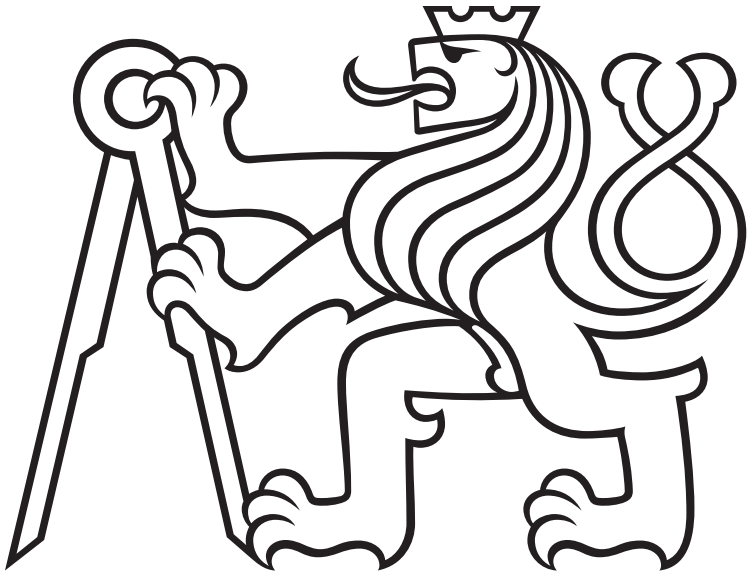
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra telekomunikační techniky



Vytvoření případových studií ICT systémů a sítí ve virtuálním prostředí

Development of a Case Studies of ICT Systems and Networks in a Virtual Environment

BAKALÁŘSKÁ PRÁCE

Autor:	Ondřej Mifka
Studijní program:	Elektronika a komunikace
Vedoucí práce:	doc. Ing. Leoš Boháč, Ph.D.
Rok:	2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Mifka** Jméno: **Ondřej** Osobní číslo: **483893**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra telekomunikační techniky**

Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Vytvoření případových studií ICT systémů a sítí ve virtuálním prostředí

Název bakalářské práce anglicky:

Development of a Case Studies of ICT Systems and Networks in a Virtual Environment

Pokyny pro vypracování:

Zhodnoťte současně existují platformy určené pro virtuální nasazení systémů ICT a datových sítí a následně vypracujte soubor několika případových pokročilých ICT systémů a sítí ve virtuálním prostředí EVE-NG. Tyto případové studie musí být vypracované s odpovídající dokumentací, která bude vysvětlovat funkci dílčích částí IT systému, sítí a protokolů tak, aby je bylo možné použít pro výuky a další výzkum. Zaměřte se primárně na technologie jako je SDN, SD-WAN, MPLS, apod. Zmapujte také soudobé trendy v nových technologiích a pokuste se je dle možností reflektovat i v provedených případových studiích. Taktéž se zamyslete nad možností využití virtualizační platformy i v jiných příbuzných oborech, jakými jsou např. automobilový průmysl nebo moderní průmyslové řídicí systémy.

Seznam doporučené literatury:

[1] SÁNCHEZ-MONGE, A.: MPLS in the SDN era. Beijing: O'Reilly, [2015]. ISBN 978-1-49190-545-6.
[2] GOOLEY, J. - YANCH, D. - SCHUEMANN, D. - CURRAN, J.: Cisco Software-defined Wide Area Networks: Designing, Deploying and Securing Your Next Generation WAN with cCsco SD WAN. 1. Hoboken: Pearson Education, 2020. ISBN 978-0-13653-317-7.
[3] GREGG, M.: The Network Security Test Lab: A Step-by-Step Guide. Indianapolis, IN: Wiley, [2015]. ISBN 1-11898-705-5.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Leoš Boháč, Ph.D. katedra telekomunikační techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **09.02.2022** Termín odevzdání bakalářské práce: **15.08.2022**

Platnost zadání bakalářské práce: **30.09.2023**

doc. Ing. Leoš Boháč, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. Datum převzetí zadání

Podpis studenta

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně. Uvedl jsem veškeré použité informační zdroje v souladu s Metodickým pokynem č. 1/2009 o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 20.5.2022

.....
autor

Poděkování

Chtěl bych především poděkovat Vojtěchu Richterovi za rady a konzultace v praktické části práce. Dále bych rád poděkoval doc. Ing. Leoši Boháčovi, Ph.D. za vedení práce a rady ke směřování práce správným směrem. A moje poslední díky patří mým rodičům za podporu během školy, protože obětovali vše, abych mohl být až zde.

Ondřej Mifka

Abstrakt

Práce se zabývá virtualizací v sítích a její možnosti využití s praktickými ukázkami. První část se zabývá výčtem vybraných možností, popisem jejich principu a porovnání jejich využití v sítích. Následuje vytvoření a popis praktických úloh využívající jednu z možností virtualizace z první části práce, EVE-NG. Úlohy se soustředí na technologie MPLS a SD-WAN. Zároveň byl kladen důraz na možnost jejich využití při výuce. Finální část práce popisuje možnosti využití fyzických i virtuálních sítí v reálných případech, zvláště pak v dopravě a průmyslovém řídicím systému.

Klíčová slova

EVE-NG, SD-WAN, MPLS, virtuální sítě, softwarově definované sítě

Abstract

This thesis focuses on virtualization used in computer networking and its use cases. The first part looks into various options of virtualization. It defines basic principles and compares its practical uses. The next part introduces practical applications using one of the options from part one, EVE-NG. In total, a set of labs was created with the main focus on MPLS, SD-WAN and STP. Labs were created with usage during lectures in mind. The final part shows the options of using both physical and virtual networks in real cases, specifically in transport and industrial control systems.

Keywords

EVE-NG, SD-WAN, MPLS, virtual network, software defined network

Obsah

Seznam obrázků

Seznam zkratek

API – application programming interface
BPDU – bridge protocol data units
CCNA – Cisco Certified Network Associate
DDoS – distributed denial-of-service
ETSI – European Telecommunications Standards Institute
EVE-NG – Emulated Virtual Environment - Next Generation
FHRP – first hop redundancy protocol
IEEE – Institute of Electrical and Electronics Engineers
IGP – interior gateway protocol
ISIS – intermediate system to intermediate system
LACP – link aggregation control protocol
LDP – label distribution protocol
MAC – media access control
MPLS – multi protocol layer switching
MST – multiple spanning trees
NFV – network functions virtualization
NFVIS – Network Functions Virtualization Infrastructure Software
OSPF – open shortest path first
PAGP – port aggregation protocol
PVST – per vlan spanning tree
RIP – routing information protocol
RPVST – rapid per vlan spanning tree
RSTP – rapid spanning tree protocol
SDN – software defined network
STP – spanning tree protocol
SVI – switch virtual interface
SD-WAN – software defined - wide area network
VMM – virtual machine manager
VNF – virtualized network functions
VPC – virtual personal computer

Úvod

Internet se stal nedílnou součástí každodenního života. Jeho využití se již dávno neomezuje na výpočetní centra a osobní počítače. Přibývá stále více zařízení propojených chytrými sítěmi k internetu, ať už v průmyslovém prostředí nebo v domácnostech. Se zvýšeným počtem zařízení se zvyšuje i zatížení přenosové sítě a je nutné efektivněji sbírat a zpracovávat data. Virtualizace je tedy logický vývojový krok, který otevírá nové možnosti nejenom v tomto směru.

Virtualizace obecně není nový koncept, její využití v sítích však poměrně ano. I když nemusíme zatím znát její plný potenciál, můžeme využívat její výhody k zajištění současných potřeb. Vedlejším cílem práce je přiblížení konceptu virtualizace, popis několika variant a ukázka možného využití ve dvou vybraných oborech, automobilový průmysl a průmyslové řídící systémy. Hlavním cílem je vytvoření úloh, které bude možno využít při výuce a dalším výzkumu.

První kapitola práce se věnuje virtualizaci. Představí její základní myšlenku a ukáže tři vybrané varianty. Na konci pak porovná možnosti využití jednotlivých variant.

Ve druhé kapitole se nachází představení praktických úloh. Ke každé z úloh je uvedena myšlenka úlohy, se kterou byla navrhována. Následně je pro jednotlivé použité technologie uveden teoretický základ vhodný pro lepší pochopení. Závěr kapitoly je věnován popisu problémů, které bylo nutné překonat při vytváření úloh. Ten slouží zároveň i jako případné dovysvětlení některých rozhodnutí učiněných při návrhu a realizaci.

Ve třetí a závěrečné kapitole je krátký náhled do možnosti využití virtuálních platforem ve vybraných odvětvích jako je doprava a průmyslové řídící systémy. Vzhledem k relativně malému stáří technologie je zde uvedeno pár příkladů, kde lze očekávat větší zapojení sítí, zvláště pak využívajících virtualizaci.

Kapitola 1

Virtualizace v sítích

První zmínky o virtualizaci pochází z přelomu 60. a 70. let minulého stolení. Jedná se o prostředek umožňující přístup uživateli ke zdrojům i jinak než pouze fyzicky. Hlavní důvod pro využívání virtualizace je možnost využívat softwarové aplikace či virtuální zařízení na jednom univerzálním hardwarovém základu. Z pohledu komunikačních technologií se často zajímáme právě o možnost virtualizovat různá zařízení a software pro potřebu testování či pro potřeby nasazení do reálné sítě. Pro virtualizaci nejčastěji používáme hypervizor.

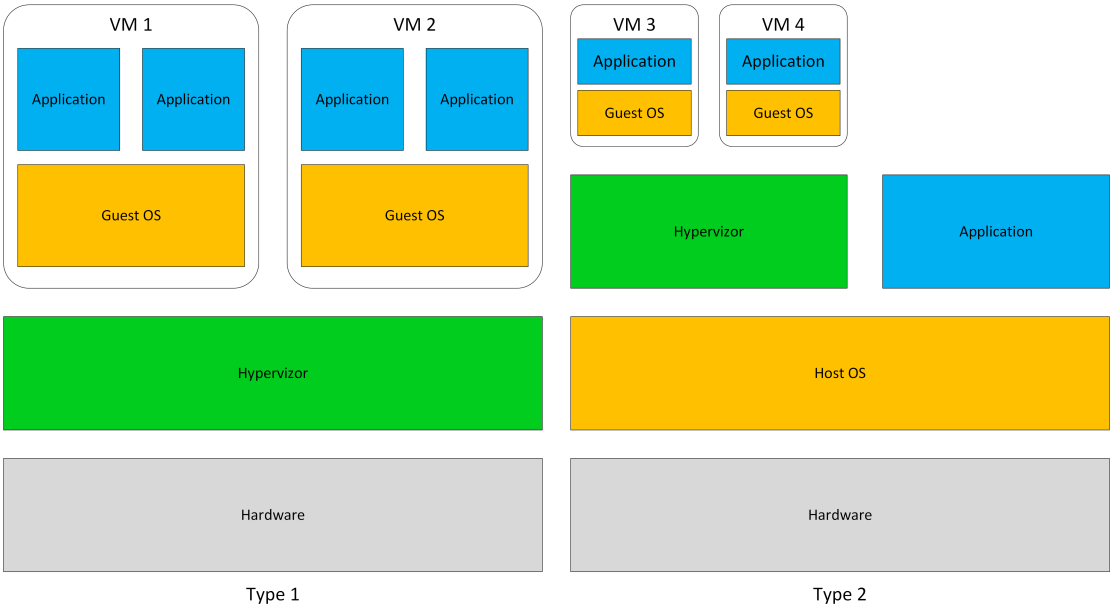
Hypervizor je speciální software, který dokáže vyvolat dojem přímého přístupu k hardwaru z nadřazeného systému. Tím umožňuje dělit přístup do zdrojů tak, že můžeme spouštět více aplikací, než na kolik bychom měli zdrojů při přímém přístupu ke zdrojům.

Existuje více možností, jak lze podle různých parametrů virtualizaci rozdělit. Tato práce se zaměří na tři typy virtualizace podle jejich použití: virtualizace s využitím hypervizora; virtualizační emulační software a network functions virtualization.

1.1 Virtualizace s využitím hypervizora

První typ virtualizace je virtualizace s použitím hypervizora. Dle Roberta P. Goldberga rozlišujeme hypervizory na Type 1 a Type 2 [1]. Type 1 hypervizor běží přímo na hostitelském hardwaru. Type 2 hypervizor je naopak spuštěn nad plnohodnotným operačním systémem. Oba typy vytváří virtuální prostředí, rozdíly mezi nimi najdeme však právě v operačním systému nad kterým je zapnutý a s ním související alokaci hardwarových zdrojů.

Type 1 provádí tuto alokaci přímo sám, tím pádem nedochází k případnému omezení ze strany hostitelského systému, musí však tyto operace umět, což zvyšuje jeho složitost. Zároveň se připravujeme o možnost spustit na jednom zařízení paralelně více hypervizorů. Type 2 dostává už jednou alokované zdroje a dále je distribuuje, což zjednodušuje jeho práci a jelikož pod ním existuje operační systém, tak je instalace jakéhokoliv pomocného softwaru určeného například pro odstraňování chyb jednodušší. Zároveň to znamená uživatelsky přívětivé prostředí, které běžným uživatelům může více vyhovovat.



Obrázek 1.1: Znázornění Type 1 a Type 2 hypervizoru [2]

1.1.1 Příklady a rozdíly ve využití

Mezi hlavní příklady Type 1 hypervizorů patří vSphere od firmy VMware nebo Hyper-V of firmy Microsoft. Pro type 2 hypervizory je hlavním příkladem Kernel-based Virtual Machine, zkráceně KVM. Dále pak Oracle VirtualBox nebo Workstation od firmy WMware [2].

Type 1 hypervizor vídáme hlavně v komerčních sítích, kde je nutné z hardwaru dostat co nejvíce výkonu za stanovené peníze a předem existuje plán sítě, který nevyžaduje flexibilitu operačního systému. Naopak Type 2 uvidíme hlavně v menších sítích či pro testovací účely. Neexistují však žádná pevně daná pravidla, ve kterých by bylo stanoveno využití jednoho či druhého typu pro specifický účel.

1.2 Emulační software

Emulační software umožňuje emulaci celé sítě se všemi náležitostmi, které k ní můžou patřit. Dále se jedná o čistý software běžící na zařízení nad operačním systémem jako každodenní programy, a nebo může být spuštěn nad hypervizorem, kde jej pak lze nazývat virtualizační emulační software. Narozdíl od přímé virtualizace, popsané v minulé kapitole, zde je emulována celá síť a nejenom jednotlivá zařízení. Kvůli tomu nepřekvapí, že hlavní využití je testování technologií před jejich nasazením nebo případné odladění funkcí. Tím může být ušetřeno na koupených zařízeních a zároveň to šetří čas potřebný k plnému zprovoznění sítě. Nestačí však samotný software a je nutné doinstalovávat jednotlivé image neboli kód zajišťující napodobení funkce reálného zařízení.

1.2.1 Varianty

Mezi příklady softwarového řešení lze zařadit program Cisco Packet Tracer, který lze nainstalovat na osobní počítač a slouží i jako výukové prostředí k cisco certifikacím. Je možné na něm provádět i komerční úlohy, ale pokročilejší technologie a protokoly nemusí být podporovány. Profesionálové pak spíše sahají po dále uvedených nástrojích. Příklady virtualizovaného řešení jsou EVE-NG, ve kterém byli i vypracovány ukázkové příklady, GNS3 nebo VIRL [3]. Všechny tyto platformy fungují na stejném principu. Jediné znatelné rozdíly lze najít v uživatelském rozhraní, kde se často jedná spíše o maličkosti což nemá zásadní vliv na funkci. Některé platformy jsou dokonce schopny spolu sdílet image [4], což jen podtrhuje jejich podobnost.

1.3 Network functions virtualization

Posledním představeným nástrojem na virtualizaci je network functions virtualization neboli NFV. Jak už název napovídá, je vhodný pro virtualizaci síťových funkcí. Pro pochopení je vhodné znát historické pozadí.

1.3.1 Historie a důvod vzniku

Vývoj zařízení určených pro využití v telekomunikacích se vždy zaměřoval na stabilitu a výkon. Nároky na stabilitu byly popsány charakteristikou tzv. High availability [5] neboli vysoká dostupnost. V souvislosti s telekomunikacemi se historicky často mluvilo o pěti devítkách neboli zaručení dostupnosti služby poskytovanou zařízením 99,999 % času. To znamená, že roční výpadek chybou zařízení mohl být pouze necelých 5,5 minuty. To zajišťovalo kvalitní síť, ale z pohledu vývoje to vytvářelo jiný problém. Zařízení, které bylo takto spolehlivé, bylo také dražší a tudíž nasazováno do sítě s předpokladem delšího životního cyklu, mimo jiné z ekonomických důvodů. V případě příchodu nové služby pak její nasazení do sítě muselo čekat do výměny zařízení, která se však konala v pravidelných intervalech. Ve výsledku tudíž nároky na stabilitu brzdily implementaci nových služeb.

Zatímco poslední pětiletí 20. století lze považovat za zlatou éru pěti devítek, 21. století už přišlo s novým pohledem na problematiku [6]. Rychle se rozvíjející systémy potřebovaly čím dál tím větší kapacitu přenosu [7]. Vedle kapacity přenosu bylo nutné myslet i na přicházející vylepšení, některá z nich bylo potřebné aplikovat i z důvodů bezpečnosti co nejdříve. Na přelomu roku 2012 a 2013 skupina operátorů vytvořila bílou knihu, která vedla k vytvoření standardu pro NFV Evropským ústavem pro telekomunikační normy [8] jako dodatek k již vytvořeným SDN. To umožnilo být více flexibilní v nasazování technologií a zároveň bylo toto řešení výhodnější pro sítě s většími nároky na rychlost a kapacitu přenosu.

1.3.2 Základní charakteristiky

Hlavní výhodou, oproti ostatním zmiňovaným možnostem virtualizace, je možnost využití i přímého přístupu k hardware. Existují služby, které nejsou optimalizované pro použití ve virtualizovaném prostředí. Možnost přístupu k hardwaru umožňuje využívat výhody virtualizace bez nutnosti omezit využívání těchto funkcí.

Zároveň u vysokorychlostních sítí lze narazit na efekt hrdla láhve, tzv. bottleneck. Hardware, popřípadě i software, nestíhá zpracovávat informace a zpomaluje tak celý proces komunikace. Tento efekt se může u virtualizace znásobit, kde mezi software a hardware přidáváme mezistupeň v podobě hypervizora a případně i dalšího operačního systému. Zde se pak projeví možnost přímého přístupu k hardwaru a zabrání se problémům, kterými by jiné možnosti virtualizace byly zasaženy. Nejedná se pouze o rychlost, ale i bezpečnost, jelikož jeden z nejčastějších kybernetických útoků je stále DDoS [9], který spoléhá na přetížení sítě.

Podobně jako u fyzické sítě lze požadovat provedení různých funkcí za sebou, je možné vyžadovat spojení více jednotlivých virtualizovaných síťových funkcí (anglicky VNF neboli virtualized network functions) pomocí service chaining procesu. Tímto spojením vznikne sekvence VNF spojené do jedné služby, kterou lze pak opakovaně používat jako jeden blok. Příklady služeb zahrnují firewally, šifrování a další zabezpečení, překlad veřejných IP adres či optimalizace provozu [10]. To pomáhá automatizaci provozního toku. Service chaining lze označit spíše jako rozšíření pro SDN, ale NFV je stále jeho častým prostředkem implementace.

NFV a SDN sice nemusí nutně fungovat společně, ale byly k tomu vytvořeny a doplňují se. Cílem tohoto spojení je vytvoření soběstačné sítě, která sama detekuje a opravuje chyby, zatímco stále zvyšuje svojí efektivitu pomocí optimalizace toku dat a s lepší škálovatelností a cenami než fyzické sítě. Navíc k její správě je využíváno modernější prostředí management API, tudíž je pro techniky jednodušší a efektivnější síť nastavit a monitorovat.

1.4 Porovnání využití jednotlivých možností

Ať už se jedná o jakékoliv nasazení, tak neexistuje jediná správná možnost. Představené varianty jsou stále všechny virtualizace, jejich základ je podobný a technicky se liší často maličkostmi. Jsou to však nejenom tyto maličkosti, které naprosto mění předpokládané využití. Příkladem může být virtualizace jednoduchého síťového prvku. Z textu je zřejmé, že je možné využít jak virtualizaci hypervizorem, tak network functions virtualization. Za předpokladu, že se jedná o nevytížený prvek, se tyto varianty mohou zdát rovnocenné. V reálném nasazení jsou však na opačném konci spektra. Zatímco software s hypervizorem lze použít i zadarmo, s dobrovolnou volbou placení za podporu, například od firmy WMWare. Network Functions Virtualization Infrastructure Software od firmy Cisco, ani nelze zakoupit samostatně bez platforem prodávaných stejným výrobcem, které najdeme s cenovkami v řádech stovek tisíc korun českých. Jsou to tedy i netechnické parametry, které určují rozsah a vhodnost nasazení.

S tímto na mysli jde najít podle zadaných požadavků doporučené postupy. Již byly zmíněné ty u emulačních softwarů, které se hodí v případě, že je požadováno testovací prostředí pro celé části sítě. NFV je připraven pro nasazení do těch nejtěžších podmínek, čemuž odpovídá i jeho cena. Omezuje se navíc na networking, kde základní virtualizace s hypervizorem může mít výhodu flexibility i mimo networking. Hlavní výhodou je však stále nízká cena.

Kapitola 2

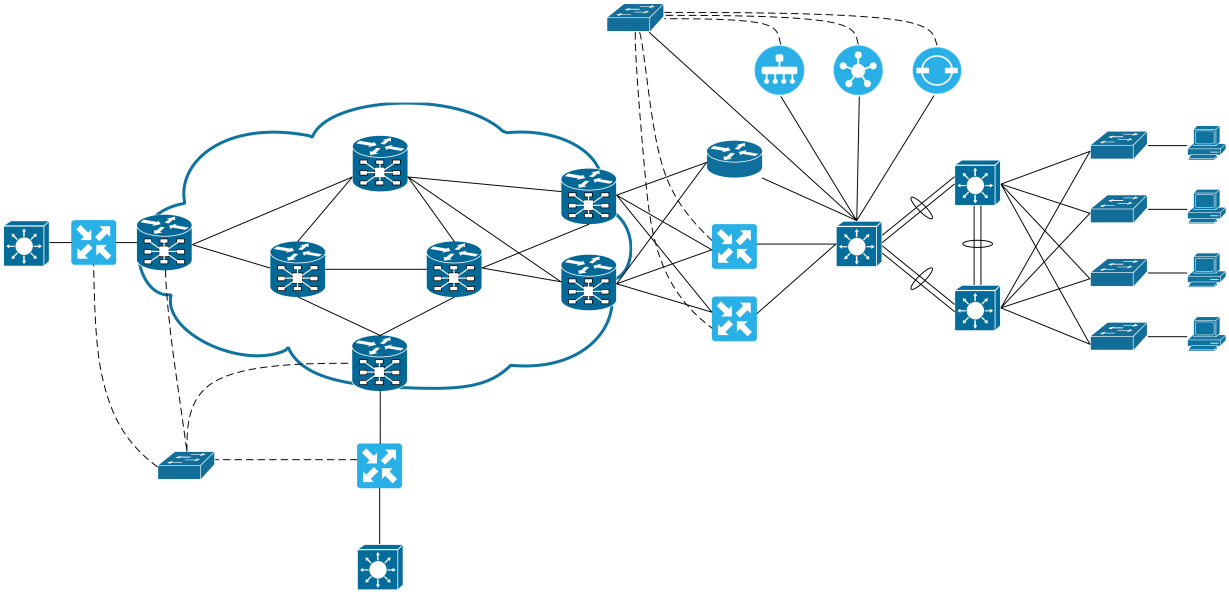
Laboratorní úlohy v EVE-NG

Hlavním, praktickým, bodem této práce bylo vytvoření případových úloh ve virtuálním prostředí EVE-NG. Úlohy a k nim náležící dokumentace byly vytvořeny tak, aby bylo možné je následně využít při výuce. Tomu musela být přizpůsobena jejich struktura. Byla snaha přizpůsobit podobu vypracování úlohám z Cisco Network Academy, které jsou ve škole při výuce již používány a jejichž kvalita je na vysoké úrovni.

2.1 Popis úloh

V rámci práce byl vypracován soubor tří úloh, které na sebe navazují. Od začátku byla snaha, aby úlohy nebyly od sebe odtržené a studenti si mohli lépe představit, jak spolu jednotlivé protokoly souvisí, a jak jeden pomáhá druhému ke správné funkci sítě. Na obrázku 2.1 je vidět výchozí topologie odvozená z reálných sítí, odpovídající středně velké firemní síti s více pobočkami.

V levé polovině obrázku vidíme oblak, který reprezentuje přenosovou síť, skládající se ze zařízení Cisco XRv 9000. K němu připojené jednotlivé pobočky, doleva a dolů menší, doprava pak hlavní. Menší pobočky se skládají pouze ze směrovače Cisco vEdge (světle modrá značka) a jednoho L3 přepínače, simulujícího koncovou infrastrukturu a zbytek pobočky. Doprava jsou zálohovaně připojeny Cisco vEdge směrovače s jedním směrovačem Cisco vIOS. Dále spojeno do centrálního L3 přepínače, ze kterého doprava pokračuje pobočka simulující koncové uživatele. V pobočce najdeme zálohované L3 přepínače, L2 přepínače a koncová zařízení. Nahoru od centrálního L3 přepínače lze najít řídící zařízení pro SD-WAN dále popsané v kapitole 2.4. Přerušovanými čárami jsou vybraná zařízení spojena s přepínači značícími síť sloužící pro správu těchto zařízení bez závislosti na stavu zbytku sítě. Lze tak zajistit dostupnost pro opravy i bez funkční sítě.



Obrázek 2.1: Topologie celé sítě

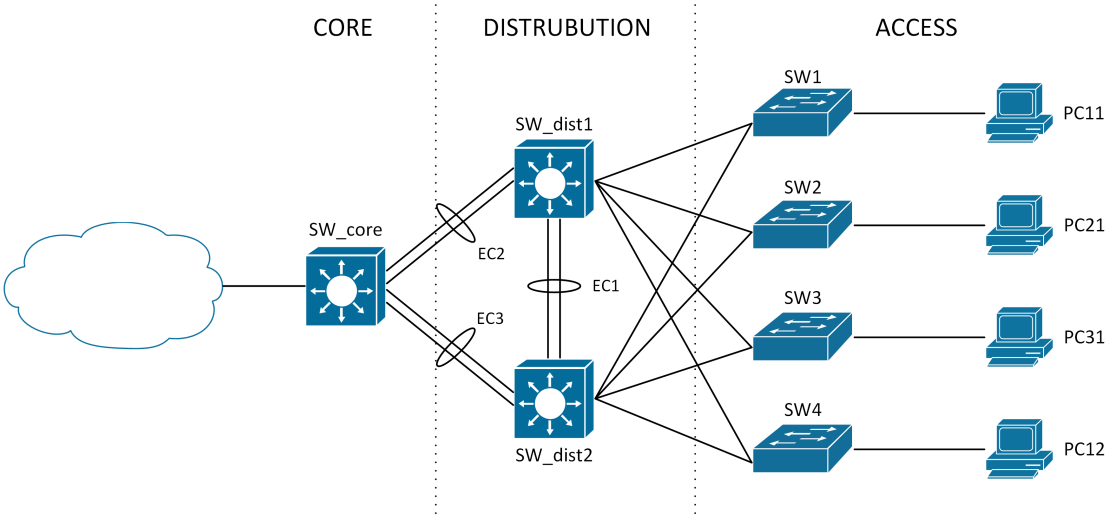
2.1.1 IP plány

V úlohách bylo použito více variant IP plánů. Efektivní a bezpečná síť stojí na přípravě IP adresace před samotnou konfigurací. V úlohách byla snaha nejenom ukázat potřebu dostatečně plánovat dopředu, ale i různé možnosti zápisu. V první úloze byla použita IP tabulka, která by mohla být uložena do excel souboru. Ve druhé úloze byl IP adresní plán zanesen přímo do topologie. I přes menší množství informací v první úloze byl použit plán, který se více hodí pro větší sítě, jelikož je jednodušší na škálovatelnost a přehlednost. Zároveň umožňuje přidávat různé poznámky a sdílet ho mezi více účastníků projektu. Plán přímo v topologii je jednodušší pochopit na první pohled, avšak s větší sítí se stává méně přehledný a spolupráce na něm s jinými techniky je náročná. Neexistuje žádný návod na vytváření plánu, pouze doporučení a zkušenosti. Proto byla snaha první takovou zkušenost ukázat již v těchto úlohách. Dobré je tyto varianty kombinovat s tabulkou jako hlavní zdroj a do topologie psát případné poznámky nebo často hledané informace.

2.2 Úloha 1 - FHRP, RSPT+, Portchannel

První úloha se zaměřuje na technologie, které tvoří základ velkého množství sítí, ale některé z nich nejsou vyučovány. Spanning tree protocol nebo jeho novější

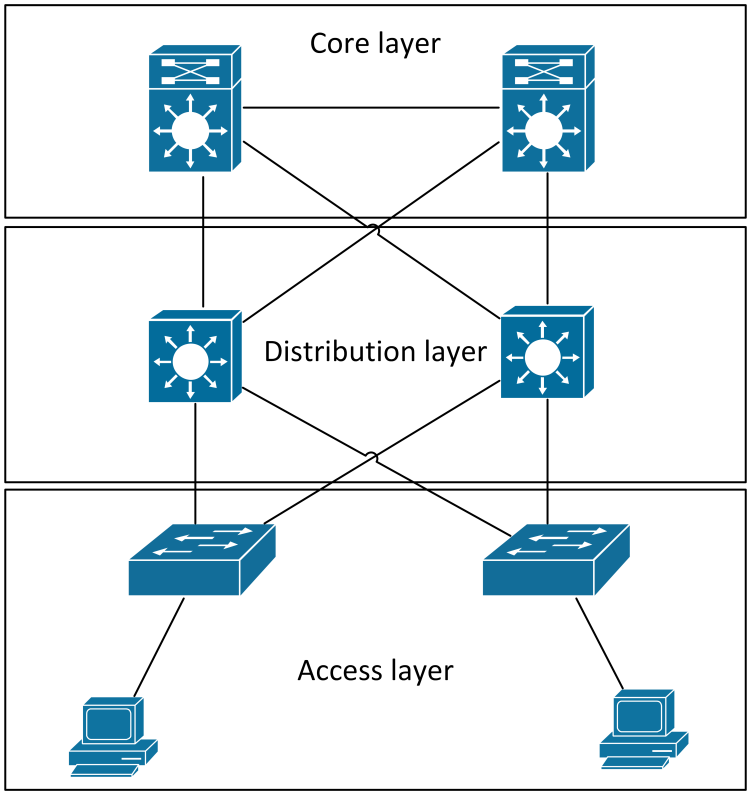
variantu rapid spanning tree protocol najdeme ve většině sítí. First hop redundancy protocol společně s portchannelem jsou zde použity jako příklad možného vyřešení zajištění spolehlivosti připojení v sítích. Obecně je spolehlivost jeden ze základních požadavků, které se musí při návrhu reálné sítě řešit. V části topologie náležící k této úloze je použit hierarchický model sítě. Přesněji je použit Cisco tříúrovňový hierarchický model. Jedná se o soubor doporučení pro stavbu kampusové sítě.



Obrázek 2.2: Topologie první úlohy

2.2.1 Tříúrovňová topologie

Tato část sítě používá hierarchický design, který se dělí do 3 úrovní. První úrovní je core, zde najdeme nejvýše postavené zařízení, přes core prochází veškerá komunikace. Důležité z pohledu networkingu je, že se pohybujeme na L3 vrstvě. V našem případě je tato část nezálohovaná, ale v reálu je plně možné a často vídané, že core přepínače/servery/směrovače jsou nasazovány alespoň v párech, aby při výpadku jednoho mohl druhý převzít komunikaci a nedošlo k výpadku v celé síti. Druhou úrovní je distribution, kde často najdeme předěl mezi L3 a L2 technologií ISO/OSI modelu. Ve středně velké či větší síti se zde může nacházet více než jeden zálohovaný pár zařízení. Dokonce lze říct, že je to žádoucí. Máme-li firmu s více pobočkami, pak v distribution úrovni je běžné vidět jeden, ve velkých sítích i více, zálohovaný pár zařízení v každé pobočce. A poslední, třetí, úrovní je access. Zde najdeme L2 přepínače a je to část se kterou se setkáváme nejčastěji. Do těchto zařízení také připojujeme koncové zařízení jako například osobní počítače, kamery, IP telefony a další.



Obrázek 2.3: Obecný hierarchický model

2.2.2 Spanning tree protocol

Spanning tree protocol má v sítích se záložními trasami roli zabránit vytváření smyček neboli posílání stejné informace mezi zařízeními dokola nebo duplicitně. Tři základní problémy, které vznikají kvůli smyčkám, jsou broadcast storms neboli broadcastové bouře, nestabilita MAC tabulky a přeposlání kopií stejného rámce. První problém způsobuje přetížení sítě posíláním broadcastového rámce skrze síť, kde s každou smyčkou znásobí svůj počet. Zbylé dva problémy se týkají hlavně odpovědí jednotlivých zařízení. V obou případech se stane, že zařízení, které dostává rozporující si informace, nebude správně odpovídat zdroji a nebude tudíž správně fungovat spojení. V posledním případě pak může dojít až k pádu aplikace, která informace přijímá [11].

Historie

Máme několik modernizací protokolu, některé samotným IEEE, jiné jednotlivými výrobci, vytvářející proprietární variantu. Původní spanning tree protocol byl definován v IEEE 802.1d v 90. letech. Rychlost konvergence činila 50 sekund a je vyvolána každou změnou v topologii sítě. Pro některé aplikace však byla tato doba dlouhá, a proto byl vytvořen standard IEEE 802.1w, který definoval rapid spanning tree protocol a v roce 2004 ho nahradil i v původním IEEE 802.1d. Rychlost konvergence u tohoto protokolu se zkrátila na méně než pětinu původního času. Nejnovějším protokolem je multiple spanning trees definován v IEEE 802.1s. Lze ho připodobnit RSTP s rozeznáváním VLAN. Cisco má své propri-

etární protokoly jako PVST nebo RPVST, které spojují STP a RSTP, v tomto pořadí, s možností per vlan. RPVST lze tedy nazvat ekvivalentem MST a jsou spolu dokonce kompatibilní [12].

Nástroje používané protokolem

Celá funkce protokolu se odvíjí od povolených a zakázaných portů. Protokol se snaží vytvořit z povolených portů stromový graf, kde kořenem je jeden prvek a to root. Root poznáme tak, že má nejnižší bridge ID, což je číselná hodnota v násobcích 4096. Dále tu pak je path cost, neboli cena cesty a každé spojení je hodnoceno na základě jeho rychlosti. S vývojem technologií a většími přenosovými rychlostmi se hranice pro cenu několikrát posouvaly, ale v současné době pro jeden gigabit za sekundu cena činí 20 000. Čím rychlejší spojení tím nižší cena. Celkový stromový graf je pak skládán od root dál podle kombinace bridge ID a ceny cesty. V případě stejného bridge ID se rozhoduje podle nejnižší MAC adresy. U starých zařízeních, stejně jako u cen cest, bývaly hodnoty bridge ID menší, a proto se po připojení do nové sítě může stát, že se prohlásí za root, což musí být ošetřeno, jinak síť nemusí fungovat, jak bychom chtěli.

Role portů

Každý port na zařízení má svou roli. Důležité je nezaměňovat role a stavy portů. Zde je popis RSTP rolí, které se od STP lehce liší, zejména názvy. Existuje pět rolí portu. Root port je port s nejmenší cenou cesty k portu z daného zařízení. Alternate port je nový pro RSTP a je to port s druhou nejnižší cenou cesty hned po root portu. V případě chyby na root portu je na alternate port přepnuto bez čekání. Designated port nevede do root portu, ale přeposílá data do jiných částí sítě s nejmenší cenou cesty. Backup port je pak záloha designated portu, vede-li do zařízení nebo části sítě více cest. Poslední role je disabled, která má zakázáno posílat jakékoliv zprávy a zároveň přijímat všechny zprávy, které by mohly měnit MAC tabulku a další. Jediné zprávy, které přijímá, jsou BPDU, aby v případě změny mohl začít proces změny z discarding stavu na forwarding stavu.

Průběh při prvním spuštění

Na začátku musí proběhnout volba kořene neboli root. Každé zařízení považuje samo sebe za root a tuto informaci posílá v hello BPDU, které se skládá hlavně z vlastního bridge ID, bridge ID zařízení, které považuje za root a root cost k tomuto zařízení. Zároveň poslouchá na svých portech hello zprávy ostatních zařízení. Jakmile přijde hello zpráva s vhodnějším kandidátem, nastaví ho jako root a změní obsah své hello zprávy. Výhodou oproti STP je, že nepřestane šířit hello zprávy, a tak dochází k rychlejší konvergenci topologie.

Stavy portů

Aby nedošlo k vytvoření dočasného loopu, změna portu z blocking na forwarding musí probíhat přes mezistavy. STP rozeznává pět stavů, v RSTP došlo ke zjednodušení na tři stavy. První dva stavy jsou si velmi podobné, disabled a blocking. V první případě se jedná o administrativně vypnutý port a v druhém případě je port

již zapnutý, ale STP rozhodl o zákazu přeposílání rámců. RSTP pak oba tyto porty označilo jako discarding. Následuje listening stav, zde port stále nepřeposílá rámce, neučí se nové MAC záznamy do tabulky a staré záznamy k danému portu z MAC tabulky vymaže. Tento stav byl označen v RSTP za nadbytečný a není využíván. Předposlední stav je pro oba protokoly stejný, learning. V tomto stavu podobně jako v listening, nepřeposílá rámce, ale učí se nové záznamy do vyčištěné MAC tabulky. Finálním stavem je forwarding stav, kde posílá rámce a učí se záznamy do mac tabulky.

V STP každém mezistavu zůstává dobu určenou Delay Timerem, který je ve výchozím nastavení nastaven na 15 vteřin a společně s ostatními časovači je celková doba konvergence obvykle 50 sekund. RSTP pak změnami v časovačích, odstraněním mezistavu a principu vytváření zpráv o změně topologie snížilo tuto dobu v nejhorších případech na 10 sekund, standardně se však uvažuje spíš o jednotkách sekund.

2.2.3 Portchannel

Portchannel umožňuje spojení více fyzických portů do jednoho logického a komunikaci přes tento logický port. Takto můžeme spojit běžně 8, na výkonnějších platformách až 16, fyzických portů. Pro takovéto spojení je nutné dodržet na všech portech stejnou přenosovou rychlost, stejnou konfiguraci a stejné nastavení L1 až L3 parametrů jako například maximální velikost IP datagramu, duplex mód nebo nastavení VLAN. Stejně jako v reálných případech i v úloze je proto doporučeno nastavovat rozhraní přes rozsah [11].

Mezi výhody využití patří lineární zvýšení kapacity provozu v závislosti na počtu rozhraní. Další výhoda se váže s využitím RSTP, v případě přerušení pouze části spojení. Nedochází tak nutně k vyvolání konvergence dokud nejsou přerušeny všechny fyzické propoje.

Známe dva standardy, které slouží pro vyjednávání portchannelů, LACP a PAGP. PAGP je proprietární protokol výrobce Cisco a je starší než LACP. LACP je definován v dokumentu IEEE 802.3ad a je univerzálně podporován širokou škálou výrobců. Samotné rozdíly mezi těmito protokoly jsou minimální a kvůli omezení PAGP na jednoho výrobce je LACP rozšířenější. Důležité je však používat na obou koncích stejný protokol, jelikož i přes svou podobnost spolu nedokážou kanál vyjednat.

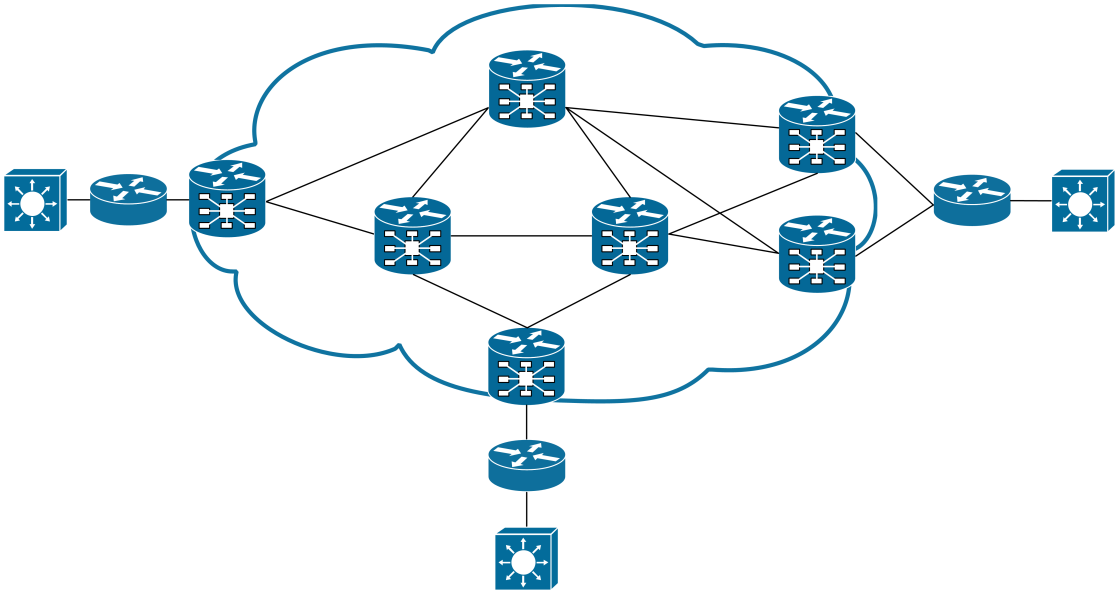
2.2.4 First hop redundancy protocol

Ve skutečnosti se jedná o skupinu protokolů, které dohromady řeší jeden problém. V sítích s redundantními prvky je potřeba určit, který z nich je primární a který je sekundární. Primární prvek obsluhuje datový tok a zajišťuje vše potřebné k chodu sítě. Sekundární prvky pak pouze poslouchají vše, co dělá primární, aby v případě problému byli schopné nahradit tento prvek bez výpadku v provozu.

Zařízení si mezi sebou pošlou zprávy, ve kterých se mimo jiné nachází priorita. Zařízení s nejvyšší prioritou se pak stane primární. Novější verze protokolu, Gateway Load Balancing Protocol, od firmy Cisco už umí i jistou formu kooperace.

2.3 Úloha 2 - MPLS

Druhá úloha se soustředí na část sítě, která se standardně nachází na straně internetového providera. V obrázku 2.4 je v oblaku ukázka MPLS sítě. Jedná se pouze o zmenšený model, protože díky škálovatelnosti tyto sítě mohou dosahovat velikosti stovek zařízení, kterým se však konfigurací nemění. Open shortest path first je směrovací protokol, který je používanější než stále vyučovaný a zastaralý RIP protokol, a zároveň jednoduchý na pochopení a konfiguraci. Pro distribuci značek je zde použit segment routing, který představuje alternativní přístup k vyučovanému label distribution protokolu s větší možností ovlivnění přiřazených značek. Následně je nastavena i MPLS. Toto vše je nastaveno na zařízeních používající Cisco IOS XR, který je protokolově centrický na rozdíl od IOS XE a jeho port centrického přístupu. Tento software byl zvolen, protože Junos OS od konkurenční firmy Juniper se více blíží IOS XR a je tedy vhodné znát oba přístupy. Zároveň je IOS XR rozšířený v zařízeních s vysokou dostupností, se kterými se často setkáme v sítích poskytovatelů internetu.



Obrázek 2.4: Topologie druhé úlohy

2.3.1 Open Shortest Path First

OSPF je směrovací protokol ze skupiny interních gateway protokolů, které směrují v autonomních systémech neboli systémech zařízení se stejnou správou a jednotnou směrovací politiku [13]. Protokol je založený na stavu linek, což znamená, že na rozdíl od RIP protokolů všechny směrovače znají topologii celé sítě, ve které se nachází. K hledání nejkratší cesty ve vytvořeném grafu se využívá Dijkstrův algorimus.

Pro zjištění nejrychlejší cesty využívá parametr metrika, který je nepřímo úměrný přenosové rychlosti. Jako referenční rychlost se v současné době používá nejčastěji 10 gigabitový ethernet, který dostane metriku jedna. S pomalejšími linkami se pak

metrika zvyšuje a vyhrává cesta s nejmenší metrikou. V minulosti se již referenční rychlost posouvala a nejde tedy tvrdit, že se v budoucnu tato hodnota nezmění.

Každé zařízení zná topologii celé sítě, a proto jsou implementovány oblasti, které zabraňují přehlcení paměti záznamy. Každý směrovač pomocí hello paketů vytváří sousedskou relaci, pokud se shodují například hodnoty čísla oblasti, subnet a subnet masky, časovačů protokolu a autentizační údaje. O relaci následně vytvoří záznam ve směrovací tabulce. Hello pakety se zasílají pravidelně podle hello časovače s výchozí hodnotou 10 sekund. Tím se udržuje aktuálnost a díky malé velikosti paketu nezatěžuje síť a ani neomezuje její propustnost. V případě, že je navázána úplná sousedská relace, dochází k výměně záznamů a tím k šíření informací o topologii. V případě, že vyprší dead timer, který je vždy resetován přijatým hello paketem, je relace považována za přerušenou a je vytvořen paket o změně topopologie, který je poslán do topologie a dochází k přepisu tabulek a přepočítávání nejkratších cest. Celý protokol používá pro posílání paketů multicast.

Dijkstrův algoritmus

Poprvé algoritmus popsal Edsger Dijkstra po kterém je i pojmenován. Slouží k hledání nejkratších cest v konečném pozitivním grafu. Algoritmus si pro počátek pamatuje nejkratší cestu ke každému vrcholu grafu a zároveň si pamatuje, zda již vrchol nebyl použit, aby nedošlo k vytvoření smyčky.

2.3.2 Segment routing

Tato metoda umožňuje distribuci značek, o kterou se již nestará Label Distribution Protocol, ale samotný Interior Gateway Protocol. Mezi IGP se řadí OSPF, ISIS a starší RIP. Zde tedy OSPF distribuuje kromě informací o cestách i informace o značkách ve speciálním paketu. Značky se nenastavují pro celé zařízení, ale pro jedno jeho rozhraní. Často se proto využívá loopback, který není závislý na stavu linky k němu připojené.

Segment routing představuje vylepšení oproti LDP, který přiřazoval značky pro každé zařízení jiné, tudíž dva různé směrovače znaly třetí pod jinou značkou. Segment routing umí pro jedno zařízení přiřadit značku tak aby byl v celé síti známý pod touto značkou. Navíc jelikož už využívá IGP tak snižujeme zátěž na control plane, která se musela s LDP synchronizovat.

Rozlišujeme dva základní typy značek: globální a lokální. Název napovídá, že globální značky slouží k označení jednoho portu v rámci sítě. Nastavuje se jako absolutní hodnota nebo jako index. Neexistuje standard sjednocující výrobce, který by zajišťoval, že ve všech zařízeních budou alokované stejné množiny, informace o nich si však zařízení vyměňují. Existuje proto zápis indexem, při kterém si zařízení přičtou hodnotu indexu k počátku své množiny a nevzniká kolize i v případě, že se v absolutních hodnotách množina liší. Lokální značky označují jednotlivé linky, které následně propagují do celé sítě. Umožňují větší kontrolu nad cestou rámců skrz síť, jelikož pro výchozí cestu využíváme OSPF, ale občas je žádoucí posílat přes např. zabezpečené trasy nebo trasy s nízkou latencí.

2.3.3 Multi protocol label switching

Tato metoda vkládá svoji hlavičku mezi L2 a L3 hlavičky, někdy označována jako L2,5. Zjednodušuje proces zpracování rámců, jelikož funguje na principu přepínání. Díky tomu má menší nároky na hardware, respektive umožňuje na stejném hardwaru větší propustnost oproti obyčejnému IP směrování. Navíc umí přenést různé, i starší, protokoly, které by bylo nemožné přenést přes standardní přenosovou síť. Tato kompatibilita je klíčová vlastnost a důvod širokého rozšíření této technologie.

Existuje mnoho způsobů jak MPLS využívat. Můžeme přes MPLS vytvářet šifrované tunely, optimalizovat provoz v síti, kontrolovat a regulovat kvalitu služby. Její hlavní využití se proto nachází v přenosových sítích na úrovni poskytovatelů internetu.

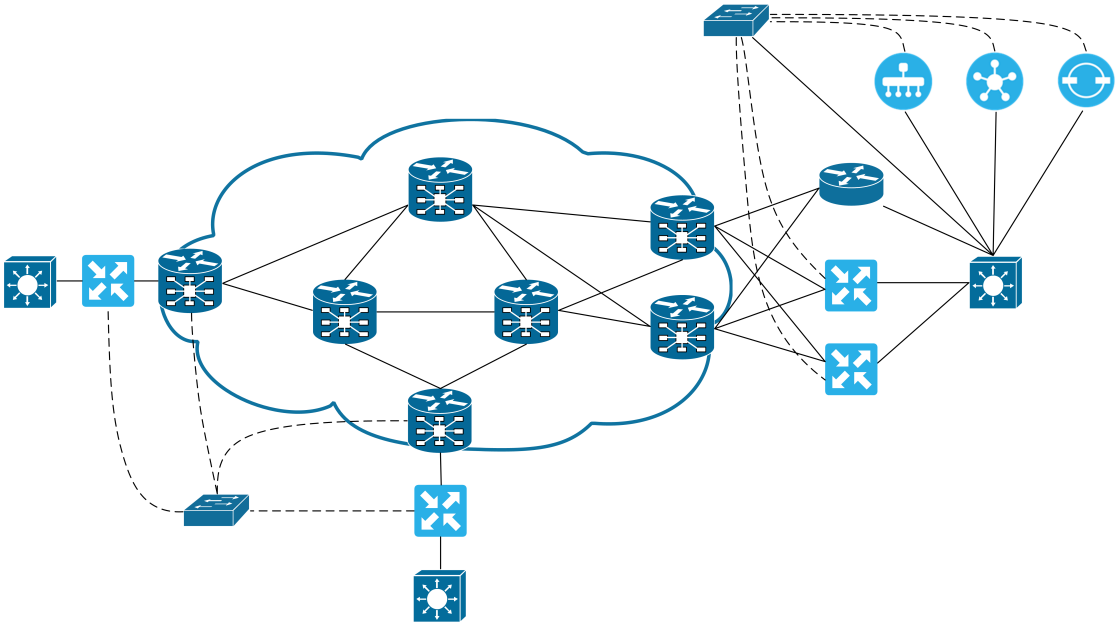
Ve své hlavičce vytváří zásobník se značkami neboli label stack. Ten slouží k určování cesty, přičemž po příchodu paketu je z tohoto zásobníku provedeno odstranění vrchní značky, čímž se zařízení dozví kudy má přesměrovat paket dále. Zjistí-li směrovač, že není cílem uvedeným ve značce, vrátí dříve popnutý label zpět do zásobníku. Díky segment routingu je potom schopný i určovat trasu paketu, jelikož jedna značka znamená v celé síti jedno zařízení. Stačí, aby se do label stacku na začátku přidaly značky trasy ve správném pořadí a správné směrovače před odesláním odstraní svoji značku a pošlou paket dál. SLDP musel label stack být u každého zařízení pozměněn, než mohl paket pokračovat.

Směrovače v MPLS síti je možné dělit do několika kategorií, je vhodné znát alespoň základní rozdělení na provider směrovač a provider edge směrovač. První z nich se nachází v rámci topologie a je obklopen pouze dalšími MPLS směrovači, komunikuje tudíž pouze pomocí značek. Druhý je pak okrajový směrovač, který má na alespoň jednom rozhraní spojené s customer edge směrovačem. Komunikuje do MPLS sítě pomocí značek, a zároveň ven z MPLS pomocí kterékoliv technologie použité na daném místě. Customer edge směrovač pak nepatří do MPLS sítě a nevyužívají pro komunikaci značky.

Jedno z vylepšení operací se značkami se nazývá penultimate hop popping. Krajní provider směrovače vědí, na které lince se nachází provider edge směrovač za pomoci speciální značky. V případě, že přijmou rámec s cílem na toto koncové zařízení, provedou před odesláním odebrání vrchní značky. Krajní směrovač následně nemusí dvakrát kontrolovat zásobník, poprvé, aby zjistil, že je paket určen pro něj a podruhé, aby určil, jak s paketem dále naložit. Stačí mu se podívat jak má rámec zpracovat, jelikož ví, že směrovač před ním už jeho značku odstranil. Tím dojde k dalšímu zefektivnění procesu a tím pádem i k jeho urychlení.

2.4 Úloha 3 - SD-WAN

Třetí a závěrečná úloha je celá věnována nastavení SD-WAN prvků sítě. Topologií se jedná o upravenou druhou úlohu, do které byla přidána tři řídící zařízení (Cisco vManage, Cisco vSmart, Cisco vBond) a čtyři Cisco vEdge směrovače. V úloze byla snaha nastavit síť tak, aby bylo možné ukázat potřebné kroky před větší automatizací sítě. Virtuální SD-WAN je velmi nový koncept, a proto nebylo možné zaručit plnou funkcionalitu.



Obrázek 2.5: Topologie třetí úlohy

2.4.1 Cíle SD-WAN

S rostoucí poptávkou po digitalizaci procesů se objevily modely softwaru, nebo dokonce infrastruktury, jako služby. Uživatel, v podobě společnosti, si koupil část výpočetního výkonu na cloudu neboli vzdáleném serveru. Díky sdílené infrastruktuře došlo ke snížení cen na provozování služby. Pro připojení k těmto cloudům však stále musí existovat vlastní firemní síť, která propojuje koncové uživatele s provozovatelem cloudu. Alternativně lze využít služeb internetu, u kterého hrozí horší stabilita specifických aplikací a otevíráme síť kyberútokům. Cílem SD-WAN je centralizace provozu přidáním centrálních prvků, které jsou schopny na základě znalosti topologie efektivně navrhovat trasy pro pakety při zvýšení bezpečnosti. Tento systém umožňuje zjednodušený pohled na nasazení a správu sítě dle zadaných požadavků snižující nároky na techniky. Zároveň je možné si tyto centrální prvky pronajmout v rámci cloudu a stavět pouze s upravenými vEdge směrovače. Nejedná se však o automatickou volbu pro všechny nové sítě, jelikož některé společnosti si stále rádi drží kritickou infrastrukturu fyzicky pod svojí správou. [14]

2.4.2 Popis technologie

Celé SD-WAN řešení lze rozdělit na čtyři úrovně. Nejnižší úroveň nazýváme datovou úrovní a najdeme zde koncové vEdge směrovače, které slouží jako samotná přenosová soustava. Další úrovní je kontrolní úroveň, kterou už lze zařadit do řídicí části a najdeme zde vSmart. Toto zařízení lze považovat za logické centrum, kde se překládají všechny vstupy uživatelů do počítačové logiky a jsou distribuovány příslušným směrovačům a ostatním kontrolním zařízením. Následuje orchestrační úroveň, ve které se nachází vBond, který zajišťuje propojení všech prvků dohromady, ať už se jedná o nahrávání konfigurací skripty, přiřazení IP adresy nebo monitorování topologie. Informace pak sdílí s ostatními vrstvami a díky tomu do-

káží vytvářet efektivnější cesty. Poslední úrovní je správa. Zde najdeme zařízení vManage, které slouží jako rozhraní pro správce. Přes jeho webové prostředí se síť nastavuje, monitoruje a případně opravuje [14].

2.5 Problémy s implementací

Stejně jako u reálných systémů, i virtualizace má nevýhody. Na rozdíl od fyzické sítě se může stát, že některé síťové funkce nejsou v dané virtualizaci podporovány. To může částečně omezit možnosti při vytváření úloh. Je to však předem zjistitelný fakt, které funkce nemusí být podporovány. Větším problémem jsou softwarové chyby. Ty nejenom, že nemusí být zmapovány, ale samotné odstranění může být nemožné. Nehledě na typ chyby je její opravení většinou nereálné a kromě jiného zařízení nebo jiné verze nad opravou nemáme velkou kontrolu.

V průběhu vytváření úloh jsem narazil na více problémů, které se týkaly softwarových chyb u zařízení a některé, spíše obecné, týkající se celé sítě. I přes zdánlivě jednoduché řešení se může stát, že řešení takového problému zabere až desítky hodin a stejně může vyústit v řešení použitím jiného protokolu či zařízení, respektive v neúspěšné vyřešení. V následující části se pokusím ukázat pouze některé problémy, se kterými jsem se setkal. Většinu z nich jsem nemohl vyřešit a musel obejít. V jistých případech by dokonce problém nebylo možné vyřešit bez zakoupení oficiální podpory, což jsem naštěstí obešel díky přístupu k těmto systémům v mém zaměstnání.

2.5.1 Nefunkčnost VPC

Hned u první úlohy jsem se setkal s problémy. První z nich se týkal zařízení VPC neboli virtuální počítač. Má se jednat o jednoduché zařízení s jedním portem, které má simulovat koncové zařízení. Po zadání několika příkazů má mít IP adresu s výchozí bránou pro komunikaci a má sloužit primárně pro kontrolu spojení mezi body. Po základní konfiguraci jsem chtěl otestovat funkčnost spojení a jejich správnou konfiguraci. Ukázalo se však, že VPC není stabilní a nepravidelně neodpovídá nebo si dokonce vymaže konfiguraci. Po kontrole, zda nedošlo z mé strany ke špatnému uložení konfigurace, bylo zjištěno, že zde problém není a VPC je skutečně nestabilní i při správném uložení konfigurace. I po několika pokusech zařízení nefungovalo ve všech případech. Mohlo by se i zdát, že celé virtuální prostředí nemá dostatečný počet hardwarových zdrojů, ale to bylo nemožné, protože VPC bylo údajně záměrně méně náročné na zdroje, a navíc nefungovalo ani po navýšení zdrojů v dalších částech úlohy.

Nakonec jsem se rozhodl problém obejít kompletní výměnou zařízení. Jelikož se jedná pouze o koncový bod, lze ho nahradit směrovačem nebo L3 přepínačem, kterému na připojeném portu nastavíme IP adresu a výchozí cestu, port následně stačí zapnout. Funkci plní správně, jen nemusí být z uživatelského pohledu tak intuitivní.

2.5.2 Problém s nevhodnými image zařízení

Ještě před dokončením první úlohy se projevilo další zařízení, tentokrát Cisco IOL. Jedná se o dva image, jeden jako L2 zařízení a druhý jako L3, dohromady

simulující přepínač a pravděpodobně směrovač. Jedná se také o výchozí zařízení v používaném školním prostředí.

Již základní přepínání a směrování neproběhlo v pořádku. Ukázalo se, že zařízení nepoužívají Cisco IOS XE, ale svůj vlastní operační systém, který měl tendenci nepravidelně zamrzat. Jakákoliv práce s ním byla tedy zdlouhavá.

Samotné zamrzání, ač nepohodlné, nijak nebránilo funkci. Hlavní problém nastal při nastavování port-channelu, kde v jedné z variant, L2 nebo L3, nevyjednával kanál. Dokonce vypisovalo chybové hlášky, které vypisuje v případě správného nastavení a nesestavení kanálu. Zpočátku jsem podezříval vlastní chybu v konfiguraci, ale ani konzultace s kolegou žádnou neodhalila. Hledání na internetových fórech odhalilo, že Cisco IOL má více problémů, které se verze od verze pouze liší, ale neopravují nikdy všechny najednou.

Uznal jsem, že Cisco IOL nejsou vhodné pro používání v úloze. V zaměstnání jsem musel zažádat o image Cisco vIOS směrovač a Cisco vIOS přepínač, protože nejsou volně dostupné a jiné varianty jsem nenašel. Navíc s těmito jsem již pracoval a měl jsem předem možnost otestovat funkčnost. Znovu tedy nebylo možné chybu vyřešit a musel jsem celý problém obejít výměnou používaného zařízení.

2.5.3 Neexistence certifikační autority v testovací síti

S jedním z dalších problémů jsem se setkal při implementaci SD-WAN, která má jisté potřeby v síti a jednou z potřeb, je certifikační autorita. Ta slouží k ověřování autenticity zařízení, aby nedošlo k jejich podvržení. Školní síť, pro kterou jsou tyto úlohy vytvářeny, je značně nepřipravená na implementaci podobných úloh. Naštěstí řešení od firmy Viptela počítalo s podobnou eventualitou a vytvořili možnost autorizace certifikáty, kde vBond slouží zároveň jako certifikační autorita. Z praktického pohledu se nejedná o nejbezpečnější řešení, avšak pro ukázku v úlohách postačí.

2.5.4 Předpoklad použité SD-WAN technologie o propojení s online účty

Celý koncept SD-WAN počítá i s kompletně cloudovým řešením. Aby se nestalo, že se vEdge směrovače spojí s vManage jiné firmy, je nutné zařízení zadat do portálu Plug and Play, kam je nutné se přihlásit pomocí cisco ID, který musí být spojen s Cisco Smart Accountem [15]. Cisco Smart Account je účet, který umožňuje správu licencí a dalších softwarových požadavků. Jedná se o účet, o který je nutné zažádat s unikátní doménou. Není vhodné, aby každý student žádal o unikátní smart account.

2.5.5 Neexistující implementace částí SD-WAN do virtuálního prostředí

Jednou z hlavních výhod SD-WAN má být nízká náročnost na konfiguraci a jednoduchý management sítě. I mimo tuto technologii využíváme Zero Touch Provisioning, který umožňuje nahrání předem připravené konfigurace. Stačí tedy zařízení připojit a podle kvality skriptu lze zařízení plně předkonfigurovat bez zásahu technika. SD-WAN má v sobě tuto možnost zabudovanou, ale ve virtuálním prostředí není na mnou používané verzi funkční [16]. Proto se všechna zařízení musí složitě

nastavovat ručně. V případě použití jiné verze by naopak nemusela fungovat síť vůbec.

Kapitola 3

Využití virtuálních platforem

Virtuální platformy nejsou navrženy jako náhrada fyzických. Stále jsou spuštěné na fyzických zařízeních a pouze rozšiřují možnosti při nasazení. Mohou sloužit jako škálovatelná záloha, která dokáže flexibilně pomáhat síti od přetížení. Nebo jsou naopak využity jako levnější varianta v místech, kde nahradí více fyzických zařízení jedním s podporou virtualizace.

3.1 Automobilový průmysl

S vývojem technologií se i v automobilech zvyšuje množství různorodých senzorů. Některé predikce počítají s nahrazením lidského řidiče počítačovými systémy [17]. Ty budou spoléhat právě na data ze senzorů a zajišťovat plynulou a bezpečnou jízdu bez zásahu člověka. To samozřejmě zvyšuje nároky na parametry datového přenosu. Současná auta využívají pro komunikaci senzorů s řídící jednotkou jednoduché sběrnice. Jednoduché protokoly dovolují těmto sběrnicím mít menší přenosové rychlosti. Jedním z příkladů je Controler area network sběrnice, která se poprvé objevila ve vozech Mercedes-Benz v roce 1991 [18] a je dodnes ve vozech využívána. Nově bude nutné se vypořádat s kamerovými systémy a senzory, kterým přenosové rychlosti starších sběrnic už nemusí stačit, hlavně pak jedná-li se o hlavní zdroj informací pro řídicí počítačový systém při rozhodování v řízení.

Rychlost přenosu není však jediný parametr, na který je nutné brát ohled. Stejnou roli může hrát i odezva systému a může to být i rozdíl mezi pokračováním v jízdě a dopravní nehodou. Začíná se proto uvažovat o využívání metalických IP sítí [19], což umožní využití ověřené technologie a zvýšení přenosových rychlostí bez nákladů na vývoj nového systému na míru.

3.1.1 Indy Autonomous Challenge [20]

Jedním více otevřeným experimentem je závod Indy Autonomous Challenge, ve kterém všechna auta byla čistě autonomní [21]. Jednalo se o spojení několika částí telekomunikací. V rámci aut byl umístěn industriální přepínač Cisco IE řady 5000 [22], který sloužil jako centrální bod pro sběr dat ze senzorů. Tato data byla následně přenášena pomocí bezdrátové technologie Cisco Ultra-Reliable Wireless Backhaul k týmu na diagnostiku [23]. Hlavní výpočetní operace však byly stále prováděny přímo ve vozidle, aby se dosáhlo co nejnižší reakční doby.

Nejednalo se pouze o test aut, ale celé infrastruktury, jelikož po závodní dráze nejezdilo po přímce jediné auto. Zatímco nelze očekávat, že podobné autonomní závody v nejbližší době nahradí závody s lidskými piloty, posloužil závod k lepšímu pochopení problematiky autonomního řízení ve vysokých rychlostech, které se může týkat například provozu po rychlostních silnicích [24].

3.2 Průmysl 4.0

Automobilový průmysl však není jediný, který hledá možnosti inovace. Lze tvrdit, že není ani největší. V minulé dekádě se objevila koncepce průmyslu 4.0 označována také jako čtvrtá průmyslová revoluce [25]. Jedná se o spojení trendu digitalizace a automatizace, který by mohl od základu změnit pohled na výrobu v továrnách a nejenom tam. Z pohledu sítí se znovu jedná o problematiku sběru velkého množství dat a jejich zpracování. Zde je již více vidět zapojení virtualizace, jelikož se předpokládá zpracování dat v cloudových úložištích [26]. Neřeší se pouze sběr dat, ale propojení několika autonomních systémů v jeden celek.

Kapitola 4

Závěr

Práce měla jeden vedlejší a jeden hlavní cíl. Prvním cílem bylo představit virtuální platformy a ukázat možnosti jejich využití. Druhým cílem bylo vytvoření pracovních úloh v prostředí EVE-NG, které bude možné použít k výuce témat SD-WAN a MPLS.

První kapitola byla věnována představení virtualizace, nejprve obecně a následně ve spojitosti se sítěmi. Byly představeny tři často využívané varianty, virtualizace s hypervizorem, emulační software a network functions virtualization. U každé byl kromě popisu uveden i příklad využití a na konci kapitoly bylo i souhrnné srovnání možnosti využití i dle netechnických parametrů. Ač byly popsány různé varianty, jedná se stále o virtualizaci, a proto je zmíněn i překryv využití.

V rámci práce byl vytvořen soubor tří praktických úloh, které byly ve druhé kapitole práce popsány a doplněny o teoretické základy. Jednotlivé úlohy byly navrženy do jedné topologie tak, aby bylo jednoduché si představit, jak na sebe jednotlivé technologie navazují. První úloha byla navržena spíše jako podpůrná, obsahující protokoly STP, portchannel a FHRP. Dobře ukazuje nižší úrovně sítě, se kterými přijde technik často do kontaktu. Zároveň poukazuje na důležitost redundance a plánování. Druhá kapitola navazuje na první v podobě sítě, která se objevuje na vyšších úrovních u poskytovatelů internetu nebo ve velkých společnostech jako soukromá přenosová síť. Mezi představené protokoly patří OSPF, segment routing a MPLS. Zároveň úloha používá často opomíjené zařízení se softwarem Cisco IOS XR, který se používá právě u dražších zařízení, připravených na funkci v nejvytíženějších částech sítě. Soubor úloh zakončuje třetí, zaměřená na technologii SD-WAN. Ta si klade za cíl vytvoření sítě s centrálními prvkem, ze kterého jde celá konfigurovat a spravovat. Snižuje nároky na znalost techniků a urychluje nasazování síťových prvků. Zároveň počítá s podporou cloudových řešení a umožňuje mít celou řídící část virtualizovanou na serveru.

Ve třetí kapitole byly uvedeny možná místa využití virtuálních platforem v zadaných oborech. V automobilovém průmyslu byla zmíněna autonomní doprava a závod autonomních formulí Indy Autonomnous Challange. Řídící systémy byly zastoupeny průmyslem 4.0, jehož velkou součástí, i když ne jedinou, je sběr dat a řízení jednotlivých částí výrobních závodů, což je jedna z možných úloh jednoduše škálovatelných virtuálních platforem.

Poslední sekcí druhé kapitoly je výpis problémů, které se musely vyřešit pro správnou funkčnost. Úlohy se až do samého závěru dařily dle očekávání. Při samotném testování, však vyplynuly na povrch další problémy mimo rámec práce.

Zařízení využívající software IOS XR se ukázaly být náročné na hardwarové zdroje, a proto navrhovaná topologie musela být rozdělena podle jednotlivých úloh. I přes toto rozdělení se však nepodařilo snížit nároky dostatečně, a proto nemůže správně fungovat celá úloha používající tyto zařízení naráz. Jmenovitě se jedná o úlohu dva a na ní navazující úlohu tři. Řešením tohoto problému by byla změna topologie a snížení počtu zařízení s IOS XR a jejich nahrazení zařízeními se IOS XE, které protokoly podporují.

Ve všech úlohách byla snaha vyhnout se prostému kopírování konfigurace a potřeba ze strany řešitele zamýšlet se nad problematikou, což bylo podporováno i kontrolními otázkami v průběhu úlohy. Třetí úloha svou náročností předčila očekávání a připomíná spíše dlouhý návod než laboratorní úlohu. V kapitole 2.5 zmíněná nepodpora některých částí reálné verze SD-WAN a relativně malé stáří technologie mají za následek neexistenci dostatečného množství podpory ze strany výrobce. Jakýkoliv proces vytváření kontrolní části by se v daném čase nepodařilo stihnout. Zároveň se cestou vyskytly problémy i s virtuálním prostředím pod správou školy, které zpomalovaly práci a donutili mě testovat část konfigurace v soukromém prostředí.

S dostatkem času a znalostí by bylo možné rozšířit a upravit úlohy pro vhodné použití při výuce. Úlohy jako takové je možné využít jako základ, na kterém jde rozšiřovat dle volby uživatele.

Literatura

[1] GOLDBERG, Robert P. *Architectural Principles for Virtual Computer Systems.* Harvard University, 1973, s. 22–26. dostupné z https://apps.dtic.mil/sti/pdfs/AD0772809.pdf [08.08.2022].

[2] *Type-1 and Type-2 Hypervisors explained* [online]. [cit. 2022-04-30]. Dostupné z: https://www.vembu.com/blog/type-1-and-type-2-hypervisor/

[3] *Top 5 Network Simulation Tools in 2022* [online]. [cit. 2022-05-06]. Dostupné z: https://ipwithease.com/top-5-network-simulation-tools-in-2020/

[4] *How To Run GNS3 VM on KVM* [online]. Techviewleo.com, 2021 [cit. 2022-05-06]. Dostupné z: https://techviewleo.com/how-to-run-gns3-vm-on-kvm/

[5] PIEDAD, Floyd a Michael HAWKINS. *High Availability: Design, Techniques, and Processes.* Prentice Hall, 2001. ISBN 9780130962881.

[6] *How Low-Cost Telecom Killed Five 9s in Cloud Computing* [online]. [cit. 2022-06-27]. Dostupné z: https://www.wired.com/insights/2013/03/how-low-cost-telecom-killed-five-9s-in-cloud-computing/

[7] SUMITS, Arielle. The History and Future of Internet Traffic [online]. 2015-08-28 [cit. 2022-06-27]. Dostupné z: https://blogs.cisco.com/sp/the-history-and-future-of-internet-traffic

[8] *Network Functions Virtualisation* [online]. [cit. 2022-06-27]. Dostupné z: https://www.etsi.org/technologies/nfv

[9] WARBURTON, David. *2022 Application Protection Report: DDoS Attack Trends* [online]. 2022-03-16 [cit. 2022-06-27]. Dostupné z: https://www.f5.com/labs/articles/threat-intelligence/2022-application-protection-report-ddos-attack-trends

[10] FITZGIBBONS, Laura. *Service chaining* [online]. 2019 [cit. 2022-06-27]. Dostupné z: https://www.techtarget.com/whatis/definition/service-chaining

[11] WENDELL, Odom. *CCNA 200-301 Official Cert Guide Library: Volume 1.* Cisco Press PTG, 2020. ISBN 9780136755449.

[12] *MST and PVST+ Interoperability* [online]. [cit. 2022-05-05]. Dostupné z: https://networklessons.com/cisco/ccie-routing-switching-written/mst-pvst-interoperability

[13] HAWKINSON, J. a T. BATES. *Guidelines for creation, selection, and registration of an Autonomous System (AS)* [online]. 1996 [cit. 2022-08-01]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc1930

[14] ROHYANS, Aaron, Ali SHAIKH, Chandra Balaji RAJARAM, et al. *Cisco SD-WAN: Cloud scale architecture*. San Jose, California: Book Sprints, 2019.

[15] *Plug and Play Connect Service* [online]. [cit. 2022-07-12]. Dostupné z: https://www.cisco.com/assets/sol/sb/RV345_Emulators/RV345_Emulator_v1-0-03-15/help/help/t_UsingPnP_Connect.html

[16] *Release Notes for Cisco vEdge Devices, Cisco SD-WAN Release 20.3.x* [online]. [cit. 2022-07-25]. Dostupné z: https://www.cisco.com/c/en/us/td/docs/routers/sdwan/release/notes/vedge-20-3/sd-wan-rel-notes-20-3.html

[17] *Profesionální řidiče nahradí „stupeň 5". Otázka je, kdy se to stane* [online]. 2017 [cit. 2022-07-09]. Dostupné z: https://www.idnes.cz/technet/veda/autonomni-auto-porsche-lidar-sea-international.A170614_083717_veda_kuz

[18] *Mercedes W140: First car with CAN* [online]. Daimler, 2016 [cit. 2022-07-09]. Dostupné z: https://can-newsletter.org/engineering/applications/160322_25th-anniversary-mercedes-w140-first-car-with-can/

[19] *IP and Ethernet in Motor Vehicles: Challenges for the development tool, illustrated by today's applications* [online]. 2012 [cit. 2022-07-09]. Dostupné z: https://cdn.vector.com/cms/content/know-how/_technical-articles/Ethernet_IP_ElektronikAutomotive_201204_PressArticle_EN.pdf

[20] *The Autonomous Challenge* [online]. [cit. 2022-07-09]. Dostupné z: https://www.indyautonomouschallenge.com

[21] Cisco, 2021, *Behind the Scenes: Cisco IoT Powers the Indy Autonomous Challenge* [YouTube video]. [cit. 2022-07-07]. Dostupné z: https://www.youtube.com/watch?v=EFvkFx1lwoY

[22] *Cisco Industrial Ethernet 5000 Series Switches: Fueling innovation on and off the track* [online]. [cit. 2022-07-09]. Dostupné z: https://www.cisco.com/c/en/us/products/switches/industrial-ethernet-5000-series-switches/index.html

[23] *Cisco Ultra-Reliable Wireless Backhaul* [online]. [cit. 2022-07-09]. Dostupné z: https://www.cisco.com/c/en/us/products/wireless/ultra-reliable-wireless-backhaul/index.html

[24] *Cisco Connected Transportation Solutions* [online]. Barcelona, 2018 [cit. 2022-07-09]. Dostupné z: https://www.ciscolive.com/on-demand/on-demand-library.html?search=IoT%20solutions&search=IoT+solutions#/session/1636059727281001 7t86

[25] KAGERMANN, H., W. WAHLSTER a J. HELBIG. *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Final report of the Industrie 4.0 Working Group* [online]. 2013 [cit. 2022-07-26]. Dostupné z: https://www.din.de/blob/76902/e8cac883f42bf28536e7e8165993f1fd/recommendations-for-implementing-industry-4-0-data.pdf

[26] *Jak rozumět konceptu Průmysl 4.0* [online]. 2019 [cit. 2022-07-09]. Dostupné z: https://www.spcr.cz/aktivity/z-hospodarske-politiky/12973-jak-rozumet-konceptu-prumysl-4-0

Příloha A

Ukázka první úlohy

Úvod

Vítejte v části jedna z laboratorních úloh. Tato část se soustředí na ukázku tříúrovňové podnikové sítě, na které ukazuje využití technologie RSTP, FHRP a portchannel. Každá z nich je nezbytnou součástí stabilní moderní sítě a navzájem se doplňují. Cílem této úlohy je:

Ukázka konfigurace a vysvětlení použití těchto technologií

Ukázka možné topologie reálné sítě

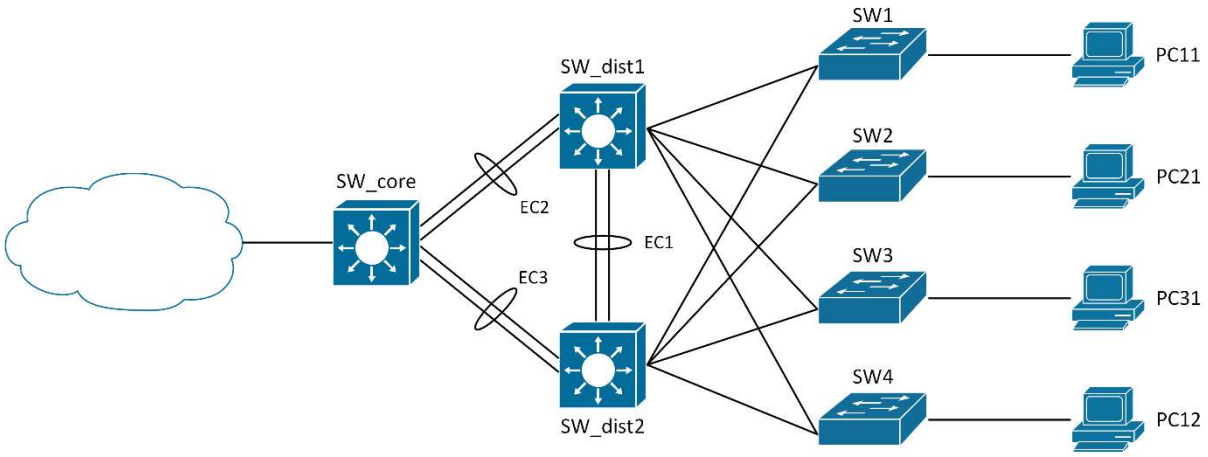
Naznačit rozdíl mezi L2 a L3

Příprava sítě použité v dalších částech laboratorní úlohy

Pro více teorie se odkažte do kapitoly 2.3 bakalářské práce.

Topologie

Na obrázku níže vidíme topologii sítě, se kterou budeme v úloze pracovat.



Jedním z důležitých faktorů u stabilní sítě je redundantnost, aby při výpadku jednoho spoje nevypadla značná část sítě, ideálně vůbec žádná. Toho docílíme použitím více paralelních zařízení, je však nutné nakonfigurovat síť, aby s těmito záložními propoji poradila. Použijeme tří úrovňovou topologii. Lze jednoznačně určit jednotlivé úrovně. V jednotlivých úrovních je také vidět ukázka zálohy pro případ výpadku, ať už jednotlivých linek (využití portchannelů nebo propojení s více zařízeními na vyšší vrstvě) nebo výpadku celých zařízení (zdvojené distribuční L3 switche). Jelikož se žádná síť neobejde bez rozdělení na VLAN tak celá úloha bude počítat s rozdělením na jednotlivé VLAN.

IP plán

zařízení	IP/maska	GW	VLAN
VPC11	192.168.10.11/24	192.168.10.2	VLAN 10
VPC12	192.168.10.12/24	192.168.10.2	VLAN 10
VPC21	192.168.20.11/24	192.168.20.2	VLAN 20
VPC31	192.168.30.11/24	192.168.30.2	VLAN 30

VLAN 10	PC	
VLAN 20	VOICE	
VLAN 30	GUEST	
VLAN 900	PORTCHANNEL_2	pozn: pouze pro distribuci a core
VLAN 901	PORTCHANNEL_3	pozn: pouze pro distribuci a core

SW_dist1	
SVI VLAN 10	192.168.10.2 /24
SVI VLAN 20	192.168.20.3 /24
SVI VLAN 30	192.168.30.2 /24
SVI VLAN 900	192.168.255.2 /30
SW_dist2	
SVI VLAN 10	192.168.10.3 /24
SVI VLAN 20	192.168.20.2 /24
SVI VLAN 30	192.168.30.3 /24
SVI VLAN 901	192.168.255.6 /30
SW_core	
SVI VLAN 900	192.168.255.1 /30
SVI VLAN 901	192.168.255.5 /30

VIP VLAN 10	192.168.10.1/24
VIP VLAN 20	192.168.20.1/24
VIP VLAN 30	192.168.30.1/24

Nastavení VLAN na zařízeních SW1-4 a SW_dist1-2

Pro to abychom mohli používat více VLAN v síti je nutné, aby všechna zařízení o těchto VLAN věděla. Dynamických propagování VLAN může propagovat i do míst kam nechceme, a proto je nastavíme ručně podle IP plánu. Nezapomeňte na access switchích přiřadit příslušné porty do příslušné VLANy.

Pro bezpečnost si vytvořte VLAN 999 s názve SHUTDOWN na všech zařízeních a všechny neaktivní porty do ní přiřaďte a zamezte jim možnost forwardovat přes trunky. Porty poté ještě vypněte. Zkuste vše přes interface range.

Příklad založení:

```
SW1(config)#vlan 10
SW1(config-vlan)#name PC
```

Příklad na trunky mezi SW a SW_dist (jak na stranu SW tak na stranu SW_dist):

```
SW1(config)#interface ethernet 3/0
SW1(config-if-range)#switchport trunk encapsulation dot1q
SW1(config-if-range)#switchport mode trunk
SW1(config-if-range)#switchport trunk allowed vlan add 10
SW1(config-if-range)#switchport trunk allowed vlan add 20
SW1(config-if-range)#switchport trunk allowed vlan add 30
SW1(config-if-range)#switchport trunk allowed vlan remove 1
SW1(config-if-range)#switchport trunk allowed vlan remove 999
```

Ke koncovým switchům není žádoucí dělat trunky, nutno nastavit access porty aby došlo k otagování paketů.

```
SW1(config)#interface ethernet 0/0
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 10
```

Vyberte si jeden z trunků a pomocí show commandu se podívejte jaké VLANy jsou povolené.

Všechny porty by měly být ve VLAN 999 ale kdyby některý nebyl a zůstal ve VLAN 1, prošel by trunkem? Proč?

Na jaké vrstvě osi modelu pracují tyto porty?

Ověřte správné nastavení pomocí show. Které jste použili?

Nápověda show commandů, které nám pomůžou.

```
show ip interface brief

show vlan

show interfaces trunk
```

Nastavení koncových zařízení (VPCs)

Koncová zařízení jsou ve skutečnosti switche, je tedy jednoduché je nastavit aby šly použít pro kontrolu funkčnosti sítě. U normálního PC by stačilo nastavit IP adresu s maskou a defaultní routu. Zde je nutné nastavit port do správné VLAN a následně nastavit SVI v příslušné vlan s požadovanou IP adresou. Na základě příkazů z minulé kapitoly si toto vyzkoušejte.

Port-channel

Port-channely nám umožňují spojit více fyzických propojů do jednoho logického, to má několik výhod. Při loadbalancingu jsme schopni dosáhnout vyššího bandwidthu. Následně pak v případě přerušení jednoho z kabelů nám nespadne celý propoj. To má vliv nejen na koncové zákazníky a stabilitu jejich připojení, ale také to například nevyvolá konvergenci topologie u RSTP (viz další kapitola). Pro zadání příkazů do více interface naráz použijeme interface range. V případě rozdílné konfigurace není možné spojit porty v port-channel. Jedná se o i výrobcem doporučovaný postup. I když se celou dobu bavíme o port-channelu tak ve skutečnosti je budeme dělat na L2 a zde se nazývají etherchannely. Důvodem je nesprávná funkčnost na použitých zařízeních. Použijeme pro každý jinou VLAN a tím pádem alespoň částečně oddělíme sítě od sebe.

Posledním příkazem nastavíme channel group. O tom, jestli je to L3 nebo L2 rozhoduje, jestli zadáme no switchport (L3) nebo switchport mode trunk/access (L2).

L2

```
interface interface-id
switchport mode {access | trunk}
switchport access vlan vlan-id
channel-group channel-group-number mode {auto | desirable | on } | { active | passive}
```

L3

```
interface interface-id
no ip address
no switchport
channel-group channel-group-number mode { auto | desirable | on } | { active | passive }
```

Pro náš případ nejdříve doplníme VLAN pokud není vytvořená z prvního kroku.

```
SW_dist1(config)#vlan 900
SW_dist1(config-vlan)#name PORTCHANNEL_2
```

A následně nastavíme port na trunk a nastavíme channel-group.

```
SW_dist1(config)#interface range ethernet 3/0, ethernet 3/1
SW_dist1(config-if-range)#switchport trunk encapsulation dot1q
SW_dist1(config-if-range)#switchport mode trunk
SW_dist1 (config-if-range)#switchport trunk allowed vlan add 900
SW_dist1 (config-if-range)#switchport trunk allowed vlan remove 1
SW_dist1 (config-if-range)#no shutdown
SW_dist1(config-if-range)#channel-group 2 mode active/passive
```

Mezi distribučníma switchema můžeme v případě výpadku potřebovat i funkci switchování.

```
SW_dist1(config)#interface range ethernet 2/0, ethernet 2/1
SW_dist1(config-if-range)#switchport trunk encapsulation dot1q
SW_dist1(config-if-range)#switchport mode trunk
SW_dist1 (config-if-range)#switchport trunk allowed vlan add 10
SW_dist1 (config-if-range)#switchport trunk allowed vlan add 20
SW_dist1 (config-if-range)#switchport trunk allowed vlan add 30
SW_dist1 (config-if-range)#switchport trunk allowed vlan remove 1
SW_dist1 (config-if-range)#switchport trunk allowed vlan remove 999
SW_dist1(config-if-range)#channel-group 1 mode active/passive
```

V případě nastavení passive na obou koncích, dojde k sestavení port channelu? Ne, passive pouze poslouchá a čeká kdo se s ním pokusí navázat port-channel.

A co v případě active na obou stranách? Ano, active aktivně domlouvá sestavení a nevadí když je na obou koncích. V praxi se používá varianta active na obou stranách nejčastěji, pokud není specifický důvod proč nechtít z nějaké strany PC nenavázat.

Vypište show příkazy, které jste použili k ověření správné funkčnosti: show port-channel summary

Rapid Spanning Tree Protocol

RSTP, rapid spanning tree protocol, je vylepšená verze spanning tree protocolu (hlavně pak v rychlosti konvergence a vylepšením zjednodušující fungování), která zabraňuje vytvoření smyček v síti. Takto zálohovaná síť by nemohla bez podobného protokolu fungovat.

Cisco virtuální zařízení, se kterými zde pracujeme, stejně jako většina jejich reálných zařízení přichází z výroby připravená k použití. Znamená to často zaplé porty nebo například zapnutý spanning tree protocol. Pro případ že není zaplý použijeme příkaz konfiguračním režimu:

```
SW_dist2(config)#spanning-tree mode rapid-pvst
```

Tím rovnou povolíme RSTP, pv ve zkratce znamená per vlan a tudíž budeme mít per vlan rapid spanning tree protocol.

Jelikož víme, že pro VLAN 10 a 30 chceme, aby root byl SW_dist1 a pro VLAN 20 chceme, aby root byl SW_dist2 tak si nastavíme prioritu pro tyto boxy příkazem

```
SW_dist2(config)#spanning-tree vlan <vlan> priority <4096>
```

Zároveň víme, že vždy protější box (pro 10 a 30 = SW_dist2 a pro 20 = SW_dist1) chceme jako záložní tak nastavíme prioritu 2x vyšší, tudíž 8192.

Zkuste pomocí show příkazu ověřit, že je správně nastaven root z access layer switche:

```
SW_dist2#show spanning-tree
```

Nastavení SVI a kontrola sítě

Jelikož pro kontrolu správnosti chceme použít ping tak musíme nastavit IP adresy, ty však na L2 nejde přiřadit přímo portům, proto je dobré podle IP plánu nastavit adresy na virtuální porty (SVI = Switch Virtual Interface), které pro nás budou plnit funkci podobnou loopbacku a po zapnutí ip routing můžeme i loopback vytvořit a zkusit konektivitu. (na access switchích nejsou potřeba, protože nikdy nebudou routovat L3)

```
SW_dist1(config)#interface vlan 10
SW_dist1(config-if)#ip address 192.168.10.2 255.255.255.0
SW_dist1(config-if)#no shutdown
```

Když ve VPC 11 zkusíme příkaz ping 192.168.30.3 jaká bude odpověď?

Co když zkusíme to samé s adresou 192.168.30.2?

Pokud alespoň na jednu z nich vidím proč? Pokud ne proč?

Jeden z příkazů, které na L3 routerech zadávat nemusíme ale bez kterých se L3 switch neobejde je ip routing. Bez něj L3 switch funguje jen jako L2 switch a neroutuje.

Co vypíše příkaz na SW_dist1 show ip route? Proč?

Pusťte tento příkaz v konfiguračním režimu a zkuste znovu tyto otázky.

Když ve VPC 11 zkusíme příkaz ping 192.168.30.3 jaká bude odpověď?

Co když zkusíme to samé s adresou 192.168.30.2?

Znovu se podívejte do routovací tabulky SW_dist1, vidíte změnu? Zkuste na VPC 11 příkaz trace 192.168.30.3 a podívejte se na cestu.

Pokud byste před zadáním ip routing zkusili ping na adresu 192.168.10.3, dostali bychom odpověď?

First Hop Redundancy Protocol

FHRP, first hop redundancy protocol, používáme pro jednodušší určování, které zařízení je hlavní, zpracovává datový provoz, a které pouze čeká až jiné zařízení přestane fungovat. Nastavujeme ho na distribution úrovni na IP interfacech. Jelikož jediné IP (L3) interface které na switchi máme jsou nastavené SVI tak je nastavujeme tam. Příkazy v subconfigu standby <číslo skupiny>. Číslo skupiny slouží jako identifikátor pro ostatní zařízení, ale není to jediný parametr.

Priorita zde funguje obráceně a to, že vyšší znamená větší (opak od STP). Zároveň nastavujeme preempt, to znamená, že v případě, že přidáme zařízení s vyšší prioritou tak převezme kontrolu.
V případě, že nenastavíme tento parametr by přidání zařízení s vyšší prioritou neznamenalo změnu až do selhání zařízení, které do té doby bylo primární.

Takto nastavíme pro všechny 3 VLANy.

```
SW_dist1(config)#interface vlan 10
SW_dist1(config-if)#standby 1 priority 200
SW_dist1(config-if)#standby 1 preempt
SW_dist1(config-if)#standby 1 ip 192.168.10.1

SW_dist2(config)#interface vlan 10
SW_dist2(config-if)#standby 1 preempt
SW_dist2(config-if)#standby 1 ip 192.168.10.1
```

Zde nenastavujeme prioritu, protože nám stačí defaultní 100, avšak nic nám nebrání

Je dokonce možné použít jenom 2 skupiny, víme že pro VLAN 10 a 30 máme stejný primární a sekundární switch. Stačí proto použít jinou VIP adresu dle IP plánu se stejnou skupinou.

Vyvolejte show standby brief pro oba switche

Jakou vMAC mají jednotlivé skupiny (show standby)?

Nyní je nutné přenastavit jako default gateway příslušnou VIP pro jednotlivé VLANy abychom viděli z počítačů ven.

Zároveň v core je dobré si udělat loopback0 s adresou v prostoru 192.168.128/17 (například 192.168.254.10/32)

A ip routy z core switche

ip route 192.168.0.0 255.255.128.0 a následně pro loadbalancing přes obě adresy SVI proti SW_core 192.168.255.2 a 192.168.255.6. Bez nich by switch nevěděl, kam má odpovídat.

```
ip route 192.168.0.0 255.255.128.0 192.168.255.2
ip route 192.168.0.0 255.255.128.0 192.168.255.6
```

Nyní je možné se z počítačů dostat do core a získat i odpověď.

Závěr

Nyní máme hotovou celou síť. Máme místo, kam můžeme připojovat počítače a další zařízení v síti. Zároveň máme zálohu distribuční úrovně, čímž zvyšujeme stabilitu.
Jedním z možných vylepšení je link state tracker. Ten by nám umožnil sledovat stav portů. Při vypnutí zařízení z core se může stát, že distribuce (primární) tuto skutečnost nezaznamená (jelikož stále dostává odpovědi přes druhé zařízení). Všechna data, která však pošle do core jsou zahozena a není

zde konektivita dále do sítě. Link state tracker nám umožní případný výpadek zjistit a přesměrovat komunikaci na aktivní zařízení.
V případě nefunkčnosti virtuálních počítačů (VPC) je lze nahradit vIOS routerem, stačí nám jeden, na připojený port nastavit adresu odpovídajícího počítače a nastavit defaultní gateway (ip route 0.0.0.0 0.0.0.0 192.168.<vlan>.1). Tímto způsobem by mělo vše již fungovat správně.

Příloha B

Ukázka druhé úlohy

Úvod

Vítejte v části dva z laboratorních úloh. Tato část se soustředí na ukázku kousku sítě, na které ukazuje využití technologie MPLS, segment routing a propojení s SD-WAN, která bude tématem třetí laboratorní úlohy. Podobné sítě se vyskytují v internetu na mnoha místech a používají je všichni bez jejich vědomí. Cílem této úlohy je:

Ukázka konfigurace a vysvětlení použití MPLS a segment routing

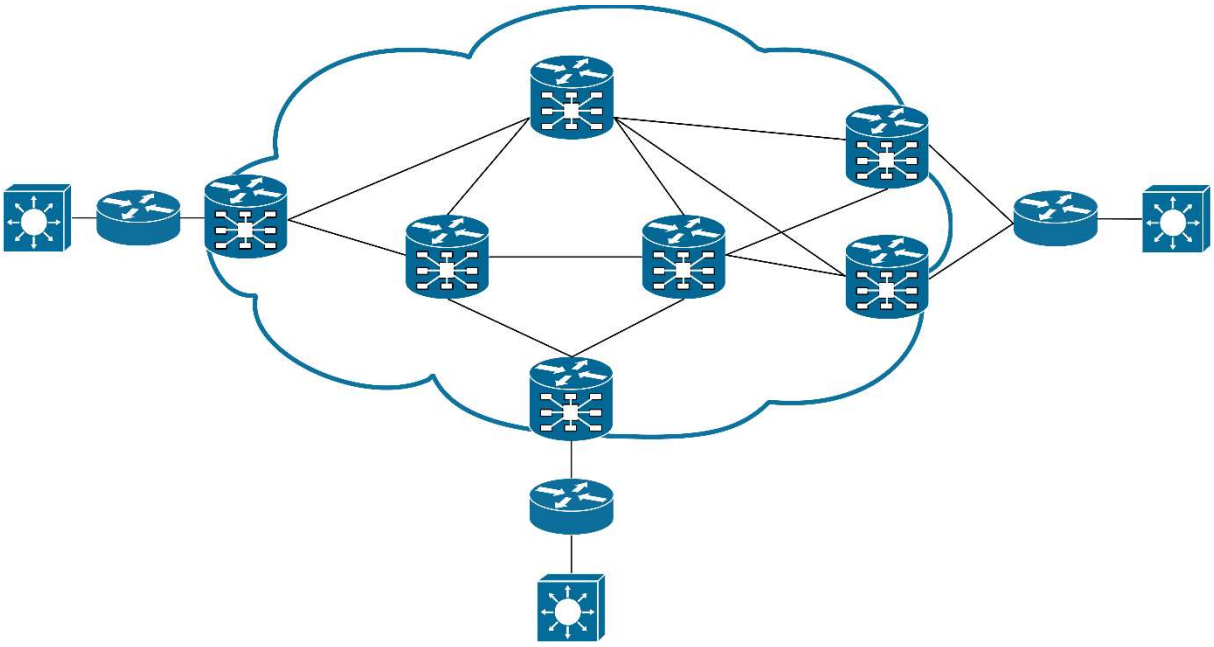
Práce s novým operačním systémem – IOS XR

Příprava sítě použité v dalších částech laboratorní úlohy

Pro více teorie se odkažte do bakalářské práce kapitoly 2.3.

Topologie

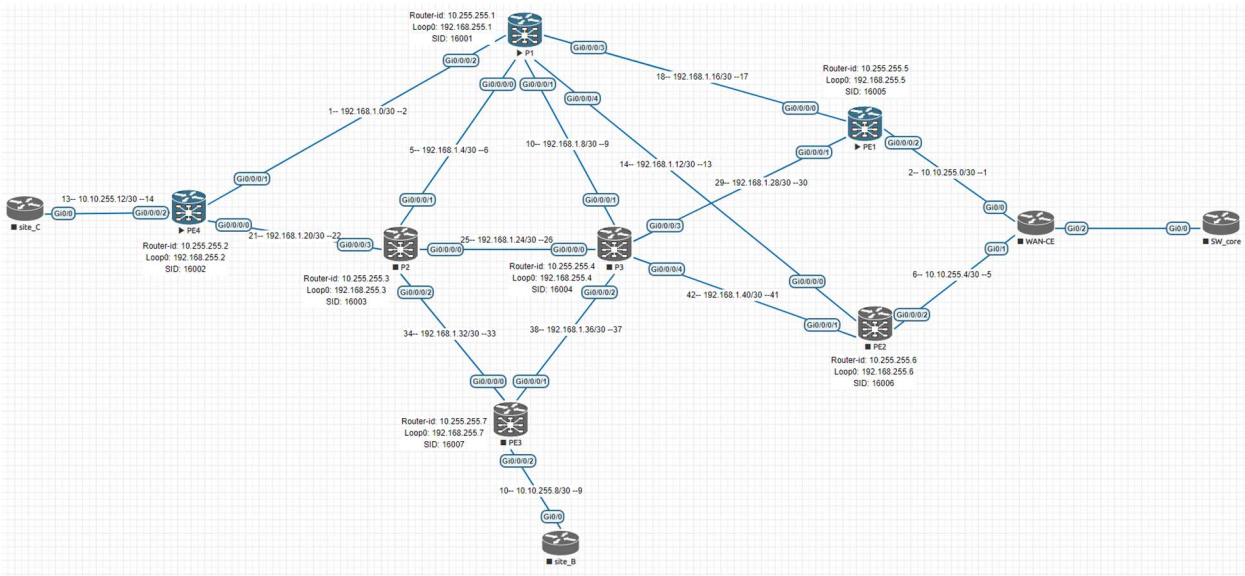
Na obrázku níže vidíme topologii sítě, se kterou budeme pracovat v této úloze.



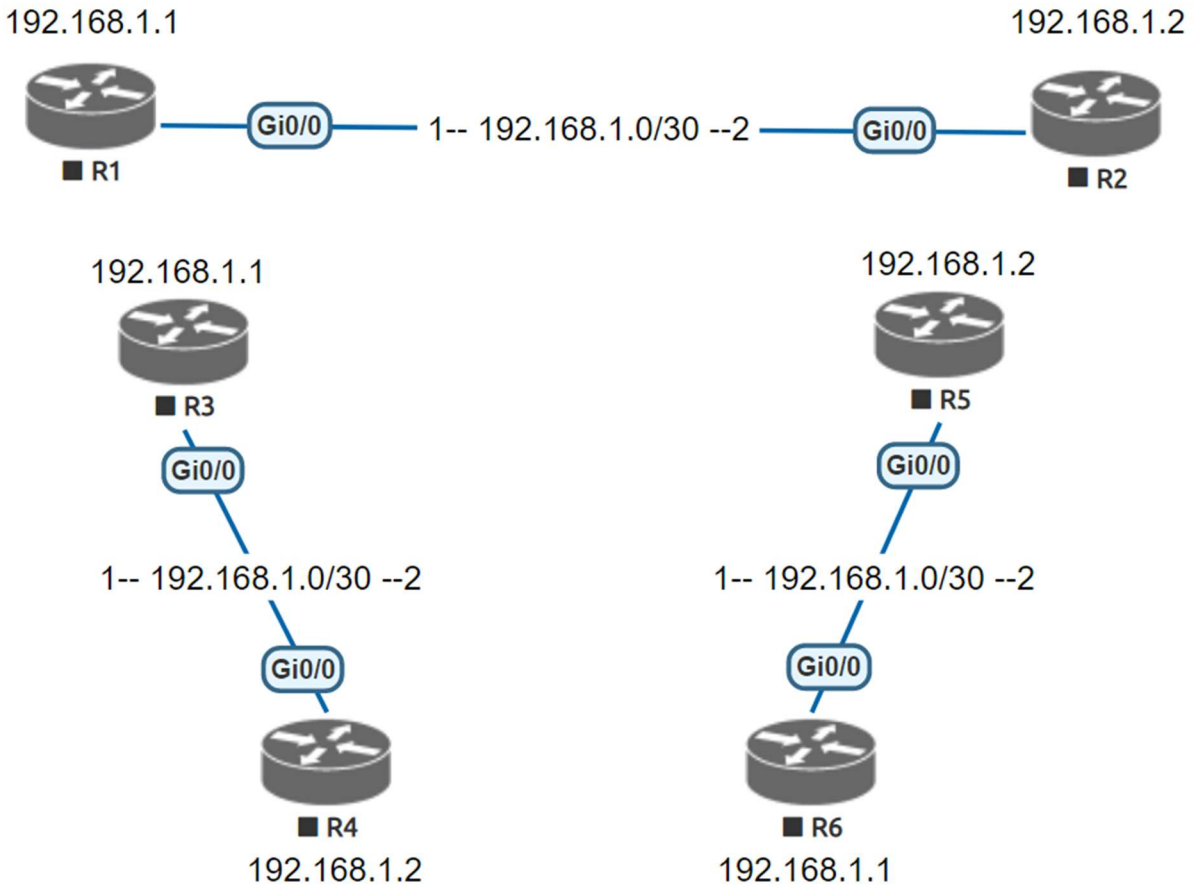
Než budete pokračovat, zapněte zařízení, hlavně ty uprostřed bubliny, jejich zapnutí trvá dlouho!

IP plán

Ip plán je napsaný přímo v úloze, je tak názornější a jednodušší na hledání než tabulka.



Při plánování je zachována snaha o systém, pro část sítě, která by standardně byla pod správou ISP, jsou použity podsítě s maskou 30 a je snaha držet v plánu vyšší číslo fyzicky nahoře, u vodorovných je pak volena adresace zleva nižší. Pro způsob zápisu je zvoleno net ID s maskou uprostřed a po stranách pouze host ID, ukázka viz obrázek.



Žádný standard neexistuje, pouze doporučení, a tudíž je na jednotlivém síťovém inženýrovi jaký způsob si vybere. Je však dobré se rozhodnout pro jednotný způsob adresace a snažit se ho držet,

může se tak předejít různým problémům vzniklých často lidskou chybou a zmatení při implementaci. Nevylučuje se ani použití obou metod zároveň.

Po zapnutí

Hlavní část bude věnována zařízením Cisco XRv 9000, někdy zkracováno jako XRv9K.

Hned na úvod jste vyzváni pro zadání jména a hesla. Doporučuji dát něco univerzálního na všechny zařízení. Heslo musí mít délku alespoň 6 znaků. Jelikož se nacházíme ve virtuálním prostředí, stačí jednoduché, např. **admin** a **Cisco123**.

Nejdříve je nutné jako vždy nastavit ip adresy k jednotlivým portům.

Již zde budou vidět změny, pokud byl v IOS XE příkaz ip address pak v XR bude ipv4 address, rozlišujeme tedy v4 a v6 adresní prostory. Je to změna způsobená přípravou na funkci v sítích, které často využívají v6 adresy.
Další změna se týká ukládání konfigurace. Ta probíhá příkazem **commit**, který musíme zadat ještě v konfiguračním módu. Pokud se pokusíme odejít příkazem exit do privilegovaného módu, vyzve nás k uložení nebo smazání konfigurace. Případné smazání konfigurace s chybou provedeme příkazem **clear**.

Je dokonce možné se podívat jaké změny byly udělány před jejich uložením příkazem **show commit changes diff** po kterém vyjede seznam změn, na začátku řádku jsou vidět + a – podle toho, zda se změny přidají do konfigurace nebo zda se uberou.

Obrázek níže ukazuje příklad výstupu. Vidíme, že byl přidán loopback 0 a nastavena ipv4 adresa, zároveň byl zadán příkaz no shutdown a proto z konfigurace zmizí (označeno mínus na začátku řádku) shutdown pro tento port. Následně vidíme, že interface gi 0/0/0/0 už byl vytvořen a proto na začátku řádku nemá žádné znaménko.

```
RP/0/RP0/CPU0:ios(config)#show commit changes diff
Wed Jul 20 07:33:50.180 UTC
Building configuration...
!! IOS XR Configuration 7.2.2
+  interface Loopback0
+   ipv4 address 192.168.255.1 255.255.255.255
-   shutdown
   !
   interface GigabitEthernet0/0/0/0
+   ipv4 address 192.168.1.6 255.255.255.252
-   shutdown
   !
end
```

Na příkladu níže vidíme, že už zadáváme masku jako /{maska sítě}

```
RP/0/RP0/CPU0:ios(config)#interface gigabitEthernet 0/0/0/1
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 192.168.1.1/30
RP/0/RP0/CPU0:ios(config-if)#no shutdown
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#commit
```

Po nastavení portů je vhodné si zkontrolovat správnost nastavení show příkazy. Které příkazy použijete?

Zároveň zde funguje command chaining jako v příkladu níže.

RP/0/RP0/CPU0:ios(config)#interface gigabitEthernet 0/0/0/3 ipv4 address 192.168.1.18/30

Nefunguje však se všemi příkazy, níže napsaný příkaz nebude fungovat.

RP/0/RP0/CPU0:ios(config)#interface gigabitEthernet 0/0/0/3 no shutdown

Všechny **no** příkazy musí no začínat jako v příkladu níže.

RP/0/RP0/CPU0:ios(config)#no interface gigabitEthernet 0/0/0/3 shutdown

Můžeme samozřejmě skládat více příkazů dohromady, takto lze zkrátit příkaz s další části:

RP/0/RP0/CPU0:ios(config)#router ospf LAB area 0 interface gigabitEthernet 0/0/0/3 network point-to-point

OSPF

Dále je potřeba nastavit OSPF, jelikož momentálně se navzájem sítě nevidí. Nastavte nejdříve pouze zařízení P1 a PE4.

RP/0/RP0/CPU0:ios(config)#router ospf LAB {LAB = název}
RP/0/RP0/CPU0:ios(config-ospf)#router-id {id dle plánu – 10.255.255.1}
RP/0/RP0/CPU0:ios(config-ospf)#area 0
RP/0/RP0/CPU0:ios(config-ospf-ar)#interface gigabitEthernet 0/0/0/1
RP/0/RP0/CPU0:ios(config-ospf-ar-if)#network point-to-point

Před nastavením zbytku sítě si uložte výstřižky výstupu jednotlivých příkazů.

Show ospf interface brief

Show ospf neighbor

Show route ipv4

show ospf database router 10.255.255.2 (router-id routeru opačnému než na kterém příkaz zadávám)

Následně nastavte i zbytek zařízení a znovu proveďte show commandy. Všimněte si, že zatím co ukazují často velmi podobnou věc, některé detaily se dozvíte až v databázi nebo záznamu neighbor.

Je dobré si u častěji používaných protokolů ukládat výstřižky, jelikož po nějaké době můžete zapomenout kde jakou informaci hledat.

Jelikož nenastavujeme multi-area ospf, používáme area 0. V případě, že bychom chtěli více area, tak area 0 slouží jako pomezí mezi nimi, protože se učí ospf cesty ze všech oblastí.

Všimněte si, že pro nastavení ospf na port není potřeba vrátit se do konfiguračního režimu a přejít pod port. Toto ukazuje protokol centrické chování, na rozdíl od IOS XE.

Segment Routing

Zapneme segment routing v globální konfiguraci.

```
RP/0/RSP0/CPU0:router(config)# segment-routing
RP/0/RSP0/CPU0:router(config-sr)# global-block 16000 22000
```

Po zapnutí segment routingu je vhodné nastavit globální segment ID. Ten nastavíme na předem připraveném portu loopback0. Pro jednoduchost použijeme stejné SID jako poslední oktet IP adresy.

Můžeme použít 2 možnosti nastavení, indexem nebo absolutní hodnotou. V případě, že by zařízení jiného výrobce mělo jiný rozsah indexů, metoda indexy upraví hodnoty správně od počátku množiny, kdežto metoda absolutní hodnoty by nemusela fungovat, pokud by se v této části množiny zařízení neshodly. Zařízení si sama pošlou informace o svých rozsazích.

Zde jsou použita stejná zařízení, můžeme využít obě metody a poté zkontrolovat jejich rovnost. Segment routing global block, neboli rozsah našich zařízení je 16000 – 22000. Od indexu 24000 až přes milion jsou rezervovány pro lokální SID, které si nastavují sama zařízení.

```
RP/0/RSP0/CPU0:router(config)# router ospf LAB {LAB = název}
RP/0/RSP0/CPU0:router(config-ospf)# area 0
RP/0/RSP0/CPU0:router(config-ospf-ar)# interface Loopback0 passive
RP/0/RSP0/CPU0:router(config-ospf-ar)# prefix-sid index 5
```

nebo

```
RP/0/RSP0/CPU0:router(config-ospf-ar)# prefix-sid absolute 16005
```

SID musí být nastaveno na portu. Nastavujeme ho na loopback, protože loopback nespadne, když dojde k přerušení linky. Kdybychom nastavili SID na jeden z portů tak spadnutím linky se může stát, že celý router nebude dostupný i přesto že bude existovat validní cesta.

MPLS

Samotné zapnutí segment routingu nestačí, ještě je nutné na zařízení nastavit, aby segment routing labely preferoval nad IP routingem. Můžeme zapnout globálně nebo pod specifickou area.

```
RP/0/RSP0/CPU0:router(config)# router ospf LAB {LAB = název}
RP/0/RSP0/CPU0:router(config-ospf)# area 0
RP/0/RSP0/CPU0:router(config-ospf-ar)# segment-routing mpls
```

Ověříme si nastavení labelů příkazem:

```
Show mpls label table
```

```
Show mpls label table detail
```

Závěr

Nyní máme hotovou dílčí část sítě, která slouží jako základ použitelný pro třetí úlohu. Seznámili jsme se s topologií. Nakonfigurovali zařízení, aby spolu komunikovala pomocí OSPF. Následně nastavili segment routing a přiřadili labely jednotlivým zařízením. A předpřipravili si síť globálním zapnutím MPLS.