VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH ROBUSTNÍ PODNIKOVÉ DATOVÉ SÍTĚ S VYUŽITÍM VIRTUÁLNÍCH SÍTÍ

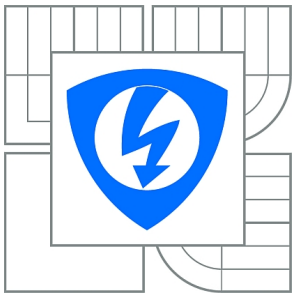
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE PAVOL VRABLIC
AUTHOR

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH ROBUSTNÍ PODNIKOVÉ DATOVÉ SÍTĚ S VYUŽITÍM VIRTUÁLNÍCH SÍTÍ

DESIGN OF RELIABLE VLAN-BASED ENTERPRISE DATA NEWORK

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

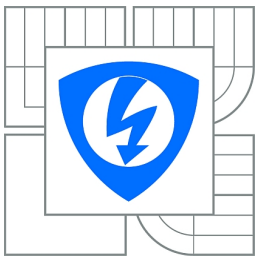
AUTOR PRÁCE — PAVOL VRABLIC

AUTHOR

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

SUPERVISOR

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Pavol Vrablic	***ID:***	154668
Ročník:	3	***Akademický rok:***	2014/2015

NÁZEV TÉMATU:

Návrh robustní podnikové datové sítě s využitím virtuálních sítí

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s problematikou současného přístupu k návrhu podnikových datových sítí. Zaměřte se jak na problematiku robustnosti architektury, podpory QoS, tak i na využití techniky virtualizace sítí a serverových služeb. Dle dostupného vybavení navrhněte laboratorní úlohu do předmětu Architektura sítí.

DOPORUČENÁ LITERATURA:

[1] SEIFERT, Rich. The switch book: the complete guide to LAN switching technology. New York: John Wiley, 2000, 698 s. ISBN 04-713-4586-5.
[2] MARCHESE, Mario. QoS over heterogeneous networks: the complete guide to LAN switching technology. Chichester: John Wiley, 2007, 307 s. ISBN 978-0-470-01752-4.

Termín zadání:	9.2.2015	***Termín odevzdání:***	2.6.2015

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

doc. Ing. Jiří Mišurec, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Cieľom tejto práce je oboznámiť sa s problematikou návrhu robustnej siete, siete ktorá musí byť vysoko spoľahlivá a odolná voči poruchám funkčnosti kritických služieb. Táto práca obsahuje popis nových metód vetvenia siete a virtualizácie ktorá je čoraz viac využívaná a nasadzovaná v podnikových sieťach. Ďalej je spomenutý clustering, ktoré zaisťuje funkčnosť kritických služieb aj pri čiastočných výpadkoch siete a QoS, ktoré potrebujeme využívať na dosiahnutie vysokej kvality týchto služieb.

KĽÚČOVÉ SLOVÁ

Robustnosť siete, VLAN, STP, SPB, TRILL, virtualizácia, kvalita služieb, Cluster, Cloud, SDN.

ABSTRACT

The main aim of this work is to identify the potential issues within a robust network to provide a highly reliable and fault free service – with a smooth and reliable functionality of critical services. This work contains new methods of branch networking and virtualization that is increasingly used in enterprise networks. It is important to note that computer clustering provides the functionality of critical services that can surpass potential partial failures and network QoS; which overall, achieves a high quality and reliable service.

KEYWORDS

Robustness network, VLAN, STP, SPB, TRILL, virtualization, Quality of Service, Cluster, Cloud, SDN.

VRABLIC, P. *Návrh robustní podnikové datové sítě s využitím virtuálních sítí.*: bakalárska práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 57 s. Vedúci práce bol doc. Ing. Vít Novotný, Ph.D.

PREHLÁSENIE

Prehlasujem, že som svoju bakalársku prácu na tému „Návrh robustní podnikové datové sítě s využitím virtuálních sítí.“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorském, o právoch súvisejúcich s právom autorským a o zmeně niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhé, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno ....

(podpis autora)

OBSAH

ZOZNAM OBRÁZKOV

ÚVOD

Počítačová sieť sa stala v posledných rokoch základom fungovania každej menšej či väčšej spoločnosti. S čoraz väčšou orientáciou firiem na informačné technológie narastajú aj nároky na rýchlosť, spoľahlivosť, bezpečnosť a kvalitu siete. Popri tom ako sa svet sieti stále vyvíja po výkonnostnej stránke, vznikajú aj nové metódy ako zvýšiť efektivitu a spoľahlivosť stávajúcich sieti. Každým rokom narastajú nároky na veľkosť dátového prenosu ako aj smerom do internetu tak aj v rámci lokálnej siete. Kľúčovými pre rýchly beh aplikácií server-klient je optimalizovanie sieti a zvýšenie ich efektivity. V pomalších uzloch siete treba zabezpečiť určitú kvalitu služieb, aby sme mohli zároveň využívať VoIP, kritické aplikácie či napr. zálohovanie. Pri väčších sietiach je výhodné uvažovať nad používaním virtuálizácie služieb a siete samotnej. Pomocou virtualizácie sa totiž zjednoduší administrácia a konfigurácia a sieť sa stáva v jej prostredí homogénnou. Táto práca stručne zhrnie základy potrebné na pochopenie stávajúcich a nových technológií a pokúsi sa preskúmať nové možnosti ktoré nám pomôžu navrhnúť kvalitnú dátovú sieť.

1 DIZAJN SIETE

1.1 Sieť bez hierarchie

Sieť pracujúca na vrstve 1, ktorá pozostáva z rozbočovačov, sa vyznačuje tým že je ju jednoduché a málo nákladné postaviť. Má však veľké kolízne domény, veľké všesmerové domény, nízku bezpečnosť, je ťažké vyhľadať problém, je kladená vyššia záťaž na zariadenia a vzniká vyššie oneskorenie prenosu.

Sieť pracujúca na vrstve 2, ktorá pozostáva z prepínačov, už nezdieľa prenosovú kapacitu a každá stanica má svoju vlastnú. Dochádza k zmenšeniu kolíznych domén ale zostávajú veľké všesmerové domény ktoré sme schopný rozdeliť len pomocou VLAN sieti.[1]

1.2 Hierarchický model

Zavedením smerovačov je možné rozdeliť sieť na rôzne podsiete a vytvoriť tak segmentáciu siete. Zmenšíme tak všesmerové domény a sme schopní riadiť prevádzku. Počet portov na smerovačoch však býva nízky a cena na 1 port je vysoká v porovnaní s prepínačmi. V hierarchickom modeli je vhodné rozdeliť zariadenia podľa funkcie, ktorú majú vykonávať.

Vrstva jadra siete Patria sem zariadenia ktoré tvoria chrbticu siete a spájajú ju s internetom. Musia zvládať veľké objemy dát s vysokou rýchlosťou spracovávania, agregovať dáta do distribučných prepínačov, zariaďovať vysokú dostupnosť a redundanciu.

Distribučná vrstva Sem patria zariadenia ktoré prepájajú prístupové zariadenia s jadrom siete. Tieto musia vedieť agregovať dáta, robiť segmentáciu a smerovanie, riadiť tok dát a využívať ACL. Taktiež musí byť vysoko-rýchlostná a s prvkami redundancie.

Prístupová vrstva Obsahuje zariadenia ktoré slúžia na pripájanie koncových staníc k sieti. Patria sem rozbočovače, prepínače, WiFi Access pointy.[1]

2 SIEŤOVÉ PRVKY

Aktívne sieťové prvky sú najdôležitejšou časťou siete. V minulosti sa používali, rozbočovače (HUB), mosty (Bridge), opakovače (Repeater), dnes sú to zvyčajne len smerovače (Router) a rôzne druhy prepínačov (switch). Rozdiely medzi prepínačom a smerovačom sa začali zmenšovať príchodom viacvrstvových prepínačov. Tie majú častokrát všetky funkcie smerovača. Definovať medzi nimi rozdielnosť je čoraz ťahšie, preto sa častokrát objavuje názov (Routing switch) čo je v podstate prepínač pracujúci na vrstve 2 aj 3 (L3 switch). Základný rozdiel je vo výkone. Smerovač vykonáva úlohu na základe nejakého software, potrebuje teda väčší výkon procesoru. Prepínač dokáže vykonávať viac operácií hardvérovo a preto býva rýchlejší. Väčšinou však nevykonáva všetky funkcie hardvérovo ale má tiež špecializovaný software.

2.1 Smerovač (Router)

Smerovač je zariadenie ktoré spája 2 a viac rôznych sieti a pracuje na 3. vrstve ISO/OSI modelu. Jeho primárnou úlohou je preposielanie paketov správnym smerom do jeho cieľa, k čomu využíva smerovacie tabuľky a smerovacie protokoly. Smerovač rozdeľuje kolízne domény, blokuje všesmerové vysielanie, vďaka tomu že pracuje na sieťovej vrstve dokáže vykonávať firewall. Má vyššiu inteligenciu ako prepínač a dokáže používať rôzne smerovacie protokoly a mechanizmy, riadiť dátový tok rôznymi pravidlami. Smerovače delia siete podľa sieťových adries a nie podľa fyzických. Sieť tvorenú smerovačmi nazývame smerovaná sieť.[7]
Smerovače môžeme deliť na:

- Hardvérový smerovač - je zariadenie s vlastným software od výrobcu.
- Softvérový smerovač - je to software ktorý beží napr. na platforme x86

2.2 Prepínač na 2.vrstve(L2 witch)

Prepínač je zariadenie ktoré pracuje na linkovej vrstve. Slúži k prepojeniu alebo oddeleniu časti siete. Prepínanie paketov je hlavnou funkciou. Prepínač obsahuje tabuľku v ktorej má zaznamenanú väzbu medzi hardvérovou adresou a portom na ktorom sa zariadenie nachádza. Do tejto tabuľky zapisuje informácie pri prvom prečítaní paketov, kedy prečíta zdrojovú aj cieľovú MAC adresu. Zariadenie obsahuje vyrovnávaciu pamäť ktorá rieši rozdielnosť rýchlostí na portoch. Tieto rýchlosti dokáže automaticky detekovať. Prepínač spôsobuje malé oneskorenie, ktoré práve vyrovnávacia pamäť spôsobuje. Prepínač delí kolízne domény unicast paketov. Dokáže detekovať porušené dáta čím odľahčuje od týchto dát zbytok siete. V súčastnosti

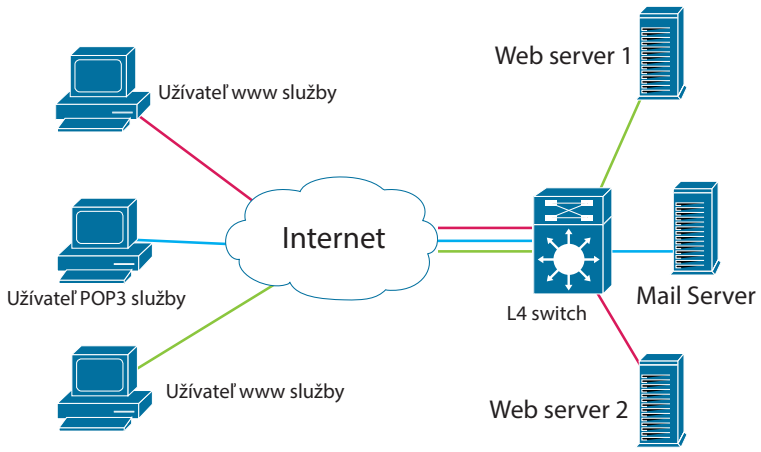
s vývojom sieti vzniklo viac druhov prepínačov, ktoré delíme podľa vrstvy ISO/OSI modelu, s ktorou dokážu pracovať.[1]

2.3 Prepínač na 3.vrstve (L3 switch)

V terajšej dobe outsourcingových riešení sa zmenili požiadavky na klasické prepínače. V minulosti používané pravidlo 80:20 znamenajúc pomer prenesených dát v sieti k pomeru prenesených dát smerom do internetu už neplatí. Čoraz viac dát sa vďaka rôznym cloudovým službám outsourcovaným mimo siete začalo prenášať do internetu. Smerovače musia zvládať oveľa väčšiu prevádzku na sieti a musia ju stíhať spracovávať rýchlo. Došlo k optimalizácií zariadení na rýchlosť. Softvérová logika smerovačov sa presunula do hardvéru. Vznikli tak zariadenia označované L3 switch. Z hľadiska logického fungovania je to smerovač, pretože sa rozhoduje podľa sieťových adries, ale jeho rozhodovacie schopnosti bývajú oveľa menšie v záujme maximálnej dosiahnuteľnej rýchlosti. L3 switch nepoužijeme tam kde potrebujeme nastavovať špeciálne oprávnenia, obmedzenia a užívateľské prístupy.[8]

2.4 Prepínač na 4.vrstve (L4 switch)

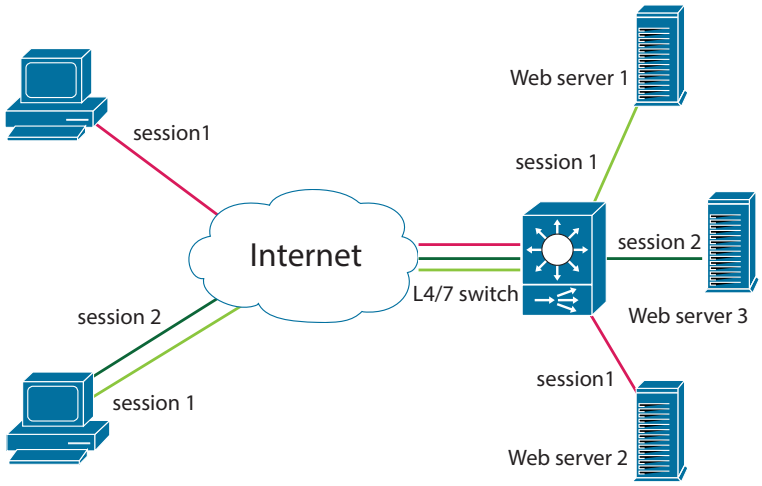
L4 switch je zariadenie ktoré pracuje na transportnej vrstve. Dokáže rozpoznať druh prevádzky (napr. HTTP, SMTP, FTP). Robí prostredníctvom čísiel portov. Napríklad web je port 80/443, mail port 25. Tieto zariadenia manipulujú s paketami na sieťovej vrstve ale rozhodujú sa podľa vrstvy transportnej.[8]



Obr. 2.1: L4 switch v sieti

2.5 Prepínač riadený obsahom (content switch)

Niekedy nám na rozhodovanie smeru toku dát nestačí sledovať len transportnú vrstvu a rozhodovať sa podľa čísel portov. Služba môže bežať na inom porte a L4 switch nevie analyzovať tieto dáta aby spoznal o akú službu sa jedná. *Content switch* je zariadenie pracujúce s vrstvami 4 až 7, ktoré dokáže tento problém vyriešiť. Využíva sa aj na *Load balancing* služby ktorý nie je možné urobiť L4 prepínačom. Sú to služby ktoré potrebujú udržať kontinuálne spojenie alebo session na jednom serveri (napr. HTTPS). Často sa označujú ako L4/7 switch alebo aj web switch (názov pre prepínač zameraný iba na sledovanie prenosu www stránok).[8]



Obr. 2.2: L4/7 switch v sieti

3 SIEŤOVÉ TOPOLÓGIE SÚČASNOSTI

Pri návrhu siete je dôležité zamyslieť sa nad tým ako bude vyzerať jej topológia, či daná topológia splní nároky na robustnosť a výkon a či je na nej možné použiť niektoré z technológií spomenutých v ďalších kapitolách.

3.1 Hviezdicová topológia

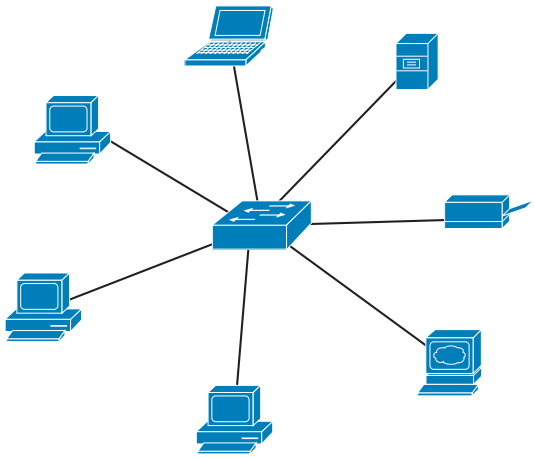
Siete zapojené do hviezdy sú jedným z najbežnejších typov sieti. Pozostávajú z jedného centrálneho uzla ktorý je tvorený prepínačom, smerovačom alebo rozbočovačom (Pri použití rozbočovača má sieť charakter zbernicovej topológie). Dáta v sieti sú prijímané a vysielané len k jednému centrálnemu uzlu.

Výhody:

- rýchlosť, odolnosť voči zahlteniu siete,
- jednoduché zapojenie a rozšíriteľnosť,
- porucha ktorá nie je na centrálnom uzli nemá vplyv na ostatné zariadenia siete,
- jednoduché detekovanie poruchy.

Nevýhody:

- porucha centrálneho uzla odstaví celú sieť,
- každé zariadenie musí mať pripojovací kábel až do centrálneho uzla.[16]



Obr. 3.1: Hviezdicová topológia

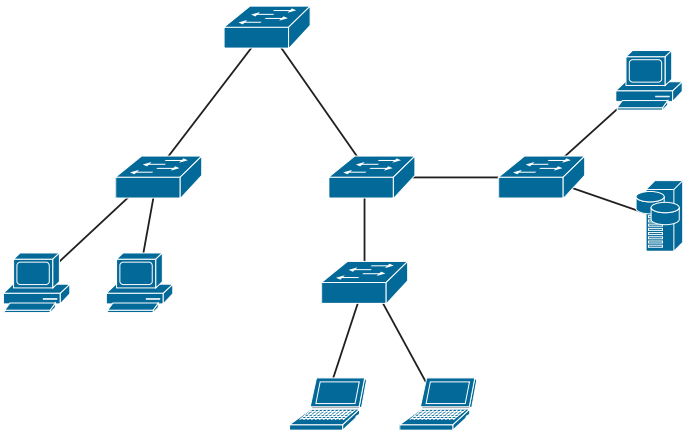
3.2 Stromová topológia

Pojmom stromová topológia sa označujú počítače zapojené do útvaru pripomínajúceho strom. Je to niekoľko hviezdicových topológii spojených centrálnymi uzlami znovu do hviezdy. Tento typ je často používaný v rozsiahlych sieťach, kedy je vhodné oddeliť napr. budovy, oddelenia či poschodia samostatnými hviezdami.
Výhody:

- odolnosť voči zahlteniu siete,
- menej potrebných káblov (káble sa ťahajú len do lokálnych centrálnych uzlov),
- pri poruche jednej hviezdy funguje zbytok siete.

Nevýhody:

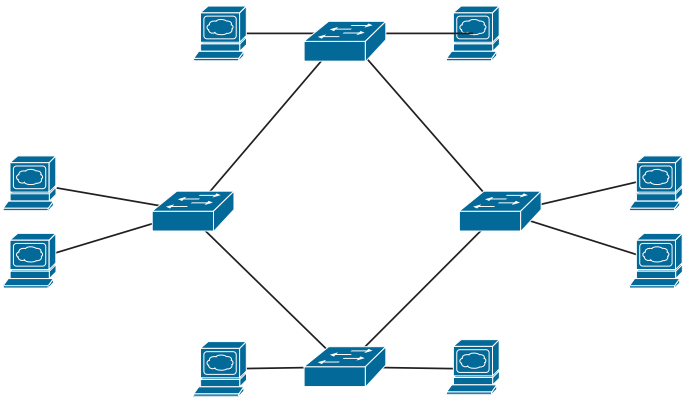
- väčší počet sieťových prvkov (prepínač, smerovač, rozbočovač). [16]



Obr. 3.2: Stromová topológia

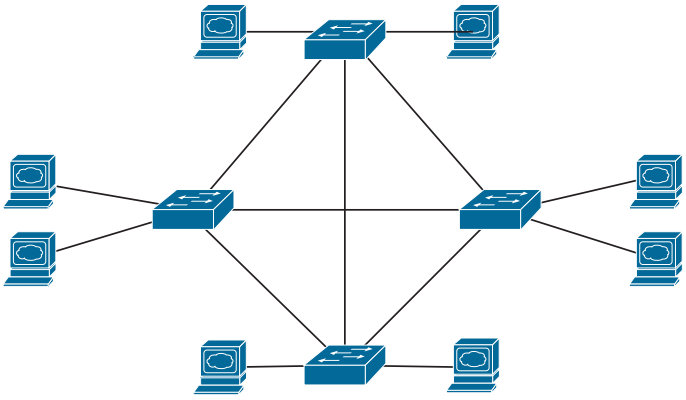
3.3 Zmiešaná topológia (MESH)

Tento typ siete je typický pre moderné robustné dátové siete. Podľa robustnosti ich delíme na dve kategórie.
Zmiešaná topológia (Partially connected MESH) je typ siete, v ktorej sú niektoré uzly siete prepojené viac ako jedným spojom. Používajú sa tam kde je nutné dosiahnuť vyššiu odolnosť siete voči výpadkom a kde je topológia každý s každým nerealizovateľná.[16]



Obr. 3.3: Zmiešaná topológia

Topológia každý s každým (Full connected MESH) je typ siete kde je každé zariadenie siete pripojené ku každému uzlu siete. Používajú sa tam kde sú najvyššie nároky na odolnosť voči výpadku.



Obr. 3.4: Topológia každý s každým

V sieti je nutné použiť metódy vhodného výberu cesty v sieti ktoré sa dokážu dynamicky meniť. Na toto nám slúžia protokoly rodiny Spanning Tree a nové technológie TRILL a SPB, ktoré sú zhrnuté v ďalšej kapitole.[16]

Výhody:

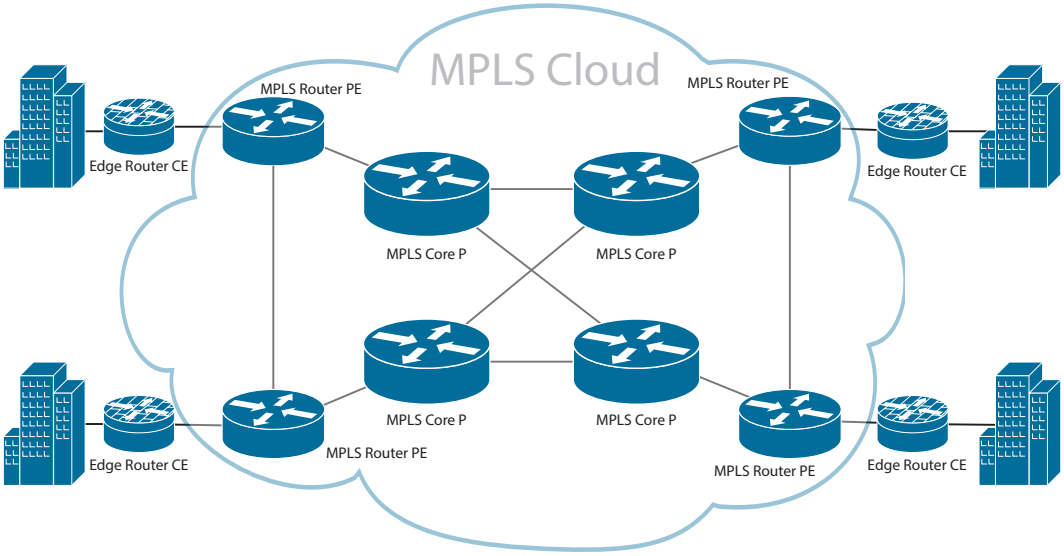
- možnosť komunikovať s uzlom siete pri zlyhaní niektorých liniek k uzlu,
- zlyhanie jedného prvku nespôsobí zlyhanie celej siete.

Nevýhody:

- vyžaduje smerovanie toku dát alebo ochranu proti zacykleniu.[16]

4 VYUŽITIE MPLS NA SPÁJANIE UZLOV SIETÍ

MPLS (MultiProtocol Label Swiching) - mechanizmus prenosu dát,ktorý emuluje niektoré vlastnosti siete s prepínaním okruhov v sieti, s prepínaním paketov. MPLS je mechanizmus prepínania, ktorý nastavuje paketom značky (čísla) a potom pomocou značiek tieto pakety prepína. Značky sa priraďujú na hranici MPLS siete a prepínanie v sieti MPLS prebieha výhradne na základe značiek. Značky obvykle odpovedajú ceste k cieľovej adrese vrstvy 3 čo odpovedá smerovaniu IP založenému na cieľových umiestneniach. Mechanizmus MPLS vznikol preto, aby umožnil prepínať aj iné protokoly ako TCP/IP. Vzhľadom k tomu prebieha prepínanie značiek v rámci siete rovnako bez ohľadu na protokol vrstvy 3. Vo väčších sietiach môžu v dôsledku značkovania MPLS vyhľadávať smerovanie iba hraničné smerovače. Všetky smerovače v jadre siete posielajú pakety podľa ich značiek, čo urýchľuje posielanie paketov po sieti poskytovateľa služieb. V súčasnosti sú nahradzované siete spoločností za siete *Frame Relay* s mechanizmom MPLS.[1]



Obr. 4.1: MPLS cloud v rozľahlej sieti

5 VYUŽÍVANIE VLAN V SIETI

Delenie sieti podľa typu prevádzky alebo iných logických kritérií nebýva vždy možné. Vznikli preto Virtuálne LAN siete, ktoré nám umožňujú deliť siete v uzloch v podstate ľubovoľne. Virtuálne LAN siete slúžia k Logickému rozdeleniu siete nezávisle na fyzickom usporiadaní. Môžeme tak vytvárať menšie segmenty siete na pôvodnej štruktúre. Ak máme napríklad jednu skupinu zariadení pripojených do jedného prepínača a inú do druhého prepínača a tieto prepínače sú navzájom fyzicky oddelené (rôzne segmenty siete), môžeme pomocou VLAN sieti dosiahnuť vytvorenie jednej siete v ktorej sa zariadenia oboch skupín navzájom vidia. S VLAN sieťami môžeme pracovať ako s normálnymi sieťami, môžeme teda na nich používať smerovanie. V L3 prepínačoch sa dnes bežne používa funkcia *inter-VLAN routing*, ktorá nám toto umožňuje.[1]

Dôvody prečo využívať VLAN siete:

- zoskupovanie užívateľov podľa určitých kritérií do skupín a izolácia komunikácie medzi týmito skupinami.
- zmenšenie broadcastových domén a zníženie prevádzky na sieti
- zmenšenie kolíznych domén (v sieti s rozbočovačmi)
- zjednodušená správa, zariadenie je možné zmenou konfigurácie presúvať zo siete do siete
- oddelenie špeciálnej prevádzky na siete, napr. sieť správcovského prístupu, transfér VoIP po sieti atď.
- využite QoS spolu s VLAN
- zníženie počtu zariadení zefektívnením využívania počtu portov.[18]

5.1 IEEE 802.1q

Tento protokol označovaný aj ako trunking protocol podporujú všetky moderné prepínače. V prepínači prebieha tagovanie a to tak že sa hlavička rozšíri o 4 bajty a prepočíta sa kontrolný súčet.

6 Bajtov	6 Bajtov	4 Bajty	2 Bajty	64-1500 bajtov	4 Bajty
Cieľová adresa	Zdrojová adresa	802.1q tag	typ alebo dĺžka	dáta	kontrolný súčet (FCS)

Obr. 5.1: Upravený VLAN rámec

Tieto 4 bajty (802.1q tag) nesú na začiatku informáciu o protokole 0x8100,potom prioritu podľa 802.1q, príznak CFI a na konci číslo VLAN siete.

2 Bajty	3 Bajty	1 Bajt	12 Bajtov
0x8100	priorita podľa 802.1p	Canonical Format Indicator (CFI)	VLAN ID

Obr. 5.2: 802.1q tag

Ak sa deje komunikácia vo VLAN sieti v rámci jedného prepínača, prepínač povoľuje smerovanie podľa informácií v pamäti a nič netaguje. Ak však nastáva komunikácia medzi viacerými prepínačmi v rámci jednej VLAN, tak prepínač taguje na trunk porte komunikáciu smerom k druhému prepínaču.
Pri trunk portoch nastavujeme aj *Natívny VLAN* ktorý nám zabezpečí že sa prichádzajúce spojenie na tomto porte netaguje a priradí sa do tejto *Native VLAN* siete.

5.2 Cisco ISL - Inter-Swicth Link

Tento protokol vznikol ešte pred vznikom štandardu IEEE 802.1q. V súčasnosti ho nepodporujú iné zariadenia ako Cisco prepínače vyššej rady. Rámce sa balia do novej hlavičky a kontrolného súčtu čím sa zväčšia o 30Bajtov a zväčšuje sa tak podstatne komunikácia.

23 bajtov		4 bajty
ISL hlavička	originálny rámec	kontrolný súčet (FCS)

Obr. 5.3: Cisco ICL hlavička

6 METÓDY ZABEZPEČENIA ROBUSTNOSTI PREPÍNANEJ SIETE

6.1 Redundantné spoje a slučky v sieti

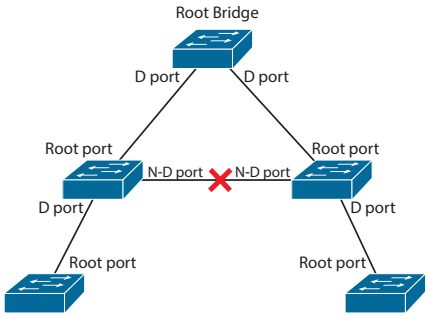
Pri vytváraní robustných sieti je potrebné zabezpečiť redundantné spoje, aby v prípade výpadku jednej cesty nehavarovala celá vetva siete ktorá je od tejto cesty závislá. Redundantné spoje ale môžu spôsobiť slučky v sieti, ktoré vedú k všesmerovým búrkam a ku kópiám rámcov. Toto je ľahké ošetriť použitím protokolov rodiny Spanning-Tree. Protokolmi SPB (Shortest Path Bridging) a TRILL (RBridge) môžme slučky dokonca využiť na rozloženie záťaže siete vrátane spomínaných redundantných spojov.V tejto kapitole je zhrnutý stručný opis funkčnosti týchto protokolov.

6.1.1 Spanning Tree Protocol (IEEE 802.1d)

Hlavnou úlohou protokolu STP je predchádzať sieťovým slučkám v sieti vrsty 2. Protokol podrobne sleduje sieť a vyhľadáva všetky linky. Vypnutím všetkých redundantných liniek zaisťuje aby sa nevyskytli žiadne slučky. Tento protokol pomocou algoritmu kostry grafu (STA – spanning-tree algorithm) najprv vytvorí topologickú databázu a potom vyhľadá a zlikviduje redundantné linky. Počas aktivity sú rámce preposielané iba po najlepších spojoch, ktoré protokol vybral.

STP nastavuje jednotlivým portom stavy:

- **Root Port** - je port s najnižšou cenou (linka spojená s root bridge alebo najkratšia cesta k nemu),
- **Designated Port** - port, ktorý je členom STP topológie a pripojuje segment,
- **Non-designated Port** - blokovaný port ktorý je redundantnou cestou .



Obr. 6.1: Spanning Tree Protocol

Root a designated porty sú porty ktoré posielajú dáta (stav forwarding) a non-designated port je port blokujúci prenos dát (stav blocked).
V STP sieti vzniká takzvaný **Root Bridge**.
Má najnižší Bridge ID (*BID*) ktorý sa skladá z priority a mac adresy. Zmenou priority teda môžeme meniť *root bridge*. Všetky jeho porty komunikujú a sú v stave *designated*. Je koreňom stromu STP a všetky rozhodnutia sa dejú podľa neho. Preto je vhodné aby *Root Bridge* bol najvýkonnejší prepínač.[1]

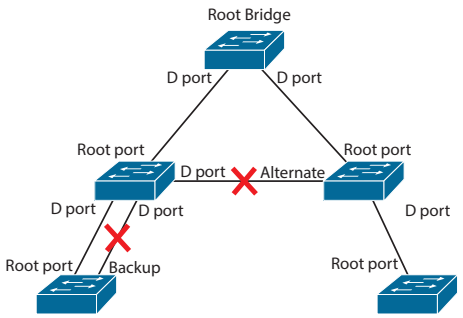
6.1.2 Rapid Spanning Tree protocol (IEEE 802.1w)

STP konverguje vo veľkých časoch, niekedy aj 30-50 sekúnd čo je extrémne veľa. Preto vznikol upravený algoritmus RSTP ktorý dokáže tento čas udržať na hodnotách okolo 1-2 sekundy. RSTP používa novšiu verziu BPDU v2. Všetky prepínače zasielajú BPDU na všetky porty.
Typy portov:

- **P2P** - slúži na pripojenie ďalšieho prepínača, linka musí byť full duplex,
- **Edge** - koncový port, v ktorom je pripojené koncové zariadenie,
- **Shared** - zdieľaná linka half duplex,

Úlohy portov:

- **Root Port** - najlepší BPDU na prepínači,
- **Designated Port** - najlepší BPDU na segmente,
- **Alternate Port** - blokovaný, je alternatívnou cestou k rootu,
- **Backup Port** - blokovaný, je redundantnou cestou k segmentu.



Obr. 6.2: Rapid Spanning Tree Protocol

Zloženie BPDU rámca je následovné:

Prepínače si medzi sebou posielajú *Agreement* a *Proposal*, ak zistia že sused má väčšie BID, automaticky si prenastavia *root* a *designated* port. Switch na ktorom beží STP automaticky zahadzuje BDU z RSTP.[9]

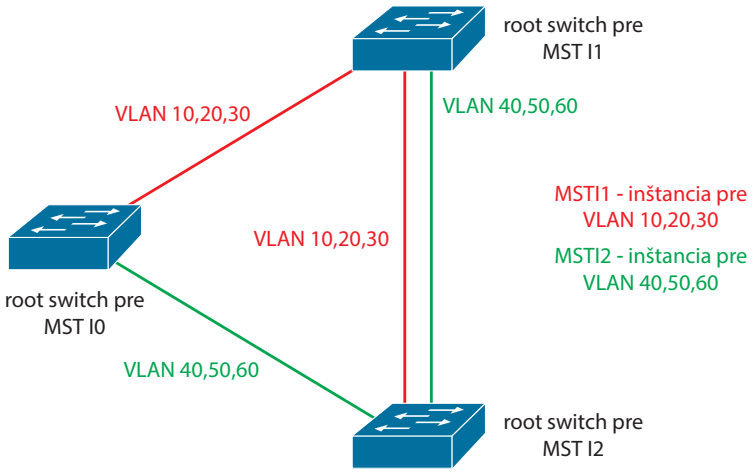
BPDU												
2 B	2 B	2 B	1 B	8 B	4 B	8 B	2 B	2 B	2 B	2 B	2 B	1 B
protocol ID	version	type of BPDU	flags	root BID	root path cost	bridge ID	port ID	message Age	maximum Age	Hello time	forward delay	version 1 lenght

Obr. 6.3: BPDU rámec

6.2 Technológie umožňujúce rozloženie záťaže

6.2.1 Multiple Spanning Tree protocol (IEEE 802.1s)

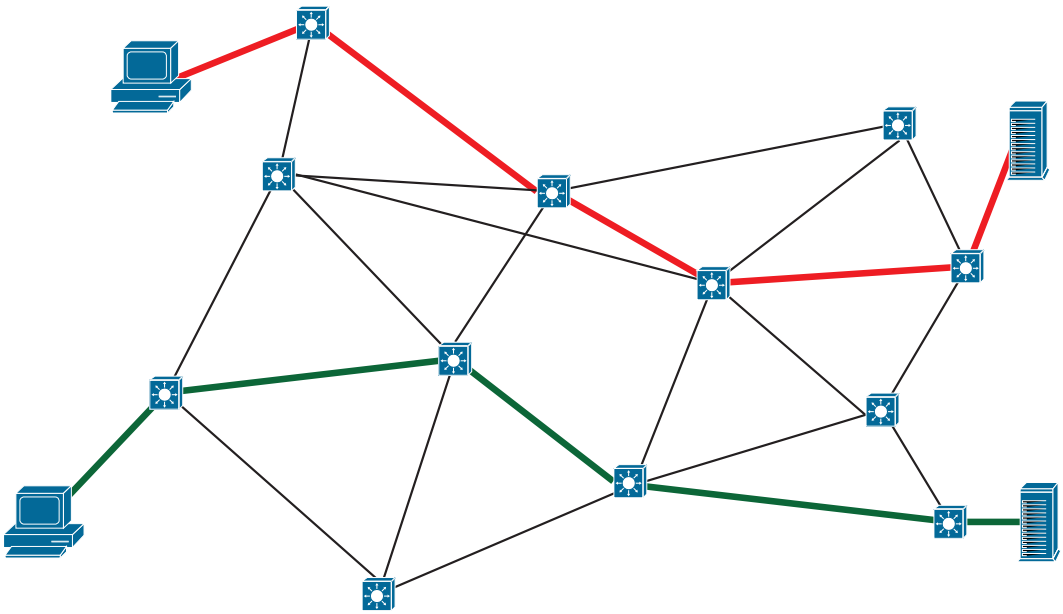
MSTP je rozšírením RSTP. Úzko súvisí s VLAN sieťami. Je rovnako rýchly ako RSTP a umožňuje mapovanie niekoľkých VLAN do jednej STP inštancie (celkovo je možné vytvoriť až 65 inštancií). Ušetrí sa tak počet STP pre veľký počet VLAN. Dokáže využívať viac ciest pre každú inštanciu a vykonávať tak jednoduchý load-balancing, ktorý je ale statický. MSTP beží nad RSTP, takže musia existovať spolu. Používa sa na chrbtové siete. U MSTP rozlišujeme MST región, ktorý je skupinkou prepínačov ktoré majú rovnako mapované inštancie na VLANy. MSTP inštancia je mapovanie VLAN do skupín. Defaultne sú všetky mapované do inštancie 0. Pri zmenách inštancie treba brať v úvahe krátkodobý výpadok v dôsledku reinicializácie. MST protokol bol pôvodne zadefinovaný v norme 802.1s, roku 2003 bol ale zlúčený do normy 802.1q ktorá sa zaoberá VLANmi.[10]



Obr. 6.4: MSTP inštancie

6.2.2 Shortest Path Bridging (IEEE 802.1aq)

Tento protokol nahradil staršie protokoly STP, ktoré dovoľovali iba jednu cestu k *Route Bridge*. SPB dovoľuje využívať všetky cesty. Vďaka nemu je možné vytvárať oveľa väčšiu a robustnejšiu topológiu. SPB môže byť použité všade kde sa už Spanning Tree práve používa. Správca siete môže vziať akúkoľvek sieť s nasadeným STP alebo MSTP a migrovať ju na SPB. Celý dátový tok je smerovaný symetricky a najkratšími cestami. Využíva sa linkový stavový protokol IS-IS pre propagáciu topo-



Obr. 6.5: SPB a šírenie dátového toku

lógie a príslušnosti k logickej sieti. Pakety sa zapuzdrujú do MAC-in-MAC popísanej v norme 802.1ah alebo do rámcov 802.1Q/802.1ad. SPB dovoľuje agregáciu liniek podľa štandardu IEEE 802.1AX a implementáciu jeho MC-LAG.[6]

Hlavné výhody využitia tejto technológie:

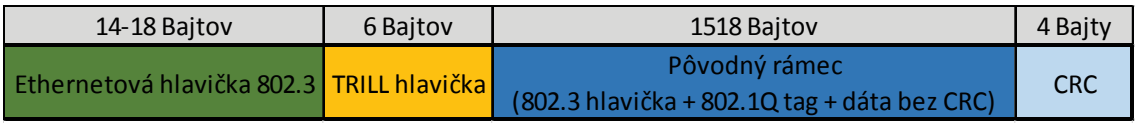
- možnosť využiť až 16 miliónov VLAN sieti oproti bežne použiteľným 4096,
- rýchla konvergencia,
- zvýšená priepustnosť,
- redundancia vyplývajúca zo zdieľania záťaže cez všetky cesty siete.

Príklad zariadení podporujúcich SPB:

- Alcatel-Lucent 7750-SR, OmniSwitch 6900, OmniSwitch 10K,
- Avaya VSP 4000,4800,7000,8000,88000,9000 Series,
- Enterasys Networks S140,S180,
- Huawei S9300,
- HP 5900,5920,11900,12500,12900.

6.2.3 TRILL

TRILL prepínač je zariadenie podobné klasickému ethernetovému prepínaču. Má 2 typy ethernetových portov. Porty ktorými sa prepájajú TRILL prepínače a porty ktorými sa pripájajú koncové stanice. Každý typ spracováva prichádzajúce a odchádzajúce rámce iným spôsobom. Na TRILL portoch dochádza k modifikácii rámcov pridaním hlavičky protokolu TRILL. Rámec sa následne zabalí do ďalšej hlavičky. Hlavička potom vyzerá takto:



Obr. 6.6: TRILL rámec

Na ostatných portoch sa spracovávajú rámce v klasickej podobe. TRILL prepínač po zapnutí automaticky vyhľadá svojich susedov a pomocou IS-IS protokolu si vybuduje topologickú mapu celej prepínanej siete. Na základe tejto mapy si vypočíta najvhodnejšie cesty ku každému TRILL prepínaču v sieti. Tieto cesty využíva pri posielaní rámcov so známou unicastovou adresou. TRILL prepínače vypočítavajú zdieľaný distribučný strom, ktorý potom používajú k k posielanie multi-desination rámcov na TRILL portoch. Na bežných portoch sa pre multi-destination rámce použije klasická metóda floodingu. Prvé rámce ktoré dorazia z koncových staníc sú popri prenášaní zaznamenávané do prepínacích tabuliek, navyše sa zaznamenávajú aj zdrojové adresy z dekapsulovaných TRILL rámcov. V tabuľke je zapísaný k mac adrese zapísaný zdrojový TRILL prepínač ktorý vykonal enkapsuláciu natívneho rámca od koncovej stanice. Všetko sa vykonáva automaticky.[5]

Známe zariadenia podporujúce TRILL technológiu:

- HP 5900,5920,11900,12900 Switch Series,
- Huawei Cloud Engine 5800,6800,7800,12800 Switches,
- Enterasys Networks S-Series Switches.

6.2.4 Cisco FabricPath

Spoločnosť Cisco vzala technológiu TRILL a prerobila ju do svojej podoby pod označením FabricPath. Pridali vylepšenia:

- podpora port-channel naprieč zariadeniami,
- podpora viacerých topológií - FabricPath prepínače môžeme nastaviť tak, aby sa učili MAC adresy len jednej VLAN siete, čo je veľmi výhodné pri veľkých sieťach (tisíce staníc - napr. dátové centrá)[5]

Cisco Nexus 5000 a 7000 podporuje ako FabricPath tak aj TRILL, nedajú sa však používať naraz na jednom prepínači.

6.2.5 Juniper Networks QFabric

Firma Juniper Networks prišla s vlastným riešením určeným na obrovské siete nazývaným *QFafbric system* čo sú v podstate programovateľné L3 prepínače navrhnuté do full-mesh sieti s cieľom využívania v stredne veľkých až obrovských datacentrách. QFabric má vlastný systém rozkladania záťaže a podporu virtualizácie siete a vôbec nepodporuje TRILL ani SPB.
V súčasnosti sú k dispozícií 2 modely:
QFK3000-M Fabric System - podpora 48 až 768 10Gbe portov, vhodný na stredne veľké dátové centrá ktoré dokážu vykryť bežné podnikové aplikácie,
QFX3000-G Fabric System - podpora až 6144 10Gbe portov, vhodné na obrovské bussiness aplikácie, analýzu dát, výskum či vysokovýkonný cluster.[15]

7 ZAISTENIE QOS V PREPÍNANEJ SIETI

Ak je sieť dostatočne dimenzovaná, zvyčajne netreba riešiť traffic management. Situácia sa ale mení ak pracujeme s obrovskými tokmi dát a ak má niektorý uzol siete obmedzenú šírku pásma.
Nastávajú rôzne poruchy prenosu:

- strata paketov,
- oneskorenie
- rozptyl
- doručenie mimo poradia

Ak máme v pláne používať v sieti napríklad VoIP, videohovor, streamované multimédia, terminálove pripojenia k serveu alebo veľmi dôležitú kritickú aplikáciu musíme myslieť na kvalitu týchto služieb. Quality of Service (QoS) je súbor technológií, ktoré riešia niekoľko problémov traffic managementu. Cieľom QoS je umožniť nastavenie určitej kvality prenosu pre dáta prenášané sieťou. QoS dokáže rozlišovať typ prenosu a nastaviť podľa typu kvalitu prenosu. QoS je obsiahnutý v norme IEEE 802.1p a 802.1q. Funkcie QoS:

- priorizuje určitý prenos pred iným,
- obmedzuje prenosové pásmo,
- vyhradzuje prenosové pásmo.

Podľa architektúry siete sa musíme rozhodnúť pre spravnz model traffic managementu.

7.1 Integrované služby (IntServ)

Model Intserv má za úlohu zaistiť požadovanú kavlitu v IP sieti. Intserv poskytuje 2 typy služieb. Garantovanú službu a službu s riadením záťaže. Je zložený z 4 hlavných častí, ktoré musia byť implementované vo všetkých smerovačoch a hostiteľoch.

- **Plánovač paketov (Packet Scheduler)** - riadi zasielanie jednotlivých prúdov dát. K tomu využíva fronty a napr. časovač. Plánovač musí byť k dispozícií tam kde sú pakety radené do front. každý tok je riešený samostatnou frontou. Plánovač následne rieši ako bude zaobchádzať s jednotlivými frontami.
- **Kontrola prístupu (Admision control)** - realizuje sa rozhodovacím algoritmom, ktorý využíva smerovač alebo hostiteľská stanica k zisteniu či vytvorenie novej rezervácie neovplivní toky ktoré su už rezervované. Kontrola prístupu je v každom uzli siete.

- **Klasifikátor (Classifier)** - slúži k identifikácií a smerovaniu paketov. Každý prichádzajúci paket je mapovaný klasifikátorom do určitej triedy. S paketami v jednej triede systém zachádza rovnako.
- **protokol rezervácie prostriedkov (RSVP)** - protokol k zostaveniu a udržiavaniu stavov v smerovačoch a koncových zariadeniach po celej trase, kde má byť rezervovaná prevádzka.

Celá prevádzka je riadená klasifikátormi a plánovačmi paketov. Ak príde do smerovača koncovej stanice žiadosť RSVP, tak táto žiadosť prejde mechanizmom, kedy sa určí či je v koncovom zariadení dostatok prostriedkov k dosiahnutiu požadovaných kvalít služieb.
Na zaistenie QoS je implementované riadenie prenosu v dvoch hlavných krokoch:

- **Klasifikátor paketov** určí kvalitu služieb pre každý paket
- **Plánovač paketov** docieli sľúbenú kvalitu služieb

Po kontrole dostatku zdrojov pre QoS sa zisťuje či má užívateľ povolenie na vytvorenie rezervácie prostriedkov. Ak má, príde k nastaveniu klasifikátorov a plánovačov. Ak nemá, zamietne sa rezervácia. Pri rezervácií sa musia dodržovať určité režimy a to hlavne pri multicastovom vysielaní. Ak existuje viac odosielateľov v jednej relácií, potom sa delí rezervácia na dva režimy:

- **Odlišná rezervácia (Distinct Reservation)** - vytvárajú sa samostatné rezervácie pre každého odosielateľa služby,
- **Zdieľaná rezervácia (Shared reservation)** - vytvára sa zdieľaná rezervácia pre určitú skupinu odosielateľov služieb.[17]

7.2 Diferencované služby (DiffServ)

Diferencované služby členia jednotlivé služby podľa ich nárokov na sieť. Služby sú priradené do tried. S každou triedou nakladá smerovač rozdielne ale s paketami v tej istej triede nakladá rovnako. Diffserv sa používa na chrbtových sieťach. V lokálnej sieti sú dáta klasifikované, upravené podľa určitých pravidiel a pridelené do skupiny s určitou agregáciou. Značka chovania je zapísaná do DSCP kódu (DiffServ Code Point), ktorý skúma klasifikátor. Ten použije pre označený paket úpravu prenosu. Vo vnútri siete je teda chovanie tokov PHB (Per Hop Behavior), ktoré je pridružené k polu DSCP. V sieti sa zjednajú určité pravidlá pre určité skupiny paketov. Tieto skupiny definuje SLA (Service Level Agreements). Podmnožinou SLA je dohoda o úprave prevádzky TCA (Traffic Conditioning Agreements), kde je podrobne špecifikovaný spôsob, akým bude s dátami nakladané, aby prenos vyhovoval SLA. TCA zahrňuje klasifikačné pravidlá, dopravné profily, značenie a pravidlá pre formovanie

dátového toku. Dopravný profil špecifikuje dočasnou vlastnosťou dopravného prúdu vybraného klasifikátorom. Poskytuje pravidlá pre určenie, či daný paket patrí alebo nepatrí do daného profilu. Pre kód služby DSCP sa využíva pole ToS v hlavičke IP datagramu. Využíva sa 6 bitov z celkových 8 pre kód služby DSCP. V hlavičke IP paketov v poli ToS určujú prvé 3 bity prioritu danej služby. Celý DSCP vytvorí potom prvých 6 bitov. Úprava prevádzky je pri Diffserv vykonávaná v koncových smerovačoch. Zariadenie vykonávajúce funkciu Diffserv predáva pakety vstupujúce do smerovača na výstup v poradí tak, aby boli splnené požiadavky pre daný typ služby.[17]

7.3 Rozdiely mezi IntServ a DiffServ

IntServ:

- na zostavenie a udržanie zaisteného spojenia používa protokol RSVP,
- precízne nastavenie garancie služby
- K zaisteniu QoS potrebuje všetky smerovače na ceste
- Vysoké nároky na kapacitu siete
- **Vhodné na veľké podnikové siete, využitie hlavne na okraji siete**

DiffServ:

- Trietdy typu prevádzky sú definované dopredu, netreba špeciálnu signalizáciu na zaistenie QoS
- Klasifikovanie typu prevádzky zaisťujú koncové zariadenia - jednoduchší management siete
- **Vhodné na rozsiahle siete (Chrbtové siete)**
- Gridový cluster - Grid cluster.[17]

8 POČÍTAČOVÝ CLUSTERING

8.1 Čo je to počítačový cluster

Počítačový cluster je zoskupenie viacerých počítačov ktoré spolu spolupracujú ale navonok sa tvária ako jeden počítač. Prepojené bývajú počítačovou sieťou.

Môžu mať viacero funkcií a podla nich rozlišujeme:

- Úložný cluster,
- Gridový cluster,
- Výpočtový cluster,
- Cluster s vysokou dostupnosťou,
- Cluster s rozložením záťaže.[14]

8.2 Úložný cluster

Úložný cluster alebo aj Storage cluster sprostredkuje prístup k diskovej kapacite rozloženej medzi viac počítačov. Využívajú sa špeciálne súborové systémy, ktoré zabezpečujú rozloženie záťaže, redundanciu dát a rôzne iné sprievodné funkcie.[14]

8.3 Gridový cluster

Gridový cluster je zložený z počítačov určených primárne na inú činnosť ako clustering. Počítače majú v sebe nejaký software, ktorý sa pospája s inými počítačmi v sieti a vytvorí cluster. Každý stroj pritom odovzdáva do clusteru svoj nadbytočný výkon ktorý nepotrebuje.[14]

8.4 Výpočtový cluster

High-performance computing pozostáva z niekoľkých počítačov ktoré sú prepojené vysoko-rýchlostnou sieťou s malými odozvami. Dosiahnú sa tak veľmi vysoké výpočtové výkony oproti vysokovýkonnému serveru. Aplikácie pre tento typ clusteru musia byť naprogramované aby ho dokázali používať.[14]

8.5 Cluster s vysokou dostupnosťou

Tento typ clusteru označovaný aj ako Failover Cluster zaisťuje nepretržitý chod kritických aplikácií. Jeho úlohou je poskytovať služby aj pri poruche alebo údržbe jedného alebo viacerých členov clusteru v závislosti od konfigurácie. Službu poskytuje len jeden počítač a v prípade nejakej závady prevezme jeho úlohu iný počítač podľa určenej priority.[14] V prostredí Microsoft Windows (od verzie 2008) je známa služba Failover Cluster ktorá sa využíva na clustering Hyper-V serverov, webových služieb a poštového serveru Microsoft Exchange.

8.6 Cluster s rozložením záťaže

Na dosiahnutie kvalitných parametrov sieťe treba predchádzať preťaženiu, maximalizovať priepustnosť, dosiahnuť efektívne a optimálne využívanie zdrojov a znížiť čas odozvy na minimum.
Najznámejšie funkcie load-balancingu:

- SSL akcelerácia a distribúcia
- DDoS ochrana
- HTTP kompresia
- TCP buffering
- TCP offload
- Uprednostnenie podľa priority
- Smerovanie podľa obsahu
- Zvýšenie bezpečnosti
- Aktivácia a de-aktivácia serverov podľa funkčnosti

Pri load balancingu je dôležitá perzistencia. V určitých aplikáciách je treba udržiavať session na jednom serveri alebo upne mimo koncového servera. Dáta na session zase napríklad v replikovanej databázi čo je ale pomalé.
Softvérový load-balancing:
Je to špeciálny software nainštalovaný na serveri. Najznámejšie sú Ultra Monkey, BalanceNG, Microsoft Network Load Balancing.
Hardvérový load-balancing:
Pozostáva z prepínačov L3,L4,L4/7 alebo smerovačov so softvérom pre load-balancing.

Riešenia load-balancingu:

- **least connection** - úlohu spracováva počítač ktorý aktuálne vybavuje najmenší počet úloh,
- **vážený least connection** - úlohu spracováva počítač ktorý vybavuje najmenší počet úloh ale riadi sa váhou, ktorá mu určuje aký počet spojení z celkového počtu bude vykonávať,
- **round-robin** - požiadavky sú cyklicky prideľované na všetky počítače, vhodné na stroje s rovnakým výkonom,
- **vážený round-robin** - požiadavky sú cyklicky prideľované ale každému zariadeniu navyše prideľujeme váhu čo je vhodné ak máme stroje s rozdielnymi výkonmi. [11]

Round-robin DNS:
Tento typ load-balancingu nepotrebuje špeciálny hardvér alebo softvér. Je to viacero IP adries priradených jednému doménovému menu. Klientská stanica si sama vyberie na ktorý server sa pripojí. Záťaž sa tak štatisticky rovnomerne rozdelí. Cache dns v klientských staniciach môže spôsobiť nerovnomerné rozloženie záťaže. Toto riešenie však pripája klientskú stanicu aj na server v poruche.[12]

9 VIRTUALIZÁCIA AKO TREND

Pojem virtualizácia je dnes veľmi rozšírený a zároveň všeobecný. Virtualizácia sa rozšírila primárne s virtualizáciou serverov, nasledovala virtualizácia aplikácií a v posledných rokoch sa úspešne rozširuje virtualizácia desktopov. A aby virtualizácie nebolo málo, prichádza do dátových centier virtualizácia sieťových prvkov.

Virtualizácia je proces pri ktorom je nahradený fyzický prostriedok softvérovou vrstvou. Prostriedok je pre systém transparentne definovaný aj keď fyzicky neexistuje. Vďaka virtualizácií môžeme nezávisle prevádzkovať niekoľko rovnakých či rôznych systémov na jednom hardvéri.

Kombinovaným riešením virtualizácie serverov a sieti vytvárame dnes často spomínané Cloudové služby.

9.1 Virtualizácia serverov a ich služieb

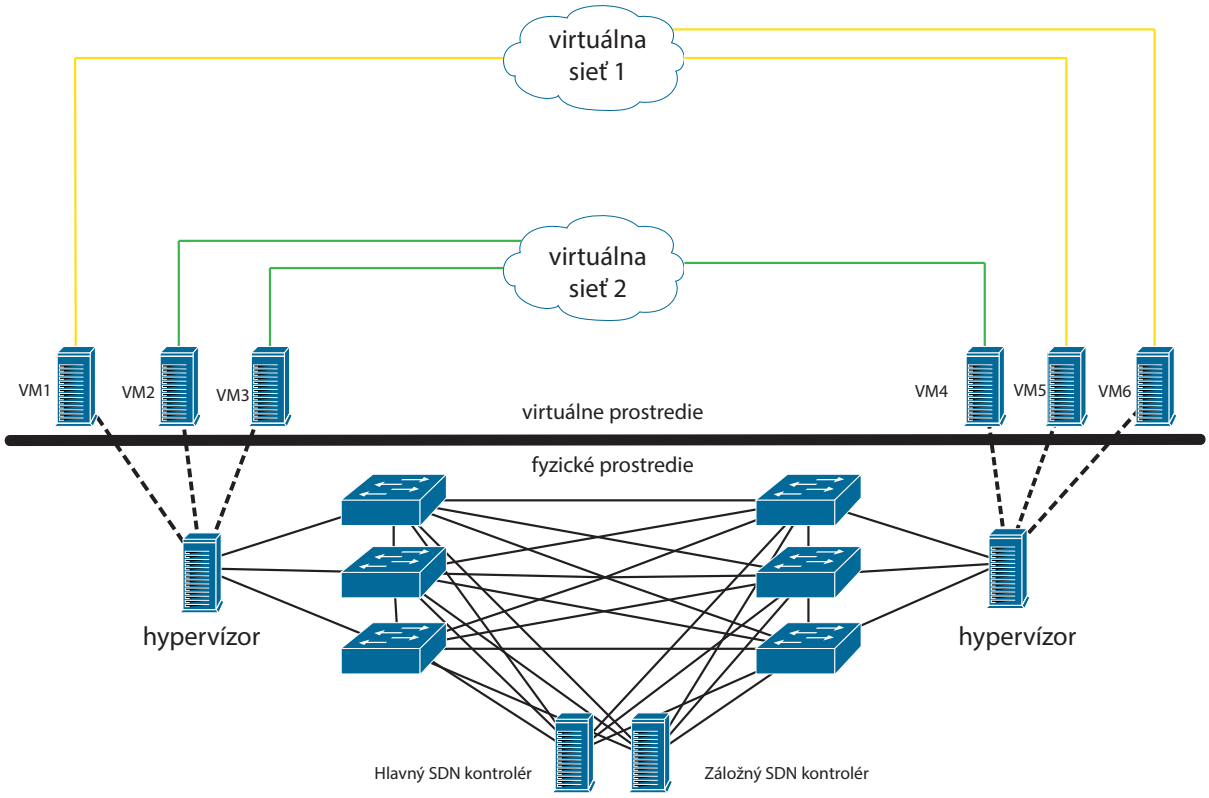
Virtualizačný software - Hypervisor vytvára štandardizovaný hardware ktorý poskytuje pre všetky virtuálne stroje. Vytvára sa napr. virtuálne CPU, operačná pamäť, disky, vstupno-výstupné zariadenia. Operačný systém nepotrebuje byť špeciálne upravený na beh vo virtuálnom prostredí. Hardware poskytnutý hypervisorom je natoľko štandardný že väčšina systémov s ním v súčastnosti vie pracovať. Inštalácia prebieha rovnako ako na fyzických strojoch cez konzoly, ktoré sú akýmsi vzdialeným monitorom, klávesnicou a myšou.

Výhody virtualizácie sú:

- úspora nákladov (energeticky efektívnejšie),
- jednoduchšia migrovateľnosť,
- možnosť živej migrácie alebo replikácie
- rýchla nasaditeľnosť nových systémov,
- automatický manažment virtuálnych strojov,
- možnosť presúvania služieb medzi rôznymi fyzickými systémami bez rekonfigurácie
- jednoduchšie zálohovanie a možnosť rýchlej obnovy
- snímkovanie systému v čase (snapshot)[13]

9.2 Virtualizácia siete - SDN

V posledných rokoch dochádza k prudkému zrýchľovaniu nasadzovania virtualizačných technológií a treba zabezpečiť lepšiu integráciu a podporu čoraz viac virtuálnych prostredí. Virtualizácia sieťových prvkov vyžaduje nový prístup k sieťovej architektúre. Fyzické prostredie nedokáže ponúknuť potrebnú dynamiku a škálovateľnosť. Ak potrebujeme napríklad zväčšiť kapacitu rozhrania siete, upraviť formu komunikácie, nastaviť blokovanie komunikácie, prípadne ak chceme len jej usmernenie, môžeme to všetko vyriešiť dynamicky a úplne automatizovať vďaka riešeniam založeným na SDN. Sieťová virtualizácia teda vytvára flexibilné logické virtuálne siete, ktoré sú oddelené od základného sieťového hardvéru. SDN siete sa využívajú takmer vždy v prostredí v ktorom sa využívajú aj virtualizované servery. Je takmer nevyhnutné ich používať kvôli flexibilite v moderných dátových centrách.[4]



Obr. 9.1: Sieťová virtualizácia

9.3 Virtualizačné platformy

Serverová virtualizácia:
Pre využitie výkonnej serverovej virtualizácie sú zaujímavé hlavne natívne (bare metal) hypervisory ktoré bežia priamo na fyzickom hardvéri a majú preto vyšší výkon a spoľahlivosť než hostované hypervisory bežiace pod nejakým operačným systémom. V súčasnosti sa najviac používajú:

- VMWare ESX,
- Citrix XenServer,
- Microsoft Hyper-V

Sieťová virtualizácia:

OpenFlow - je najrozšírenejší protokol podporujúci sieťovú virtualizáciu. Pomocou neho sa plnia vyhľadávacie tabuľky, inštalujú pravidlá (čo zahadzovať, čo kam poslať, obmedzenie priepustnosti, zmena hlavičky atď).

Voľne dostupné riešenia:

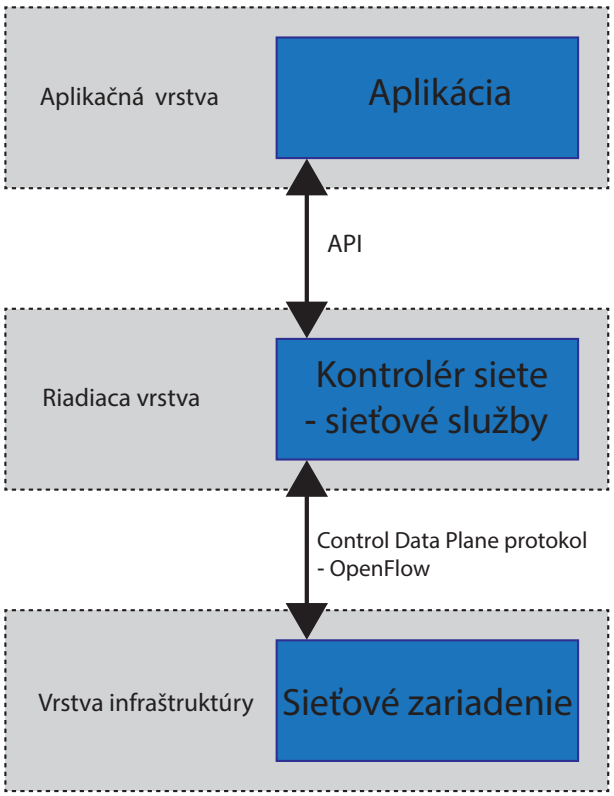
- Open Daylight
- Project Floodlight
- Beacon
- NOX/POX

Predávané špecifické riešenia:

- Cisco APIC
- Juniper Contrail
- Nuage Virtualized Services Controller (VSC)

10 NÁVRH LABORATÓRNEJ ÚLOHY - SDN

Rozmach virtualizácie priniesol novinku zvanú SDN. Neustály vývoj tejto technológie priniesol výhody, ktoré sú veľmi zaujímavé hlavne v dátových centrách, ale aj v rozľahlých sietiach kde potrebujeme mať kontrolu nad dátovým tokom alebo aj kde vyžadujeme pružnosť v možnosti rekonfigurácie siete v prípade rôznych výpadkov, porúch či preťažení uzlov siete. Vďaka SDN môžeme testovať rôzne experimentálne technológie na funkčnej sieti ktorá využíva už nejakú technológiu ako napríklad MPLS.
Výsledkom môjho skúmania a testovania tejto technológie je navrhnutá laboratórna úloha, ktorá je súčasťou prílohy. V nej sa snažím zdôrazniť to čo považujem za najväčšiu výhodu tejto technológie a tou je práve jednoduchosť správy a kontroly toku dát v rozľahlejšej sieti a tiež jej rozšíriteľnosť o rôzne funkcie ktoré si môžeme aj sami naprogramovať.
Ako vlastne SDN sieť funguje? Potrebujeme kontrolér, SDN aplikáciu, univerzálny komunikačný protokol ako napr. OpenFlow a hardware v dobre navrhnutej fyzickej topológii, ktorý OpenFlow podporuje.



Obr. 10.1: SDN architektúra

10.1 Kontrolér

Kontrolér je počítač ktorý má znalosť celej siete ktorú riadi. Jeho zlyhanie by bolo kritické, preto sa prevádzkuje v clusteroch alebo v strojoch s takzvanou vysokou dostupnosťou. Sieť je štandardne nastavená tak aby dokonca krátkodobý výpadok služby kontroléru dokázala prežiť. Kontrolér býva vybavený užívateľským interface (web GUI alebo shell) a API rozhraním. Ako Externé API sa používa REST API, čo sú jednoduché webové volania, ktoré je možné realizovať z aplikácií naprogramovaných v ľubovolnom jazyku a prostredí. Toto API je ideálne pre úlohy ktoré nemusia bežať v reálnom čase. Ako Natívne API sa používa Java API, ktorým môžeme vytvárať moduly v rámci OSGi architektúry. Cez toto API môžeme nahrať moduly do kontroléru za behu avšak musia byť naprogramované v Jave. Aplikáciou cez Java API môžeme meniť nastavenia v reálnom čase v závislostí od udalostí. Kontrolér SDN pridáva a odstraňuje záznamy z tabuľky tokov dát v prepínačoch. Tieto záznamy môžeme nastaviť staticky alebo aj dynamicky pomocou nejakej aplikácie využívajúcej spomínané API.[21] Podľa toho ako pracuje kontrolér so sieťovými prvkami, môžeme rozdeliť jeho chovanie na 3 režimy:

Reaktívne chovanie
v sieťovom prvku nie sú po spustení SDN žiadne inštrukcie, s každým novým tokom je informovaný kontrolér. Ten doručí paket a nastaví inštrukcie tak, že pakety v sieti už nebude spracovávať ASIC prepínača sám. Vďaka tomu má kontrolér prehľad o každom toku v sieti.[21]

Proaktívne chovanie
kontrolér nastaví dopredu tok pre spracovanie ASIC čipom na základe udalostí z nejakej inej aplikácie. Využitie je hlavne tam, kde nevyžadujeme dynamické aplikácie riadenia toku dát v reálnom čase.[21]

Hybridné chovanie
v tomto režime je nastavené ako posledné pravidlo v tabuľke toku dát „forward Normal“ namiesto „forward Controller“. Sieťové prvky tak spracovávajú data vlastným riadením toku. Môžeme tak využívať sieť, ktorá už má aplikovanú technológiu ako napríklad STP či OSPF. Špecifickým pravidlom ale môžeme posielať data do kontroléru, kde môžeme nejakou aplikáciou zaistiť napríklad ochranu siete alebo aj QoS.[21]

10.2 Protokol OpenFlow

Tento protokol označovaný ako south API, komunikuje z pohľadu API z infraštruktúrou. OpenFlow umožňuje oddeliť control plane od data plane. Pomocou neho môžeme poslať akýkoľvek paket do ktorejkoľvek časti siete, priamo ho presmerovať, zahodiť či replikovať. [21]
OpenFlow pozostáva zo správ:

packet_in
Dôležitá správa reaktívneho módu SDN. Paket, alebo len jeho úvodná časť sú zabalené v správe Packet IN a zaslané do kontroléru.[20]

packet_out
Taktiež dôležitá správa reaktívneho módu SDN. Kontrolér zasiela pakety od seba smerom do ľubovolného zariadenia na ľubovolný port (fyzický/logický) zabalené do správy Packet OUT.[20]

flow_mod
Pre verziu OpenFlow 1.0 obsahuje táto správa filtrovacie pravidla L2-L4 a akcie ktoré má pri danom filtri vykonať. (Pri verzii OpenFlow 1.3 už pribudlo aj MPLS, IPv6, inštrukcie a sety akcií a cookies.)
V týchto správach sa nachádza „idle timeout“ (platnosť záznamu po dobu kedy nastala zhoda), „hard timeout“ (pevne stanovený čas po ktorom sa záznam vymaže) a priorita ktorá určuje v akom poradí sa pravidlá vyhodnocujú.[20]

Záznam v prietokovej tabuľke ma 3 polia:

- hlavička definujúca tok
- akcia podľa ktorej sa pakety spracujú
- štatistiky o počte paketov a času od zavedenia pravidla alebo poslednej zhody paketu.[19]

Vybrať vhodný tok môžeme podľa:

- Vrstvy 1 - Tunel ID, Vstupný port, QoS priorita
- Vrstvy 2 - MAC adresa, VLAN ID, ethernet typ
- Vrstvy 3 - IPv4/IPv6, ARP
- Vrstvy 4 - TCP/UDP , ICMP.[19]

Akcie ktoré môžeme vykonávať:

- výstup na port
- zahodenie
- manipulácia s paketom
- zaslanie do kontroléru.[19]

10.3 Prepínač s podporou OpenFlow

Na výber sú v súčasnosti rôzne „hlúpe“ prepínače, ktorým dodá inteligenciu práve SDN kontrolér, ale aj rôzne klasické L2 prepínače či Multi-vrstvové prepínače. OpenFlow protokol je častokrát zavádzaný do hardvéru aktualizáciou firmware a preto ho v súčasnosti podporuje aj starší hardware. Vďaka tomu dokážeme v starom zariadení využívať nové funkcie, ktoré sú ale ohraničené pôvodným výkonom čipov. Pri výbere vhodného prepínaču musíme zvážiť či chceme sieť prevádzkovať v normálnom alebo hybridnom režime. V hybridnom režime môžeme naďalej používať technológie ako napr. STP, SPB, TRILL, OSPF. V normálnom režime však nič takéto neexistuje. V súčasnosti môžeme využívať OpenFlow protokol napríklad na týchto zariadeniach:
Extreme Networks – BlackDiamond X8, Summit X670
HP 2920/3500/3800/5130/5400/5500/5900/5920/5930/8200/10500 Switch Series
Mikrotik routers and switches (podpora zatiaľ len OpenFlow v1.0).

10.4 Návrh fyzickej topológie

Význam SDN závisí od štruktúry na ktorej je aplikovaná. Ak máme k dispozícií dostatočný počet redundantných liniek, môžeme uvažovať o využívaní load-balancingu. Vo veľkej sieti poskytovateľov služieb je veľká výhoda ochrany siete pred hrozbami ako napr. DDoS, nebezpečné stránky obsahujúce škodlivý kód, phishing stránky, kde SDN aplikácia zaistí ochranu už na okraji siete. V malej sieti bude zohrávať SDN hlavnú úlohu skôr v zabezpečení QoS, prípadne k presmerovaniu toku dat cez paketový analyzátor. Pri návrhu novej infraštruktúry sa oplatí investovať do polygonálnej štruktúry siete vďaka čomu môžeme nasadiť nové aplikácie bez toho aby sme museli zasahovať do fyzickej časti siete. Podstatou SDN je totiž manipulovať s tokom dát z centrálneho miesta (kontroléru) bez toho aby sme museli posielať technika prehadzovať káble v prepínačoch.

10.5 Mininet (Emulátor siete)

Aj keď v súčasnosti už veľa zariadení podporuje technológiu OpenFlow, na experimentálne účely nám bude stačiť aj program na emuláciu siete zvaný Mininet. Ten dokáže emulovať sieťové prepínače s podporou OpenFlow verzie 1.0 ale aj verzie 1.3, ďalej koncové stanice a spoje medzi jednotlivými prvkami siete. V prostredí Mininetu je navyše možné posielať ICMP pakety, spúšťať rôzne služby na emulovaných staniciach a pripájať sa k ním, môžeme dokonca nastaviť rôzne parametre linky ako napr. rýchlosť a stratovosť.

11 ZÁVER

V tejto práci sú zhrnuté základné znalosti a myšlienky pri návrhu dátovej siete. Základom je vybrať vhodnú hierarchiu, vymyslieť vhodnú topológiu, ktorá plne vykryje požiadavky na výkon a nakoniec zvoliť vhodné zariadenia ktoré podporujú techniky rozloženia záťaže, agregácie ale aj redundancie liniek. Je jasné že pre malé siete využitie topológie typu MESH nebude zaujímavé, ale pri väčších sietiach je toto využitie veľmi výhodné. Na svete sú totiž techniky rozloženia záťaže ako MSTP, TRILL či SPB vďaka ktorým dokážeme zefektívniť výkon siete a rozložiť záťaž na všetky sieťové prvky. Zachová sa robustnosť siete ktorá je veľmi dôležitá pri súčasnom trende Clusteringu a orientácií aplikácií a služieb do Cloudov.
Zaujímavý je do budúcnosti nástup virtualizovaných sieti (SDN). V praktickej časti práce som otestoval viacero druhov najznámejších SDN riešení (OpenDaylight, Floodlight, HP VAN SDN Controller) ktoré boli zaujímavo spracované ale užívateľsky ťažkopádne. Všetky riešenia napredujú a postupne im pribúda funkcionalita čo naznačuje potrebu tejto technológie. SDN sa už teraz momentálne uplatňujú hlavne v dátových centrách, ale vzhľadom na svoje súčasné možnosti je predpoklad že sa začnú čoraz viac uplatňovať aj v rozľahlejších sietiach poskytovateľov služieb (svoju vlastnú verziu SDN už používa napríklad Google). Podľa mnohých názorov sa dokonca práve SDN riešenie stane v budúcnosti základom každej firemnej siete. Firmy však budú musieť na toto myslieť pri návrhu topológie a zabezpečiť dostatočnú redundanciu liniek.

LITERATÚRA

[1] LAMMLE, Todd. *CCNA: výukový průvodce přípravou na zkoušku 640-802. Vyd. 1.* Brno: Computer Press, 2010, 928 s. ISBN 978-80-251-2359-1.

[2] SEIFERT, Rich. *The all new switch book: the complete guide to LAN switching technologyy. 2nd ed.* Indianapolis: Wiley Publishing, 2008, 784 s. ISBN 978-0-470-28715-6.

[3] MARCHESE, Mario. *QoS over heterogeneous networks: the complete guide to LAN switching technology.* Chichester: John Wiley, 2007, 307 s. ISBN 978-0-470-01752-4.

[4] ŠPIČKA, Vladimír. *Virtualizácia siete bude ďalšia fáza* [online]. 2012, [cit. 20. 11. 2014]. Dostupné z URL: <`http://www.itnews.sk/2013-08-28/c155582-virtualizacia-siete-bude-dalsia-faza`>.

[5] Internet Engineering Task Force (IETF). RFC7177 *Transparent Interconnection of Lots of Links (TRILL)* [online]. 2014, [cit. 23. 11. 2014]. Dostupné z URL: <`http://tools.ietf.org/html/rfc7177`>.

[6] SPONSOR, LAN/MAN Standards Committee of the IEEE Computer Society. *IEEE standard for local and metropolitan area networks media access control (MAC) bridges and virtual bridged local area networks.* New York: Institute of Electrical and Electronics Engineers, 2012, 324 s. ISBN 978-0-7381-7262-0.

[7] SOSINSKY, Barrie A. *Mistrovství – počítačové sítě. Vyd. 1.* Brno: Computer Press, 2010, 840 s. ISBN 978-80-251-3363-7.

[8] PETERKA, Jiří. *Báječný svět počítačových sítí, Část XIX. : Propojování na síťové a aplikační vrstvě* [online]. 2014, [cit. 26. 11. 2014]. Dostupné z URL: <`http://www.earchiv.cz/b06/b1100001.php3`>.

[9] Cisco Systems. *Understanding Rapid Spanning Tree Protocol (802.1w)* [online]. 2006, [cit. 28. 11. 2014]. Dostupné z URL: <`http://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/24062-146.html`>.

[10] Cisco Systems. *Understanding Multiple Spanning Tree Protocol (802.1s)* [online]. 2007, [cit. 28. 11. 2014]. Dostupné z URL: <`http://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/24248-147.html`>.

[11] University of Tennessee. *Selecting a load balancing method* [online]. 2014, [cit. 11. 12. 2014]. Dostupné z URL: <https://help.utk.edu/kb/index2.php?func=show&e=1699>.

[12] KEMP Technologies, Inc. *When to use Round Robin DNS load balancing* [online]. 2014, [cit. 11. 12. 2014]. Dostupné z URL: <http://kemptechnologies.com/emea/load-balancing/round-robin-load-balancing/>.

[13] PAVLIS, Martin., VÁVRA Jan. *Možnosti virtualizace, přínosy virtualizace serverů - Virtualizace v praxi (1. díl)* [online]. 2009, [cit. 11. 12. 2014]. Dostupné z URL: <http://www.systemonline.cz/virtualizace/moznosti-virtualizace-prinosy-virtualizace-serveru.htm>.

[14] BUYYA, R. *High performance cluster computing. Vyd. 1.* New Jersey: Prentice-Hall, 1999, 849 s. ISBN 01-301-3784-7.

[15] Juniper Networks. *QFABRIC SYSTEM* [online]. 2014, [cit. 14. 12. 2014]. Dostupné z URL: <http://www.juniper.net/assets/us/en/local/pdf/datasheets/1000393-en.pdf>.

[16] GROTH, David. *Network study guide. 4th ed.* London: SYBEX, 2005, 519 p. ISBN 07-821-4406-3.

[17] ČÍKA, P. *Multimediální služby.* Brno: Vysoké učení technické v Brně, 2012. s. 1-127. ISBN: 978-80-214-4443- 0.

[18] BOUŠKA, Petr. *QVLAN - Virtual Local Area Network: Co je to VLAN* [online]. 2014, [cit. 15. 12. 2014]. Dostupné z URL: <http://www.samuraj-cz.com/clanek/vlan-virtual-local-area-network/>.

[19] BRZOZA, Martin. *Open Flow, Open vSwitch* [online]. 2013, [cit. 27. 5. 2014]. Dostupné z URL: <http://wh.cs.vsb.cz/sps/images/8/87/OpenFlow.pdf>.

[20] OPEN NETWORKING FOUNDATION: *OpenFlow Switch Specification Version 1.0.0 - Wire Protocol 0x01.* [online]. 2009, [cit. 12. 5. 2015]. Dostupné z URL: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf>.

[21] KUBICA, Tomáš. *Vývoj aplikací pro HP VAN SDN kontrolér.* [online]. 2014, [cit. 12. 5. 2011]. Dostupné z URL: <http://www.netsvet.cz/cs/download/hp-sdn-python-lab-1.02.pdf>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

IETF	Internet Engineering Task Force - Komisia techniky internetu
IEEE	Institute of Electrical and Electronics Engineers - Inštitút pre elektrotechnické a elektronické inžinierstvo
TRILL	Transparent Interconnection of Lots of Links (protokol ktorý rieši rozloženie záťaže a slučky v sieti)
SPB	Shorthest Path Bridging (protokol ktorý rieši rozloženie záťaže a slučky v sieti)
DiffServ	Differentiated Services (Diferencované služby v spojení s kvalitou služieb)
IntServ	Integrated Services (Integrované služby v spojení s kvalitou služieb)
SDN	Software-Defined Networking (Softvérovo definovaná sieť)
IS-IS	Intermediate System to Intermediate System (smerovací protokol k určovaniu najlepších ciest)
MC-LAG	Multi-Chassis Link Aggregation Group (protokol určený k agregácií liniek rôznymi zariadeniami)
ACL	Access Control List (Zoznam pre riadenie prístupu)
VLAN	Virtual Local Area Network (Virtuálna sieť)
MPLS	Multiprotocol Label Switching (prepínací protokol)
QoS	Quality of Service - kvalita služieb
DDoS	Distributed Denial of Service (útok ktorý znefunkční služby)
STP	Spanning Tree Protocol (protokol pre zamedzenie slučiek v sieti)
RSTP	Rapid Spanning Tree protocol - protokol pre zamedzenie slučiek v sieti
MSTP	Multiple Spanning Tree Protocol -protokol pre zamedzenie slučiek v sieti

ZOZNAM PRÍLOH

A SDN - SOFTWAROVĚ DEFINOVANÁ SÍŤ

A.1 Cíl

Cílem úlohy je seznámit se s kontrolérem SDN sítě HP VAN SDN a s jeho konfigurací, ověření jeho funkce a schopností řídit provoz sítě na přepínačích podporujících protokol OpenFlow.

A.2 Vybavení pracoviště

- počítač s operačním systémem Windows
- software HP VAN SDN kontrolér a Mininet
- router Mikrotik s podporou protokolu OpenFlow 1.0.

A.3 Úkoly

1. Otestovat funkčnost kontroléru a citlivost sítě na jeho výpadek.
2. Seznámit se s kontrolérem v hybridním režimu a s řešením smyček v síti.
3. Testovat průtok dat síti nejkratší cestou.
4. Modifikovat výchozí proudění toku dat pomocí skriptů.
5. Nastavit Mikrotik router pro komunikaci s kontrolérem.

A.4 Teoretický úvod

SDN (Software Distributed Network) tvoří síťové prvky řízené kontrolérem, který zná celou topologii sítě, neustále komunikuje s jednotlivými prvky a realizuje programování jejich chování. Díky tomu je schopen celou síť efektivně spravovat. Jde o centralizované pojetí síťové infrastruktury oproti dřívějšímu distribuovanému. Kontrolér může snadno nasadit potřebná průtoková pravidla napříč velkou a rozlehlou sítí. Kontrolér je v podstatě síťový middleware který umožňuje při správě abstrahovat od konkrétních komponent – přepínačů, routerů, load-balancerů nebo firewallů. Kontrolér komunikuje se síťovými prvky prostřednictvím protokolu OpenFlow, kterým lze u zařízení s jeho podporou diktovat pravidla pro směrováni dat v síti. Protože kontrolér řídí celou síť, jeho selhání je kritické. Jeho funkce bývá provozována v clusteru (na studijní účely je dostačující kontrolér který není součástí clusteru), nebo je kontrolér provozován virtuálním strojem který má zajištěnou vysokou dostupnost. Síť je standardně nakonfigurována tak aby byla schopna směrovat provoz i při krátkodobém výpadku kontroléru.[1]

U námi testovaného kontroléru je možné nastavit dva základní režimy chování:

Reaktivní chování
v síťovém prvku nejsou po spuštění SDN žádné instrukce, s každým novým tokem je informován kontrolér. Ten doručí paket a nastaví instrukce tak že pakety v síti už bude zpracovávat ASIC přepínačů úplně sám. Díky tomu má kontrolér přehled o každém toku v síti.[1]

Proaktivní chování
kontrolér dopředu nastaví tok zpracovávaný ASIC čipem na základe vnějších událostí. Využití je hlavně při statickém nastavení sítě kde nevyžadujeme dynamické aplikace řízení toku dat v reálném čase.[1]

Hybridní chování
Tento režim nenastaví jako poslední pravidlo „forward Controller“, ale „forward Normal“. Prvky zpracovávají data i vlastním „control plane“ díky čemu můžeme využívat stávající síť již s aplikovanými technologiemi jako například STP, OSPF. Specifickým pravidlem pak můžeme posílat konkrétní data do kontroléru kde můžeme aplikaci zajistit například ochranu sítě nebo i QoS.[1]

OpenFlow přepínač ve verzi protokolu 1.0 obsahuje:
průtokovou tabulku - ta obsahuje sadu flow záznamů (hodnoty hlaviček paketů pro srovnáváni s filtrem), počítadlo aktivity a jednu nebo více akcí k aplikováni při shodě s pravidlem. Všechny zpracované pakety přepínačem jsou srovnávány s průtokovou tabulkou. Když se najde shoda, paket je zpracován podle akce zadané průtokovým pravidlem. Když se shoda nenajde, paket je posílán zabezpečeným kanálem do kontroléru sítě. Ten rozhodne o tom jak naloží s daným paketem bez flow shody.[2]
zabezpečený kanál ke kontroléru sítě - prvky vSwitch komunikují běžně s kontrolérem sítě přes zabezpečený kanál, není to však podmínkou.[2]

OpenFlow protokol podporuje tři typy zpráv:
Controller-to-Switch - zprávy vytváří kontrolér sítě a slouží k přímému ovládání nebo k zjištění stavu přepínače.
Asynchronous - zprávy vytváří přepínač a informuje nimi kontrolér o síťových událostech a o změnách stavu přepínače.
Symmetric - zprávy vytváří přepínač i kontrolér sítě a jsou vysílána bez žádostí.[2]

Záznam v průtokové tabulce má 3 pole:

- hlavička definující tok
- akce dle které se pakety zpracovávají
- statistiky o počte paketů a čas od zavedení pravidla nebo od poslední shody paketu.[3]

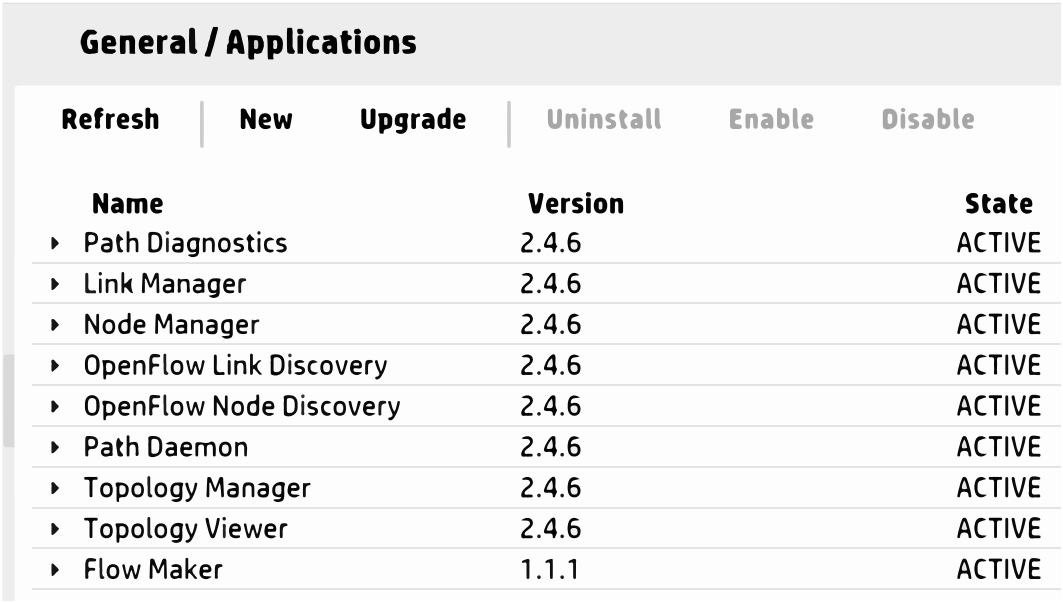
Zjistit shodu toku můžeme podle:

- Vrstvy 1 - Tunel ID, Vstupní port, QoS priorita
- Vrstvy 2 - MAC adresa, VLAN ID, ethernet typ
- Vrstvy 3 - IPv4/IPv6, ARP
- Vrstvy 4 - TCP/UDP , ICMP.[3]

Akce k vykonávaní:

- výstup na port
- zahození
- manipulace s paketem
- zaslání do kontroléru.[3]

Základní moduly které HP VAN SDN kontrolér obsahuje:



Obr. A.1: moduly HP VAN SDN kontroléru

Link Manager

Modul je zodpovědný za objevování spojů mezi přepínači OpenFlow. Aplikace pravidelně generuje speciální pakety které zasílá do síťových prvků zprávou Packet_Out. V nich jsou identifikátory DPID. Po přijetí paketu OpenFlow prvkem je zaslán tento paket do kontroléru zprávou Packet_In. Tímto se vyhledají linky mezi prvky a informace se uloží do databáze.[2]

Topology Manager

Modul bere informace z Link Manageru, sbírá hlášení s prvků (typicky pád linky). Poskládá mapu sítě a vypočítá nejkratší trasu mezi dvěma koncovými body.[2]

Node Manager
Modul, který poslouchá ARP, DHCP a IP pakety. Na základe získaných informací udržuje informace o návaznosti sítě, tedy kde je k čemu která stanice připojena a jakou má MAC a IP adresu.[2]

Path Daemon
Tento modul se stará o generování reaktivních pravidel. Pokud dostane z OpenFlow přepínače paket zabalený v správě Packet_In, z Node manageru zjistí, kde se nachází stanice, z Topology Manageru zjistí nejkratší cestu. Následně tento modul naprogramuje celou cestu napříč všemi uzly sítě. Modul se podle konfigurace řídí MAC adresami nebo IP adresami. Je důležitou aplikací reaktivního režimu.[2]

Path Diagnostics
Modul diagnostiky sítě. Umí vstřikovat uměle vytvořené pakety s identifikátorem a libovolnou zdrojovou a cílovou adresou. Každý prvek sítě informuje kontrolér o úspěšném průchodu paketu. Jeho funkce je podobná L2 traceroute a IP SLA a to pro všechny body průběžně po cestě.[2]

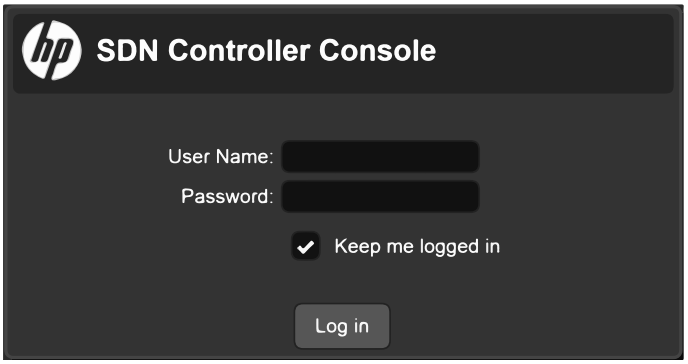
Topology Viewer
Modul zobrazuje grafickou reprezentaci topologie v GUI kontroléru.[2]

Mininet
Protože výhody SDN se ukazují hlavně při rozlehlých sítích, bylo by zbytečně komplikované vytvořit síť z velkým počtem prvku pro studijní účely. Stačí nám program pro emulaci sítě zvaný Mininet. Ten používá skripty napsané v Pythonu, které je možné libovolně editovat pro vytvoření vlastní sítě.
Důležité příkazy:
mn - -topo single,2 - -controller=remote,ip=192.168.1.101 - vytvoří síť s jedním přepínačem a dvěma hosty a nastaví používaní kontroléru s IP adresou 192.168.1.101.
pingall - pošle icmp paket mezi všemi hosty.
h1 ping h2- pošle icmp z h1 to h2
iperf h1 h2 - otestuje se šířka pásma mezi hosty h1 a h2
exit - ukončí spuštěný skript emulované sítě

A.5 Postup řešení

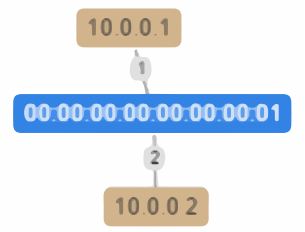
A.5.1 Úkol č.1

1. Spustíme VirtualBox a nastartujeme virtuální stroj HP VAN SDN Controller.
2. Pomocí programu Putty se připojíme k službě SSH na IP 192.168.1.101.
   Výchozí přihlašovací jméno **root** a heslo **sdn**
3. Počkáme přibližně 5 minut než naběhnou všechny služby kontroléru. (pomocí příkazu **top** můžeme odhadnout zda kontrolér běží a to tak že virtuální stroj využívá přibližně 2,8 GB paměti a vytížení procesoru klesne pod 20 procent).
4. Pomocí libovolného webového prohlížeče otevřeme webovou aplikaci pro správu kontroléru:
   https://192.168.1.101:8443/sdn/ui/



Obr. A.2: GUI login do kontroléru

   Výchozí přihlašovací jméno je **sdn** a heslo **skyline**
5. V SSH konzole spustíme příkaz:
   mn --topo single,2 --controller=remote,ip=192.168.1.101.
   Vytvoří se síť jednoduché struktury:



Obr. A.3: Síť skriptu 1.py

6. Zasílejte pakety ICMP z hostu 1 na host 2 příkazem **h1 ping h2**.

7. Opakovaně se připojte pomocí programu Putty k službě SSH na IP 192.168.1.101 (root/sdn).
8. Shoďte službu kontroléru příkazem **service sdnc stop**.
9. Sledujte ICMP pakety zasílané z h1 na h2. Po čase se spojení rozpadne. Zapište čas od vypnutí služby po výpadek spojení.
10. Službu opětovně nastartujte příkazem **service sdnc start**,
11. V GUI kontroléru zkontrolujte v sekci:
    Configurations - com.hp.sdn.ctl.path.impl.PathDaemon - hard.timeout.
    Náš zapsaný čas by měl souhlasit s tímto nastavením. Díky tomuto parametru může běžet síť i po výpadku kontroléru po námi nastavený čas. Tento čas totiž udává jak dlouho je průtokové pravidlo v platnosti na přepínači po zavedení.
12. Jednoduchou síť ukončete příkazem **exit** v programně Mininet

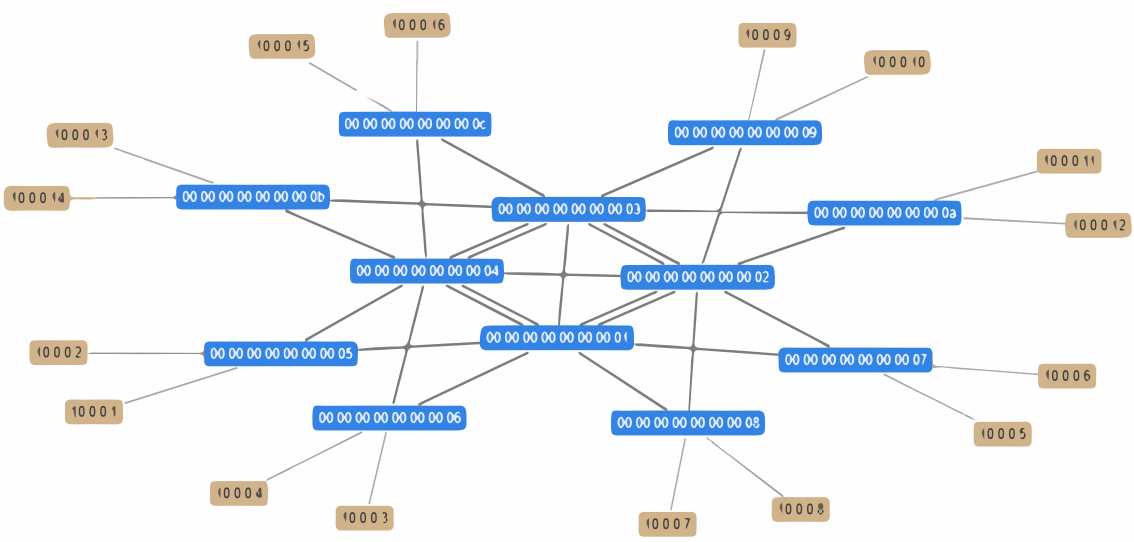
A.5.2 Úkol č.2

1. Nastartujte skript **python /mininet/datacenter_net.py**. Vytvoří se polygonální síť.
2. Zašlete příkaz **pingall** a sledujte co se děje v mapě sítě.
3. Nyní přepnete kontrolér do hybridního režimu.
   V GUI kontroléru změňte nastavení:
   Configuration - com.hp.sdn.ctl.of.impl.PathDiagnosticManager - hybrid mode = true.
4. Znovu zašlete **pingall** a sledujte na mapě sítě co se děje.
   Najednou se síť začne rozpadat protože kontrolér už neřeší smyčky v síti a síť se řídí vlastními pravidly na přepínačích kde není aplikovaná technologie STP.
   V programu Mininet po zasláni příkazu např. **h1 ping h10** vidíte že se zvýšili odezvy, paketu vůbec neprocházejí nebo je při nich značka DUP co znamená duplicitu způsobnou smyčkami.
5. Zkontrolujte flow tabulku na jednom z přepínačů v sekci **OpenFlow - Flows** pro **Data Path ID: 00:00:00:00:00:00:00:01**.
   Zapište „Akce“ které jsou vykonány pro ARP,BDDP,DHCP. (Přibyla jedna která způsobí všesměrovou bouři v síti).
6. Síť shoďte příkazem **exit** a kontroléru zakážeme **hybridní řežím na false** (podle kroku 3).

A.5.3 Úkol č.3

1. V SSH konzole nastartujte skript **python /mininet/custom/2.py**. Vytvoří se síť s několika přepínači, kde některé jsou spojeny redundantními linkami

s nižší odezvou (kolem 200 ms) a nižší propustností (10 Mbit/s). Kontrolér vyhledá a zmapuje všechny přepínače v síti a zobrazí propojení mezi nimi.



Obr. A.4: Síť skriptu 2.py

2. Nyní můžete vidět, že kontrolér vyhledal síťové prvky OpenFlow, nevyhledal však koncové stanice v síti. Na zmapování stanic v síti musí modul kontroléru „Node Manager“ obdržet nějaké pakety ARP nebo IP z přepínačů. To dosáhneme tím že v programu Mininet zašleme příkaz **pingall.**
3. V GUI kontroléru v sekci OpenFlow Topology si vyberte dvě stanice mezi kterými chcete zkontrolovat průtok paketů sítí. Např. stanici h7 a h13.
4. Upravte zobrazení mapy stisknutím klávesy N a P.
   Najděte stanici s IP adresou 10.0.0.7 (poslední bajt zodpovídá číslu stanice). Označte jí ikonkou **Src**.
   najdeme stanici s IP adresou 10.0.0.13. Označte jí ikonkou **Dst**.
   Pokud máte vybraný „Shortest Path“ tak vám kontrolér zobrazí červenou barvou nejkratší cestu kterou využívá i při programování prvků.
5. Zdvojené propoje mezi přepínači jsou nastaveny jako linky s rychlostí 10Mbit a odezvou kolem 200ms. Zkontrolujte, či kontrolér vybral co nejmenší počet skoků v síti a přitom vybral nejkvalitnější trasu (vyhnul se pomalým spojům).
6. Otestujte příkazem **h7 ping h13** odezvu a příkazem **iperf h7 h13** rychlost linky mezi stanicemi.
7. Vypněte linku mezi s2 (00:00:00:00:00:00:00:02) a s4 (00:00:00:00:00:00:00:04) příkazem **link s2 s4 down**.
   Kontrolér je změní trasu na pomalou linku aby se vyhnul dalšímu skoku sítí.
8. Znovu otestujte odezvu a rychlost mezi stanicemi.

A.5.4 Úkol č.4

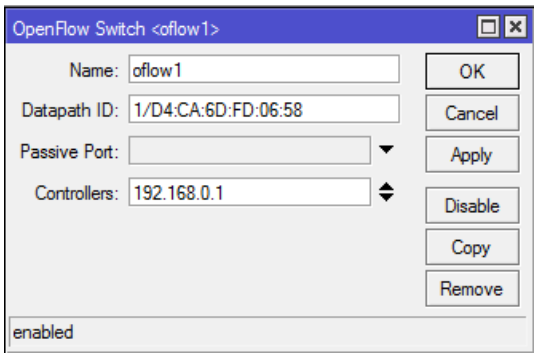
Tento úkol má za úlohu ukázat jednoduchost využívání různých aplikací pro kontrolu sítě. V GUI kontroléru je možné instalovat aplikace obdržené z „HP App Store“, my však využijeme ukázkové skripty napsané programovacím jazykem Python. Tyto skripty využívají knihovnu *hpsdnclient* která usnadňuje připojení k API kontroléru.

1. V SSH konzole spustíte opět skript **python /mininet/custom/2.py**.
2. Zadáte **pingall**. Z druhé SSH konzoly spustíte skript:
   python /sdn/scripts/where.py --ip 10.0.0.5[1]
   Skript vypíše informace o tom, na kterém přepínači a portu se nachází, jakou má MAC adresu.
3. Zkontrolujte přes GUI v mapě topologie zda je daný host doopravdy připojen k danému uzlu sítě.
   (Klávesa P zapne zobrazování portů na mapě, klávesa N přepne zobrazení stanic na IP nebo MAC)
4. Pomocí SSH konzoly spustíte skript:
   python /sdn/scripts/stanice.py --ip 10.0.0.5[1] (ujistěte se že ve Windowsu běží Xming server)
5. Aplikace zobrazí všechny známe stanice a informace o připojení do přehledné tabulky. Okno zaznamenejte.
6. Nyní si zvolte dvě stanice v síti mezi kterými začneme zasílat ICMP. Např. **h2 ping h5**. Vidíte že stanice navzájem komunikují.
7. Spusťte skript kterým zakážeme jedné ze stanic komunikovat po určitou dobu:
   python /sdn/scripts/kill.py --ip 10.0.0.5 - -penalty 30[1].
8. Do přepínačů je nyní zavedeno pravidlo s platností času penalty=30s. Ověřte přerušení spojení příkazem **h2 ping h5**. Najdete přes GUI pravidlo na přepínači které skript zavedl pro umlčení stanice h5.
9. Zkontroluje zda po 30 sekundách se vrátí vše do normálu.
10. Dalším skriptem:
    python /sdn/scripts/killGUI.py[1]
    otestujte pohodlnost „killování“ stanic z okna.
11. Zkuste upravit skript **killGUI.py** libovolným editorem tak, aby umlčel stanice jen na 10 vteřin.

A.5.5 Úkol č.5

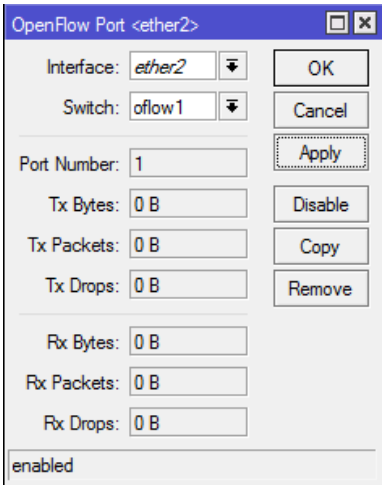
V tomto úkolu jde jen o ukázku rychlosti konfigurace OpenFlow na hardwaru, rychlosti spojení a odezvy kontroléru a správnosti identifikace portů, verze zařízeni a verze použitého protokolu OpenFlow.

1. Pomocí aplikace Winbox se připojte k směrovači Mikrotik přes IP adresu **192.168.0.2**. Výchozí jméno: **admin** heslo:**bez hesla**
2. V menu vyberte **OpenFlow**, zobrazí se okno kde v záložce **Switches** přidáme připojení ke kontroléru: **Name: oflow1 Controllers: 192.168.0.1** a stiskněte OK.



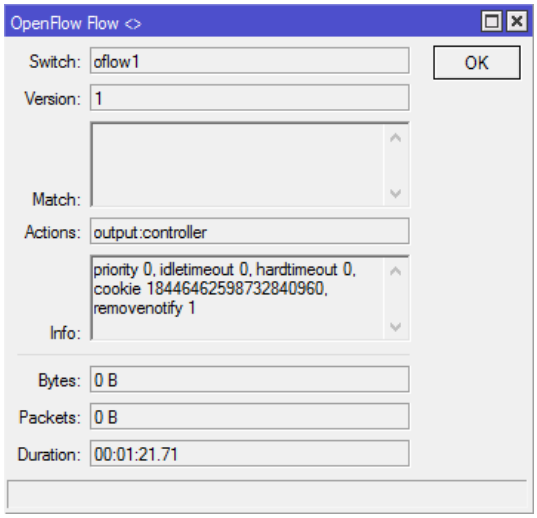
Obr. A.5: vytvoření OpenFlow přepínače v Mikrotiku

3. V záložce **Ports** zvolíme:
   New Open Flow Port - vybete **Interface: ether2** a **Switch: oflow1**
   New Open Flow Port - vyberte **Interface: ether2** a **Switch: oflow1**



Obr. A.6: Přidávaní portu do OpenFlow na Mikrotiku

4. Mikrotik se okamžitě spoji s kontrolérem a přidá si výchozí pravidlo které najdeme v záložce „Flows“. Zapište pravidlo.



Obr. A.7: Přidané flow pravidlo v Mikrotiku

5. V GUI kontroléru zvolte OpenFlow Monitor a najděte připojený router Mikrotik. V záložce „Flows“ najděte Flow pravidlo a srovnejte ho s pravidlem zapsaným dříve.
6. Zkontroluje shodu i v záložce „Summary“ a „Ports“.

A.6 Kontrolní otázky

1. Proč nedojde k všesměrové bouři způsobené ARP paketama když je v síti několik smyček?
2. Vysvětlete zprávy typu Packet_In a Packet_Out. Jaký je mezi nimi rozdíl?
3. Co se stane v případe výpadku některé linky?
4. Jak proudí data sítí v základním režimu?
5. V čem je výhoda SDN sítě oproti běžné síti.

A.7 Seznam zkratek

SDN Software-Defined Network

ASIC Application Specific Integrated Circuit (integrovaný obvod pro pro konkrétní využití)

STP Spanning Tree protocol (protokol odstraňující slučky v síti)

OSPF Open Shorthest Path First (směrovací protokol)

QoS Quality of Service (kvalita služeb)

GUI Graphical User Interface (grafické uživatelské rozhraní)

A.8 Literatura

[1] KUBICA, Tomáš. *Vývoj aplikací pro HP VAN SDN kontrolér.* [online]. 2014, [cit. 12. 5. 2011]. Dostupné z URL: <`http://www.netsvet.cz/cs/download/hp-sdn-python-lab-1.02.pdf`>.

[2] OPEN NETWORKING FOUNDATION: *OpenFlow Switch Specification Version 1.0.0 - Wire Protocol 0x01.* [online]. 2009, [cit. 12. 5. 2015]. Dostupné z URL: <`https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf`>.

[3] BRZOZA, Martin. *Open Flow, Open vSwitch* [online]. 2013, [cit. 27. 5. 2014]. Dostupné z URL: <`http://wh.cs.vsb.cz/sps/images/8/87/OpenFlow.pdf`>.

B INŠTALÁCIA POUŽITÉHO SOFTWARE

B.1 Konfigurácia Virtualboxu

Vo VirtualBoxi vytvoríme virtuálny stroj ktorému pridelíme:

- Pamäť - 3 GB (minimun)
- Disk - 4 GB
- Adapter 1 - Bridged Adapter
- Adapter 2 - Host-Only Adapter
- Názov - HP VAN SDN Controller
- Typ - Linux
- Verzia - Ubuntu (64 bit

V globálnych nastaveniach VirtualBoxu zmeníme nastavenie Virtualbox Host-Only Adaptéru a to tak že zmeníme IP adresu na 192.168.1.100.

B.2 Inštalácia OS Ubuntu 12.04.5 LTS

Spustíme inštaláciu OS v minimálnej konfigurácii.
Nainštalujeme z inštalátora OpenSSH server.
Po spustení odblokujeme root účet príkazmi - **passwd root**, opakovane zadáme heslo **sdn** a následne zadáme príkaz **passwd -u root**.
Nastavíme konfiguráciu sieťových adaptérov v súbore **/etc/network/interfaces** : eth0 na dhcp a eth1 na 192.168.1.101/24
Pridáme smerovacie pravidlo pre úlohu č.5: **ip route add 192.168.0.0/24 via 192.168.1.100 dev eth1**

B.3 Inštalácia Mininet 2.2.1

Nainštalujeme balíček Mininet:
apt-get install mininet
Vytvoríme adresár */mininet/custom/*
Nakopírujeme do tohto adresára skripty zo zložky *mininet_scripts*.

B.4 Inštalácia HP VAN SDN Controller

Nainštalujeme potrebné balíčky:
apt-get update
apt-get install python-software-properties ubuntu-cloud-keyring
Pridáme potrebný repozitár:
add-apt-repository cloud-archive:icehouse.
Nainštalujeme licenčný server:
apt-get update
apt-get install keystone
Pomocou sftp nakopírujeme inštalačný balíček a nainštalujeme:
dpkg -–unpack hp-sdn-ctl_2.4.6.0627_amd64.deb
apt-get install -f
Spustíme službu sdn kontroléru:
service sdnc start
Spustíme skript ktorý vytvorí základného užívateľa:
/opt/sdn/admin/config_local_keystone
Ak sa neotvára web GUI a píše chybu SSL, treba vygenerovať nový SSL kľúč:
mv keystore keystore.orig
mv truststore truststore.orig keytool -genkey -alias serverKey -keyalg rsa -keysize 2048 -keystore keystore -validity 1780
Zadáme heslo **skyline**.
keytool -exportcert -keystore keystore -alias serverKey -file serverkey.cer
keytool -importcert -trustcacerts -keystore truststore -file serverkey.cer -alias CARoot
Zadáme heslo **skyline**.
mkdir/sdn/scripts/ a zkopírujeme z DVD skripty zo zložky *controller_scripts*.
Doinštalujeme zopár závislostí pre úlohu č.4:
apt-get install python-pip python-tk ebtables tshark bridge-utils
pip install hp-sdn-client

B.5 Konfigurácia Mikrotiku a počítača

Na poslednú úlohu treba nastaviť jeden z ethernetov počítača na IP 192.168.0.1/24, ktorý je prepojený káblom k vyresetovanému Mikrotik smerovaču do eth1.
Konfigurácia Mikrotiku je následovná:
/ip address add address=192.168.0.2/24 interface=ether1
/ip route add dst-address=192.168.1.0/24 gateway=192.168.0.1
Do PC nainštalujeme z DVD zložky *tools* **Xming** aplikáciu ktorá umožní ssh X.

C OBSAH PRILOŽENÉHO DVD

Na priloženom DVD sa nachádzajú tieto dáta:

V adresári *BP*:
Bakalárska práca + laboratórna úloha vo formáte docx.

V adresári *install_files*:
HP VAN SDN Controller
VirtualBox
Ubuntu 12.04.5 LTS

V adresári *virtualbox_machine:*
Hotová inštalácia HP VAN SDN Controller+mininet naOS Ubuntu 12.04.5 LTS.

V adresári *tools:*
Program Putty - klient SSH pre pripojenie ku kontroléru.

V adresári *mininet_scripts:*
Skripty použité na vygenerovanie siete programom Mininet.
V adresári *controller_scripts:*
Python skripty pre API komunikáciu s SDN kontrolérom.