku.
14. Vo Wiresharku odchytíme komunikáciu na rozhraní lo0 a do filtra zadáme of. Teraz by sme mali vidieť OpenFlow 1.0 správy, ktoré prebiehajú medzi prepínačom a kontrolérom.
15. Zastavíme kontrolér POX príkazom CTRL+C a spustíme POX tentokrát ako L3 smerovač príkazom ./pox.py forwarding.l3_learning --fakeways=10.0.0.254, 10.0.1.254, kde 10.0.0.254 je predvolená brána pre počítače zo siete 10.0.0.0/24 a 10.0.1.254 je predvolená brána pre počítače zo siete 10.0.1.0/24.
16. Nastavíme predvolené brány smerovačom h1 až h4 v Mininet CLI rozhraní príkazmi:
    - h1 route add default gw 10.0.0.254
    - h2 route add default gw 10.0.1.254
    - h3 route add default gw 10.0.0.254

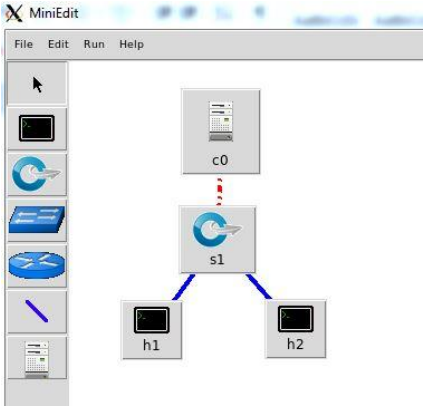
- h4 route add default gw 10.0.1.254

17. Overíme konektivitu medzi h1 a h4, h2 a h3, h1 a h2.
18. Pozrieme si toky pomocou príkazu dpctl dump-flows.
19. Príkazom sh ovs-vsctl set Interface s1-eth1 ingress_policing_rate=1000 nastavíme portu s1-eth1 maximálnu povolenú rýchlosť na 1000 kb/s.
20. Príkazom sh ovs-vsctl set Interface s1-eth1 ingress_policing_burst=100 nastavíme portu s1-eth1 maximálne množstvo dát, ktoré môže pretiecť portom nad rámec povolenej rýchlosti.
21. Otvoríme si okno miniedit a otvoríme si terminál počítačov h1 a h3 (pravé tlačítko myši na počítač a klikneme na terminal).
22. Funkčnosť QoS politiky overíme pomocou programu iperf. V termináli h3 zapneme iperf server príkazom iperf –s a v termináli h1 zapneme klienta príkazom iperf –c 10.0.0.3. Zhodnotíme výsledky meraní.

Cvičenie 4: Napojenie topológie v Mininete na kontrolér Floodlight

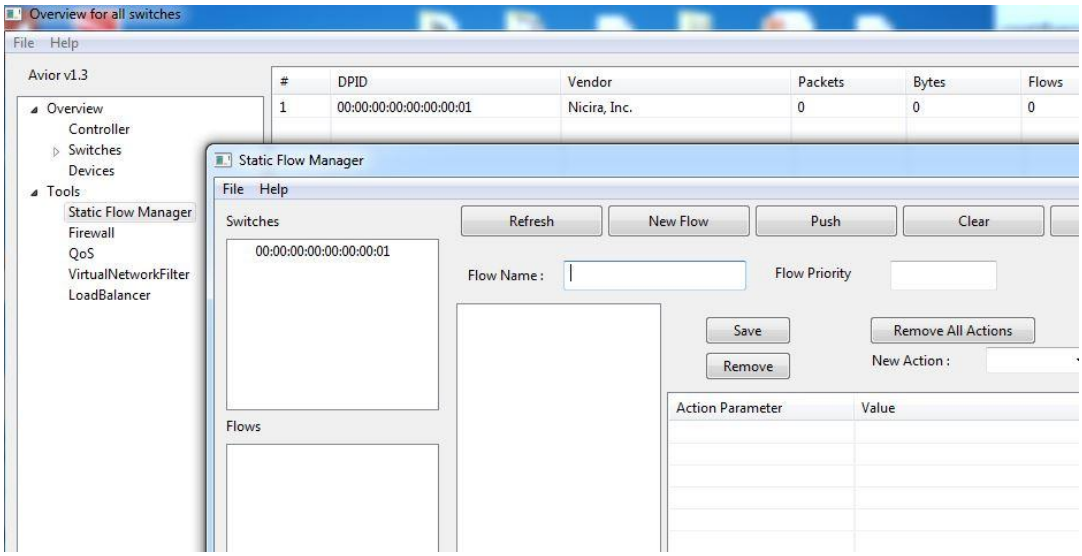
1. Spustíme Windows VM.
2. Stiahneme, nainštalujeme a spustíme aplikáciu Xming zo stránky *https://sourceforge.net/projects/xming/* vo Windows VM.
3. Spustíme si Mininet a miniedit.
4. Spustíme si Ubuntu VM, ktorú sme vytvorili na prvom cvičení.
5. Do Ubuntu nainštalujeme potrebné súčasti pre kontrolér Floodlight príkazom sudo apt-get install build-essential default-jdk ant python-dev eclipse.
6. Stiahneme Floodlight verziu 0.90 a skompilujeme príkazmi:
   - wget https://github.com/floodlight/floodlight/archive/v0.90.tar.gz
   - tar xvf v0.90.tar.gz
   - cd floodlight-0.90
   - ant
   - sudo mkdir /var/lib/floodlight
   - sudo chmod 777 /var/lib/floodlight
7. Príkazom ifconfig si pozrieme a zapamätáme IP adresu Ubuntu VM, ktorú nám priradil DHCP server.
8. Spustíme Floodlight príkazom java -jar target/floodlight.jar.

9. Cez ľubovoľný prehliadač sa pripojíme na webové rozhranie Floodlightu pomocou adresy http://[ip adresa Ubuntu VM]:8080/ui/index.html
10. V Mininete cez mininedit si vytvoríme topológiu podľa obrázka.



11. Kontrolér c0 nastavíme na typ Remote Controller a IP adresu na adresu Ubuntu VM. (stlačíme pravé tlačítko myši na kontrolér a vyberieme properties).
12. V Edit->Preferences zaškrtneme Start cli a IP Base nastavíme na 10.0.0.0/24.
13. Počítaču h1 nastavíme IP adresu 10.0.0.1/24 a počítaču h2 nastavíme IP adresu 10.0.0.2/24 (stlačíme pravé tlačítko myši na počítač a vyberieme properties).
14. V Run spustíme topológiu kliknutím na Run.
15. V prehliadači vo webovom rozhraní kontroléra Floodlight by sme mali vidieť prepínač s1. Keď spravíme ping z h1 na h2, mali by sme vidieť aj virtuálne počítače.
16. Na odchytenie komunikácie zo strany Mininetu môžeme použiť Wireshark alebo tcpdump. Pre použitie Wiresharku si potrebujeme otvoriť novú SSH reláciu v putty s Mininet VM. Ako obyčajne musíme povoliť X11 forwarding. Prihlásime sa a zadáme príkaz sudo wireshark, ktorý nám otvorí okno Wiresharku.
17. Vo Wiresharku odchytíme komunikáciu na rozhraní eth0 a do filtra zadáme of. Teraz by sme mali vidieť OpenFlow 1.0 správy.
18. Na odchytenie komunikácie zo strany Ubuntu môžeme použiť aplikáciu tcpdump príkazom tcpdump –i eth0 –w capture.pcap. Musíme si však otvoriť ďalšiu tty reláciu pre Ubuntu vo VirtualBoxe pomocou klávesovej skratky pravé CTRL + F2.
19. Na získanie súboru capture.pcap potrebujeme v zložke s týmto súborom spustiť jednoduchý http server príkazom python -m SimpleHTTPServer. Na tento server sa pripojíme cez prehliadač adresou http://[ip adresa ubuntu]:8000/, stiahneme súbor capture.pcap a otvoríme vo Wiresharku. Do filtra zadáme openflow. Teraz by sme mali vidieť OpenFlow 1.0 správy.

20. Stiahneme a použijeme aplikáciu Avior 1.3 vo Windows VM zo stránky *http://openflow.marist.edu/avior*. Pre spustenie aplikácie Avior, musíme mať nainštalované prostredie Java Runtime Enviroment (JRE). (*http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html*).
21. Spustíme Avior a zadáme IP adresu kontroléra (Ubuntu VM).
22. Po pripojení na kontrolér, otvoríme Static Flow Manager.

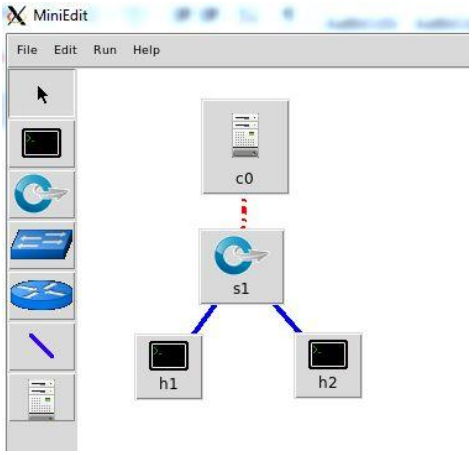


23. Našou úlohou bude cez toto rozhranie znemožniť prechod ICMP paketov z h1 na h2. Moje riešenie je nasledujúce. Klikneme na prepínač, ktorý máme v kolonke Switches. Klikneme na pridanie novej položky New Flow. V tabuľke úplne vpravo vyberieme Match Parameter vyplnením Network Source na 10.0.0.1 a Network Protocol na 0x01. Zvolíme New Action na output a parameter akcie Port vyplníme na 1. Ak máme tok nakonfigurovaný vložíme ho do tabuľky prepínačov možnosťou Push. V tomto momente by mal byť tok pridaný do prepínača a ping medzi h1 a h2 by nemal ísť.

Cvičenie 5: Napojenie topológie v Mininete na kontrolér OpenMUL

1. Spustíme Windows VM.
2. Stiahneme, nainštalujeme a spustíme aplikáciu Xming zo stránky *https://sourceforge.net/projects/xming/* vo Windows VM.
3. Spustíme si Mininet a miniedit.
4. Spustíme si Ubuntu VM, ktorú sme vytvorili na prvom cvičení.

5. V Ubuntu nainštalujeme revízny systém Git príkazom sudo apt-get install git.
6. Stiahneme a skompilujeme OpenMUL kontrolér:
   - git clone https://github.com/openmul/openmul.git
   - cd openmul
   - ./build.sh
7. Spustíme OpenMUL príkazom ./mul.sh start l2switch
8. Príkazom ifconfig si pozrieme a zapamätáme IP adresu Ubuntu VM, ktorú nám priradil DHCP server.
9. Pripojíme sa na CLI rozhranie kontroléra OpenMUL pomocou putty cez adresu Ubuntu VM na port 10000. Ovládanie rozhrania je zhrnuté v dokumente na adrese *https://github.com/openmul/openmul/blob/master/docs/openmul-cli-guide.pdf*.
10. V Mininete cez mininedit si vytvoríme topológiu podľa obrázka.



11. Kontrolér c0 nastavíme na typ Remote Controller, IP adresu na adresu Ubuntu VM a port na 6653. (stlačíme pravé tlačítko myši na kontrolér a vyberieme properties).
12. V Edit->Preferences zaškrtneme Start cli, OpenFlow 1.3 a IP Base nastavíme na 10.0.0.0/24.
13. Počítaču h1 nastavíme IP adresu 10.0.0.1/24 a počítaču h2 nastavíme IP adresu 10.0.0.2/24 (stlačíme pravé tlačítko myši na počítač a vyberieme properties).
14. V Run spustíme topológiu kliknutím na Run.
15. Na odchytenie komunikácie zo strany Mininetu môžeme použiť Wireshark alebo tcpdump. Pre použitie Wiresharku si potrebujeme otvoriť novú SSH reláciu v putty s Mininet VM. Ako obyčajne musíme povoliť X11 forwarding. Prihlasíme sa a zadáme príkaz sudo wireshark, ktorý nám otvorí okno Wiresharku.

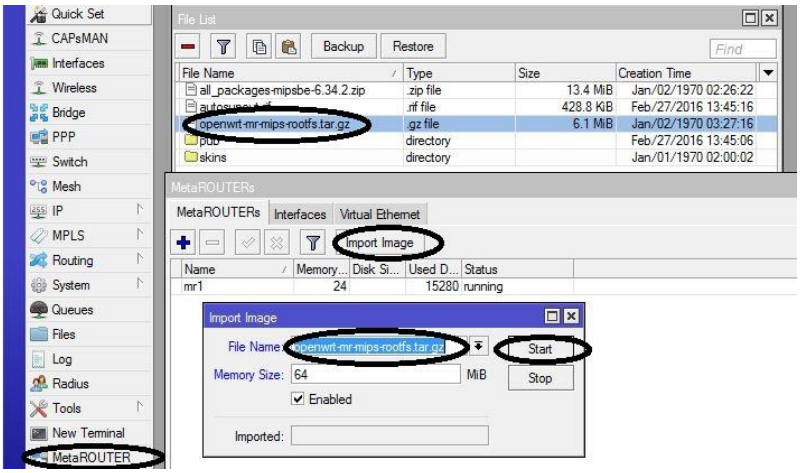
16. Vo Wiresharku odchytíme komunikáciu na rozhraní eth0 a do filtra zadáme of. Teraz by sme mali vidieť OpenFlow 1.3 správy.
17. Na odchytenie komunikácie zo strany Ubuntu môžeme použiť aplikáciu tcpdump príkazom sudo tcpdump –i eth0 –w capture.pcap. Musíme si však otvoriť ďalšiu tty reláciu pre Ubuntu vo VirtualBoxe pomocou skratky pravé CTRL + F2.
18. Na získanie súboru capture.pcap potrebujeme v zložke s týmto súborom spustiť jednoduchý http server príkazom python -m SimpleHTTPServer. Na tento server sa pripojíme cez prehliadač adresou http://[ip adresa ubuntu]:8000/. Stiahneme súbor capture.pcap a otvoríme vo Wiresharku. Do filtra zadáme openflow_v4. Teraz by sme mali vidieť OpenFlow 1.3 správy.
19. Teraz máme na výber ako toky medzi počítačmi h1 a h2 ovplyvníme. Ja som si zvolil klasickú možnosť zakázať ICMP pakety z h1 na h2. Postup je nasledujúci. Otvoríme si OpenMUL CLI rozhranie a pozrieme si DPID prepínača pomocou príkazu sh of-switch all. V konfiguračnom móde zadáme:
    - mul-conf
    - of-flow add switch [DPID prepínača] smac * dmac * eth-type 0x0800 vid * vlan-pcp * mpls-label * mpls-tc * mpls-bos * dip * sip 10.0.0.1/32 proto 1 tos * dport * sport * in-port 1 table 0
    - instruction write
    - action-add drop
    - action-list end
    - commit
20. Overíme, či prechádzajú ICMP pakety z h1 na h2 nástrojom ping.

Cvičenie 6: Mikrotik zariadenie a kontrolér OpenMUL

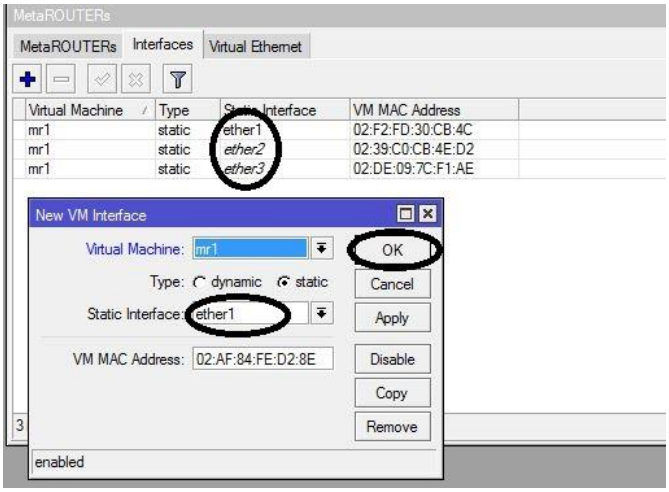
1. Spustíme Windows VM.
2. Stiahneme a spustíme aplikáciu winbox (*http://www.mikrotik.com/download*), ktorou sa budeme pripájať na Mikrotik smerovač.
3. Vytvoria sa dvojice. Každá dvojica bude mať k dispozícii jeden Mikrotik smerovač. Študenti si zapoja svoje počítače do portu 2 a 3 na Mikrotik smerovači a port 1 pripoja do lokálnej siete katedry. Jeden z dvojice si nastaví na Cisco sieťovej karte IP adresu

10.0.0.1/24 a druhý IP adresu 10.0.0.2/24. Sieťovú kartu na pripojenie do internetu dočasne obaja vypnú.

4. Pomocou aplikácie winbox sa pripojíme na Mikrotik cez MAC adresu zariadenia.
5. V smerovači by mal byť nahratý súbor *openwrt-mr-mips-rootfs.tar.gz*, ktorý je priložený na DVD k mojej diplomovej práci. Ak nie, určite bude súbor sprístupnený študentom na nahratie do zariadenia.
6. Tento súbor importujeme do funkcie Metarouter Mikrotik zariadenia podľa obrázka.



7. K vytvorenej virtuálnej inštancii systému OpenWrt priradíme rozhrania smerovača ether1, ether2 a ether3.



8. Otvoríme si konzolu systému OpenWrt a prepíšeme súbor /etc/config/network do tejto podoby pomocou príkazu joe /etc/config/network:

config interface loopback
option ifname lo
option proto static

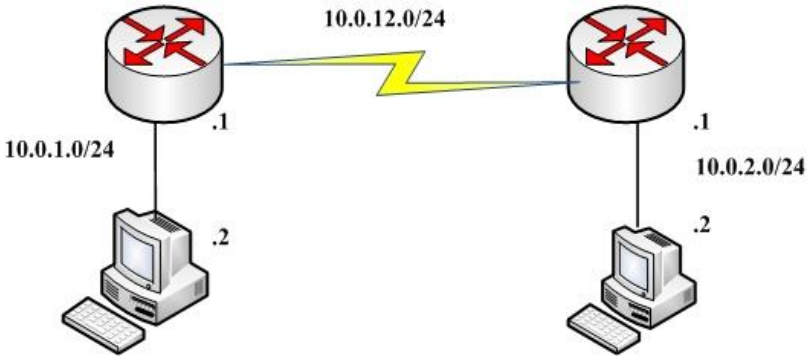
option ipaddr 127.0.0.1
option netmask 255.0.0.0

config interface lan
option ifname eth0
option proto dhcp

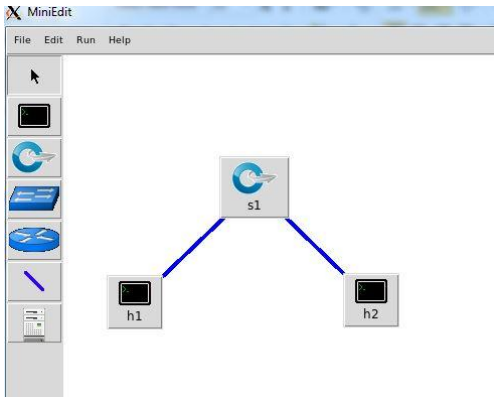
9. Reštartujeme OpenWrt príkazom reboot. Po reštarte by sme mali dostať IP adresu na rozhranie eth0 od DHCP servera.
10. Občas sa konzolové okno v Metaroutri správa zvláštne. Preto je vhodné pripojiť sa na OpenWrt prostredníctvom IP adresy rozhrania eth0 cez telnet. Musíme však na počítačoch povoliť sieťovú kartu do internetu.
11. V OpenWrt zapneme rozhrania eth1 a eth2 príkazmi ifconfig eth1 up a ifconfig eth2 up.
12. Vytvoríme nové bridge rozhranie pomocou príkazu ovs-vsctl add-br ovsbr. Overíme, či rozhranie je funkčné príkazmi ovs-vsctl show a ovs-dpctl show. Ak aplikácia Open vSwitch padne, stačí ju reštartovať príkazom /etc/init.d/openvswitch restart.
13. Pridáme do bridge rozhrania rozhrania eth1 a eth2 príkazmi
    - ovs-vsctl add-port ovsbr eth1
    - ovs-vsctl add-port ovsbr eth2
14. Spustíme si Ubuntu VM a dostaneme sa do zložky s kontrolérom OpenMUL.
15. Spustíme OpenMUL príkazom ./mul.sh start l2switch
16. Príkazom ifconfig si pozrieme a zapamätáme IP adresu Ubuntu VM, ktorú nám priradil DHCP server.
17. Pripojíme sa na CLI rozhranie kontroléra OpenMUL pomocou putty cez adresu Ubuntu VM na port 10000. Ovládanie rozhrania je zhrnuté v dokumente na adrese *https://github.com/openmul/openmul/blob/master/docs/openmul-cli-guide.pdf*.
18. Napojíme Open vSwitch bežiaci v OpenWrt na kontrolér OpenMul príkazom ovs-vsctl set-controller ovsbr tcp:[ip adresa Ubuntu]:6653.
19. Príkazom ovs-vsctl show skontrolujeme, či vo výpise sa nachádza *Controller is_connected: true.*
20. V CLI rozhraní OpenMUL konfigurujeme toky medzi počítačmi študentov podľa vlastných potrieb, napríklad tým istým spôsobom ako v predošlom cvičení.

Cvičenie 7: Prepojenie dvoch Mininet topológií cez GRE tunel

1. Študenti budú robiť po dvojiciach. Zapoja si topológiu, ako je znázornená na obrázku. Smerovače nakonfigurujú tak, aby bola možná konektivita medzi počítačmi.



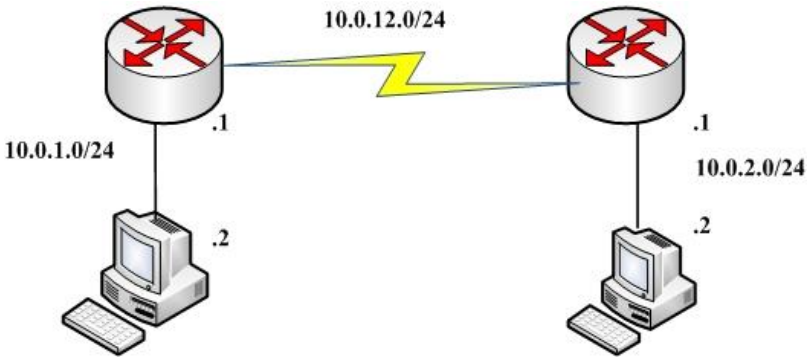
2. Spustíme Windows VM.
3. Stiahneme, nainštalujeme a spustíme aplikáciu Xming zo stránky *https://sourceforge.net/projects/xming/* vo Windows VM.
4. Zmeníme typ sieťovej karty (bridged adapter) pre Mininet VM vo VirtualBoxe zo sieťovej karty, pomocou ktorej sa pripájame na internet, na Cisco sieťovú kartu.
5. Spustíme Mininet.
6. Prvý z dvojice si nakonfiguruje rozhranie eth0 a predvolenú bránu nasledovne:
   - ifconfig eth0 10.0.1.2 netmask 255.255.255.0 up
   - route add default gw 10.0.1.1 eth0
7. Druhý z dvojice si nakonfiguruje rozhranie eth0 a predvolenú bránu nasledovne:
   - ifconfig eth0 10.0.2.2 netmask 255.255.255.0 up
   - route add default gw 10.0.2.1 eth0
8. V Mininete cez mininedit si vytvoríme topológiu podľa obrázka.



9. V Edit->Preferences zaškrtneme Start cli, IP Base nastavíme na 10.0.0.0/24.
10. Jeden z dvojice nastaví počítaču h1 IP adresu 10.0.0.1/24 a počítaču h2 10.0.0.2/24. Druhý z dvojice nastaví počítaču h1 IP adresu 10.0.0.3/24 a počítaču h2 10.0.0.4/24 (stlačíme pravé tlačítko myši na počítač a vyberieme properties).
11. V Run spustíme topológiu kliknutím na Run.
12. V CLI rozhraní Minetu každý z dvojice overí konektivitu s druhou stranou pomocou príkazu sh ping [IP adresa rozhrania eth0 druhého počítača].
13. Každý z dvojice vytvorí GRE tunel pomocou príkazu sh ovs-vsctl add-port s1 gre -- set interface gre type=gre options:remote_ip=[IP adresa rozhrania eth0 druhého počítača].
14. Ak sa tunel vytvoril mala by fungovať konektivita medzi h1 a h2 počítačmi jedného študenta s h1 a h2 počítačmi druhého študenta.
15. Na odchytenie komunikácie môžeme použiť Wireshark. Pre použitie Wiresharku si potrebujeme otvoriť novú SSH reláciu v putty s Mininet VM. Ako obyčajne musíme povoliť X11 forwarding. Prihlásime sa a zadáme príkaz sudo wireshark, ktorý nám otvorí okno Wiresharku.
16. Vo Wiresharku odchytíme komunikáciu na rozhraní eth0. V komunikácii by sme mali vidieť GRE pakety.

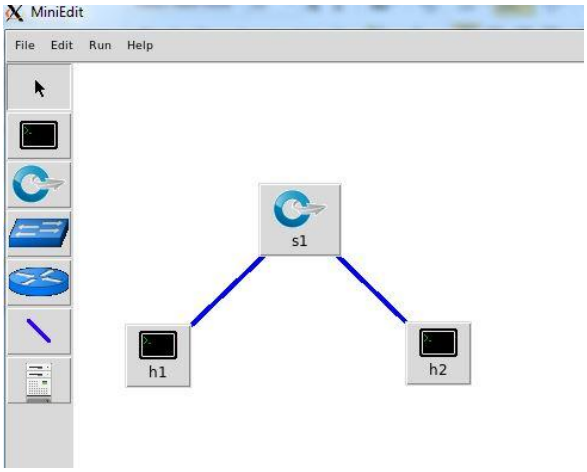
Cvičenie 8: Prepojenie dvoch Mininet topológií cez VXLAN tunel

1. Študenti budú robiť po dvojiciach. Zapoja si topológiu, ako je znázornená na obrázku. Smerovače nakonfigurujú tak, aby bola možná konektivita medzi počítačmi.



2. Spustíme Windows VM.
3. Stiahneme, nainštalujeme a spustíme aplikáciu Xming zo stránky *https://sourceforge.net/projects/xming/* vo Windows VM.

4. Zmeníme typ sieťovej karty (bridged adapter) pre Mininet VM vo VirtualBoxe zo sieťovej karty, pomocou ktorej sa pripájame na internet, na Cisco sieťovú kartu.
5. Spustíme Mininet.
6. Prvý z dvojice si nakonfiguruje rozhranie eth0 a predvolenú bránu nasledovne:
   - ifconfig eth0 10.0.1.2 netmask 255.255.255.0 up
   - route add default gw 10.0.1.1 eth0
7. Druhý z dvojice si nakonfiguruje rozhranie eth0 a predvolenú bránu nasledovne:
   - ifconfig eth0 10.0.2.2 netmask 255.255.255.0 up
   - route add default gw 10.0.2.1 eth0
8. Spustíme si miniedit.
9. V Mininete cez mininedit si vytvoríme topológiu podľa obrázka.



10. V Edit->Preferences zaškrtneme Start cli, IP Base nastavíme na 10.0.0.0/24.
11. Jeden z dvojice nastaví počítaču h1 IP adresu 10.0.0.1/24 a počítaču h2 rovnakú IP adresu 10.0.0.1/24. Druhý z dvojice nastaví počítaču h1 IP adresu 10.0.0.2/24 a počítaču h2 rovnakú IP adresu 10.0.0.2/24.
12. V Run spustíme topológiu kliknutím na Run.
13. Jeden z dvojice v CLI rozhraní Minetu nastaví rovnakú MAC adresu počítačom h1 a h2 príkazmi:
    - h1 ifconfig h1-eth0 hw ether 00:00:00:00:aa:01
    - h2 ifconfig h1-eth0 hw ether 00:00:00:00:aa:01
14. Druhý z dvojice v CLI rozhraní Minetu nastaví rovnakú MAC adresu počítačom h1 a h2 príkami
    - h1 ifconfig h1-eth0 hw ether 00:00:00:00:aa:02
    - h2 ifconfig h1-eth0 hw ether 00:00:00:00:aa:02

15. Každý z dvojice vytvorí VXLAN tunel pomocou príkazu sh ovs-vsctl add-port s1 vxlan -- set interface vxlan type=vxlan option:remote_ip=[IP adresa rozhrania eth0 druhého počítača] option:key=flow ofport_request=10
16. Jeden z dvojice si vytvorí súbor príkazom sh nano flows.txt, ktorý bude obsahovať následujúce toky:

table=0,in_port=1,actions=set_field:100->tun_id,resubmit(,1)
table=0,in_port=2,actions=set_field:200->tun_id,resubmit(,1)
table=0,actions=resubmit(,1)

table=1,tun_id=100,dl_dst=00:00:00:00:aa:01,actions=output:1
table=1,tun_id=200,dl_dst=00:00:00:00:aa:01,actions=output:2
table=1,tun_id=100,dl_dst=00:00:00:00:aa:02,actions=output:10
table=1,tun_id=200,dl_dst=00:00:00:00:aa:02,actions=output:10

table=1,tun_id=100,arp,nw_dst=10.0.0.1,actions=output:1
table=1,tun_id=200,arp,nw_dst=10.0.0.1,actions=output:2
table=1,tun_id=100,arp,nw_dst=10.0.0.2,actions=output:10
table=1,tun_id=200,arp,nw_dst=10.0.0.2,actions=output:10
table=1,priority=100,actions=drop

17. Druhý z dvojice si vytvorí súbor príkazom sh nano flows.txt, ktorý bude obsahovať následujúce toky:

table=0,in_port=1,actions=set_field:100->tun_id,resubmit(,1)
table=0,in_port=2,actions=set_field:200->tun_id,resubmit(,1)
table=0,actions=resubmit(,1)

table=1,tun_id=100,dl_dst=00:00:00:00:aa:02,actions=output:1
table=1,tun_id=200,dl_dst=00:00:00:00:aa:02,actions=output:2
table=1,tun_id=100,dl_dst=00:00:00:00:aa:01,actions=output:10
table=1,tun_id=200,dl_dst=00:00:00:00:aa:01,actions=output:10

table=1,tun_id=100,arp,nw_dst=10.0.0.2,actions=output:1

table=1,tun_id=200,arp,nw_dst=10.0.0.2,actions=output:2
table=1,tun_id=100,arp,nw_dst=10.0.0.1,actions=output:10
table=1,tun_id=200,arp,nw_dst=10.0.0.1,actions=output:10
table=1,priority=100,actions=drop

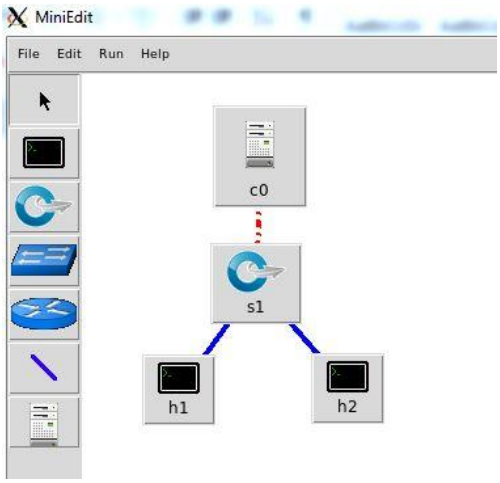
18. Obaja aplikujú toky príkazom sh ovs-ofctl add-flows s1 flows.txt.
19. Ak sa toky aplikovali správne mala by fungovať konektivita medzi h1 počítačom prvého študenta s h1 počítačom druhého študenta, takisto aj medzi h2 počítačom prvého študenta s h2 počítačom druhého študenta. Dôležité je si uvedomiť, že prebiehajú dve rozdielne komunikácie odlíšené cez VXLAN idendifikátor VNI.
20. Na odchytenie komunikácie môžeme použiť Wireshark. Pre použitie Wiresharku si potrebujeme otvoriť novú SSH reláciu v putty s Mininet VM. Ako obyčajne musíme povoliť X11 forwarding. Prihlásime sa a zadáme príkaz sudo wireshark, ktorý nám otvorí okno Wiresharku.
21. Vo Wiresharku odchytíme komunikáciu na rozhraní eth0. V komunikácii by sme mali vidieť UDP pakety. Tieto UDP pakety treba dekódovať ako VXLAN pakety.

Cvičenie 9: Zabezpečenie komunikácie prepínač-kontrolér pomocou SSL

1. Spustíme Windows VM.
2. Stiahneme, nainštalujeme a spustíme aplikáciu Xming zo stránky *https://sourceforge.net/projects/xming/* vo Windows VM.
3. Zmeníme typ sieťovej karty (bridged adapter) pre Mininet VM vo VirtualBoxe z Cisco sieťovej karty na sieťovú kartu, pomocou ktorej sa pripájame na internet.
4. Spustíme Mininet VM.
5. Vygenerujeme certifikáty pre prepínač aj pre kontrolér príkazmi:
   - cd /etc/openvswitch
   - sudo ovs-pki req+sign ctl controller
   - sudo ovs-pki req+sign sc switch
6. Nastavíme ssl parametre pre Open vSwitch príkazom:
   - sudo ovs-vsctl set-ssl \
     /etc/openvswitch/sc-privkey.pem \
     /etc/openvswitch/sc-cert.pem \

/var/lib/openvswitch/pki/controllerca/cacert.pem

7. Spustíme kontrolér ovs-controller s podporou SSL príkazom
   - sudo ovs-controller -v pssl:6633 \
     -p /etc/openvswitch/ctl-privkey.pem \
     -c /etc/openvswitch/ctl-cert.pem \
     -C /var/lib/openvswitch/pki/switchca/cacert.pem
8. Otvoríme si novú tty reláciu vo VirtualBoxe stlačením klávesovej skratky pravé CTRL + F2.
9. Spustíme si miniedit.
10. V mininedite si vytvoríme topológiu podľa obrázka



11. Kontrolér c0 nastavíme na typ Remote Controller a protokol nastavíme na SSL (stlačíme pravé tlačítko myši na kontrolér a vyberieme properties).
12. V Edit->Preferences zaškrtneme Start cli a IP Base nastavíme na 10.0.0.0/24.
13. Počítaču h1 nastavíme IP adresu 10.0.0.1/24 a počítaču h2 nastavíme IP adresu 10.0.0.2/24 (stlačíme pravé tlačítko myši na počítač a vyberieme properties).
14. V Run spustíme topológiu kliknutím na Run.
15. Pomocou príkazu sh ovs-vsctl show skontrolujeme, či sme pripojený na kontrolér.
16. Na odchytenie komunikácie použijeme Wireshark. Pre použitie Wiresharku si potrebujeme otvoriť novú SSH reláciu v putty s Mininet VM. Ako obyčajne musíme povoliť X11 forwarding. Prihlásime sa a zadáme príkaz sudo wireshark, ktorý nám otvorí okno Wiresharku.
17. Vo Wiresharku odchytíme komunikáciu na rozhraní lo. Mali by sme vidieť šifrovanú komunikáciu.

Príloha B: Obsah DVD

Priložené DVD obsahuje:

- Prácu v elektronickej podobe (formát PDF)
- Skompilovaný systém OpenWrt pre platformu Metarouter
- Vypracované teoretické témy pre prednášky (formát DOCX)