FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání diplomové práce

Název:	Nasazení a analýza chování Softwarově definovaných sítí v simulovaném prostředí
Student:	Bc. Josef Zápotocký
Vedoucí:	Ing. Alexandru Moucha, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Počítačové systémy a sítě
Katedra:	Katedra počítačových systémů
Platnost zadání:	do konce letního semestru 2023/2024

Elektronicky schválil/a prof. Ing. Pavel Tvrdík, CSc. dne 13. února 2023 v Praze.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Pokyny pro vypracování

Softwarově definované sítě (SDN) dávají výhody oproti tradičním sítím v oblastech škálovatelnosti, dostupnosti, řiditelnosti a segmentace. Nicméně, řešení rozlehlých sítí, jako jsou SD-WAN a SD-Access od Cisco, Mikrotiku (OpenFlow), Huawei a dalších, nejsou vždy dostupné pro účely škol z důvodu nedostupnosti či vysokých finančních nákladů na hardwarové prostředky. Cílem této diplomové práce je, na základě poznatků bakalářské práce [1], nasadit řešení SD-WAN různých poskytovatelů a jejich simulaci a otestovat případnou vzájemnou kompatibilitu komunikace použitím veřejně dostupných simulátorů síťové infrastruktury (EVE-NG, GNS3, případně Cisco VIRL).

Analyzujte tok dat, způsoby monitorování řešení technologií SD-WAN od společnosti Cisco a jiných společností a otevřených řešení na veřejně dostupných simulátorech. Výsledkem analýzy bude srovnání jednotlivých funkcionalit a efektivity komunikace mezi jednotlivými implementacemi a otestování možné spolupráce zařízení různých výrobců v oblasti tvoření SD-WAN. Následně zjistěte, jak jednotlivé implementace různých výrobců jsou kompatibilní s otevřenými implementacemi SD-WAN. Jednotlivé příklady implementujte do simulátorů, na kterých budou jednotlivé implementace porovnány, popište možné použití simulátorů ve výuce. Nakonec zjistěte možnost konfigurace vnějších zařízení ze zařízení uvnitř simulátoru a případné napojení simulátoru na reálná zařízení.

[1] M. Lanča: Analýza a implementace simulovaného prostředí pro SDN, Bakalářská práce FIT ČVUT, 2022.

Elektronicky schválil/a prof. Ing. Pavel Tvrdík, CSc. dne 13. února 2023 v Praze.

Diplomová práce

NASAZENÍ A ANALÝZA CHOVÁNÍ SOFTWAROVĚ DEFINOVANÝCH SÍTÍ V SIMULOVANÉM PROSTŘEDÍ

Bc. Josef Zápotocký

Fakulta informačních technologií
Katedra teoretické informatiky
Vedoucí: Ing. Alexandru Moucha, Ph.D.
9. května 2024

Obsah

Poděkování ix

Prohlášení x

Abstrakt xi

Seznam zkratek xii

Úvod 1

1 Problematika softwarově-definovaných sítí 2
1.1 Dnešní implementace sítí . . . 2

2 Volba virtuálních prostředí 6
2.1 Virtualizace síťových zařízení . . . 6
2.2 Simulátory sítí . . . 6
2.2.1 Emulated virtual environment (EVE-NG) . . . 7
2.2.2 Graphical Network Simulator (GNS3) . . . 7
2.3 Použitý hypervizor Cisco ESXi . . . 8
2.4 Cisco SDN . . . 8
2.4.1 SD-WAN . . . 9
2.4.2 SD-ACCESS . . . 11
2.5 Huawei SDN . . . 12
2.5.1 SD-WAN . . . 13
2.5.2 CloudCampus . . . 15
2.6 Mikrotik/FOSS SDN . . . 15
2.6.1 OpenFlow . . . 15
2.7 FOSS SDN . . . 16
2.7.1 Zařízení . . . 16
2.7.2 Kontroléry . . . 18

3 Přístup Cisca k SD-WAN 21
3.1 Fyzické sítě . . . 22
3.2 Logické sítě . . . 22
3.2.1 Segmentace logických sítí . . . 23
3.2.2 Detekce obousměrného směrování . . . 23
3.3 Zařízení/součásti SD-WAN . . . 24
3.3.1 vManage - Řídící vstva . . . 24
3.3.2 vSmart - Kontrolní vstva . . . 25
3.3.3 WAN-edge zařízení - Datová vrstva . . . 26
3.3.4 vBond - Orchestrační vrstva . . . 28
3.3.5 Ostatní volitelná zařízení . . . 30

4 Přístup Huawei k SD-WAN 32
4.1 Prvky řešení Huawei SD-WAN . . . 32
4.2 Fyzické sítě . . . 33
4.3 Logické sítě . . . 33
4.3.1 Příklady logických sítí . . . 34
4.3.2 Segmetace logických sítí . . . 36
4.4 Vrstva prezentace služeb . . . 37
4.5 Vrstva řízení a orchestrace sítě . . . 37
4.6 Vrstva síťového připojení . . . 38

5 Přístup Mikrotiku/FOSS k SD-WAN 39
5.1 OpenFlow . . . 39
5.1.1 Jak OpenFlow funguje . . . 40
5.1.2 Výhody OpenFlow . . . 41
5.2 Referenční přepínač OpenFlow . . . 42
5.2.1 Mikrotik . . . 43

6 Nástroje na analýzu toku dat 44
6.0.1 Wireshark . . . 44
6.0.2 Netflow analyzer . . . 45

7 Příprava implementací 46
7.1 Požadavky laboratoře . . . 46
7.1.1 Cisco SD-WAN laboratoř . . . 46
7.1.2 Rozsáhlá Cisco SD-WAN laboratoř . . . 46
7.1.3 Mikrotik/OpenFlow SDN laboratoř . . . 47
7.1.4 Řešení laboratoře OpenFlow SDN s více výrobci . . . 48
7.2 Potřebné licence . . . 49
7.3 Instalace EVE-NG . . . 50
7.3.1 Dodatečný software . . . 51
7.3.2 Obrazy zařízení . . . 51
7.3.3 Přístup do vnější sítě . . . 54
7.4 Instalace GNS3 . . . 55
7.4.1 Dodatečný software . . . 56
7.4.2 Obrazy zařízení . . . 57
7.4.3 Přístup do vnější sítě . . . 59

8 Cisco Laboratoře 62
8.1 Příprava fyzické sítě . . . 62
8.1.1 WAN okruhové přepínače . . . 62
8.1.2 Výchozí brána . . . 63
8.1.3 Docker Jumpbox . . . 65
8.1.4 VPN přepínač . . . 65
8.1.5 VPN směrovač . . . 65
8.1.6 Směrovač s NATem . . . 65
8.1.7 Směrovač se Statickým NATem . . . 65
8.2 Správa certifikátů a začleňování směrovačů . . . 68
8.2.1 vManage . . . 69
8.2.2 vSmart . . . 69
8.2.3 vBond . . . 70
8.2.4 Certifikáty . . . 72
8.3 vEdge a jejich začleňování . . . 77
8.3.1 Manuální konfigurace . . . 77

8.4 cEdge . . . 78
8.5 Předlohy . . . 82
8.5.1 Předlohy příkazové řádky . . . 82
8.5.2 Předlohy vlastností . . . 83
8.6 ZTP . . . 88
8.7 Procházení NATem . . . 92
8.8 Směrování . . . 93

9 FOSS Laboratoře 102
9.1 Kontroléry OpenFlow . . . 102
9.1.1 OpenDaylight . . . 102
9.1.2 ONOS . . . 104
9.1.3 Floodlight . . . 104
9.2 Zařízení OpenFlow . . . 106
9.2.1 Mikrotik . . . 107
9.2.2 Cisco . . . 108
9.2.3 Open vSwitch . . . 109
9.3 Funkcionality OpenFlow SDN . . . 110

Obsah příloh 118

Seznam obrázků

2.1 Cisco SD-WAN Multi-Tenancy 10
2.2 Huawei SD-WAN Multi-Tenancy 13
2.3 OpenFlow SD-WAN Architektura 16
2.4 Open vSwitch Architektura 17

3.1 Cisco SD-WAN Architektura 21
3.2 Cisco SD-WAN Segmentace 23
3.3 Cisco SD-WAN vManage 24
3.4 Cisco SD-WAN řídící vrstva 26
3.5 Cisco SD-WAN STUN 29
3.6 Cisco SD-WAN Orchestrační vrstva 31

4.1 Huawei SD-WAN Architektura 32
4.2 Huawei SD-WAN řídící vrstva 33
4.3 Huawei SD-WAN mapování logické sítě 34
4.4 Huawei Datacenter mapování datacentrové logické sítě 35
4.5 Huawei SD-WAN mapování logické SD-WAN 36
4.6 Huawei SD-WAN segmentace 37
4.7 Huawei SD-WAN řídící vrstva 38
4.8 Huawei iMaster NCE 38

5.1 OpenFlow 40
5.2 OpenFlow pipeline 41
5.3 OpenFlow tabulka toků 41
5.4 OpenFlow zpracování komunikace 42
5.5 OpenFlow přepínač 43

6.1 Grafolean analyzér 45

7.1 Cisco SD-WAN EVE-NG 47
7.2 Cisco SD-WAN GNS3 48
7.3 Rozšířená Cisco SD-WAN EVE-NG 49
7.4 Rozšířená Cisco SD-WAN GNS3 50
7.5 OpenFlow SDN EVE-NG 51
7.6 OpenFlow SDN GNS3 52
7.7 Rozšířená OpenFlow SDN EVE-NG 53
7.8 Rozšířená OpenFlow SDN GNS3 54
7.9 EVE-NG cloud 55
7.10 GNS3 VM 1 56
7.11 GNS3 VM 2 56
7.12 Vytváření zařízení v GNS3 1 57
7.13 Vytváření zařízení v GNS3 2 58
7.14 Vytváření zařízení v GNS3 3 59
7.15 Vytváření zařízení v GNS3 4 59

7.16 Vytváření zařízení v GNS3 5 60
7.17 Vytváření zařízení v GNS3 6 60
7.18 GNS3 vnějšek 1 61
7.19 GNS3 vnějšek 2 61

8.1 vManage úvodní konfigurace 70
8.2 Generování CA 72
8.3 Distribuce CA do kontrolérů 73
8.4 Instalace kořenového CA 74
8.5 vManage Enterprise CA 74
8.6 Generování CSR 75
8.7 Distribuce certifikátů 75
8.8 Cisco SD-WAN Smart 76
8.9 Cisco SD-WAN sériový soubor 1 76
8.10 Cisco SD-WAN sériový soubor 2 77
8.11 Cisco SD-WAN sériový soubor 3 77
8.12 Cisco SD-WAN sériový soubor 4 78
8.13 Cisco SD-WAN sériový soubor 5 78
8.14 Cisco SD-WAN sériový soubor 6 79
8.15 Cisco SD-WAN sériový soubor 7 79
8.16 Cisco SD-WAN sériový soubor 8 80
8.17 Cisco SD-WAN sériový soubor 9 80
8.18 Cisco SD-WAN sériový soubor 10 80
8.19 Cisco SD-WAN sériový soubor 11 81
8.20 Cisco SD-WAN sériový soubor 12 81
8.21 Cisco SD-WAN sériový soubor 13 82
8.22 Cisco SD-WAN sériový soubor 14 82
8.23 Cisco SD-WAN sériový soubor 15 83
8.24 Cisco SD-WAN sériový soubor 16 83
8.25 Nahrání sériového souboru 84
8.26 Ruční nahrání sériového souboru 84
8.27 Seznam zařízení 85
8.28 Cisco předlohy Cli 1 86
8.29 Cisco předlohy Cli 2 86
8.30 Cisco předlohy Cli 3 87
8.31 Cisco předlohy Cli 4 88
8.32 Cisco předlohy vlastností 1 88
8.33 Cisco předlohy vlastností 1 89
8.34 Cisco předlohy vlastností 2 90
8.35 Cisco předlohy vlastností 3 90
8.36 Cisco předlohy vlastností 4 91
8.37 Cisco předlohy vlastností 5 91
8.38 Cisco předlohy vlastností 6 92
8.39 Cisco předlohy vlastností 7 92
8.40 Cisco předlohy vlastností 8 93
8.41 Cisco předlohy vlastností 9 93
8.42 Cisco předlohy vlastností 10 94
8.43 Cisco předlohy vlastností 11 94
8.44 Cisco Feature Template 12 95
8.45 Cisco předlohy vlastností 13 95
8.46 ZTP Server validní zařízení 96
8.47 ZTP vEdge 96

8.48 NAT Traversal . . . 97
8.49 OMP spojení . . . 97
8.50 OMP spojení 1 . . . 97
8.51 OMP spojení 2 . . . 98
8.52 OMP spojení 3 . . . 98
8.53 OMP spojení 4 . . . 98
8.54 OMP spojení 5 . . . 99
8.55 OMP spojení 6 . . . 99
8.56 OMP spojení 7 . . . 100
8.57 OMP spojení 8 . . . 100
8.58 OMP VPN segmentace . . . 101

9.1 OLD přihlášení . . . 104
9.2 Cisco OFM . . . 106
9.3 ONOS přihlášení . . . 106
9.4 Floodlight přihlášení . . . 108
9.5 Openflow Architetura . . . 108
9.6 Mikrotik 1 . . . 109
9.7 Mikrotik 2 . . . 109
9.8 Mikrotik 3 . . . 110
9.9 Mikrotik 4 . . . 111
9.10 Mikrotik 5 . . . 112
9.11 Mikrotik v kontroléru Onos . . . 112
9.12 Ukázková konfigurace YANG . . . 113

Seznam tabulek

2.1 Srovnání sp . . . 8
2.2 SD-Access . . . 12

7.1 Cisco SD-WAN . . . 47
7.2 Cisco SD-WAN . . . 48
7.3 Cisco SD-WAN . . . 49
7.4 Cisco SD-WAN . . . 51

Seznam výpisů kódu

8.1 WAN přepínač . . . 63
8.2 Gateway směrovač . . . 64
8.3 VPN přepínač . . . 66
8.4 VPN směrovač . . . 66

8.5 NAT směrovač . . . 67
8.6 Směrovač realzující statický NAT . . . 68
8.7 Konfigurace vManage . . . 69
8.8 Konfigurace vSmart . . . 71
8.9 Konfigurace vBond . . . 71
8.10 Generování vlastních kořenových certifikátů . . . 72
8.11 Možná distribude kořenových certifikátů do kontrolérů . . . 72
8.12 Instalace kořenového certifikátu . . . 73
8.13 Podepisování certifikátů . . . 73
8.14 Minimální manuální konfigurace vEdge . . . 85
8.15 Konfigurace ZTP serveru . . . 89
9.1 Instalace Open Daylght . . . 103
9.2 Instalace Cisco OFM . . . 104
9.3 Instalace ONOS . . . 105
9.4 Instalace a nasazení Floodlight . . . 107
9.5 Minimální konfigurace OpenFlow na Cisco IOS zařízení . . . 112

Chtěl bych poděkovat především vedoucímu práce Ing. Alexandru Moucha, Ph.D., za zpřístupnění svého hardwaru a softwaru virtuálních cisco zařízení pro vypracovní této diplomové práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 9. května 2024

Abstrakt

Tato práce se zabývá simulací nasazením softwarové-definovaných sítí ve virtuálních prostředí a srovnáním jednotlivých řešení softwarově-definovaných sítí od různých výrobců, jako jsou Cisco, Huawei a Mikrotik. Jelikož se jedná o analýzu ve virtuálních prostředí, práce se zaměřuje na implementaci a analýzu především na SD-WAN. Jelikož v komerních prostředích se ne vždy můžeme setkat pouze s jedním výrobcem řešení softwarově-definovaných sítí, je tudíž potřeba, aby jednotlivé stroje, ne nutně stejného výrobce, fungovali pod jedním ekosystémem. Cílem práce je vysvětlit postupi nasazení softwarově-definovaných sítí ve virtuálních prostředí různých řešní, zjistit případnou možnost komunikace simulace s reálnými stroji a analyzovat tok dat a způsoby monitorování jednotlivý řešení SD-WAN. Všechny postupi zde jsou detailně popsány pro případnou replikaci. Tato práce navazuje na bakalářskou práci Analýza a implementace simulovaného prostředí pro softwarově-definované sítě v oblasti simulovaných prostředí a rozšiřuje ji o analýzu toku dat v softwarově-definovaných sítí.

Klíčová slova SDN, SD-WAN, NetFlow analýza, OpenDaylight, ONOS, EVE-NG, GNS3, Cisco, síťová simulace, virtualizace, NETCONF, OpenFlow

Abstract

This thesis deals with the simulation of software-defined networking deployment in virtual environments and compares different software-defined networking solutions from different vendors such as Cisco, Huawei and Mikrotik. Since this is an analysis in a virtual environment, the thesis focuses on the implementation and analysis mainly on SD-WAN. Since in commercial environments we cannot always encounter only one manufacturer of software-defined networking solutions, it is therefore necessary that individual machines, not necessarily from the same manufacturer, operate under one ecosystem. The aim of this paper is to explain the deployment of software-defined networking in virtual environments of different solutions, to investigate the possible communication of the simulation with real machines and to analyze the data flow and monitoring methods of individual SD-WAN solutions. All procedures are described here in detail for possible replication. This thesis builds upon the bachelor thesis Analysis and Implementation of Simulated Environments for Software-Defined Networks in the area of simulated environments and extends it by analyzing data flow in software-defined networks.

Keywords SDN, SD-WAN, NetFlow analysis, OpenDaylight, ONOS, EVE-NG, GNS3, Cisco, network simulation, virtualization, NETCONF, OpenFlow

Seznam zkratek

ACL	Access-Control List
AP	Access Point
AS	Autonomous System
BGP	Border gateway protokol
BSS	Business Support System
CA	Certification Authority
CIS	Cybersecurity Intelligence System
CLI	Command Line Interface
CORD	Central Office Re-architected as a Datacenter
CPE	Customer Premises Equipment
CSR	Cloud Service Router
ČVUT	České vysoké učení technické
CUCM	Cisco Unified Communications Manager
DC	Datacenter
DNA	Digital Net Architecture
DTLS	Datagram Transport Layer Security
ECMP	Equal-cost multipath
EOL	End of Life
ESXi	Elasti Sky X intergrated
EVE-NG	Emulated Virtual Environment
EVPN	Ethernet Virtual Private Network
FIT	Fakulta informačních technologií
FOSS	Free & Open Source Software
GRE	Generic Routing Encapsulation
GUI	Graphical User nterfae
HSRP	Host Standby Router Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypetext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IDN	Intent-Driven Network
IP	Internet Protocol
ISE	Identity Service Engine
ISR	Integrated Service Router
IWG	InterWorking Gateway
KVM	Kernel Virtal Machine
LAN	Local Area Network
LACP	Link Aggregation Control Protocol
LAN	Local Area Network
LTS	Long Term Support
MPLS	Multi-Protocol Labeled Switching
MSP	Managed Service Provider
NCE	Network Cloud Engine
NMS	Network Monitoring Software
NVO3	Network Virtualization Overlays
NVGRE	Network Virtualization using Generic Routing Encapsulation
OEM	Original Equipment Manufacturer
OFM	OpenFlow Management
OMP	Overlay Management Protocol
ONOS	Open Network Operating System
OS	Operating System
OSS	Operating Support System
OSPF	Open Shortest Path First
P4	Programming Protocol-independent Packet Processors
PNP	Plug and Play
QEMU	Quick Emulator
QoE	Quality of Experience

QoS	Quality of Service
RAM	Random Access Memory
RR	Route Reflector
SAL	Service Abraction Layer
SDN	Software-Defined Network
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SSH	Secure SHell.
SST	Single Spanning Tree
TCP	Trasmission Control Protocol
TLS	Transport Layer Security
VIRL	Virtual Internet Routing Lab
VLAN	Virtual Local Area Network
VN	Virtual Network
VRF	Virtual Rounting and Forwarding
VTEP	Virtual Tunnel End Point
VXLAN	Virtual eXtensible Local Area Network
WAN	Wide Area Network
ZTP	Zero Touch Provisioning

Úvod

V dnešním světe jsou počítačové sítě téměř nedílnou součástí našich každodenních životů, proto jejich funkčnost, či připadná nefukčnost, má velký dopad na chod našich životů. Však čirá velikost sítě v počtu nacházejících se aktivních a její případná komplexita nastavení dělá správu těchto sítí vcelku obtížnou a energeticky vytěžující úlohu, a nakonec tyto faktory mohou vést k častým výpadkům na síti a dlouhými časy zjištění jejich příčin a následnou opravou daných problémů. V takových to problémech přichází na scénu softwarově-definované sítě, jejichž výhoda je centralizovaná kontrola nad datovou a řídící vrstvou sítě, čímž usnadňují a zlepšují spoustu aspektů sítě. Důvod proč jsem se rozhodl psát tuto práci právě o SDN je, že právě všechny velké sítě se upírají tímto směrem, jako jsou například datová centra, školní sítě, sitě národních poskytovatelů síťových služeb či podnikové sítě a to je jen pár příkladů. V této práci navážu na bakalářskou práci Matěje Lanči Analýza a implementace simulovaného prostředí pro softwarově definované sítě [1], kde se bakalářská práce zaměřila na nasazení softwarově-definované sítě od Cisca se zařízenímy viptela vManage, vSmart a vEdge pro implementaci SD-WAN v simulovaném prostředí EVE-NG [2].

Tato práce předpokláká minimální znalost sítí v oblastech ISO/OSI modelu a TCP/IP síťového modelu, jiné znalosti potřebné pro vytvoření simulovaného prostředí a simulovaných sítí bude do jisté míry citováno z předchozích prací [1] a vysvětleno v kapitolách předcházející implementaci. Zde také budou do podrobna zmíněné výhody a důvody použití SDN oproti klasickým sítím a zmíněna možnost simulace zařízení cisco, mikrotik a huawei v simulovaném prostředí. Tím chci ukázat, že možnost simulovat reálná zařízení pomocí virtuálních zařízení v simulovaném prostředí není pouze skvělý výukový nástroj ale i upřednostňovaná vlastnost, jež mnoho firem používá k implementaci svých sítí a mnoho výrobců zařízení některé aktivní prvky sítě už nabízí pouze ve virtuální variantě. Jak funguje propojení virtuální sítě se sítí reálnou bude popsáno.

Práce byla implementována na zařízení R210-2121605W od společnosti Cisco s dvěmi procesory čiřjádrovými Intel(R) Xeon(R) CPU L5630 @ 2.13GHz, 191.97 GB, s operačním systémem VMware-ESXi-6.7.0-17700523-Custom-Cisco-6.7.3.1 (CISCO) upravený pro zařízení hypervizoru Cisco. Na tomto virtualizačním hardwaru poté běží virtualizační nástroje jako je EVE-NG pro následnou virtualizaci jednotlivých implementací SDN. Tyto laboratoře následně poslouží jako referenční implementace a na nich se pak bude analyzovat tok dat. Jak laboratoře byly implementovány budou popsány v následných kapitolách.

Kapitola 1

Problematika softwarově-definovaných sítí

V této kapitole se uvedu čtenáře do tématu softwarově-definovaných sítí v simulovaném prostředí, kde představím jednotlivé prostředky pro možnou simulaci softwarově-definovaných sítí. Dále v této kapitole vysvětlím, jak dnešní sítě fungují a s jakými překázkami se můžeme v nich setkat, proč vyžadují změnu a proč dává smysl tyto problémy řešit z pohledu firem. Dále zmíním jaké obecné problémy řeší softwarově-definované sítě, jaké komponenty jsou potřeba pro funkční SD-WAN ad SD-Access a představuji stručný pohled jejich funkcionalitu v SDN. Jelikož se vyskytuji převážně v simulovaném prostředí, SD-Access zde bude zmíněno krátce oproti SD-WAN, které bude popsáno více do podrobna. Důvodem je, že komponenty SD-WAN jsou na trhu nabízeny ve virtuální podobě oproti jejich variantám z SD-Access.

1.1 Dnešní implementace sítí

Jak postupem času se obory informačních technologií neustále mění a běří ku předu stále novým technologiím, tak neustále narůstají nároky na počítačové sítě, ať z hlediska propustnosti, latence či jakosti služeb, pak staré modely jsou nahrazovány modely novými. Nové technologie jsou vyvíjeny a následně zahrnuty do prostředí sítí. Nové technologie jsou pak následně vytvářeny aby přinesly rychlejší inovace a schopnosti přijmutí příslušných technologií zjednodušenou cestou.

Spolu s počtem nových technologií a jak rychle na trh přicházejií, se mohou administrátoři, odpovedni za chod počítačové sítě, jsou čím dál více zatíženi, jelikož nasazení danných technologií do stávajícího produkčního ekosystému často přináší nepříjemnou výzvu, jeliž stávající ekosystém nemusí být kompatibilní s novou technologií. Dalším problém je, že zahrnutí nové technologe nesmí ohrozit ani nijak omezit denní chod aplikací, kterých se zahrnutí nové technologie týká. Bohužel však v mnoha případech je nevyhnutelné, aby systémy zůstaly nezměněny a nová abstrakce je nutno implemetovat. Proto v dnešní době se mnohdy snažíme zmírnit cenu provozu, což jde ruku v ruce se snízením komplexity problému.

Můžeme říct, že každa síť, do určité míry, má nějakou úroveň komplexnosti, a tudíž vlastnit nástroje, které mohou pomoci tyto komplexní stě řídit, se stává dnem ode dne potřebnější a následlá vzniklá automatizace se stává cílem, kterého se podniky dosáhnou. Důvodem je, že dnešní počítačové se průběžně stávají čím dál více komplikované a nedopustit se chyby ze strany administrátora je téměř nemožné, jelikož podniky často nechávají velké sítě spravovat pouze malou hrstkou lidí. Tím, že se podniky snaží šetřit kde se dá, vede k situacím, že ve správě sítí je málo lidí a síťová zařízení často tvoří zařízení různých výrobců, navzdory tomu, že administrátoři se nemusí nutně specializovat v konkrétním ekosystému,než na jaký byly zvyklí. Dalším důvodem

přijetí technologií SDN je obecné zlepšení QoE v rámci spravované sítě, jež zahrnuje možnost uživatele sítě mít jednoduchý přístup ke kritickým aplikacím odkudkoliv ze sítě. Kromě zlepšení QoE pro uživatele, automatizace se aktvně snaží zjednodušovat operace v síti.

Zde je představen pohled, jak většina sítí je řízena. Jedním z hlavních nástrojů na řízení síťových zařízení dlouho zastává tradiční konsole (CLI) jelikož nevyžaduje grafické prostředí, které mnohdy zabírá desítky megabytů RAM paměti. Podle zdrojů z Cisca až 95% všech jejich zákazníků, stále nastavují svá ařízení manuálně. Tato čísla jsou však v dnešní době docela dost velká, jelikož, jak by zmíněno dříve, většina problémů vzniká právě ručním nastavení zařízení (user error).

V tomto listu vyjmenuji právě několik problémů, které dokáží SDN řešit. U každého z problémů obecně napíšu, jak SDN daný problém řeší.

Segmentace sítě v dnešním světě kdy se velmi často používají adhoc sítě a sítě pro hosty má smysl dělat, jelikož je potřeba aby se rozlišilo k čemu jednotlivý uživatelé sítě mají přístup, například právě zmíněné rozdělené na síť pro hosty a sít pro pracovníky v malé firmě či domácnosti. U větších sítí je už potřeba mít schopnost rozdělit sítě do vícero segmentů podle oddělení, administrace či DMZ. Segmentaci sítě dále rozdělujeme na:

- **Mikrosegmentace** je schopnost rozdělení datové komunikace podle jednotlivých zařzení v síti. Bohužel v dnešních datových sití je to pořád velmi těžká úloha k realizaci. Jeden ze způsobů jak dneska řešit mikrosegmentaci na síti, jde dobře pouze na 2. vrstvě ISO/OSI modelu za použití Access-Control Listů (ACL), a to ještě jen v případech, zařízení se nepohybují po síti. Zkusme si představit případ, kdy zařízení je připojeno pomocí bezdrátového signálu a zařízení se přesouvá mezi jednotlivými přístupovými body (AP). V takových to případech by bylo potřeba, aby daný ACL pravidla byla nasazena na všechny aktivní prvky sítě spravující ACL, nemluvě o tom, že každé z techto zařízení má jen omezenou velikost ACL tabulky, a představa provádět změny v takovéto síti je pomalu nereálná. Dalším problémem, který může tento způsob řešení rozbýt je změna vysílané MAC na bezdrátovém rozhraní klienta. Z opačného pohledu může být samotný návhr sítě problém, kdy navrhovaná síť se může chovat jinak než jak na první pohled vypadá. Příkladem může být případ, kdy v síti se nachází řada dynamicky vytvořených tunelů. Z pohledu mikrosegmentace je to nespravovatelné, ale když máme kontrolér, který dynamické tunely vytváří na základě námi daných politik, tak mít takto vytvářené point-to-point spojení v sítích má mnoho výhod, jako například pravidla mající vyšší granuralitu. Zajímavá myšlenka je zde identifikace zařízeni na základě přihlášovacích údajů nikoliv pouze IP adresou.
- **Makrosegmentace** je schopnost rozdělovat datové komunikace podle sítě. Tento problém se v počítačových sítí řeší pomocí virtuálními lokálními sítěmi (VLAN) a ACL na třetí vrstvě. Tento způsob řešení pak ale trpí na škálovatelnost, manipulovatelnost a schopnost se přispůsobit pod společná pravidla. Tyto problémy navíc také postihují i mikrosegmentaci. Zde je právě lepším řešním použití technologií SD-Access, kde v tradičních sítí se komunikace mezi jednotlivými segmenty řeší pomocí VRF na směrovačích a L3 přepínačích tak zde například Cisco SD-Access to nazývá Virtuální sít (VN) a realizuje to v Cisco DNA Center.

Verzování nastavení na aktivních prvích sítě je jedna z ne příliš častých praktik požívaných v dnešních počítačových sítí, a proto konzistence verzí je v praxi velmi těžké docílit, přesto v dnešní době je verzování nastavení zařízení klíčová vlastnost, kterou by měl mít k dispozici každý síťový administrátor. Jednou z možností je použití SNMP pro zjišťování změn v konfiguraci oproti nějaké lokální kopii a pak případně aplikovat kopii na postižené zařízení. SDN však nabízí mnohem elegantnější řešení v podobě centrálního řízení všech zařízení.

Společný přístup a aplikační stabilita v dnešních sítích je velmi žádanou vlastností. V mnoha podnikových sítí je nutnost přístupu do podnikové sítě nejen pomocí kabelu či bezdrátového spoje, ale také z Internetu, kvůli nutnosti zajistit přístup pro administrátory nejen z vnitřku

sítě. Samozřejmě toho si administrátor může docílit různými způsoby tunelů a podobně, zde je však kladen důraz na to, aby spojení bylo možno co nejjednodužší. „Více než 90 % podniků nyní využívá multicloudová a hybridní (veřejná/soukromá) cloudová prostředí jako kritickou součást své síťové a obchodní strategie. Výsledkem je, že důležitá data a kritické podnikové aplikace mohou být umístěny kdekoli - ať už v areálu firmy, na pobočce, v datovém centru nebo v cloudu. A samozřejmě by mělo být pro koncového uživatele transparentní, kde tyto aplikace sídlí. Zabezpečená síť SD-WAN poskytuje organizacím spolehlivou a bezpečnou konektivitu a uživatelské prostředí, které potřebují pro svou cestu do cloudu.“ [3] Vlastními slovy, síť by se měla chovat jako kdyby administrátoři byly bezpečně přímo připojeni do firemní sítě poskytující kritické firemní služby. Toto by měla být možnost, kterou dodává minimálně SD-WAN od společnosti Cisco, zprostředkovat SLA pro kritické služby s možností použít šifrovaných tunelů a QoE.

Pomalé nasazení tradičních sítí je právě jeden z důsledků nutnosti škálovat síť. Nakonec pak samotná pracovní síla se stává úzkým hrdlem tradičních sítí. Toto může postihnou v podnicích kritické systémy, buďto jejich přístupnost z jiných sítí, či zcela znemožnit jejich působení. Také právě jeden z důvodů proč automatizace, která přišla společně s SDN, je nepostradatelný nástroj.

Škálovatelnost je jeden z hlavních důvodů, proč vůbec SDN dělat. S přibívajícím aktivními prvky sítě se stává jejich správa a konfigurace stále více neefektivní úlohou administrátora, zvlástě když škálovatelnost nemusí nutně být jen přidání nových zařízení. Dalším příkladem škálování muže být přidání nových technologií, a konfigurace se musí promítnou do všech nebo podmnožiny zařízení v síti. Samozřemě se dá argumentovat, že úpravy provede skrypt, ale jelikož v síti se nachází zařízení různých výrobců, tak zjišťovat, jaký skrypt použít po příhlášení přes SSH/konzoli, se stává časově náročnou úlohou. Právě automatizace v SDN společně s automatickým zřizováním zařízení v síti v technologii SD-WAN jako PNP a ZTP spolu s ostatními zlepšeními, která přináší SDN, umožnňují nasazení nových sítí a jejich řízení být lehce škálovatelnými.

Redundance je jeden ze základních pilířů nejen v síťových technologií. V dnešní době se očekává, že počítačové sítě fungují 24 hodin denně bez přestání a výpadek jen části sítě bez redundance spojení může v dnešním světě vést až k milionovým škodám. Jelikož i tradiční sítě sítě se skládají z mnoha prvků v síti, tak každý tento prvek se přináší několik bodů, kde může síť zkolabovat, například selhání prvku nebo jednoho z síťového rozhraní. V klasických sítí může být redundance docílena řešením jako japříklad Host Standby Router Protocol (HSRP), agregací spojů (LACP), BGP a další. Potřeba mít redundantní spoje napříč vícero zařízeními je také způsob dosáhnutí zabezpečení vůči chybám. V případě použití technologie SD-WAN sice stále musíme stále použít hardwarové metody pro redundanci sítě, ale tradiční způsob připojení sítě WAN pomocí BGP může být zjednodušeno použitím řešení SD-WAN. SD-WAN navíc jednoduše řeší rozdělování zátěže napříč všemi přístupy do WAN. Jinými slovy, využití vytváření dynamických tunelů napříč koncovými body pomocí SD-WAN může naše síť pro zařízení z WAN vypadat a chovat se jinak než jak je ve skutečnosti vytvořena.[1]

Chyby v nastavení může síťový administrátor udělat téměř kdykoliv. Stačí když v tradičních sítích administrátor častokrát píše jeden a ten samý příkaz na spousty zařízení zvlášť, pak je téměř nevyhnutelné, že administrátor se přehlídne a vytvoří chybu, která se pak velmidlouho hledá, protože ji na první pohled adminstrátor nevidí. Takovéto chyby zpusobené lidským faktorem pak firmu stojí velké peníze. V některých případech, například na Cisco zařízeních, může být rozdíl mezi očekávanou funkcí příkazu pouze jedno klíčové slovo a dokumentace příkazu na oficialních stránkách výrobce také nemusí být nutně dobře napsaná nebo přehledná. Tyto faktory pak vedou k nechtěným výpadkům sítě a případně služeb na síti. Právě zde SDN

[1] Logická vs fyzická topologie sítě.

excelejule v tom, že kontrolér umožňuje kontrolu syntaxe, chyb a navíc disponuje možností zvrátit změny na konfigurovaném zařízení bez výpadku na síti. Tato vlastnost bude dále podrobněji popsána.

Zde představené problémy tradičních sítí by měly být řešitelné pomocí softwarově definovaných sítí, buďto od proprietárních řešení jako například Cisco, tak řešení ze světa FOSS jako je například OpenDaylight, který implementuje komunikaci se zařízeními pomocí protokolů jako je NETCONF. Jakékoliv zařízení, které umí být nastaveno pomocí NETCONF protokolu, by v teorii mělo by být schopno být součástí SD-WAN, SD-Access je stále příliš vázané na proprietární řešení výrobců zařízení.

Kapitola 2
Volba virtuálních prostředí

Tato kapitola se zaměřuji na volbu virtuálního prostředí pro simulaci SDN. Zde budou představeny dvahlavní virtualizační nástroje pro použití nasazení virtuálního prostředí a to EVE-NG a GNS3, k nimž budou zmíněny alternativy, které vybrány nebyly. Cílem této kapitoly je najít, čeho všeho jsou virtuální prostředí schopny, co všechno se dá virtualizovat aniž by musela být nutnost pořizovat drahá SDN zařízení.

2.1 Virtualizace síťových zařízení

Cíl této kapitoly je seznámit se způsoby vrtualizace síťových zařízení v dnešní době a představit možnosti nasazení různých implementací SDN. Simulátory sítí jsou v dnešní době jen obyčejné počítače, které musí být schopny virtualizovat různá zařízení. Těmto počítačům pak v praxi říkáme hypervizor, na rozdíl však od hrubého hypervizoru, zde nám virualizační nástroje nabídnou přívětivé grafické rozhraní uspůsobenou pro přehlednou visualizaci sítí a aktivních prvcích v nich. Zde opět zmíním stroj Cisco ESXi, který v této práci bude hlavní hypervizor nasazených virtuálních strojů.

2.2 Simulátory sítí

V dnešní době si společnosti už taká vytvářejí vlastní simulovaná prostředí, ať už je to Cisco Packet Tracer, Cisco VIRL či Enterprise Network Simulator Platform (eNSP) od Huawei, tak zde se především zaměřím na nekomerční způsoby simulace počítačových sítí. Nejznámější, zdarma dostupný, software na simulaci počítačových sítí jsou platformy Graphical Network Simulator-3 (GNS3) a Emulated Virtual Environment - Next Generation (EVE-NG). Pro účely diplomové práce budou tyto dvě simulační platformy použity pro testování SDN a bude popsáno nasazení simulovaného hardwaru na virutalizační platformy.

Ačkoliv tyto platfomy plně virtualizují reálný hardware pomocí komerčně používáných nástrojů pro Kernel Virtual Machine (KVM) jako je QEMU, i přesto se doporučuje tyto nástroje používat pouze ke studijmín a testovacím účelům, a pro účely nasazení pak použít nástroje jako je OpenStack či jiných komerčních nástrojů.

Na rozdíl od vlastnění fyzických zařízení, která mohou stát až nekolik desítek tisíc korun, virtualizace těchto zařízení nám umožňuje, že tato zařízení mají vždy nejaktuálnější zabezpečení v oblasti hardwarové bezpečnosti, tak výkonu zařízení či transparentní redundanci daného zařizení. Navíc díky vizualizaci sítě ve virtuálním prostředí nám umožňuje lépější nahlédnou do problematiky počítačových sítí, které jsou pak tvořený reálným hardwarem.

2.2.1 Emulated virtual environment (EVE-NG)

Emulated virtual environment - next generation je síťový virtualizační nástroj, který nepoužívá softwarového klienta k vytváření laboratoří. Celé grafické rozhraní, které EVE-NG nabízí je totiž zpřístupněno přes webové rozhraní, hostovaném pomocí interního HTTP/HTTPS[1] serveru nacházejícím se na EVE-NG hypervizoru. Některé funcionality EVE-NG jsou však, pro správnou funkčnost, nutné doinstalovat jako například software pro sledování síťové komunikace WireShark čí telnet/SSH klienta jako je například PUTTY. EVE-NG je nabízeno ve variantách PRO a Community edition, zde byla použita varianta PRO, ale veškeré funkcionality nutné k replikaci zde vytvořených laboratoří, jsou dostupné i ve zdarma dostupné komunitní edici.

EVE-NG je momentálně dostupná ve variantách formátu obrazu virtuálního stroje OVF a instalačního ISO obrazu. EVE-NG umožňuje a doporučuje instalovat virtualizační nástoj přímo na reálný hardware, protože platforma EVE-NG je také hypervizor, který simuluje reálný hardware. Toto doporučení spočívá v tom, že pokud instalujeme EVE-NG jako virtuální stroj nad jiným hypervizorem, tak se v koncovém výsledku dopouštíme takzvané vnořené virtualizace, který má negativní dopad na celkový chod simulovaného hardwaru. Přesto však je toto řešení pro naše účely nevyhnutelné a proto EVE-NG nabízí pro snadné nasazení obraz virtuálního stroje ve formátu Open Virtual Format (OVF). Jak bylo zmíněno dříve, tak veškeré simulované prostředí jsou nasazeny na Cisco ESXi stroj, tak zde byla zvolena varianta OVF.

Jeden z hlavních vlastnost, které EVE-NG nabízí, je široká podpora různých zařízení, přes kontainerizované aplikace pomocí Docker, tak zařízení používající složitější emulaci pomocí dynamips. Bohužel kvůli licenčním právům, EVE-NG nedodává obrazy simulovaných zařízení, pokud nejsou pod nějakou otevřenou licencí, a tak je uživatel odkázán sám na sebe, aby nějakým způsobem získal nutné obrazy chtěných zařízení pro simulaci. Z důvodů rostoucí popularity EVE-NG, škála podporovaných zařízení roste, kdežto v prostředí GNS3 spousta předpřipravených šablon přestává fungovat.

2.2.2 Graphical Network Simulator (GNS3)

Graphical Network Simulator (GNS3) je software navržen pro simulaci aktivních prvů sítě a jejich připadnou topologii zapojení. Na rozdíl od EVE-NG, GNS3 se skládá ze serveru, kde se emulují síťové prvky a jejich zapotjo, a klienta, který se připojuje k serveru a poskytuje uživateli grafické prostředí pro návrh simulované sítě. Jednotlivé obrazy síťových prvků jsou spouštěny pod emulátorem MIPS procesorové architektury, čili veškeré funkcionality reálných přepínačů a směrovačů jsou zde dostupny, a tudíž tvorba komplexnějších laboratoří je zde možna na jednom zařízení, místo použití zařízení více.

Klient GNS3 podporuje platformy Windows OS, Linux and MacOS. Pro server, pokud není lokální, tak podobně jako u EVE-NG, lze virtualizovat či nasadit na reálný stroj. Tento způsob nasazení virtuálního prostředí GNS3, kdy klient je na koncovém zařízení a veškerá zátěž se přesouvá na jeden nebo více serverů, je doporučen. Podbně jako EVE-NG, některé utility jsou potřeba doinstalovat pro použití všech funkcionalit virtuálního prostředí GNS3 jako je například už předem zmíněný nástroj pro sledování síťové komunikace WireShark, narozdíl však od EVE-NG, GNS3 nabízí instalaci potřebných nástrojů během instalace klienta a není nutno je dohledávat ručně.

▶ **Příklad 2.1.** Příkladem optimálního nasazení a používání GNS3 je instalace GNS3 serveru na výkonném stroji se spoustou volných výpočetních prostředků nikoliv jako operační systém, nýbrž jako virtuální stroj ve Virtual Boxu, VMWare či přímo pomocí QEMU. Grafické prostředí pak může uživatel naistalovat na jakékoliv zařízení, jelikož veškerou zátěž emulace síťových zařízení pak vykonává server a nikoliv klient. Bohužel tento scénář není vždy uživateli dostupný.

GNS3 je na rozdíl od EVE-NG je open-source platforma a je zdarma dostupna k použití, ale stejně jako v případě EVE-NG, GNS3 nedodává obrazy síťových zařízení a uživatel je tudíž

[1] Záleží na verzi či nastavení

Tabulka 2.1 Srovnání EVE-NG a GNS3 [4]

Parametr	GNS3	EVE-NG
Původ	Open-source, zdarma client/server rozhraní určené pro virtualizaci a emulaci sítí, napsané v Pythonu a podporuje Cisco	Bez klientový, virtuální simulátor sítí, který byl vytvořen pro jednotlivce a malé podniky. Nabízí placenou a zdarma edici
Přístup k softwarovým obrazům	Přístupný přes výrobce nebo školu	Přístupný přes výrobce nebo školu
Specializovaný síťový simulátor	Vyžaduje instalace aplikací pro sledování síťových zařízení	Záleží podle edice

odkázán sám na sebe. V následných kapitolách bude zmíněno, jaké rozdíly a případné problemy byly v nasazení SD-WAN technologií v prostředí GNS3 oproti EVE-NG.

2.3 Použitý hypervizor Cisco ESXi

Samozřejmě simulační prostředí nejsou jediným způsobem, jakým lze dosáhnout simulování síťových zařízení a topologií, i když mají připravenou celou sad nástrojů k jednoduché emulaci zařízení, které každý uživatel ocení, tak pořád můžeme simulovat síť přimo na hypervizoru.

Možnost použít ESXi hypervizor přímo na nasazení jednotlivých virtuálních síťových prvků a následně je propojit pomocí vestavěné funkcionality viruálních přepínačů, nám totiž nabízí mnohem stabilnější implementaci, použitelnou do produkčního prostředí, i když ne nutně jednoduchou na nastavení či přehlednost jakou nám simulační prostředí nabízí. Protože se jedná o komerční zařízení, celkový počet výpočetních prostředků alokovatelné na virtuální stroj je omezen koupenou licencí, v našem případě používáme zdarma licenci, která omezuje počet použitelných virtuálních jader na virtuální stroj na polovinu dostupných vláken procesoru a to konkrétně na 8 vCPU. Samozřejmě v produkčních prostředí se snažíme, aby ne všechna zařízení byla virtualizována na jednom konkrétním hardwaru, ať to z hlediska finančního, že nemáme možnost platit za drahou ESXi licenci, či nechceme abychom uměle vytvářeli jeden bod potenciálního selhání. Bohužel na rozdíl od GNS3, EVE-NG vyžaduje profesionální licenc pokud by mělo rozprostřeno do clusteru.

▶ Poznámka 2.2. Samozřejmě ESXi není jediný hypervizor na trhu, který je tohoto schopen. Virtualizaci můžeme dokonce lépe než takto realizovat na svém hardwaru pokud máme dostatečné vědomosti, pak můžeme utilizovat paravirtualizaci KVM a pro velké virtuální stroje utilizovat Hugepages pro zrychlení přístupu do paměti RAM. Tyto možnosti mnohdy nemáme v komerčních zařízení a pokud jsou k dispozici, tak často jejich funkcionalitu a nasazení nedokážem ovlivnit. V této práci používám ESXi, protože mi byla dodána vedoucím práce, jelikož můj domácí server indisponuje 128GiB RAM

Veškeré simulace budou nasazeny na oba zmíněná simulační prostředí. Tím chci zamezit možnosti, že analýza toku dat v sítích bude ovlivněna chodem virtuálního prostředí, proto popis jednotlvých laboratoří bude u jednoho simulátoru popsán více do podrobna než druhý, jelikož koncová implementace by měla zůstat zachována.

2.4 Cisco SDN

Prvním představeným řešením softwarově-definovaný sítě je řešením od Cisca. Společnost Cisco nabízí SDN ve třech vydání, SD-WAN, SD-Access a SD-Branch, kde každý z nich přinášejí koncept SDN do různých odvětví sítí, kde SD-WAN se více zaměřuje na WAN, SD-Access se zaměřuje na koncová zařízení sítí a SD-Branch se zaměřuje na virtualizaci síťových funkcí.

2.4.1 SD-WAN

Před tím než začneme simulovat technologie SD-WAN, je potřeba zajistit, aby veškeré nutné komponenty potřebné k implementaci SD-WAN byly virtualizovatelné. V této sekci se proto zaměřuji na možnosti virtualizace Cisco SD-WAN a zbytek sekce zaměřím na nasazení do virtualáních prostředí, s tím že zde popíši funkcionality implementace.

Řešení SD-WAN je založeno na strojích, ať virtuální či fyzická, takzvané kontroléry, které řídí jednotlivá SDN zařízení, spadající do řídící vrstvy SD-WAN a WAN-Edge směrovačích. Cisco nabízí obrazy Viptela SD-WAN ve formátu QCOW2, čili simulátory musí být schopny s tímto formátem pracovat, nebo být konvertibilní do jiného formátu aniž by se poškodila funkcionalita obrazů. Naneštěstí Cisco nabízí obrazy ve vícerofromátů takže i když je zrovna formát QCOW2 potřeba pro chod v prostředí EVE-NG, tak nejsme odkázáni pouze na jeden formát virtuálního disku.

Cisco tyto SD-WAN zařízení nabízí ve dvou skupin verzí a to:

Kontroléry/vEdge routery které jsou založeny na ViptelaOS

cEdge jsou Cisco IOS-XE zařízení.

Bohužel je pořeba při nasazování těchto zařízení, aby zařízení měly stejsou verzi softwaru, v rámci skupin.

▶ **Příklad 2.3.** Jako příklad z praxe vEdge, vSmart, vBond a vManage zařízení používají verzi X a cEdge zařízení používa verzi Y, kde verze X a Y jsou od sebe různé, což je příklad korektního nasazení.

Přestože použití různých verzí nemusí nutně znefunkčnit nasazení SD-WAN, zvyšujeme tím riziko, že chyby spjaté práve s rozdílnými verzemi zařízení v rámci skupiny, mohou být velmi těžké na odhalení. Stejně tak není doporučováno, aby byly používány cisco cEdge zařízení společně s vEdge zařízeními, kvůli tomu, že jejich správa a funkce se drobně liší a rozdílný operační systém zařízení neumožňuje automatickou správu zařízení být jednoduchou.

▶ Poznámka 2.4. Cisco umožňuje si zjistit doporučované verze zaříení na jejich oficiálních stránkách [5]

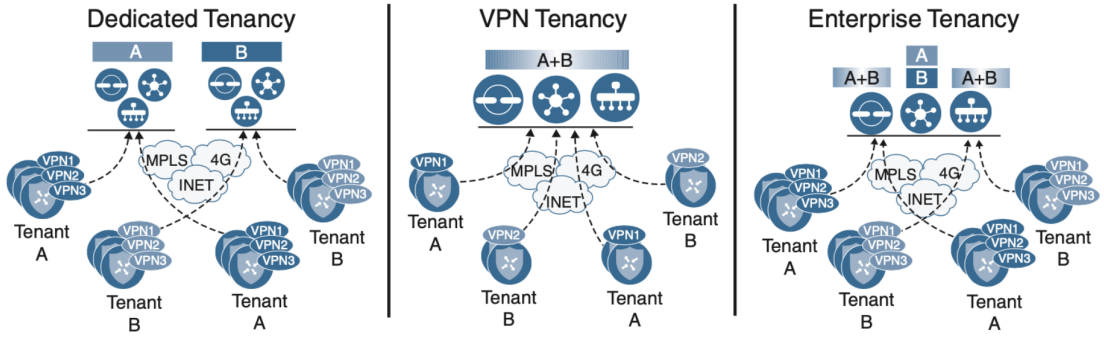
vBond kontrolér, vEdge směrovač a Zero Touch Provisioning (ZTP) server používají stejný obraz softwaru. Hlavní rozdíl mezi jednotlivými zařízeními je jejich nastavení, kdežto vSmart a vManage používají své vlastní obrazy. To platí i v případě použití Cisco IOS-XE zařízení.

Když se snažíme na Internetu hledat zařízení od Cisca z oblasti SD-WAN, jediná zařízení, která můžeme najít, jsou pouze Edge zařízení. Hlavním důvodem, proč tomu tak je, je, že vBond, vManage a vSmart jsou navrženy tak, aby byly virtualizovány, což je ideální případ pro naši práci, jelikož témeř veškerá použitá zařízeni jsou součástí virtualizovaného simulovaného prostředí. V praxi se poté setkáváme, že kontroléry jsou buď to lokálně virtualizovány na firemních serverech, či nasazeny v cloudu. Dneska existuje spousta způsobů, jak nasadit SD-WAN do cloudu, například AWS a Azure cloud podporují tyto operace, ale samozřejmě Cisco doporučuje Cisco Cloud.

▶ **Příklad 2.5.** Když někdo chce používat SD-WAN ve své sítí, musí si buď vytvořit vlastní server ze svých komponentů a na ně následně nasadit vManage, vSmart a vBond, nebo si zaplatit cloud a potřebne výpočetní prostředky na něm. V případě použití lokálního hardwaru si poté musí uživatel spravovat zařízení sám nebo si pořídit správu od Cisca, která do jisté míri přesune správu zařízení do jejich cloudu.

Cisco SD-WAN podporuje takzvaný Multi-Tenancy, což spočívá v tom že jeden vManage zvládne řídit vícero zákazníků zvaných Tenant, tito zákazníci pak zdílí prostředky sítě. Následující obrázek 2.1 ukazuje, jakým způsobem Cisco Cloud nabízí službu Multi-Tenancy.

Zde jsou stručné popisy jednotlivých způsobů použití SD-WAN v cloudu dle 2.1



Obrázek 2.1 Způsoby nasazení Cisco SD-WAN Multi-Tenancy [6]

Dedicated Tenancy je případ, kdy všechny SD-WAN kontroléry jsou řízeny jedním zákazníkem. Toto je nejčastější způsob nasazení v cloudovém nasazení.

VPN Tenancy je případ, kdy pouze datová vstva ve VPN topologii je segmentována. Užvatelé s právem čtení si zde mohou monitorovat provoz ve vManage

Enterprise Tenancy je převážně cílený na podnikové cloudové nasazení SD-WAN. V tomto případě vManage a vSmart jsou sdíleny mezi vícero zákazníků, kde finální výsledek může být zcela znepřehledněn koncovému uživateli. Každému zákazníku je zde přiřazen jedno vSmart zařízení.

▶ Poznámka 2.6. Pro potřeby SD-WAN jsou pojmi VPN a VRF zaměnitelné a popisují stejný problém.

Implementační náležitosti a hardwarové požadavky pro nasazení SD-WAN ve simulačních prostředí budou probrány v následných kapitolách zaměěřující se na nasazení. Zde zmíním seznam funkcí, které se dají simulovat, jelikož SD-WAN se dá zcela simulovat, spolu se všemi jejími funkcemi. Laboratoře pak zahrnují od celý postup jak nasadit a zprovoznit Cisco SD-WAN, spolu se základními SD-WAN funkcemi.

- NAT traversal
- VPN segmentace v datové vrstvě
- Nasazení nadbytečných kontrolérů
- Statický a připojitelný distribuování směrů pomocí Overlay Manageme Protocol (OMP)
- CLI šablony
- Šablony vlastností SD-WAN
- Manuální nastavování jednotlivých komponent
- Zero Touch Provisioning (ZTP)
- Stavění datové vrstvy pomocí IPSec tunelů
- Správa certifikátů
- Správa konfigurací příslušných aktivních prvků
- Správa všech SD-WAN kontrolérů a vEdge směrovačů

- Možnost vytváření řídících tunelů Datagram Transport Layer Security (DTLS)

Nicméně, použití a vysvětlení všech možných funkcí SD-WAN by bylo nad rámec této diplomové práce a převážne textu by byly citace již publikovaných příruček firem jako je Cisco. Další práce mohou navázat tuto práci a přidat další, z zjevně nekončícího, listu funkcí SD-WAN.

2.4.2 SD-ACCESS

V první čati této podkapitoly jsem se zaměřil na popis Cisco SD-WAN, jaká zařízení jsou potřeba pro implementaci SD-WAN a která z nich se dají simulovat pomocí dostupných simulačních prostředí. Stejným postupem budu postupovat i v případě SD-Access, proto před tím než budu moci simulovat SD-Access je potřeba zjisti co musí SD-Access obsahovat a co z toho se dá simulovat.

DNA Center kontrolér je zodpovědný za celkový chod SD-Access a její automatizaci. Nově od minulého roka Cisco nabízí DNA-Center kontrolér jako virtuální stroj nasaditelný na ESXi. [7]

▶ Poznámka 2.7. Panují však zvěsti o DNA-Center volně se šířícím ISO obrazu kolující po Internetu, který by se dal rozeběhnout na hypervizoru za malou cenu.

Fusion router je směrovač zajišťující makrosegmentaci v SD-Access. Jakýkoliv hardware či virtuální zařízení schopno provozu mBGP s VRF propouštění směrů by mělo postačovat.

Síťové uzly jsou samozřejmostí každé sítě v podobě aktivních prvků a také nutností v simulaci SD-Access.

- Cisco SD-Access edge uzel
- Cisco SD-Access border uzel
- Cisco SD-Access uzel řídící vrstvy

Bohužel zde jde virtualizovat pouze zařízení řídící vrstvy a to v podobě směrovače Cisco CSR 1000v.

Cisco ISE je zařízení zajišťující autentikaci, autorizaci a správu účetnictvý (správa cen služeb a použití). Toto zařízení plně podporuje virtualizaci a v praxi je velmi často virtualizováno, ale tento systém je velmi náročný na hardwarové prostředky.

2.4.2.1 Požadavky pro simulaci Cisco SD-Access

Bohužel předchozí drobná analýza nutných prvků pro zprovoznění Cisco SD-Access prokázala být nerealizovatelnou čistě pomocí virtualizačních nástrojů.

Virtualizovatelné komponenty Virtualizace co nejvíce zařízení je jak z hlediska finanční, tak z hlediska dostupnosti lidem, je nejlepší variantou pro simulované prostředí vrámci laboratoří, nikoliv produkčního prostředí. Právě pro domácí prostředí bohatě stačí starý a nepotřebný, však funkční hardware. Pro účely této diplomové práce byl pro virtualizaci dodaný server Cisco R210-2121605W s VMWare ESXi operančím systémem hypervizoru.

▶ Poznámka 2.8. Rozhodně pro osobní použití je vhodné použít hardware z druhé ruky, který převážně bývá stejně dobrý jako produkční hardware, až na to, že jsou starší a často na okraji nebo po End of Life (EOL). Jen pro porovnání, cena nového zařízení se Cisco DNA-Center potřebný pro SD-Access, je nabízen od 80000 amerických dolarů, což je značně více, když bazarové zařízeni se dá pořídít okolo 800 dolarů.

Tabulka 2.2 Minimální hardwarové nároky na implemetaci SD-Access

Virtuální zařízení

Zařízení	Platforma	RAM	CPU	Velikost disku	Poznámky
ISE[8]	VM na ESXi nebo HW	16GiB	12	300GiB	
DNA-Center[9]	VM na ESXi nebo HW	256GiB	32	3TiB	
WLC[10]	Cisco Catalyst 9800-CL	8GiB	4	16GiB	Vyžaduje dedikovaný NIC
Fusion router	FTDv nebo Cisco 1921	8GiB	4	50GiB	
Uzly řídící vrstvy	Cisco CSR 1000v	4GiB	4	8GiB	

Virtuální zařízení

Zařízení	Platforma
Edge uzly	Cisco Catalyst 3850/3650/9300 série
Border uzly	Cisco Catalyst 3850 nebo Cisco ISR 1000/1100/4321 série

V době kdy tato práce byla napsána, Cisco DNA-Center je nabízeno jako virtuální stroj, proto se v dnešní době dá nasadit Cisco DNA-Center SD-Access.

Již předem zmiňované Cisco ISE je plně virtualizovatelné.

Řídící vrstvu SD-Access lze realizovat vrtualizací Cisco CSR 1000v směrovači, jež virtualizace dostupná.

Jelikož každá virtualizace snižuje jak nároky na hardware, tak případnou cenu, virtualizaci doporučuji dělat tam, kde je možna.

V případě Fusion routeru, jediná podmínka na něj je aby podporoval mBGP s VRF propouštění směrů. Dobrou variantou pro tento připad je Firepower Threat Defense virtual (FTDv) firewall. Pokud by nestačily systémově prostředky pro tento firewall, alternativa může být použití směrovače Cisco 1921.

V případě, že do naši SD-Access sítě budeme vměstnávat bezdrátová zařízení, Wireless Lan Controler (WLC) bude potřeba, buďto jako virtuální stroj nebo stroj reálný.

Nevirtualizovatelné komponenety Jak bylo zmíněno v mnoha místech sekce Cisco SDN tak ne všechny komponenty Cisco SD-Access jsou k dispozci k virtualizaci, naneštěstí spousta zařízení v podobě směrovačí a přepínaču se dá sehnat za vcelku přijatelnout cenu z druhé ruky. Největším problémem celé implementace SD-Access je právě DNA-Center, jen tato položka je samotná nejdražší ze všech ostatních komponent, která se nedá jednoduše virtualizovat.

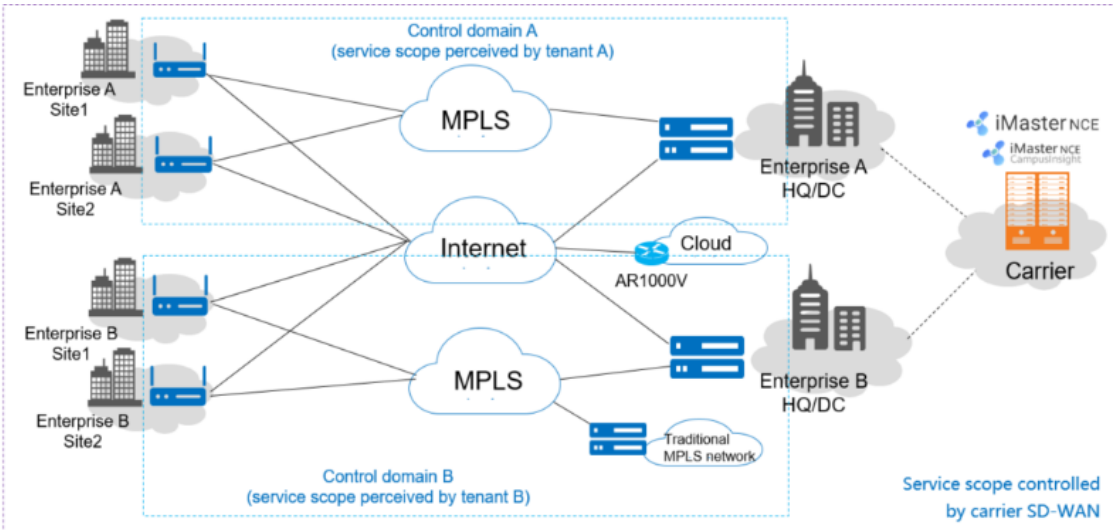
Edge uzly se dají řešit pomocí Cisco Catalyst přepínačů ze serie 3850 nebo 3650. Kdybychom potřebovaly implementovat bezdrát, tak Cisco Catalist 9300 je zařízení obsahující vestavěný WLC.

V případě border směrovačů již zmíněný Cisco Catalyst 3850, může být kromě jako edge uzel tak i jako border uzel. Dalším možností je použít Cisco Intergrated Serivce Router (ISR) směrovači ze serií 1000, 1100 nebo 4321. Jelikož funkcionalita mezi zde zmiňovanými zařízeními je velmi podobná, nejlevnější znich je adekvátní volbou.

V následující tabulce 2.2 se nachází jedna z možných konfigurací implementací. Zde se zaměřuji na co nejmenší cenu nákladů.

2.5 Huawei SDN

V této sekci se zaměřuji nad možností simulace řešní SDN od společnosti Huawei pro studijní účely. Na rozdíl od společnosti Cisco, Huawei nabízí tři způsoby řešení SDN a to SD-WAN a



Obrázek 2.2 Způsoby nasazení Huawei SD-WAN Multi-Tenancy [6]

CloudFabric/CloudCampus. Jedná se o připravená řešení pro různá odručí softwarově definovaných sítí.

2.5.1 SD-WAN

Stejně jako v případě Cisca, Huawei SD-WAN podporuje virtualizaci téměř všech svých zařízení. Tudíž i zde je potřeba zzajistit aby byly veškeré nutné komponenty potřebné k implementace SD-WAN byly virtualizovatelné a dostupné. V této sekci se proto zaměřuji na možnost virutalizace SD-WAN a zbytek sekce zaměřím na nasazení do virtuálních prostředí, s tím, že zde popíši funcionality implementace.

Řešení SD-WAN je založeno na strojích, hlavní řídící kontroléry, spadající do takzvané řídící vrstvy SD-WAN a CPE směrovače. Huawei nabízí obrazy kontrolérů iMaster NCE-Campus a iMaster NCE-WAN jako iso instalovatelné na hypervizor na ESXi. Bohužel, pro účely práce se nezdařilo sehnat obraz Huawei NCE. Zdařilo se pro účely práce pořídit obraz AR 1000v nutný pro realizaci řídící a síťové vrstvy Huawei SD-WAN. Huawei zařízení mohou být kategorizovány do těchto skupin.

Kontroléry a vCPE jsou zařízení iMaster NCE, iMaster NCE - CampusInsight, vCPE AR1000V a RR, které mohou být vCPE tvořeny

CPE jsou zařízení NetEngine AR8000, NetEngine AR6000 a AR600 směrovače, jako například NetEngine AR651 a NetEngine AR6280 a virtuální zařízení AR1000v

Huawei doporučuje, aby verze zařízeni byly stejné rámci verze SD-WAN, nebo alespoň na kompatibilní verzi s řešením SD-WAN dle tabulek.

Stejně jako v případě Cisco produktů, v případě rozdílných zařízení nemusí nutně vést k celkové nefunkčnosti řešení, tak opět rozte riziko vzniku těžko odhalitelné chyby, nemluvě o možné nekompatibilitě jednotlivých prvků řešení SD-WAN. Na rozdíl od Cisca, zde se můžou bezproblémově zaměňovat virtualní zařízení s fyzickýma s doporučením zajištění patřičné verze firmwaru.

▶ Poznámka 2.9. Předem než začnete se simulací Huawei SD-WAN, doporučuji zajistit verzi zařízení shodnou s danou verzí SD-WAN řešení podle jejich doporučení. [11]

Na obrázku 2.2 je graficky znázorněno, jak řeší Huawei SD-WAN problém správy více podniků jedním administrátorem, nebo také multi-tenancy. Na rozdál od Cisca, zde je pohledz zjednoduše do jednoho ohebného řešení.

AR1000v je vrtuální platforma ralizující koncové WAN-edge zařízení, v Huawei známo jako vCEP, který se dá použít také jako Route Reflector, realizující kontrolní/orchestrační vrstvu řěšení Huawei SD-WAN, proto k němu bude stačit jeden obraz virtuálního zařízení. Jediným rozdílem v nasazení bude totiž jen jejich konfigurace. iMaster NCE potřebují svůj vlastní obraz virtuálního stroje.

V případě, že bychom chtěli implementovat Huawei SD-WAN čistě na reálném hardwaru, všechny virtuální zařízení by se měly dát zprovoznit na architektuře x86_64.

Celková architektura Huawei SD-WAN se zkládá ze 3 částí. Těmi jsou síťová vrstva, řídící vrstva a kontrolní, kde ve vrstve řídící se používá jako hlavní kontrolér zažízení iMaster NCE. Toto zařízení může být realizováno buďto pomocí virtualizace on premise, nasadit ho v cloudu, či nasadit ho na reálném hardwaru. Huawei rozlišuje iMaster NCE-Campus a NCE-WAN jejich zakladni funkce jsou stejné.

V kontrolni vsrtve se nacházejí takzvané router reflektory, tyto reflektory jsou určeny pro předavani informací o směrech uložené v řídící vrstvě do síťové vrstvy. Tyto zarizeni opet mohou byt buďto fyzická zařizeni nebo virtualní. Pro účely této práce se podařilo obstarat vCPE zařízení AR1000v.

Zde jsou vyjmenovány hlavní funcionality Huawei SD-WAN, které, pokud by se podařilo obstarat iMaster NCE, by se daly implementovat do síťových laboratoří.

- Správa a řízení MSP
- Zero Touch Provisioning (ZTP)
- Síťování
- Správa tunelů
- Správa konektivit
- Inteligentní úpravy provozu datové komunikace
- Různé služby (QoS, ACL a podobně)
- Monitorování
- Optimizace WAN
- Cloudová konektivita
- Nativní IPv6
- Širokosáhlé monitorování
- Otevřenost

Nicméně, použití a vysvětlení všech možných funkcí Huawei SD-WAN by bylo nad rámec této diplomové práce, i v případě kdyby se podařilo sehnat iMaster NCE, převážne textu by byly citace již publikovaných příruček firem jako je Huawei. Další práce mohou navázat na tuto práci a přidat další, se zjevně nekončícího, listu funkcí SD-WAN.

2.5.2 CloudCampus

V první čati této podkapitoly jsem se zaměřil na popis Huawei SD-WAN, jaká zařízení jsou potřeba pro implementaci SD-WAN a která z nich se dají simulovat pomocí dostupných simulačních prostředí, pokud máme k dispozici jejich obrazy. Stejným postupem budu postupovat i v případě SD-Access, proto před tím než budu moci simulovat SD-Access je potřeba zjisti co musí SD-Access obsahovat a co z toho se dá simulovat. Huawei SD-Access se jmenuje CloudCampus a je to jejich řešení na otázku konkurenčního Cisco SD-Access, řešeným pomocí Cisco DNA. Proto pro tuto část textu budu vycházet z [12] abychom měli přímé srovnání s Cisco SD-Access.

iMaster NCE kontrolér je zodpovědný za celkový chod jak Huawei SD-WAN tak Huawei CloudCampus a její automatizaci. Toto zařízení Huawei nabízí iMaster NCE kontrolér jako virtuální stroj nasaditelný na do cloudů. [12]

Huawei CIS je software řešení zajišťující autentikaci, autorizaci a správu účetnictvý (správa cen služeb a použití). Toto zařízení plně podporuje virtualizaci a v praxi je velmi často virtualizováno, ale tento systém je velmi náročný na hardwarové prostředky.

2.5.2.1 Požadavky pro simulaci Huawei ClousCampus

Bohužel předchozí drobná analýza nutných prvků pro zprovoznění Huawei CloudCampus prokázala být nerealizovatelnou čistě pomocí virtualizačních nástrojů, z důvodů nedostupnosti iMaster NCE.

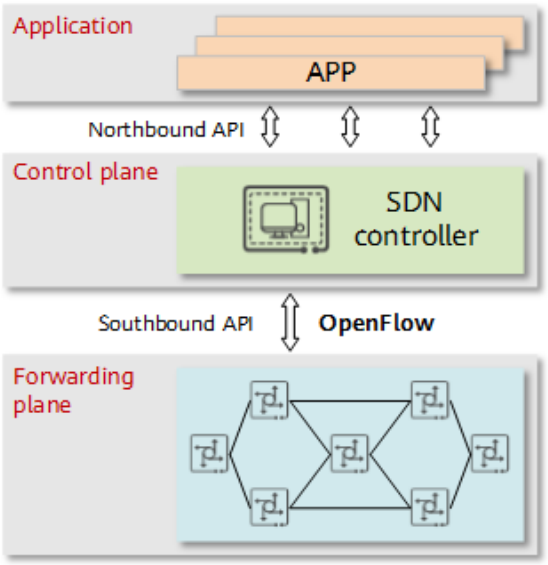
2.6 Mikrotik/FOSS SDN

Poslední představenou řešením softwarově definovaných sítí je platforma Mikrotik a jejich integrace open-sourcového řešení softwarově definovaných sítí pomocí protokolu OpenFlow jež vyvíjí Linux Foundation. Proto tato sekce se zaměří na kontroléry implementující OpenFlow a zařízení schopné používat OpenFlow.

2.6.1 OpenFlow

OpenFlow [13] je, dá se říci, standardní protokol, který umožňuje OpenFlow kontrolérům komunikovat a řídit OpenFlow přepínače a routery pod správou OpenFlow kontroléru [14]. OpenFlow je takto podporován mnohou výrocnů síťových pepínačů a směrovačů[15][16]. Kromě hardwaru, také existují čistě softwarové přepínače, jako je například Open vSwitch[17] nebo virtuální Mikrotik RouterOS. Existují též vývojové platformy OpenFlow, jako jsou NOX[18], POX[19], Trema[20] a Ryu[21], které umožňují vytvářet kontroléry. Další variantami přímo kontrolérů jsou, OpenDaylight, ONOS, Floodlight, Faucet či Beacon. Některé platformy kontrolérů dodávají vlastní rozhraní pro připojení s více úrovňovými komponentami, které se nazývá Northbound API. Avšak implementace Northbound API jsou na každé platformě OpenFlow kontroléru odlišné a nestandartizované.

Na obrázku 2.3 lze vidět jaká je architektura OpenFlow sítě. Celkové chování OpenFlow sítě je určeno spoluprácí jednotlivých dílčích kontrolérů a přepínačů implementující OpenFlow standart. V okamžik, kdy přijde nový tok do OpenFlow přepínače, přepínač se podívá do své tabulky toků, která obsahuje záznamy o tocích složené z porovnávacích podmínek a k nim odpovídajícími akcemi. Pokud tok souhlasí s odpovídající podmínkou záznamu toku v tabulce toků, tak se přepínač zachová podle příslušné akce odpovídající záznamu toku. Pokud se však nepodaří najít odpovídající záznam toku z tabulky toků, tak přepínač ohlásí tuto událost o přišlém toku OpenFlow kontroléru zasláním zprávy typu OFPPacketIn a zažádá o příslušnou akci kontrolér. Následně kontrolér rozhodne, jak by se mělo se s novým tokem naložit a pošle odpověď v paketu



Obrázek 2.3 Architektura OpenFlow SDN. [22]

typu OFPFlowMod, včetně dotyčného toku. Přepínač si pak nově přidá tok do tabulky toků a vykoná akci nově získanou z kontroléru. Když pak v budoucnu přijde tento tok znovu, tak přepínač už ví jak na něj a přímo jej zpracuje. Tímto způsobem tak dokáže OpenFlow kontrolér přímo ovlivňovat zpracování paketů napříč přepínači ve spravovaé sítí.

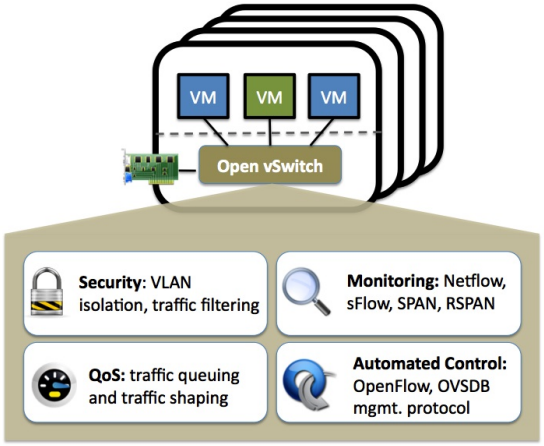
V dalších kapitolá se zaměřím na podrobnější popsání fungování OpenFlow, jelikož veškerá architektura se skládá z jednotlivých kontrolérů a přepínačů/směrovačů. Zde neexistují specifická zařízení implementující jednotlivé aspekty jako je řešení směrování, autorizace, authentikace a podobně. Veškeré funcionality jsou řešeny softwarově přímo na kontroléru nebo taženy od dedikovaných výpočetních serverů. Toto je výrazná změna od řešení, jako je Cisco SD-WAN a Huawei SD-WAN, který mají dedikované síťové prvky pro specifické operace. Tím mohou proprietární řešení lepších výsledků než čistě softwarová záležitost.

2.7 FOSS SDN

Tento svět je rozsáhlý, tvořen malými projekty jako Nante-WAN[23] založeným na Free Range Routing[24], přes standarty jako je například OpenFlow. Protože standardů je dost, a jak bývá zvykem tak svět open-source je často doprovázen nestandardními implementacemi podle potřeb konkrétního programátora či skupiny programátorů. Proto v následujících podsekcí představím několik OpenFlow kontrolérů, na některých z nich pak budou implementovány jednotlivé laboratoře.

2.7.1 Zařízení

Následující podkapitoly popisují několi známích implementací přepínačů a smerovačů, ať virtuálních či fyzických, s otevřených zdrojovým kódem a volnou dostupností na Internetu. Tyto zařízení mohou sloužit v SD-WAN řešení, jelikož implementují potřebné funcionality pro správnou funkci SD-WAN. Zařízení implemntující OpenFlow jsou zatím pouze zařízení OpenWRT a Open vSwitch. Protože VyOS je dedikován pro pouze směrování nežli přepínání, tak se momentálně uvažuje o implementaci RouteFlow[25], projekt snažící se implementovat směrování přes OpenFlow komunikaci na zařízeních s podporou OpenFlow.



Obrázek 2.4 Open vSwitch architektura. [26]

2.7.1.1 Open vSwitch

Open vSwitch je vícevrstvý softwarový přepínač s otevřeným zdrojovýk kódem pod licencí Apache 2. Open vSwitch se snaží implementovat otevřený přepínač s produkční kvalitou, která by podporovala standardní správcovské rozhraní a tím umožnila přesunout funkce obstarávající směrování a přepínaní do programovatelných a řídících rozšíření.2.4

Open vSwitch je vhodným pro fungování jako virtuální přepínač ve virtuálních prostředích. Kromě poskytování standardních ovládacích a správcoských rozhraní do virtuální síťové vrsty, byl open vSwitch navržen tak, aby podporoval nasazení napříč vícero fyzických serverů. Open vSwitch také podporuje několik linuxových virtualizačních technologií, včetně KVM a VirtualBoxu.

Většina kódu je napsána v platformně nezávislém jazyce C a díky tomu je snadno přenositelný na jiné platformy operačních systémů. Současná verze Open vSwitch podporuje následující funkce:

- Standardní 802.1Q VLAN model s trunk a access porty
- NIC bonding s nebo bez LACP na upstream přepínači
- NetFlow, sFlow(R) a zrcadlení pro zvýšenou viditelnost
- Konfigurace QoS (Quality of Service) a policing
- Tunelování pomocí protokolů Geneve, GRE, VXLAN, STT, ERSPAN, GTP-U, SRv6, Bareudp a LISP
- Správa konektivity chyby podle standardu 802.1ag
- OpenFlow 1.0 plus mnoho rozšíření
- Transakční konfigurační databáze s vazbami pro jazyky C a Python
- Vysokovýkonné přeposílání pomocí modulu jádra Linuxu

Open vSwitch může také fungovat zcela v userspace bez potřeby modulu jádra. Tato implementace v userspace by měla přinést snazší možnost přenášení mezi OS než přepínač, který by byl závislý jádře jako modul. OVS v userspace může přistupovat k zařízením Linuxu nebo DPDK. Je třeba poznamenat, že Open vSwitc s zařízeními non-DPDK je považován za experimentální a má znatelný vliv na výkon. [26]

2.7.1.2 VyOS

VyOS je směrovací operační systém s otevřeným kódem, zdarma dostupným všem, s placenou licencí za podporu a LTS vydání. VyOS přímo konkuruje ostatním komerčně dostupným řešením od známích dodavatelů síťových řešení. Jelikož VyOS běží zatím pouze na platformách x86_64, může být použit jako směrovač a firewall pro cloudová řešení.[27][28]

VyOS vznikl po tom co Brocade Communications přestala vyvíjet Vyatta Core edici Vyatta Routing software. Tak malá skupina nadšenců v 2013 převzala poslední komunitní edici, a začli pracovat na vytvoření otevřené větve jako náhrada za tehdá končící Vyatta Core.

Vlastnosti VyOS: [29]

- BGP (IPv4 and IPv6), OSPF (v2 and v3), RIP and RIPng, směrování na bázi politik.
- IPsec, VTI, VXLAN, L2TPv3, L2TP/IPsec a PPTP servry, tunelová rozhraní (GRE, IPIP, SIT), OpenVPN v klientském, servrovém nebo síť-siť módy, WireGuard.
- Stavový firewall, zónový firewall, všechny typy zdrojového a cílového překládání adres NAT (1:1, 1:N, N:N).
- DHCP a DHCPv6 server a relay, IPv6 RA, DNS přeposílání, TFTP server, webová proxy, shlukovač PPPoE přístupů, NetFlow/sFlow senzor, QoS.
- VRRP pro IPv4 a IPv6, schopnost spuštět vlastních kontrol funkčnosti a transitivních skriptů. ECMP, Stavové vyvažování zátěže.
- Vestavěné verzování.

2.7.1.3 OpenWrt

OpenWrt je vysoce rozšiřitelná Linuxová distribuce určená pro vestavěná zařízení (typicky bezdrátové směrovače). Na rozdíl od mnoha jiných distribucí pro tyto typy zařízení, OpenWrt je vybudován od základů tak, aby byl plnohodnotným a snadno upravitelným operačním systémem pro jakýkoliv směrovač. V praxi to pak znamená, že můžou být všechny potřebné funkce bez zbytečných funkcí od výrobce. Toho bylo docíleno díky aktualizovatelnosti Linuxového jádru, které je aktuálnější než u zařízení, jemž již došla životnost a podpora výrobce.

Namísto pokusů vytvořit jediný stálý firmware, OpenWrt poskytuje plně zapisovatelný filesystém s možností správy balíčků. Tím uživatele zbavuje omezení softwarové výbavy konkrétního zařízení včetně omezené konfigurovatelnosti poskytované výrobcem. Dále OpenWrt umožňuje uživateli instalovat balíčky k rozšíření základního firmwaru vestavěného zařízení o libovolné funkce. Pro vývojáře poskytuje OpenWrt platformu pro budování aplikací, aniž by museli vytvářet vlastní obraz firmwaru zařízení a řešiť hardwarovou kompatibitu. Pro uživatele to znamená úplné se zbavení všech omezení, které originální firmwary od výrobce často vytváří. To umožňuje použití vestavěného zařízení způsoby, na které výrobce v daný okamžik neměl v plánu zahrnout. [30]

2.7.2 Kontroléry

V této podsekci se zaměřím na přestavení několika známích OpenFlow kontrolérů, které se používají různě ve světě. V následujících kapitolách budou probrány nasazení některých z nich a jakým způsobem komunikují s zařízeními schopny provozovat OpenFlow. Stejně jako v předchozí podsekci zde představím jednotlivé kontroléry velmi stručně s tím, že zmíním, jaké vlastnosti daný kontrolér má, na jakém jazyku je postaven a jaké jsou způsoby nasazení včeně minimálních nároků na hardware.

2.7.2.1 OpenDaylight

Kontrolér OpenDaylight je software na bázi Java Virtual Machine (JVM) a tudíž jej lze spustit z libovolného operačního systému a hardwaru jen tehdy, pokud podporují Javu. Kontrolér implementuje koncept softwarově definovaných sítí (SDN) a její realizaci používá následující nástroje:

Maven: OpenDaylight používá Maven pro snadnější automatizaci sestavení. Maven používá pom.xml (Project Object Model) ke skriptování závislostí mezi svazky a také k popisu toho, jaké svazky se mají načíst a spustit.

OSGi: Tento framework je back-endem systému OpenDaylight, protože umožňuje dynamické načítání svazků a balíčků JAR a propojování svazků pro výměnu informací.

JAVA: Rozhraní Java se používají pro naslouchání událostem, specifikace a vytváření vzorů. Jedná se o hlavní způsob, jakým konkrétní svazky implementují funkce zpětného volání pro události a také pro označení povědomí o konkrétním stavu.

API REST: Rozhraní jsou Northbound API jako je například manažer topologie, host trackery, programátory toků, statické routování a podobně. Jedná se o severně orientovaná rozhraní API, jako je správce topologie, sledování hostitelů, programátor toků, statické směrování atd.

Kontrolér vystavuje otevřený Nothbound API, které dále vyžívají aplikace. Platforma OSGi a obousměrný REST jsou podporovány v rámci Nothbound API. Plastforma OSGi se používá pro aplikace, které běží ve stejném adresním prostoru jako OpenDaylight kontrolér, kdežto rozhraní REST (webové) API se používá pro aplikace, které neběží ve stejném adresním prostoru, či dokonce stejném systému, jako kontrolér. Pracovní logika a algoritmy jsou součástí jednotlivých aplikacích. Tyto aplikace využívají OpenDaylight kontrolér ke shromažďování informací o síti, na kterých spouštějí vse analitické algoritmy, ze kterých pak vyvozují nová pravidla, která se pak nasazují do celé sítě. V rámci Southbound je spousta prokolů podporováno jako zásuvné moduly jako jsou například OpenFlow 1.0, OpenFlow 1.3, BGP-LS a podobně. Kontrolér OpenDaylight má předinstalovaný zásuvným modul OpenFlow 1.0. Tyto moduly jsou dynamicky připojeny do vrstvy abstrakce služeb (SAL).

SAL vystavuje služby, do kterých zapsují moduly na sever od SAL. SAL zjistí, jak má splnit požadovanou službu nezávisle na protokolu používaným mezi kontrolérem a aktivními prvky sítě. Tato vlastnost poskytuje aplikacím ochranu proti změnám, které postupem času přicházeji s postupným vyvíjením OpenFlow a dílčích protokolů. Aby mohl kontrolér řídit zařízení v rámci své doméně, potřebuje znát informace o zařízení, jejich schopnostech, dosažitelnost zařízení a tak dále. Tyto získané informace o zařízeních následně v rámci kontroléru ukládá a spravuje správce topologie (Topology manager). Ostatní komponenty, jako je například ARP, Host Tracker, Device Manager a Switch Manager, pomáhají při vytváření databáze topologie uvnitř Topology Manager.[31]

2.7.2.2 Open Networking Operating System

Open Network Operating System (ONOS) je OS navržen, aby pomohl poskytovatelům síťových služeb vybudovat softwarově definované sítě úrovně páteřních poskytovatelů, navržen pro velkou škálovatelnost, dostupnost a výkonost.

Přestože byl navržen tak, aby vyhověl všem nárokům a potřebám poskytovatelů služeb, ONOS umí též implementovat řídící vrstvu softwarově definovaných sítí pro podnikové lokální sítě (LAN) a sítě datových center. Open Networking Lab (ON.Lab) vydal zdrojový kód operačního systému ONOS, napsaný v Javě, pro open source komunitu v prosinci roku 2014. Nakonec v říjnu roku 2015, projekt ONOS se stal souástí Linux Foundation jako spolupracující Linuxový projekt s otevřeným kódem.

Nové verze operačního systému ONOS vycházejí každý čtvrt roku a to v únoru, květnu, srpnu a listopadu a často bývají abecdně pojmenovány podle ptáků, což je také ve znaku ONOSu.

Názvy vydání zahrnují Tenkozobce, Potáplice, Datel a Velociraptor. Podobně jako to je u většiny softwarů s otevřeným zdrojovým kódem, tak i ONOS má svoji stránku na GitHubu, kde vývojáři mohou přispívat změnami zdrojého kódu.

Mezi poskytovateli služeb přispívající do iniciativy ONOSu jsou AT&T, NTT Communications a SK Telecom. Přispívající významní prodejci do ONOSu zahrnují Cisco, Ericsson, Intel, NEC, Ciena a Huawei. Partneři ON.Lab a ONOSu našli mnoho míst uplatnění operačního systému ONOS v praxi. Jedno ze známějších míst uplatnění je projek ON.Labu Central Office Re-architected as a Datacenter (CORD). CORD byl vytvořen jako prostředek pro změnu ústředen telekominkačních operátorů do více škálovatelných a ohebných prostředí, podobným nastupujícím datovým centrům. CORD dociluje této změny například skrze virtuální síťové funkce a zařízení v prostorách zákazníka (CPE). ONOS je jeden ze systémů s otevřeným zdrojovým kódem, který je používán pro řízení CORDu.[32]

Od roku 2022/2023, kdy proběhl na ONOS ransomware útok, se zastavil vývoj na verzi z roku 2.7.0 z roku 2022. Vzhledem k současnosti to nevypadá, že by se někdy v brzkém obzoru projekt znovu rozjel, a proto projekt stagnuje.

2.7.2.3 Floodlight

Floodlight Kontrolér je kontrolér SDN vyvíjen otevřenou komunitou vývojářů, kde mnoho z nich přišlo ze společnosti Big Switch Networks. Floodlight využívá protokolu OpenFlow pro orchestraci datových toků v prostředí softwarově definovaných sítí (SDN). OpenFlow je jeden z prvních a tudíž nejvíce používaný standard pro implementaci SDN, jež definuje otevřený komunikační protokol v prostředí SDN, který povoluje kontroléru SDN (mozek operace) komunikovat s datovou vrstvou (forwarding plane) (switches, routers, atd.), pro propagaci změn do sítě.

Kontrolér SDN je zodpovědný za správu všech síťových pravidel a poskytování nezbytných příkazů do fyzické infrastruktury pro správné řízení datového provozu. Toto umožňje podnikům lépe se adaptovat jejím měnícím se potřebám a zároveň mít lepší kontrolu nad spravovanémi sítěmi. Kontrolér Floodlight byl původně nabídnut veřejnosti společností Big Switch jako součást projektu OpenDaylight, ale v červnu roku 2013 Big Switch od projektu odešel kvůli navyšujícím se střetům zájmů s Ciscem. Přestože Floodlight kontrolér je pořád open source, tak není nijak zapojen do projektu OpenDaylight.[33]

Stejně jako ONOS, tak i tento projekt stagnuje již od roku 2020/2021. Neví se jestli ještě někdy se projekt rozjede, avšak Github repozitář není ještě archivován, takže možná naděje je.

2.7.2.4 Faucet

Poslední zde představený projekt implementující kontrolér softwarově definovaných sítí je Faucet SDN. Faucet je kompaktní open source kontrolér OpenFlow, který umožňuje provozovatelům sítí provozovat jejich sítě stejným způsobem jako serverové clustery. Faucet přesouvá řídicí funkce sítě (jako jsou směrovací protokoly, zjišťování sousedů a přepínací algoritmy) do softwaru založeného na serveru nezávislém na dodavateli oproti tradičnímu vestavěnému firmwaru směrovače nebo přepínače, kde lze tyto funkce snadno spravovat, testovat a rozšiřovat pomocí moderních osvědčených postupů a nástrojů pro správu systémů. Faucet řídí hardware s OpenFlow 1.3, který poskytuje vysoký výkon při přeposílání dat.

Faucet má dvě hlavní součásti kontroléru OpenFlow, samotný Faucet a Gauge. Faucet řídí veškerý stav forwardingu a přepínačů a vystavuje svůj vnitřní stav, např. naučené hostitele, prostřednictvím Promethea (takže jej může grafovat open source NMS, jako je Grafana).

Gauge má také připojení OpenFlow k přepínači a monitoruje stav portů a toků (exportuje je do Prometheus nebo InfluxDB, případně i do plochých textových souborů protokolu). Gauge však nikdy nemění stav přepínače, takže funkce monitorování přepínače lze upgradovat, restartovat, aniž by to mělo vliv na přeposílání dat.

Kapitola 3

Přístup Cisca k SD-WAN

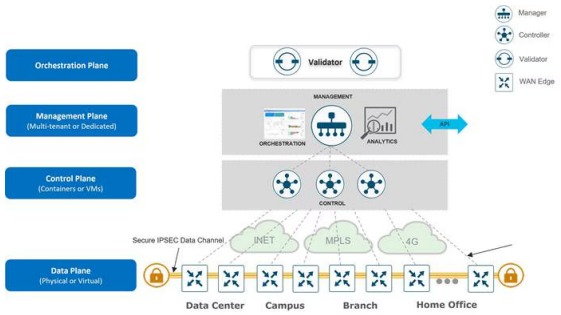
Softwarově definované sítě jsou nedílnou součástí současných řešení správy zařízení a řízení toků, a samozřejmě společnosti Cisco není pozadu. V této kapitole se proto zaměřím na definovaní pojmů, zařízení, rozdělení a jejich funcionalit v rámci Cisco SD-WAN.

Veškerá řešení SD-WAN od spocečnosti Cisco započala akvizice spocečnosti Viptela a převzetí jejího SD-WAN řešení, včetna pojmenovávání zařízení, proto „v“ v názvech zařízení právě znamená Viptela namísto obviklého virtual.

V oblastech síťové komunikace jde velmi často vidět, jak technologičtí giganti jako je práve Cisco, skoupí menší podniky jako byla práve Viptela. Cisco se zbaví konkurence a zároveň získá nové a inovativní technologie do svého repertoáru již existujících funkcionalit a zařízení.

Funkcionalita Cisco SD-WAN může být rozdělena do 3 vstev, těmi jsou: řídící vrstva, kontolní vstva a datová vsrtva.[1] V tradičních směrovačích datových sítí, se o přepínání a předávání datových toků starají linkové karty, ty zde reprezentují datovou vstvu. Centrální výpočetní jednotka (CPU), obvykle stojící za výpočty směrovací tabulky, předávání směrů a konečným rozhodování přeposílání dat, zde realizuje kontrolní vrstvu. Nakonec pak CLI směrovače reprezentuje jeho řídící vrstvu. Cisco SD-WAN spočívá v tom, že jednotlivé vrstvy se dají realizovat jako jednotlivá zařízení a ne jako jeden monolitický článek. Dá se proto říci, že takto řešení sítě dělá z celé množiny zařízení jeden monolit řízené sjednoceným grafickým rozhraním distribuovaným napříč prvky řdící vrstvy. Cisco SD-WAN pak konkrétně slučuje vrstvu kontrolní a řídící, kde řídící vrstva centralizuje řízení celé SD-WAN řešení do vManage zařízení, která mohou být virtualizována. vManage pak stojí za jednoduchou správou, konfigurací a zálohou konfiguračních

[1] Vrstvy mohou být 4, pokud zahrneme vrstvu orchestrační a vnich ležící vBond zařízení



Obrázek 3.1 Architektura Cisco SD-WAN [34]

souborů všech prvků v SD-WAN a nabízí uživateli jednoduchou a přehlednou správu přes grafické prostředí. Centralizovaná kontrolní vstva je pak obstarána pomocí vSmart zařízení, která stejně jako vManage, mohou být buďto fyzická nebo virtuální. Možnost mít veškeré informace o síti a jejich směrech na jednom místě je velmi užitečné, a všechny změny ve smerovacích tabulkách tak mohou být realizovány v jednom centrálním bodu namísto, jak je zvykem v tradičních sítích, aby byly změny v routovacích tabulkách přepočítávány na jednotlivých smerovačích pomocí obvyklých směrovacích protokolů. Tímto se dá ušetřit na drahých výpočetních prostředcích, v podobě CPU a RAM, které by museli být v jednotlivých směrovačích. Správa klíčů a prosazování pravidel je také řešeno na vSmart. V neposlední řadě datová vrstva Cisco SD-WAN je reprezentována WAN-edge[2] zařízeními, která budou podrobněji popsány v následující sekcí věnující se logickými sítěmi a WAN-edge směrovacími zařízeními. Architektura SD-WAN přidává nový koncept správy sítí jejímž je orchestrační vsrtva. Tato vstva je v systému SD-WAN reprezentována vBond kontroléry a jejich funcionalita bude podrobněji popsána níže. V obrázku 3.1 je zobrazeno jak jednotlivé vrstvy jsou Cisco SD-WAN jsou propojeny.

▶ Poznámka 3.1. Stejně jako v realitě, když se podíváme na takový počítač, pro většinu lidí se jedná o prostředek jak se podívat na Internet či hrání her. Vetšina si však neuvědomuje, že se zde jedná o pouze softwarovou vstvu položenou na konkrétní hardwarové architektuře kde jednotlivé řadiče či čipsety, jsou propojeny sběrnicemi různých typů. Pokud bychom chtěli takto bagatelizovat pohled na počítačové sítě, stačilo by nám pouze specifikovat učel použití například pomocí jakostí služeb (QoS) napříč sítí. Konfigurací jednoho prvku, v tomto případě vManage, získáme k dispozici celý abstraktní počítač, umožnující nám efektivně používat a nastavovat naší síť.

3.1 Fyzické sítě

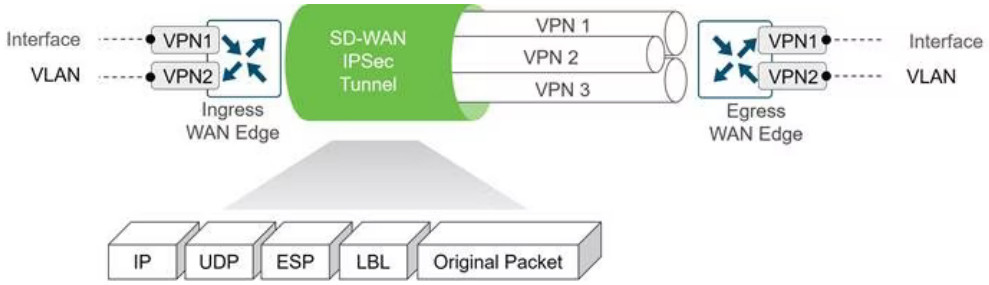
Fyzické sítě (též Underlay network) je ta část SD-WAN, která realizuje IP konektivitu napříč jednotlivými zařízeními SD-WAN. Zde nezáleží na tom, jak jsou jednotlivá zařízení propojena pokud jsou schopna jednotlivá WAN-edge zařízení mezi sebou schopna vytvořit komunikační kanály a přeposílat pakety mezi sebou. Následná dobře fungující fyzická síť může zvýšit celkový běh na niž postavené logické sítě. Toto chování je logické, jelikož se nedá postavit lepší virtuální tunel než je nejlepší možná cesta existující ve fyzické síti. Ve fyzických sítí takto zůstavají klasické způsoby návrhů sítí. Jedna z hlavních výhod nasazení a použití Cisco SDN je možnost zachování návrhu sítě bez nutnosti kompletního původního návrhu. Dobré praktiky pro LAN a datová centra (DC) jsou tradiční 3 a 2 vrstvé architektury či architektury páteře a listů (spine-leaf). Co se ale týče WAN, tak znovu zdůrazním, je důležitá právě a pouze IP konektivita.

Jedním ze způsobů jak podniky mohou dosáhnout lepších výkonů po přístup do WAN je míti redundantních zařízení a na nich redundaních přípojek různých poskytovatelů, tím docílí nejenom ochraně proti výpadkům ale získají možné lepší cesty mezi jednotlivými přípojnými body, celkově zvýšující funkčnost logické sítě. Právě svyšující se komplexita se neustále se navyšujícím počtem WAN poskytovatelů donutilo některé podniky migrovat do řešení jako je SD-WAN, kde právě tyto problémy z veklé části zmizely. Další věcí, která s použitím SD-WAN zmizela, je nutnost používat BGP na WAN routerech a rozdělování zátěze (load-balancing).

3.2 Logické sítě

Logické sítě slouží k přeposílání datových komunikací. Zde právě abstrakce SD-WAN, je realizována a umožňuje nám vytvářet sítě, které na první pohled vypadají nějak a chovají se zcela jinak. Způsob, kterým je toto chování docíleno, je právě vytváření dynamických šifrovaných

[2] Zahrnují zařízení vEdge a cEdge



Obrázek 3.2 Segmentace topologie Cisco SD-WAN [34]

tunelů, které spolu s WAN-edge zařízeními efektivně reprezentují datovou vrstvu celého řešení SD-WAN.

▶ Poznámka 3.2. Jeden by si mohl pomyslet, že abstrakce SD-WAN a datová vsrtva SD-WAN jsou vlastně jedno a to samé, a v tomto kontextu by i měl pravdu.

3.2.1 Segmentace logických sítí

Datová vstva SD-WAN přirozeně podporuje možnost segmentace, tudíž je jí nedílnou součástí. Segmentace v logické síti je řešeno standartem RFC 4023. Obsah IPv4/6 je obalen v MPLS komunikaci, která je pak vložena do IP nebo GRE komunikace. Hlavička pak obsahuje informace pro konečné edge zařízení, které jsou nutné pro dobrou segmentaci sítě. Tím, že jednotlivá komunikace je ještě zabalena do šífrované komunikace umožňuji dodat komunikaci vysněnou bezpečnost a skrytí filtrovatelného obsahu (obvykle MPLS komunikace je filtrovaná či přímo nefukční).

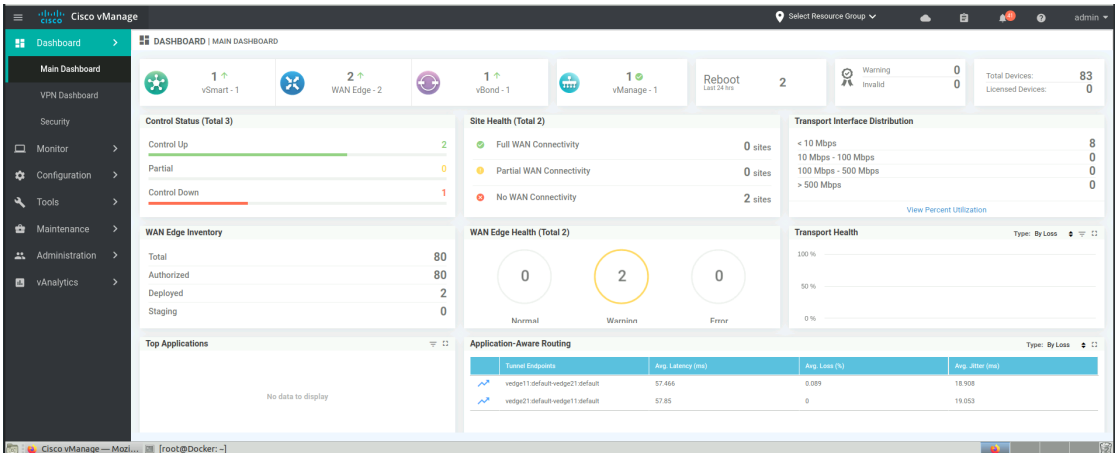
„VRF a VPN nám poskytují umožňuje rozdělení řídící a datové vrstvy do různých logických částí. Segmentace v datové vrstvě je docíleno vytvořením vícero izolovaných instancí směrovacích tabulek a přiřazení specifických rozhraní těmto instancím.“[6]

▶ **Příklad 3.3.** Jedním z příkladů jak tato segmentace může být použita, je když existuje potřeba mít vícero rozdělených topologií, například jednu pro podnikovou VPN, jednu pro síť pro hosty a podobně. Prakticky se můžeme dívat na segmentaci jako na klasické VLANy, které jsou posílány přes WAN. Obrázek 3.2 pak zobrazuje celý koncept segmentace.

Podporavné směrovací protokoly jsou OSPF, EIGRP a BGP. Když žádný z těchto směrovacích protokolů není použit, first-hop gateway redundancy protokol (FGRP) v podobě virtual router redundancy protokol (VRRP) je použit.

3.2.2 Detekce obousměrného směrování

Detekce obousměrného směrování (Bidirectional Forwarding Detection [BFD]) je použit ve všech dynamických tunelech datové vstvy, a zde popíši stručně jeho chování. Hello pakety jsou posílány aby zjistily, zda linka je pořád dostupná a jaké parametry daná linka má (parametry jako je například zpoždění, ztrátovost či jitter). Samořejmě žádné pakety není nutné zpracovávat cílovým WAN-edge zařízením, čili BFD Hello pakety jsou jednoduše posílány zpět odesílateli jako v případě ICMP, čímž hodně ulehčí práci cílového zařízení. Tento návrh BFD pro pocesor znamená, že nemusí přerušit svou stálou činnost kvůli BFD. Na základě BFD a vnitřních definovaných politik ve vManage je pak vybrána optimální cesta pro komunikaci mezi body. BFD má nastavitelné parametry, ačkoliv jeho funkcionalita nemůže být přímo vypnuta. BFD je také nástroj, pomocí kterého SD-WAN, kromě zjištění metrik cest, dokáže zajistit jakost služeb (QoS). Na obrázku 3.4 je pak vidět příklad, jak se mohou buduvat tunely na kontrolní a datová vstvě, kde právě BFD se používá.



Obrázek 3.3 Úvodní stránka ve vManage Cisco SD-WAN.

3.3 Zařízení/součásti SD-WAN

Veškerá funkcionalita Cisco SD-WAN je reprezentována pomocí dedikovaných strojů schopných implementovat SD-WAN či možná částečná virtualizace. Zde budou podrobněji popsány jednotlivé funcionality zařízení vManage, vSmart, WAN-edge, vBond.

3.3.1 vManage - Řídící vstva

vManage implemntuje řídící vrstvu Cisco SD-WAN a slouží jako centralizovaný bod pro řízení a konfiguraci jednotlivých dílčích zařízení. Cisco vManage je tvořen grafickým prostředím přístupným přes HTTPS a CLI. Ukázka uvodní stránky vManage je na obrázku 3.3.

Grafické prostředí, neboli frontend, vManage komunkuje s backendem pomocí REST programovacím rozhraním (REST API) a protokolem NETCONF. Použití rozhraní REST API umožňuje další možnosti ovládání backendu vManage, jako je například POSTMAN, a dává nám možnost spouštět různé skripty. Další vlastností vManage je možnost jeho virtualizace a veškerá konfigurace může být vyhotovena skrze právě vManage. Jednotlivá zařízení pak mohou být ve stavu řízení pomocí CLI nebo pomocí vManage. Samozřejmě pro jednodužší a standartizovanou správou zařízení napříč sítí je vhodné aby vše co může aby bylo řízeno pomocí vManage. Tím se dá zaručit konzistence sítě a její škálovatelnost.

Redundance a škálovatelnost také není problém co se týče nasazení vManage, jelikož Cisco umožňuje vytvářet klastry (spolupracující skupinu zařízení) vManage zařízení, jedinou podmínkou pro správnou funkčnost je, aby v síti byl lichý vManage zařízení kvůli synchronizaci. V případě, že se v síti objeví vadná konfigurace, tak o správnosti rozhoduje většinové hlasování a proto ten lichý počet. Každý vManage zvládne řídit až 2000 WAN-edge zařízení (nezahrnuje vSmart). Jedna z vlastností klastru vManage je, že jednotlivé řídící tunely jsou zátěžově vyvážené mezi jednotlivé uzly. vManage pak komunikuje s WAN-edge zařízeními pomocí dynamckých DTLS tunelů. V případech kdy máme dostupno více tansportních spojení na WAN-edge zařízení, pouze jedno spojení je vyhrazeno pro komunikaci s vManage. V případě ztráty spojení WAN-edge s vManage se WAN-edge pokouší vytvořit nový tunel jinou cestou pokud je to možné. Vzniklý výpadek spojení na WAN-edge však neohrožuje funčnost SD-WAN sítě jelikož konfiguraci si zařízení drží aktualní konfiguraci a při zprovoznění tunelu se buďto nahraje nová konfigurace z vManage do WAN-edge nebo naopak pokud byly provedeny lokální změny. Toto je však závislé na tom kdo spravuje konfiguraci zařízení, čili v jakém konfiguračním stavu je.

Doporučované praktiky jak vytvářet a nasazovat klastry vManage je, držet si jeden vManage

„on premise“/lokálně a další pak ve veřejném cloudu pro případ výpadku. Tímto se dá zaručit i to, že vManage je různě po světě podle toho kterého poskytovatele cloudu zvolíme a kde všude má své obrazy.

Další vlastnosti vManage je možnost takzvaného „multi-tennancy“, kde v základu je nasazení pouze jeden tennant. V práci se pouze zaměřuji na jednoho tennanta.

3.3.2 vSmart - Kontrolní vstva

Kontrolní vrstva, reprezentující zařízením vSmart, slouží ke správě směrování, bezpečnost, segmentace a authentikaci zařízení různých vstev Cisco SD-WAN. Samozřejmě i zde bylo myšleno na škálovatelnost a tudíž každé vSmart zařízení zvládne obsloužit až 5400 různých spojení s jednotlivými zařízení SD-WAN, s tím že každá logická síť umožňuje mít až 20 vSmart zařízení. Výsledkem je schopnost vytvoření velmi velkých sítí. Přestože WAN-edge zařízení se dokáže spojit až s 3 vSmart zařízeními, pouze jedno je potřeba pro autentikaci a získání politik sítě.

„Rozdělení kontrolní vrstvi od datové a řídící nám dává možnost rozšiřovat síť do větších rozměrů přestože zjednodušují síťové operace. Například když se podíváme na klasické směrovací protokoly zjišťující stav linek jako jsou například OSPF nebo IS-IS, tak každý směrovač ví o každém stavu linky napříč celou sítí a z toho přepočítává každý směr směrovací tabulky. Jak je už z textu cítit, tak toto může být nezanedbatelná přítěž na procesor směrovače navzdory tomu, že směrovači poskytne limitovaný pohled na síť. Na druhou stranu protoky vzdálených vektorů, znají pouze přímo připojené sítě a to co jim sousedé poví. Jako reakce na tuto logiku je to, že směrovače mohou dělat ne vždy optimální směrovací rozhodnutí. S Cisco SD-WAN řešením, všechny směry jsou naučené ve vSmart, který následně spočítá směrovací tabulku, kter následně rozešle mezi všechny WAN-edge zařízení. Jelkož vSmart má podrobný přehled o celé síti, tak zvládne udělat výpočet nejlepší cesty a snížit tak celkovou komplexitu sítě přestože umožní větší škálovatelnost.“[6]

Funkcionalita vSmart v SDN je velmi podobná BGP route reflektoru v iBGP. Informace o směrech se sbírají z WAN-edge zařízení do vSmart a vSmart na ně aplikuje nastavené politiky z vManage ještě předtím než se pošlou zpět na WAN-edge zařízení.

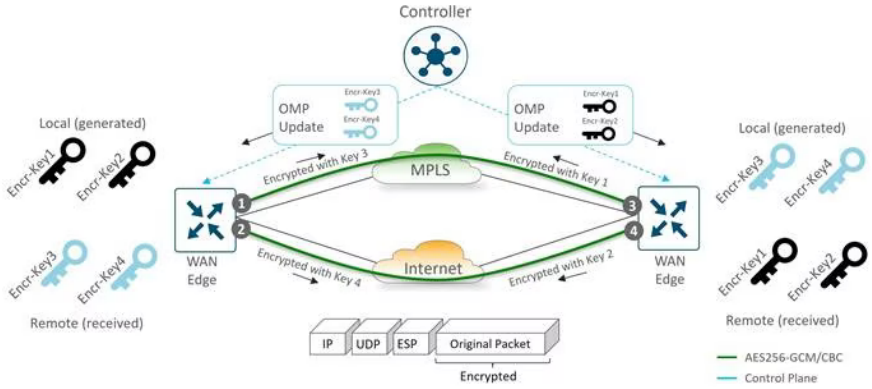
vSmart také stojí za definováním různých topologií za každé VPN rozhraní. Opět řídící politiky jsou definovány ve vManage, které pak distribuje politiky do všech vSmart zařízení přes NETCONF, kde poté vSmart aplikuje politiky do WAN-edge zařízení pomocí soukromého Cisco protokolu Overlay Management Protocol (OMP). Řízení distribuci směrů davá vSmart schopnost rozhodnout jak jednotlivé tunely se vytvoří. Úplný mesh, částečná mesh, PTP či jakákoliv jiná permutace tunelů mezi WAN-edge zařízeními.

Další důležitou funkcí kterou vSmart vykonává je výměna klíčů mezi WAN-edge zařízeními. Datová vrtva, kterou právě WAN-edge zařízení implementují, je práve tvořena šifrovanými tunely. Obsluhu výmeny klíčů, tím že obsluhuje vSmart výrazně ulehčuje výpočetní prostředky WAN-edge a tím přímo zvyšuje možnou velikost sítě.[3]

Zde je doporučeno nasazení alespoň 3 vSmart v SDN s tím aby každy bylo někde různě po světě. Zde je pak nutná jednotná konfigurace politik napříč všemi vSmart zařízeními. Rozdílé politiky pak mohou vést k až nežadoucílmu a posílání komunikace do neexistujících míst (takzvaný blackholing of traffic). Úplná mesh topologie OMP tunelů mezi WAN-edge zařízeními pak zabraňuje vzniku rozdílných konfigurací. V případech kdy jeden z vSmart zkolabuje, tak WAN-edge se snaží automaticky rebalancovat do zbylích funčních vSmart zařízení.[4]

[3] Typický případ metody „zoděl a panuj“ použitý v praxi

[4] Zdůrazňuji, že vSmart zařízení nemusí být nutně fyzická zařízení.



Obrázek 3.4 Řídící vrstva Cisco SD-WAN [34]

3.3.3 WAN-edge zařízení - Datová vrstva

WAN-edge zařízení reprezentují datovou vrstu a tvoří ji buď Cisco vEdge zařízení implementující Viptela edge a Cisco cEdge zařízení implementující Cisco XE SD-WAN směrovače[5].

Tato zařízení nejsou rozdílná pouze ve jméně, ale také v konkrétních detailech, které představím v příslušných podsekcí. Jak už název WAN-edge napovída, tato zařízení v SD-WAN datové vrstvě se nasazují jako okrajové (edge anglicky) směrovače, kde často bývá eBGP nasazen.

S bezpečností na mysli, tradičně implicitně zakazují[6] přístup do zařízení odkudkoli a specificky implicitně povolují komunikaci s SD-WAN zařízeními. Existují zde možnosti zapnout naslouchací služby jako je například SSH, NETCONF, NTP, OSPF, BGP, UDP, STUN a podobně. Povolené odchozí služby jsou DHCP, DNS a ICMP.

Když WAN-edge se prvotně připojí do sítě (proběhl například DHCP nebo administráto nastavil zařízení staticky), tak se nejprve se snaží dotázat na Cisco Plug and Play (PnP) server typicky s doménou devicehelper.cisco.com, nastaven jako vBond (orchestrátor) pro zařízení typu cEdge. Pro zařízení typu vEdge je výrobní nastavení na takzvaný Zero Touch Provisioning (ZTP) server typicky s doménou ztp.viptela.com. Obrázek 3.4 zobrazuje jak proces automatického začlenění probíhá. Pokud tento proces neuspěje, tak pořád existuje proces manuální konfigurace či bootstrapu.

▶ Poznámka 3.4. Více do hloubky na téma začlenění zařízení do SD-WAN je vysvětleno v sekci vBond orchestračního zařízení a ZTP serveru. Praktická čast pak popíše postup krok za krokem jak nastavit ZTP server.

3.3.3.1 vEdge

Zařízení typu vEdge jsou zařízení, která implementovala původní Viptela SD-WAN před jeho akvizicí společností Cisco. Tato zařízení jsou plně vybavené směrovače s dodatečnými funcionalitami pro realzaci SD-WAN. Standardní funkce tohoto směrovače jsou směrovací protokoly OSPF a BGP, Access Control Listy (ACL), jakost služeb (QoS) a další.

Cisco vEdge i po akvizici pořád beží na Viptela OS, modikace Linuxu pro tyto routovací zařízení. Jejich účel je rozšíření Cisco SD-WAN do řešení veřejných prostředí cloudů. Tudíž virtualizace je podporována pouze pro tato zařízení a Cisco poskytuje obrazy vEdge Cloud virtuálních zařízení. Běh těchto virtualizovaných zařízení taky poskytuje jedinečnou ohebnost pro hybridní cloudová řešení. Pořád však existují i fyzická zařízení a zde je představím.

Repertoár zařízení podle Cisco: [35]

[5] V praxi jsou často právě Cisco XE směrovače referovány jako cEdge.
[6] Pokud není povoleno tak zakaž

1. vEdge-100: pět stálých 10/100/1000 Mbps portů. Jsou ve třech variantách:
   - vEdge 100b: Pouze ethernet
   - vEdge 100m: Ethernet a integrovaný 2G/3G/4G modem
2. vEdge-1000: 8 portů GE SFP
3. vEdge-2000: 2 sloty pro moduly
4. vEdge-5000: 4 síťové moduly
5. ISR 1100-4G: 4 GE WAN porty
6. ISR 1100-4GLTE: 4 GE WAN porty, 4G LTE (CAT4)
7. ISR 1100-6G: 6 GE WAN porty (4 GE and 2 SFP)

Modely Cisco ISR 1100 jsou také zařazeny jako vEdge zařízení, jelikož jejich softwarovou výbavou je práve Viptela OS, tudíž funcionalita těchto zařízení je ve směs stejná jako v případech vEdge. Zásadně se zařízení liší pouze rozhraními.

3.3.3.2 cEdge

Zařízení WAN-cEdge jsou založeny na bázi vEdge, s tím že mají dodatečné funcionality. Většina dodatečných funcionalit právě spočívá v bezpečnosti jako je například Direct Internet Access (DIA), Intrusion Detection System (IDS), od Cisca Advanced Malware Protection (AMP) a Intrusion Prevention System (IPS). Na rozdíl od vEdge zařízení tato zařízení mají modulární IOS-XE OS, který je opět založen an Linuxu. Jelikož se jedná o jiný operační systém, tak se změnila i CLI a pokud chceme používat SD-WAN funkcionality tak je musíme explicitně povolit v administrativním módu zařízení cEdge pomocí příkazu a následně zařízení rebootovat.

```
router# controller-mode enable
```

Nebo se dá přímo nahrát SD-WAN podporovaný image na zařízení, pak tento příkaz neexistuje a zařízení je schopné používat funkce SD-WAN.

Fyzická a logická zařízení směrovačů cEdge jsou:[35]

1. Série ISR & ASR: S softwarovým obrazem IOS XE SD-WAN, schopnost provozovat SD-WAN může být povolena na vybraných směrovačích ze série ISR 1000, ISR 4000 a ASR 1000.
2. ENCS: S softwarovým obrazem IOS XE SD-WAN, schopnost provozovat SD-WAN může být povolena na vybraných platformách ze série ENCS 5000.
3. vEdge Cloud a CSR 1000V jsou cloudové prvky ze řešení SD-WAN.

Virtualizovatelné cEdge zařízení jsou také směrovače Cisco ISRv a Cisco Catalyst 8000v. Tyto směrovače mají minimální rozdíly oproti ostatním zařízením a mezi sebou a mají všechny stejný výhody jako vEdge Cloud virtuální zařízení. Samozřejmě největší vlastností je právě podpora Cisco SD-WAN. Pokud tedy má podnik v síti již zařízení ISR a ASR, tak pouhá koupě nových licencí umožňuje zařízením se stát součástí SD-WAN.

▶ Poznámka 3.5. Podpora směrovacího protokolu Enhance Interior Gateway Routing Protocol (EIGRP) byla přidána, tudíž pokud je potřeba tohoto routovacího protokolu, tak je možno jej použít.

3.3.4 vBond - Orchestrační vrstva

vBond zařízení realizují v SD-WAN takzvanou orchestrační vrstu a těmto zařízením se obecně říká orchestrátor, v Cisco SD-WAN se jedná o zcela nezbytnou součást. Orchestrační vrstva je, co se týče SDN, zcela nový koncept. Orchestrátoři plní v SD-WAN následující funkce:

Autorizace, Authentikace a Validace komponent SD-WAN: V okamžik, kdy se začleňuje nové zařízení do SD-WAN, tak jedinou informaci, které nové zařízení potřebuje je být schopné navázat spojení s vBond. Už během začleňování se vytváří DTLS tunely mezi nově začleněném zařízení a vBondem. Poté co je zařízení authorizováno, vBond o jeho přítomnosti oznámí ostatní kontroléry sítě[7] a postupně se formují nové DTLS tunely mezi vManage a vSmart. V okamžik, kdy zařízení je začleněno, tak se DTLS tunely s vBond zruší.

Počáteční konfigurace WAN-edge: Způsoby, jak se vůbec má WAN-edge zařízení dovědět o vBond a pak jeho následného začlenění, jsou 4.

Plug & Play: Je plně automatický způsob úplného začlenění do sítě cEdge zařízení. V tomto případě musí být zařízení schopno vidět do internetu aby se dokázalo spojit s devicehelper.cisco.com.

Zero Touch Provisioning: Je plně automatický způsob úplného začlenění vEdge zařízení do SD-WAN. V tomto případě se snaží zařízení domluvit s vBond orchestrátorem, který musí být dostupný přes doménu ztp.viptela.com. Ukázka konfigurace je zobrazena zde 3.6.

Konfigurace pomocí bootstrap souboru: V tomto případě je nutná minimální na vManage, kde se pak musí vygenerovat pro začleňované zařízení Multipurpose Internet Mail Extension (MIME) soubor. Tento soubor se pak nějakým způsobem nahraje na začleňované zařízení do bootflash. Začleňované zařízení pak použije informace ze souboru pro konfiguraci a začlenění do SD-WAN.

Manuální konfigurace: Zde není co dodat. Použití tohoto způsobu je doporučeno pouze pokud v síti SD-WAN není mnoho zařízení, a tudíž není nutno automatizovat do hloubky. Zde bude popsán proces minimální manuální konfigurace zařízení tak, aby bylo schopno vidět do internetu. Celkový postup jak manuálně začlenit zařízení bude popsáno v dalších kapitolách, tudíž tento návod bereme s rezervou.

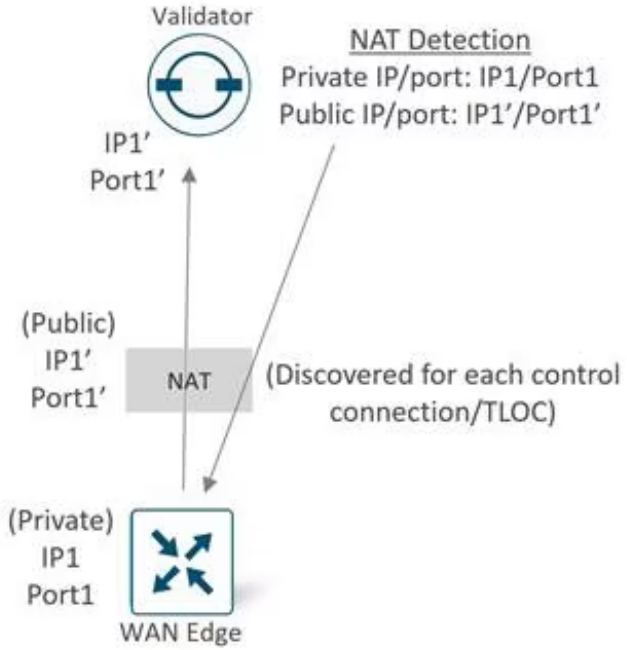
```
vedge# config
vedge(config)# system
vedge(config-system)# host-name vedgeXX
vedge(config-system)# site-id XXX
vedge(config-system)# system-ip X.X.X.X
vedge(config-system)# organization-name YYY
vedge(config-system)# vbond ZZZ
vedge(config-system)# vpn 0
vedge(config-vpn-0)# ip route 0.0.0.0/0 XX.XX.XX.XX
vedge(config-vpn-0)# dns XX.XX.XX.XX
vedge(config-vpn-0)# interface Y
vedge(config-interface-Y)# ip address XXX.XXX.XXX.XXX/X
vedge(config-interface-Y)# no shutdownt
vedge(config-interface-Y)# tunnel-interface
vedge(config-tunnel-interface)# allow-service all
vedge(config-tunnel-interface)# encapsulation ipsec
vedge(config-tunnel-interface)# commit and-quit
vedgeXX# request root-cert-chain install /home/admin/XX.pem
vedgeXX# request vedge-cloud activate chassis-number [UUID] token T
```

[7] Těmi se myslí vManage, vSmart a ostatní vBond.

Jak je vidět, tak samotná kofigurace vEdge zařízení není až tak dlouhá, když vezmeme v potaz že konfigurujeme pouze pár zařízení. Samozřejmě manuální konfigurace může zavést chyby ze strany administrátora tak, že se může upsat a následné hledání chyby může být náročné a často vyžaduje fyzickou přítomnost administrátora.

▶ Poznámka 3.6. V dalších kapitolách bude také podrobně, ukázáno jak probíhá začlenění zařízení pomocí ZTP.

NAT traversal: Poslední věcí, kterou je schopno vBond zařízení dodat je NAT traversal. vBond jako takový operuje jako server STUN podle RFC 5389. [36] Tudíž WAN-edge zařízení musí fungovat jako klienti STUN.3.5 Obecně to znamená, že vBond dokáže komunikovat a obstarat tunely s WAN-edge zařízeními, která jsou za NATem. Když takové WAN-edge zařízení se pak následně snaží vytvořit DTLS tunel s vBondem, jeho známá IP adresa (s velkou pravděpodobností neveřejná) je použita jako zdrojová adresa a také je tato adresa součástí obsahu zprávy. Když pak vBond dostane tuto zprávu, tak provede operaci XOR mezi zdrojovou IP a IP zadanou ve zprávě a pokud výsledek je nenulový tak je jistota toho, že po cestě je alespoň jedno zařízení implementující překlad adres. vBond pak informuje WAN-edge o tom, že je za NATem a WAN-edge zařízení o tomto faktu obeznámí zbytek sítě. Existují však případy kdy nelze přes NAT komunikovat, třeba v případech kdy je použit symetrický NAT.



Obrázek 3.5 STUN Cisco SD-WAN [34]

Samotná dosažitelnost vBond zařízení je potřeba brát v potaz, jelikož jeden z požadavků na vBond je, že vlastní veřejnou IP adresu nebo je za 1:1 NATem s veřejnou IP adresou.[8] Ostatní součásti SDN pak mohou být za statickým či dynamickým[9] NATem a pořád budou schopny komunikovat se zbytkem SDN. vBond pak zjišťuje, kdo je za NATem a kdo ne a umožňuje jim vytvářet spojení navzdory býti za NATem.

[8] V podstatě to samé jako kdyby měl veřejnou IP adresu.
[9] Též známí jako Port Address Translation (PAT)

3.3.5 Ostatní volitelná zařízení

V předchozích sekcí byly zmíněny zařízení, která jsou nutná pro správný chod Cisco SD-WAN, v této sekci se zaměřím na čistě zařízení SDN, která v síti mohou být, ale nemusí. vManage umí pracovat i s zařízeními, která nejsou součástí SDN jako je Cisco Unified Communications Manager (CUCM) či virtual Wireless Lan Controller (vWLC). Tato zařízení však nejsou čistě součástí SDN a proto se jim zde nebude práce věnovat.

3.3.5.1 Kořenová certifikační autorita

Kritickou součástí Cisco SD-WAN jsou certifikáty, které zajišťují bezpečnou komunikaci napříč internetem. Ne jenom, že toto řešení používá whitelisty pro authentikaci komponent, ale také každý kontrolér se rozpoznává pomocí unikátního certifikátu. Každý kontrolér se totiž vzájemně authentizuje s ostatními přes tyto certifikáty a z ních buduje kontrolní vrstvu SDN. Podobným procesem projdou i jednotlivá WAN-edge zařízení. Zařízení certifikační authority je zde volitelné ve smyslu, že tuto funkci může vykonávat vManage.

Identita WAN-edge Každé fyzické WAN-edge zařízení mý z výroby svůj certifikát totožnosti, podle kterého se dají jednotlivý zařízení identifikovat pomocí devicehelper.cisco.com a začlenění je takřka bez práce. Co se ale týče virtuálních tak to není ten samý případ. V případě virtuálních vEdge Cloud zařízení a podobně, tak nemají žádný certifikát přeinstalovaný z výroby. Místo toho používají One Time Password (OTP)/Token, který je generován uvnitř vManage a nastaven během nasazování zařízení za účelem dočasné identity. Jakmile zařízení je dočasně authorizované tak, stálá identita je zařízení dodána z vManage, jež může pracovat též jako certifikační autorita pro generování a instalaci certifikátů těmto zařízením. [34]

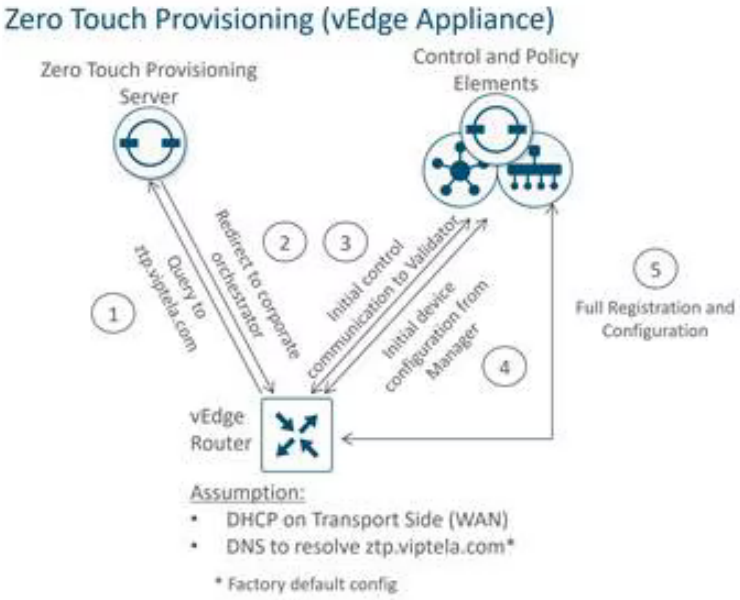
Identita kontrolérů Zde existují 3 možnosti jak dodat kontrolérům jejich eidentity. Symantec/DigiCert a Ciscem podepsané certifikáty jsou dvě nejvíce používané možnosti, v našem simulované prostředí však použijeme třetí možnost a tou je vlastně podepsané certifikáty použité jako podnikové CA. Protože tato metoda je použita v našich laboratořích, nebude zde zmínění, nýbrž bude podrobněji probrána v následujících kapitolách. Každá součast SD-WAN musí mít lokálně nainstalován kořenový certifikační řetěz důvěry pro certifikační autoritu (CA). Přeinstalované klíče zařízení Cisco SD-WAN nejsou a nemohou být použity pokud používáme podnikovou CA.

3.3.5.2 vAnalytics

Další volitelnou složkou SD-WAN je vAnalytics, jež vyžaduje dodatečnou licenci, pokud chceme toto zařízení používat. vAnalytics je součástí řídící vrstvi Cisco SD-WAN a poskytuje možné predikce potenciálních budoucích požadavků na potřeby vylepšení v síti. Jedna z hlavních parametrů sledování je celková propustnost linek. Vše je zakládáno na základě dat získaných z vManage. Hlavní nevíhodou těchto zažízen je velmi drahá licence.

3.3.5.3 ZTP server

Zero Touch Provisioning nebo také bezzásahové zajišťování služeb je způsob jak začlenit Viptela vEdge zařízení do sítě SD-WAN. Tento proces vyžaduje přítomnost ZTP serveru. Na obrázku 3.6 je vidět, kde v síti by se měl ZTP server nacházet. ZTP server je často realizován na vBond zařízení, proto ZTP funcionalita a server spadá pod Orchestrační vrstvu. Z počátku, kdy Viptela ještě nebyla součástí Cisco, byly vytvářeny klastry těchto serverů. Zákaznící kteří chtěli používat ZTP pro začleňování zařízení do SD-WAN tak jednodužše museli zaplatit Viptele drobný poplatek pro přístup na jejich ZTP servery. Cisco na to ale šlo jinak. Cisco nově už nezakládá ZTP servery, nýbrž tu možnost vytvoření ZTP serveru dává zákazníkům. Naneštěstí nastavení a azprovoznění ZTP serveru je jednoduchý proces, který bude téže vysvětlen v následujících kapitolách.



Obrázek 3.6 Orchestrační vsrtva Cisco SD-WAN [34]

▶ Poznámka 3.7. Jelikož ZTP je nástavbou vBond zařízení a vBond zařízení je nástavbou vEdge zařízení, jejich zprovoznění stojí jen jeden softwarový obraz a jediné co se liší je konfigurace prvků.

Kapitola 4

Přístup Huawei k SD-WAN

Jak již bylo zmíněno, softwarově definované sítě jsou nedílnou součástí současných řešení správy zařízení a řízení toků, a proto dalším představitelem komerční sférey je v této diplomové práci Huawei. V této kapitole se proto zaměřím na definovaní pojmů, zařízení, rozdělení a jejich funcionalit v rámci Huawei SD-WAN.

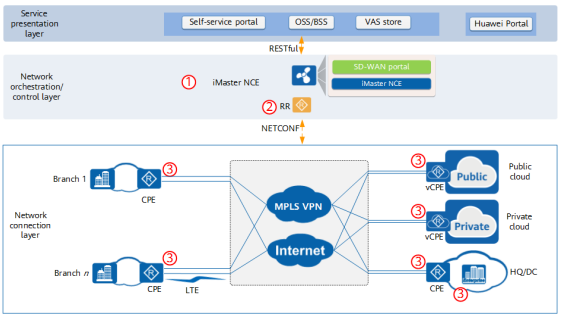
Produkt Huawei SD-WAN má třívrstvou architekturu sestávající z vrstvy prezentace služeb (Service Presentation), orchestrační a řídící vrstvy(Orchestration and Control) a vrstvy síťového připojení (Network Connection).

První vrstva je to, s čím uživatelé pracují. Řídící a orchestrační vrstva je to, co spravuje různá zařízení CPE a tunely VPN. Poslední vrstva, tedy vrstva síťového propojení, umožňuje zařízením CPE fungovat jako brány ke zbytku sítě SD-WAN a cloudům. Prvky řešení Huawei SD-WAN

4.1 Prvky řešení Huawei SD-WAN

Řešení Huawei SD-WAN propojuje všechny uzly sítě organizace. Pobočky se mohou připojit k jiným pobočkám, k centrále, k datovým centrům a ke cloudu. Software sítě SD-WAN si uvědomuje aplikace a dosahuje inteligentního řízení provozu aplikací. Takový provoz je posílán optimální cestou napříč síťovými spoji v závislosti na tom, jak je aktuálně využívána šířka pásma daného spoje.

Síť Huawei SD-WAN má tři vrstvy. Na vrcholu je vrstva prezentace služeb. Pod ní je vrstva orchestrace a řízení sítě. Ve spodní části je vrstva síťového připojení. 4.1

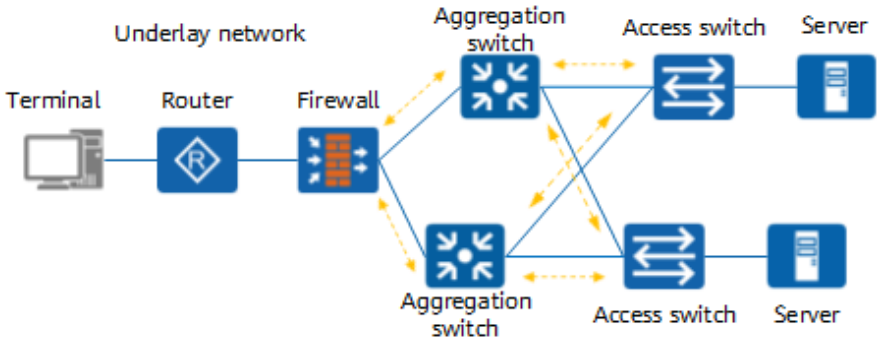


Obrázek 4.1 Architektura sítí Huawei SD-WAN [37]

4.2 Fyzické sítě

Fyzická síť, jak už název napovídá, je základní fyzickou infrastrukturou logických sítí.

Jak je znázorněno na následujícím obrázku, jedná se o fyzickou síť, která se skládá z více typů zařízení a je zodpovědná za přenos datového toku mezi sítěmi. 4.2



Obrázek 4.2 Fyzická síť Huawei SD-WAN [38]

Ve fyzické síti mohou být propojena zařízení, jako jsou přepínače, směrovače, vyrovnávače zátěže a firewally. K zajištění propojení IP mezi těmito zařízeními je však nutné použít směrovací protokoly.

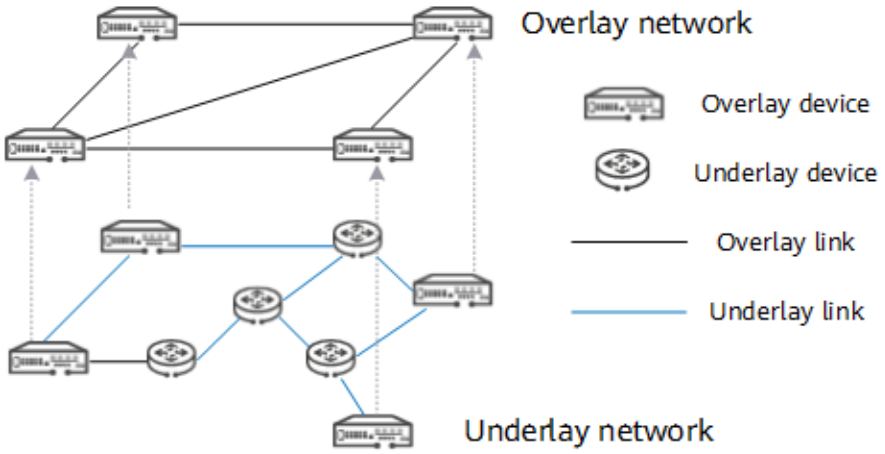
Fyzická síť se může rozkládat na 2. nebo 3. vrstvě ISO/OSI modelu. Typickým příkladem fyzické sítě na 2. vrstvě je síť Ethernet, na které jsou vytvořeny virtuální sítě LAN (VLAN). Typickou fyzickou sítí 3. vrstvy je internet. Protokol OSPF (Open Shortest Path First) nebo IS-IS (Intermediate System to Intermediate System) se používá pro řízení směrů v rámci autonomního systému (AS), zatímco protokol BGP (Border Gateway Protocol) se používá pro přenos směrů a propojení mezi AS. S rozvojem technologií lze fyzické sítě vytvářet také pomocí technologie Multiprotocol Label Switching (MPLS), kde směrovače na základě značky (barvy) mohou nakládat s pakety jako přepínač, což je technologie rozsáhlé sítě (WAN), která funguje mezi 2. a 3. vrstvou.

Nicméně, v tradičních síťových zařízení se předávají jednotlivé pakety na základě použitého hardwaru. Fyzická síť vytvořená na základě tradičních síťových zařízení má následující problémy:[38]

- Fyzická zařízení předávají pakety na základě cílových IP adres. Předávání paketů je proto do značné míry závislé na datových cestách.
- Při přidávání nebo změně služeb v rámci sítě, je třeba upravit stávající fyzická síťová připojení. Překonfigurování je časově náročné, a náchylnéna chyby.
- Na internetu nelze obecně zajistit bezpečnost soukromé komunikace.
- Rozdělování a segmentace sítě jsou složité a nelze jimi dosáhnout rozdělování síťových zdrojů na vyžádání.
- Vícecestné předávání je komplikované a k implementaci vyrovnávání zátěže nelze integrovat více podkladových sítí.

4.3 Logické sítě

Aby se odstranila omezení fyzických sítí, lze nad fyzickými sítěmi vytvořit virtuální logické sítě pomocí technologií, schopné virtualizovat sítě, jako jsou například VPN tunely.4.3



Obrázek 4.3 Logická Huawei SD-WAN [38]

Zařízení ve fyzických sítích jsou podle potřeby propojeny virtualními spoji, které tvoří logické topologie.

Mezi propojenými zařízeními logických sítí jsou vytvářeny tunely. Při vytváření datového provozu, přidá si zařízení do do komunikace novou hlavičku IP a hlavičku tunelu do části dat, tím se odstíní vnitřní hlavičku IP od směrování. Komunikace je pak předávána na základě vnější hlavičky IP. Když druhá strana přijme naši komunikaci, odstraní vnější hlavičku IP a komunikaci pošle podle původní hlavičky IP. V tomto procesu logická síť neví nic o fyzické síti.

Logické sítě podporují různé síťové protokoly a standardy, včetně virtuální rozšiřitelné sítě LAN (VXLAN), virtualizace sítě pomocí zapouzdření generického směrování (NVGRE), Single Spanning Tree (SST), zapouzdření generického směrování (GRE), virtualizace sítě na třetí vrstvě (NVO3) a virtuální privátní sítě Ethernet (EVPN).

Se zavedením technologie softwarově definovaných sítí (SDN) mají logické sítě s nasazeným kontrolérem následující výhody:

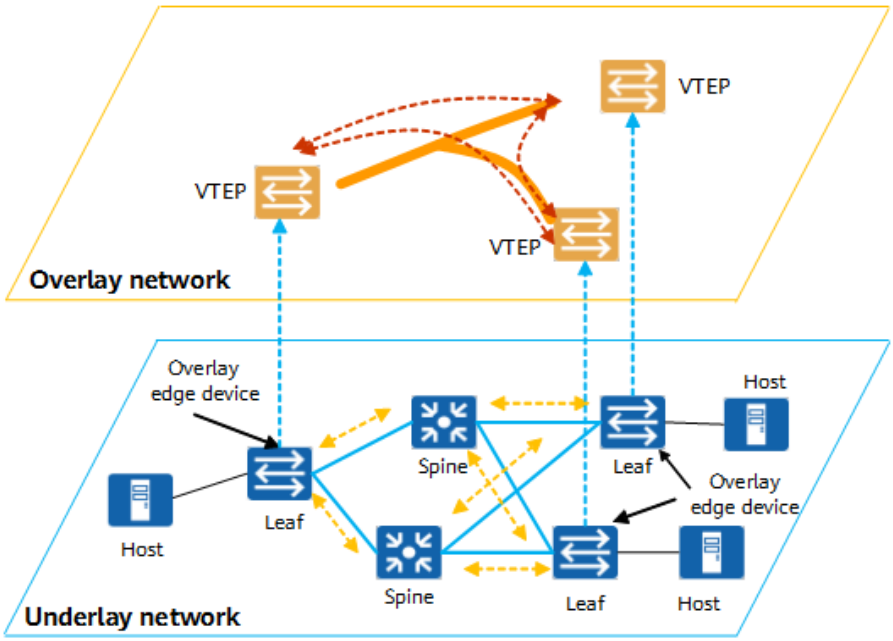
- Přenos provozu není závislý na konkrétních linkách. Logické sítě využívají tunelovacích technologií k flexibilnímu výběru z různých fyzických linek a používají více metod k zajištění stabilního přenosu provozu.
- Pro logické sítě lze podle potřeby vytvářet různé virtuální topologie, aniž by bylo nutné měnit původní síť.
- K zajištění bezpečnosti soukromého provozu na internetu lze použít metody šifrování.
- Podporováno je rozdělení sítě na segmenty. Různé služby lze oddělit tak, aby se dosáhlo optimálního rozdělení síťových zdrojů.
- Je podporováno vícecestného předávání dat. V logických sítích může být provoz přenášen od zdroje k cíli více cestami než jednou, čímž se prakticky vytváří vyrovnávání zátěže a maximalizuje se tak využití dostupné šířky pásma jednotlivých linek.

4.3.1 Příklady logických sítí

Logické sítě se hojně využívají v řešeních sítí SD-WAN a sítí datových center. Topologie logické sítě se liší podle architektury fyzické sítě.

4.3.1.1 Logické sítě datových center

S postupným vývojem architektury datových center, většina datových center používá architekturu topologe typu spine-leaf pro výtváření logických sítí a technologii VXLAN pro realizaci propojených jednotlivých logických sítí. Datová komunikace je přenášena v logických sítích VXLAN, které jsou reálně odděleny od fyzických sítí. 4.4



Obrázek 4.4 Mapování datacentrové Huawei SD-WAN [38]

Koncové a páteřní uzly jsou plně propojeny, takže jsou nám k dispozici cesty ECMP, které zajišťují vysokou dostupnost sítě.

Koncové uzly fungují jako přístupové body, které dodávají různým síťovým zařízením připojení na fyzické síti k síti VXLAN. Koncové uzly jsou také edge zařízeními logické sítě a fungují jako koncové body tunelů VXLAN (VTEP).

Páteřní uzly jsou core uzly sítí datového centra. Zajišťují vysokorychlostní předávání IP komunikace a připojují se k jednotlivým koncovým uzlům prostřednictvím vysokorychlostních rozhraní.

4.3.1.2 Logické sítě SD-WAN

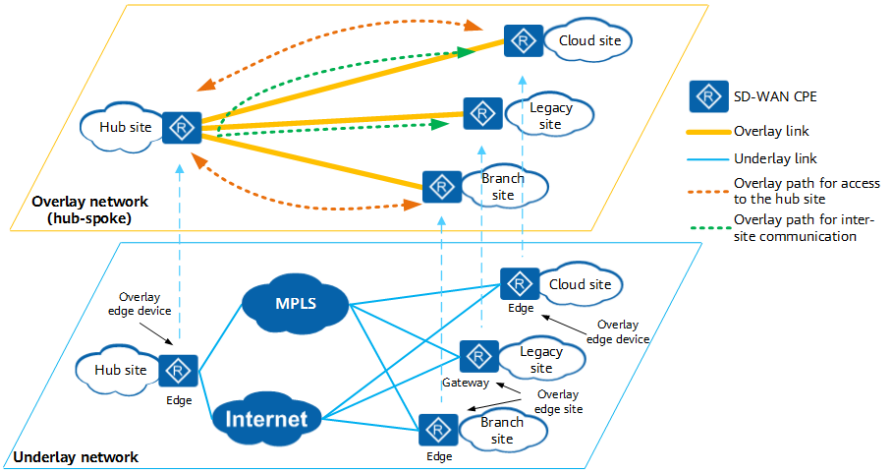
Fyzická síť SD-WAN je postavena na bázi sítě WAN a využívá hybridních spojů k realizaci propojení mezi ústředím, pobočkami a cloudovými lokalitami. Topologie logických sítí jsou sestaveny tak, aby splňovaly požadavky na propojení v různých scénářích. 4.5

Logická síť SD-WAN se skládá ze zákaznických zařízení (CPE), která se dělí na koncová zařízení a brány.

Edge: egress sítě SD-WAN.

Gateway: zažízení které připojuje sítě SD-WAN do ostatních sítí.

Na základě rozsahu podnikové sítě, počtu uzlů a požadavků na komunikaci mezi lokalitami lze vytvořit různé typy logických sítí.[38]



Obrázek 4.5 Mapování Huawei SD-WAN [38]

Sítě Hub-Spoke : Tato síť je vhodná pro podniky, které mají jedno nebo dvě datová centra. Pobočky přistupují ke službám nasazeným v centrále nebo v datových centrech prostřednictvím sítě WAN. Mezi pobočkami se přenáší malé množství provozu nebo pobočky mezi sebou nemusí vůbec komunikovat. Provoz mezi pobočkami prochází skrze ústředím nebo datová centra.

Síť Full-Mesh: Tato síť je použitelná pro malé podniky s malým počtem pracovišť nebo pro velké podniky, jejichž pobočky potřebují vzájemně spolupracovat. Spolupracující služby velkých podniků, například vysoko prioritní aplikace včetně VoIP a videokonferencí, mají vysoké požadavky na výkonnost sítě, jako je ztrátovost, zpoždění a jitter. Pro splnění požadavků takových služeb se doporučuje, aby pobočky mezi sebou komunikovaly přímo.

Hierarchické sítě: Tato síť se vyznačuje jasnou strukturou sítě a vynikající škálovatelností, a proto je použitelná pro podniky, které mají velký počet poboček, nebo pro nadnárodní podniky s pobočkami široce rozmístěnými v různých zemích či regionech.

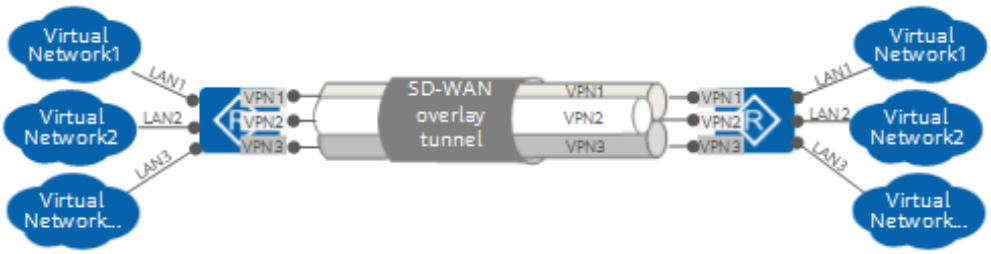
Sítě s více uzly: Tento druh sítě je použitelný pro podniky, které mají více datových center a v každém datovém centru nasazují servisní servery pro poskytování služeb pobočkám.

Sítě PoP: V případě, že operátoři nebo MSP poskytují podnikům přístupové služby SD-WAN, mohou mít některé podniky jak starší pobočky, tak i pobočky SD-WAN, které musí vzájemně komunikovat. V této síti lze nasadit bránu IWG (Interworking Gateway), která umožňuje komunikaci mezi lokalitami SD-WAN a staršími lokalitami MPLS VPN pro více podnikových nájemců.

4.3.2 Segmetace logických sítí

Pokud oddělení podniku nepotřebují být navzájem izolována, je zapotřebí pouze jedna síť VPN a všechny lokality se přidají do sítě VPN a vytvoří se překryvná topologie.

V mnoha případech je kvůli stále vyšším požadavkům na zabezpečení nutné rozdělit síť na více segmentů, aby bylo možné realizovat jemnou správu služeb a zvýšit tak bezpečnost. Služby uživatelů v různých odděleních musí být izolovány od ostatních. Řešení SD-WAN využívá sítě VPN k oddělení sítí a služeb uživatelů ve více odděleních v rámci jednoho tenanta. Každá VPN je nezávislá privátní síť na vrstvě 3 ISO/OSI. Více sítí VPN, včetně tunelů spojujících lokality a zařízení CPE v lokalitách, jsou od sebe logicky izolovány. 4.6



Obrázek 4.6 Segmentace topologie v rámci Huawei SD-WAN [39]

Chceme-li izolovat více oddělení podniku, musíte pro každé oddělení definovat virtuální síť. Každá virtuální síť má pak své odpovídající VPN tunely. Každá VPN má, na ostatních nezávislou, logickou topologii (buďto hub-spoke, full-mesh, partial-mesh nebo hierarchickou topologii). Nastavení v rámci sítě LAN, politiky provozu a politiky zabezpečení jednotlivých lokalit, kde je potřeba izolovat služby, se realizují pomocí VPN tunelů napříč WAN. Pro různé VPN sítě lze nakonfigurovat, aby měli různé politiky provozu a zabzpeěení.

Pokud oddělení podniku nepotřebují být navzájem izolovány, stačí na propojení jedna VPN a všechny lokality se přidají do této VPN sítě, realizující logickou síť takto. [39]

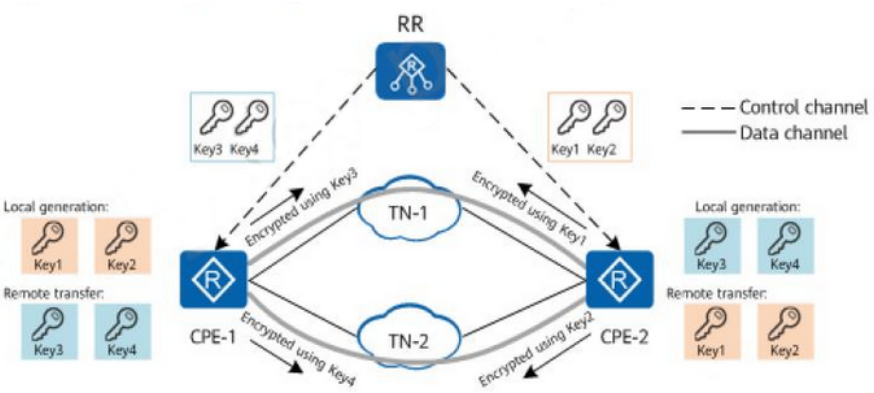
4.4 Vrstva prezentace služeb

Na této vrstvě provádějí správci sítě konfiguraci a zpracování koncových služeb SD-WAN. Společnost Huawei vyvinula ovládací panel, který je určen pro dopravce, poskytovatele spravovaných služeb a velké podnikové zákazníky (kteří pravděpodobně disponují početným personálem IT schopným samostatně spravovat síť SD-WAN). Prezentační vrstva služeb má otevřené rozhraní API, které integruje Huawei SD-WAN s portály třetích stran, aplikačními úložišti a jakýmkoli systémem podpory provozu (OSS) nebo systémem podpory podnikání (BSS). [37]

4.5 Vrstva řízení a orchestrace sítě

Agile Controller-Campus (AC-Campus) dodávané Huawei SD-WAN zpracovává většinu funkcí ve vrstvě orchestrace a řízení sítě. Jejím prostřednictvím jsou tradiční zařízení CPE, univerzální zařízení CPE (uCPE) a virtuální zařízení CPE (vCPE) jednotně spravována prostřednictvím jižního kanálu HTTP 2.0 a NETCONF.4.8

Standardizované a levnější uCPE může dodavatel zakoupit od výrobců originálního vybavení (OEM). Zařízení uCPE je také schopno provozovat libovolnou síťovou funkci, kterou schválí prodejce, což zákazníkovi nabízí větší výběr. Ne každý dodavatel SD-WAN používá uCPE. SD-WAN společnosti VMware a SD-WAN společnosti Nuage jsou příklady SD-WAN s uCPE ve své infrastruktuře. Prodejci mohou používat proprietární zařízení uCPE, aby měli větší kontrolu nebo si účtovali vyšší cenu. 4.7

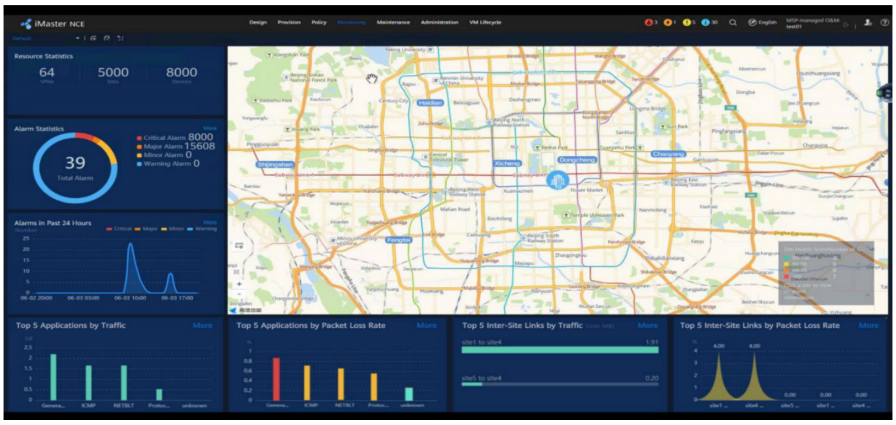


Obrázek 4.7 Řídící vrstva Huawei SD-WAN [11]

AC-Campus má také kontrolu nad virtuálními logickými sítěmi a může automaticky poskytovat služby. Pro komunikaci s prezentační vrstvou služeb používá kontrolér Nortbound RESTful rozhraní.

V případě služeb s přidanou hodnotou (VAS) třetích stran v uCPE lze do AC-Campus integrovat systém správy prvků (EMS) třetí strany.

V této vrstvě se nachází také virtuální směrový odražeč (vRR), který vytváří trasu VPN a informace o tunelování. To se provádí na vyžádání a slouží k bezpečnému připojení CPE kdekoli v síti SD-WAN na základě zásad topologie VPN definovaných správci sítě.



Obrázek 4.8 iMaster NCE GUI Huawei SD-WAN [11]

4.6 Vrstva síťového připojení

Vrstva síťového připojení umožňuje různým typům zařízení CPE fungovat jako brány. Tradiční zařízení CPE nebo uCPE mohou být použita ústředím a pobočkami organizace k připojení k síti SD-WAN. Jinými slovy, zařízení CPE nebo uCPE propojuje lokality organizace s jinými lokalitami v síti organizace. Zařízení vCPE se používá jako brána do veřejných a soukromých cloudů. U bran poboček lze při připojení k centrále, datovému centru nebo cloudům kombinovat linky MPLS, internet a LTE různými způsoby. [37]

Kapitola 5

Přístup Mikrotiku/FOSS k SD-WAN

Tato kapitola se zaměřuje na zařízení schopna používat řešení SDN pod zastřešením řešení OpenFlow, otevřeném protokolu pro realizaci distribuce a administrace toků, tabulek toků a pravidel reakcí na toky. V této kapitole budou podrobněji popsány jednotlivá zařízení a kontroléry realizující OpenFlow protokol.

Následující text se bude zaměřovat protokolu OpenFlow a několika konkrétním implementací kontroléru a zařízení. Jedná se opět o třívrstvý model, s tím že nejvižší vrstva je čistě software. Kontroléry jsou též čistě softwarová záležitost s tím, že se jakýkoliv kontrolér nasadit na jakoukoliv podporovanou verzi Linuxu, proto se o kontrolérech budu bavit jako o čstě softwarové záležitosti pro operační systém Linux, ačkoliv může být podpora rozšířena ja jíné platformy. Většina kontrolérů zde zmínených jsou postaveny na Javě.

5.1 OpenFlow

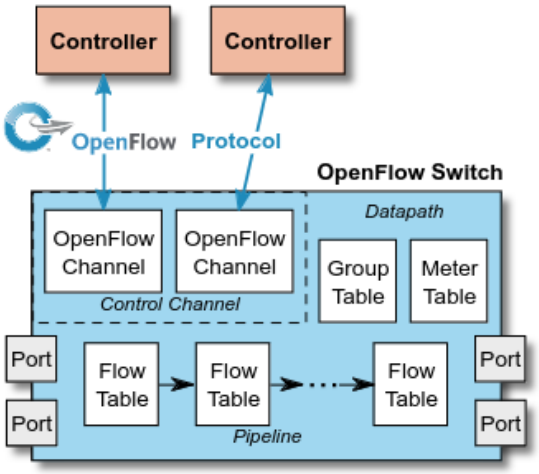
V dnešním rychle se rozvíjejícím digitálním prostředí se správa sítě a řízení toku dat staly pro podniky všech velikostí zásadními. OpenFlow je jednou z technologií, která si získala značnou pozornost a mění způsob správy sítí. V tomto příspěvku na blogu se budeme zabývat konceptem OpenFlow, jeho výhodami a důsledky pro řízení sítí.

OpenFlow je otevřený standardní komunikační protokol, který v síťové architektuře odděluje řídicí a datovou vrstvu. Umožňuje správcům sítě přímou kontrolu nad chováním síťových zařízení, jako jsou přepínače a směrovače, pomocí centralizovaného kontroléry.

Tradiční síťové architektury se řídí uzavřeným modelem, kdy síťová zařízení činí nezávislá rozhodnutí o předávání paketů. Naproti tomu OpenFlow zavádí centralizovanou kontrolní vrstvu, která poskytuje globální pohled na síť a umožňuje správcům definovat síťové zásady a pravidla z centrálního místa.

OpenFlow funguje na základě vytvoření zabezpečeného kanálu mezi centralizovaným kontrolérem a síťovými přepínači. Kontrolér je zodpovědný za správu tabulek toků v přepínačích a definuje, jak má být provoz předáván na základě předem definovaných pravidel a zásad. Toto oddělení řídicí a datové vrstvy umožňuje dynamickou správu sítě a usnadňuje implementaci inovativních síťových protokolů.

Jednou z klíčových výhod technologie OpenFlow je její schopnost zjednodušit správu sítě. Díky centralizaci řízení mohou správci snadno konfigurovat a spravovat celou síť z jednoho místa. To snižuje složitost a zvyšuje škálovatelnost síťové infrastruktury. OpenFlow navíc umožňuje



Obrázek 5.1 Architektura OpenFlow přepínače. [14]

programovatelnost sítě, což umožňuje vývoj vlastních síťových aplikací a služeb přizpůsobených konkrétním požadavkům.

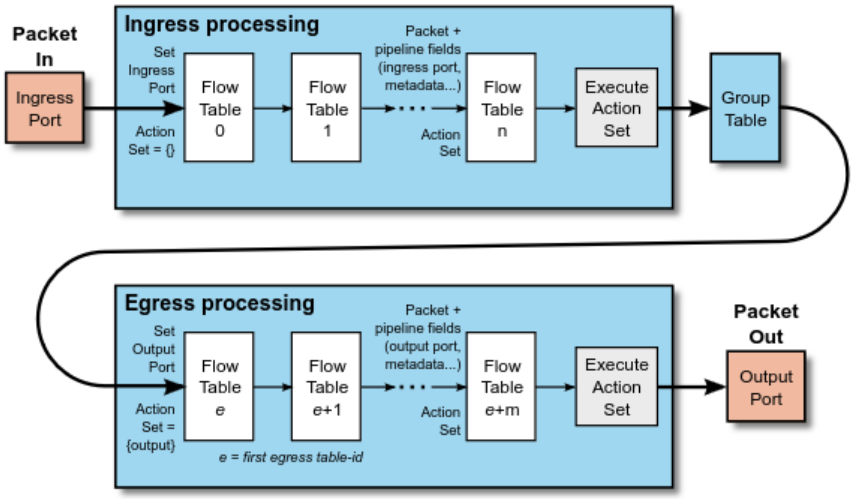
OpenFlow hraje klíčovou roli v oblasti virtualizace sítí, protože umožňuje vytvářet a spravovat virtuální sítě nad fyzickou infrastrukturou. Díky abstrakci základní sítě umožňuje OpenFlow organizacím optimalizovat využití zdrojů, zlepšit zabezpečení a zvýšit výkon sítě. Otevírá dveře dynamickému poskytování, izolaci a efektivnímu využívání síťových zdrojů. [40]

5.1.1 Jak OpenFlow funguje

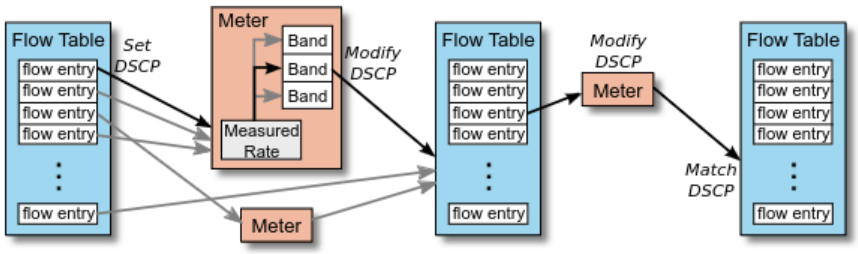
OpenFlow umožňuje síťovým kontrolérům určovat cestu síťových paketů v síti přepínačů. Mezi přepínači a kontroléry je rozdíl. Díky oddělenému řízení a předávání může být řízení provozu sofistikovanější než seznamy řízení přístupu (ACL) a směrovací protokoly. Protokol OpenFlow umožňuje vzdáleně spravovat přepínače různých výrobců, často s proprietárními rozhraními a skriptovacími jazyky. Softwarově definované sítě (SDN) jsou podle názoru jejich vynálezců umožněny právě pomocí OpenFlow.

Díky OpenFlow mohou přepínače vrstvy 3 vzdáleně přidávat, upravovat a odebírat pravidla a akce pro porovnávání paketů. Tímto způsobem mohou být rozhodnutí o směrování prováděna periodicky nebo ad hoc řídicí jednotkou a převedena na pravidla a akce s nastavitelnou dobou životnosti, které jsou poté nasazeny do tabulky toků přepínače, kde jsou pakety předávány rychlostí vedení po dobu trvání pravidla. Pokud přepínač nemůže pakety porovnat, mohou být odeslány do kontroléru. Řídicí jednotka může upravit stávající pravidla tabulky toků nebo nasadit nová pravidla, aby zabránila strukturálnímu toku provozu. Může dokonce sám předávat provoz, pokud je přepínači nařízeno předávat pakety, a ne pouze jejich hlavičky.

OpenFlow používá zabezpečení transportní vrstvy (TLS) nad protokolem TCP (Transmission Control Protocol). Přepínače, které se chtějí připojit, by měly naslouchat na portu TCP 6653. V dřívějších verzích OpenFlow se neoficiálně používal port 6633. Protokol se používá především mezi přepínači a kontroléry.



Obrázek 5.2 OpenFlow pipeline. [14]



Obrázek 5.3 Pohled na interake jednotlivých tabulek toků mezi sebou OpenFlow přepínače. [14]

5.1.2 Výhody OpenFlow

Softwarově definované sítě mění architekturu kontrolní a datové vrstvy.

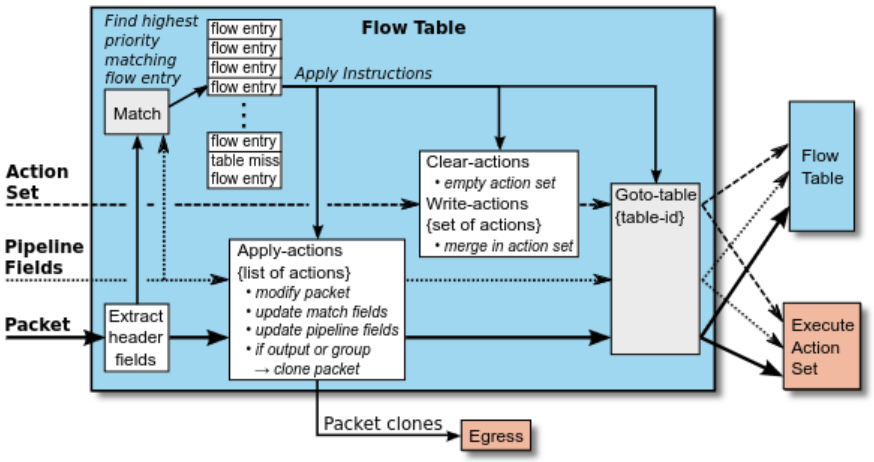
Koncept SDN tyto dvě vrstvy odděluje, tj. kontrolní a datová vrstva jsou odděleny. To umožňuje síťovým zařízením v přenosové cestě soustředit se výhradně na předávání paketů. Síť mimo pásmo používá k nastavení zásad a řízení samostatný kontrolér (orchestrační systém). Proto má vrstva předávání správné informace pro efektivní předávání paketů.

Kromě toho umožňuje přesunout kontrolní vrstvu sítě do centralizovaného kontroléry na serveru namísto toho, aby se nacházela ve stejné skříni provádějící předávání. Přesunutí inteligence („kontrolní vrstvy“) síťových zařízení datové vrstvy do kontroléru umožňuje společnostem používat v cestě předávání levný, komoditní hardware. Významnou výhodou je, že SDN odděluje datovou a kontrolní vrstvu, což umožňuje nové případy použití.

Centralizovaná výpočetní a kontrolní vrstva dává větší smysl než centralizovaná kontrolní vrstva.

Kontrolér udržuje přehled o celé síti a komunikuje pomocí Openflow (nebo v některých případech BGP s BGP SDN) s různými typy síťových boxů s podporou OpenFlow. Část datové cesty zůstává na přepínači, například most OVS, zatímco rozhodování na vysoké úrovni se přesouvá na samostatný kontrolér. Datová cesta představuje čistou abstrakci tabulky toků a každá položka tabulky toků obsahuje sadu polí pro porovnání paketů, což vede ke konkrétním akcím (drop, redirect, send-out-port).

Když přepínač OpenFlow obdrží paket, který nikdy předtím neviděl a nemá odpovídající záznam toku, odešle paket ke zpracování kontroléru. Kontrolér pak rozhodne, co s paketem udělá.



Obrázek 5.4 Zpracování komunikace vůči tabulce toků. [14]

Nad tímto kontrolérem pak mohou být vyvíjeny aplikace, které provádějí čištění zabezpečení, vyrovnávání zátěže, řízení provozu nebo přizpůsobené předávání paketů. Centralizovaný pohled na síť zjednodušuje problémy, které bylo obtížné překonat s tradičními protokoly kontrolní vrstvy.

Jediný kontrolér by mohl potenciálně spravovat všechny přepínače podporující OpenFlow. Namísto individuální konfigurace každého přepínače může kontrolér předávat zásady více přepínačům současně - přesvědčivý příklad virtualizace mnoho k jednomu.

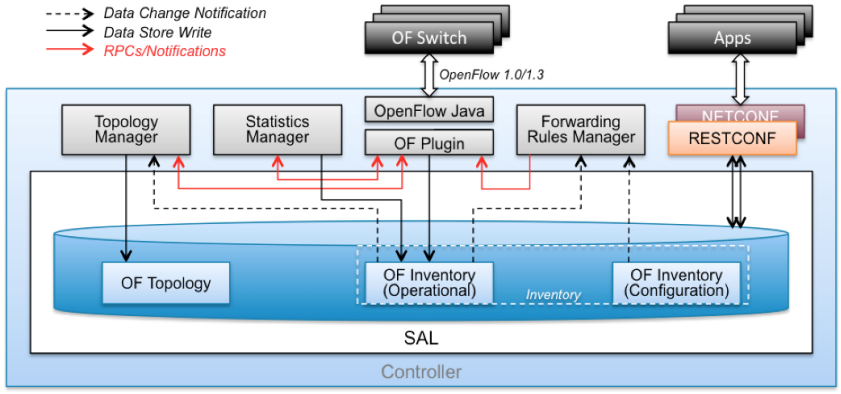
Nyní, když SDN odděluje datovou a kontrolní vrstvy, používá operátor centralizovaný kontrolér k výběru správných informací o předávání pro jednotlivé toky. To umožňuje lepší vyvažování zátěže a oddělení provozu v datové vrstvě. Kromě toho není třeba vynucovat oddělení provozu na základě VLAN, protože kontrolér by měl sadu zásad a pravidel, které by umožňovaly předávat provoz z jedné „VLAN“ pouze jiným zařízením v rámci téže „VLAN“.

5.2 Referenční přepínač OpenFlow

Protokol a rozhraní OpenFlow umožňují přístup k přepínačům OpenFlow jako k základním prvkům předávání. Architektura SDN založená na tocích, jako je OpenFlow, zjednodušuje přepínací hardware. Přesto může vyžadovat další tabulky pro předávání, vyrovnávací paměť a statistické čítače, které je obtížné implementovat v tradičních přepínačích s integrovanými obvody přizpůsobenými konkrétním aplikacím.

V síti OpenFlow existují dva typy přepínačů: hybridy (které umožňují OpenFlow) a póry (které podporují pouze OpenFlow). OpenFlow podporují hybridní přepínače a tradiční protokoly (L2/L3). Přepínače OpenFlow se při rozhodování o předávání dat spoléhají výhradně na kontrolér a nemají starší funkce ani vestavěné řízení.

Hybridní přepínače tvoří většinu přepínačů, které jsou v současné době na trhu k dispozici. Toto spojení musí zůstat aktivní a zabezpečené, protože přepínače OpenFlow jsou ovládány přes otevřené rozhraní (prostřednictvím relace TLS založené na protokolu TCP). OpenFlow je protokol pro zasílání zpráv, který definuje komunikaci mezi přepínači OpenFlow a kontroléři, což lze považovat za implementaci interakce mezi kontroléry a přepínači na bázi SDN.



Obrázek 5.5 Pohled na logiku OpenFlow přepínače. [14]

5.2.1 Mikrotik

MikroTik je světově rozšířená firma, která svým zákazníkům poskytuje síťové prvky postavené na svém vlastním operačním systému Router OS. Podílí se tak na vývoji vlastního software i hardware. MikroTik taktéž poskytuje certifikované kurzy pod svou značkou MikroTik Academy. Prvky MikroTik nejsou moc využívané pro lokální sítě, ale spíš pro páteřní spoje mezi budovami. Konfigurační prostředí Router OS je velmi intuitivní a rychlé. Nastavení routeru lze udělat během pár minut. Tato firma vyvíjí všechna síťová řešení od malých switchů až po vysokorychlostní routery a vysoko výkonné antény. Původně MikroTik používal specifická zařízení tzv. routerboard, což jsou základní desky, které lze rozšířit o další fyzické moduly jako antény, paměti, sloty a jiné porty. Dnes již tato firma každému svému výrobku říká routerboard.

Mikrotik momentálně implementuje OpenFlow jako switch, s tím že není známo jako a jestli reaguje na pravidla v tabulce toků.

Kapitola 6

Nástroje na analýzu toku dat

Pro práci byly zvoleny nástroje pro analýzu datových toků jako jsou netflow analyzéry a wireshark. Na základě pozorování chování specifických linek byly byly vyvozeny závěry o funkčnosti prokolu, jako jsou počty zpráv, typy paketů a podobně. Netflow analyzéry, dokáží pěkně vyzualizovat průběh komunikace, pro učelyanalýzy však nebyl potřeba a bude v prázi zmíněn pouze okrajově.

6.0.1 Wireshark

Wireshark je analyzátor síťového toku a konkrétních paketů. Síťový analyzátor zobrazující zachycenou síťovou komunikaci včetně obsahů jednotlivých paketů[1] v co nejpodrobnější podobě.

Analyzátor síťových paketů si můžete představit jako měřicí zařízení pro zkoumání toho, co se děje uvnitř síťového kabelu, podobně jako elektrikář používá voltmetr pro zkoumání toho, co se děje uvnitř elektrického kabelu (ale samozřejmě na vyšší úrovni).

V minulosti byly tyto nástroje buď velmi drahé, nebo proprietární, případně obojí. To se však s příchodem nástroje Wireshark změnilo. Wireshark je k dispozici zdarma, má otevřený zdrojový kód a je jedním z nejlepších analyzátorů paketů, které jsou dnes k dispozici.[41]

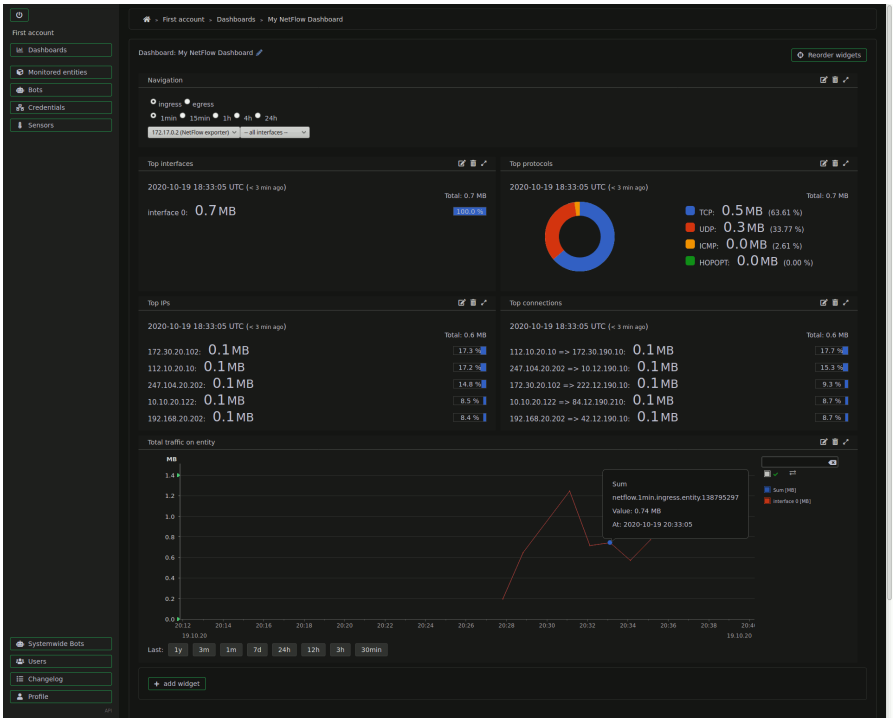
Wireshark poskytuje následující vlastnosti uživately:

- Zachytávání paketů z ruzných typů rozhraní
- Importování souborů pro zachytávání paketů od různých výrobů.
- Exportování zachycené komunikace do formátů různých výrobců.
- Preparátor síťových protokolů, jako jsou audio dekodéry a podobně.
- Je to software s otevřeným zdrojovým kódem.
- Wireshark však není IPS, IDS ani manipulátor síťové komunikace.

[1] Pokud data byla šífrována tak vidíme pouze jejich šifrovanou podobu

6.0.2 Netflow analyzer

Použitým nástrojem pro analýzu dat byl vybrán, snad jediný, otevřený nástroj pro zběr dat z SNMP, ICMP a NETFLOW Grafolean. Grafolean slouži k vyzualizaci nasbíraných dat v čase, na něž pak vytváří jednoduché statistiky. Grafolean má grafické prostředí jehož ukázka je zobrazena na následujícím obrázku 6.1



Obrázek 6.1 Grafolea.[42]

Dalším použítým nástrojem byl zdarma nabýzená komunitní edice netflow kolektoru ElastiFlow[43]. Data pouze sbírá a proto jeho výldeky jsou potřeba zobrazit. K tomu právě byl použit Wireshark.

Kapitola 7

Příprava implementací

Tato kapitola popisuje simulovaná zařízení SD-WAN a jejich požadavky na úspěšnou konfiguraci, proces instalace simulátoru EVE-NG a GNS3, softwarové licencování a jak propojit jednotlivé laboratoře s vnějším světem. Bude zde představeno několik topologií pro SD-WAN sloužící jako základ pro další kapitoly.

7.1 Požadavky laboratoře

Připojených SD-WAN laboratoří v této práci je 10, jedna Cisco SD-WAN jako proof of concept a druhá rozsáhlejší implementace, na které se testují různé funcionality, tak tomu je i pro Mikrotik/OpenFlow. Bohužel pro účely diplomové práce se nepodařilo zařídit SD-WAN řešení od Huawei, proto zde je pouze pozkoumána možnost připojení Huawei AR1000V do stávající sítě OpenFlow. Každá laboratoř je implementována v prostředí EVE-NG a následně v prostředí GNS3.

7.1.1 Cisco SD-WAN laboratoř

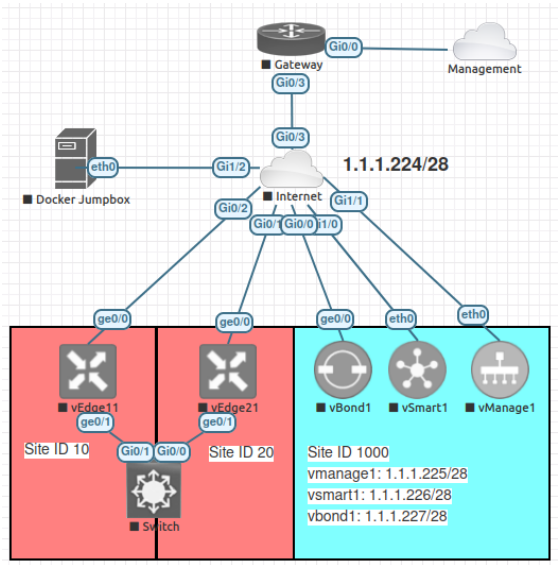
Zde je představena jednoduchá topologie jaká může být pro implementaci SD-WAN. Obrázky7.1 a 7.2. Cílem této laboratoře je vyzkoušet si nasazení kontrolérů a manuální začlenění WAN-edge zařízení. Dále je zde důraz na seznámení s grafickým prostředí vManage. Topologie je nejmenší možná pro účely otestování i na horších hypervizorech než jaký mi byl dostupný. Proces, který bude popsán v následujících kapitolách bude aplikovatelný i na tuto topologii, jelikož se jedná o redukovanou variantu rozsáhlé laboratoře. Následující tabulka 7.1 obsahuje zařízení, a jejich hardwarové nároky, použité v dané topologii.

7.1.2 Rozsáhlá Cisco SD-WAN laboratoř

V daší kapitole se provede postup krok za krokem postupné implementace jednotlivých dílčích funkcionalit Cisco SD-WAN, zde se naváže na postup bakalářské práce[1] a rozšíří postup o analýzu toku dat. Obrázek 7.3 představuje zkoumanou topologii. Veškerý přehled použitých zařízení této topologie je znázorněn v tabulce 7.2. Veškeré komponenty laboratoře však nepořebují běžet najednou ve stejný okamžik. Vícero vSmart a vBond zařízení jsou čistě daný do topologie jen jako názorná ukázka. Jak už bylo představeno víše, pouze jedno zařízení z každé vrstvy Cisco SD-WAN je potřeba pro účely zprovoznění SD-WAN. Pokud bychom chtěli takto rozjet celou topologii najednou, tak tím zatížíme procesor na maximum, a zpomalení bude mnohonásobně

Tabulka 7.1 Tabulka použitých zařízeni v Cisco SD-WAN topologii

Zařízení	Počet	Název obrazu EVE-NG/GNS3	RAM GiB	vCPU	Ethernet
vManage	1	vtmgmt-20.5.1	16/32	4/6	2/4
vSmart	1	vtsmart-20.5.1	2/4	2	2/4
vBond	1	vtbond-20.5.1	1/2	1	2/5
vEdge	2	vtedge-20.5.1	1/2	1	4/6
IOSv switch	1	viosl2-adverterprisek9-m	0.5	1	8/16
IOSv router	1	vios-adverterprisek9-m	0.5	1	4
Docker Jumpbox	1	-	4	2	1
Celkem			26/47	15	



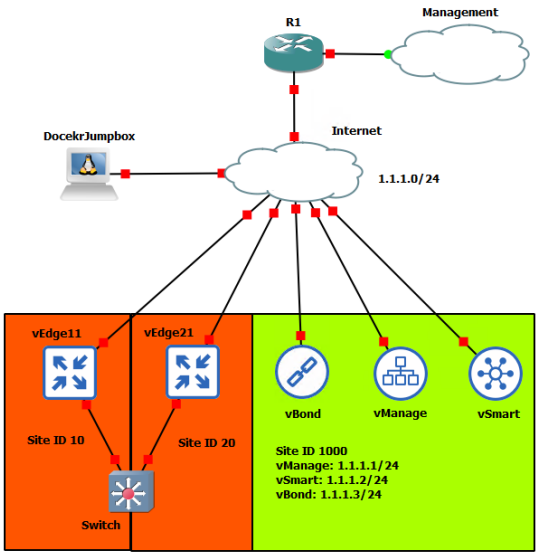
Obrázek 7.1 Topologie Cisco SD-WAN v EVE-NG

větší kvůli přepínání kontextu. Jednotlivé tabulky pro malou topologii7.1 a rozsáhlejší topologii7.2 pak ukazují jednotlivé teoretické nároky na laborku, za předpokladu že zařízení nejsou nikdy ve stavu idle a paměť se alokuje ne virtuální.

Ačkoliv je vidět na obrázcích rozsáhlejší implementace topologie v EVE-NG7.3 a GNS37.4, že nejsou zcela totožné, tak v laboratořích se to projeví pouze jako změna konkrétních parametů. Proto bude vždy popsán pouze jeden způsob implementace, protože parametrické změny nejsou zásadní pro chod laboratoří.

7.1.3 Mikrotik/OpenFlow SDN laboratoř

Tato laboratoř se zaměřuje na použití OpenFlow kontrolérů postavené nad jednoduchou topologie jaká může být pro implementaci SDN. Obrázky 7.5 a 7.6. Cílem této laboratoře je vyzkoušet si nasazení kontrolérů a manuální začlenění jednotlivých zařízení. Dále je zde důraz na seznámení s dostupnými grafickými prostředí OpenFlow kontrolérů. Topologie je nejmenší možná pro účely otestování i na horších hypervizorech než jaký mi byl dostupný. Proces, který bude popsán v následujících kapitolách bude aplikovatelný i na tuto topologii, jelikož se jedná o redukovanou variantu rozsáhlé laboratoře. Následující tabulka 7.3 obsahuje zařízení, a jejich hardwarové nároky, použité v dané topologii.



Obrázek 7.2 Topologie Cisco SD-WAN v GNS3

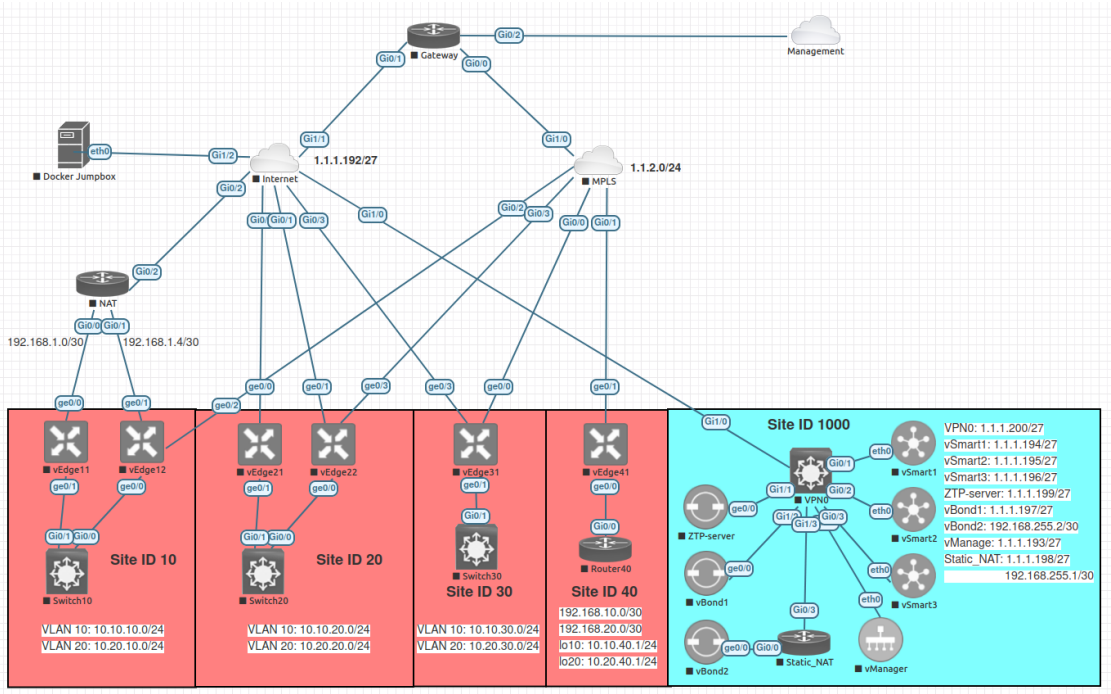
Tabulka 7.2 Tabulka použitých zařízeni v Cisco SD-WAN topologii

Zařízení	Počet	Název obrazu EVE-NG/GNS3	RAM GiB	vCPU	Ethernet
vManage	1	vtmgmt-20.5.1	16/32	4/6	2/4
vSmart	3	vtsmart-20.5.1	2/4	2	2/4
vBond	3	vtbond-20.5.1	1/2	1	2/5
vEdge	6	vtedge-20.5.1	1/2	1	4/6
IOSv switch	6	viosl2-adverterprisek9-m	0.5	1	8/16
IOSv router	4	vios-adverterprisek9-m	0.5	1	4
Docker Jumpbox	1	-	4	2	1
Celkem			40/71	33	

7.1.4 Řešení laboratoře OpenFlow SDN s více výrobci

Tato laboratoř se zaměřuje na síť s použitím zařízení více výrobců,

V závěrečné kapitole se provede postup krok za krokem postupné implementace jednotlivých dílčích funkcionalit OpenFlow SDN, kde hlavní důraz je na možnost použití jednoho OpenFlow kontroléru pro správu zařízení různých výrobcu. Obrázek 7.7 představuje zkoumanou topologii. Veškerý přehled použitých zařízení této topologie je znázorněn v tabulce 7.4. Veškeré komponenty laboratoře však nepořebují běžet najednou ve stejném okamžiku. Vícero použitých zařízení v síti jsou čistě daný do topologie jen jako názorná ukázka, a není dopuročené spoštět všchny i přesto, že většina zařízení běží jako kontejnerová aplikace. Jak už bylo představeno víše, pouze jedno zařízení z každé vrstvy OpenFlow SDN je potřeba pro účely zprovoznění SDN, čili stačí dvě. Pokud bychom chtěli takto rozjet celou topologii najednou, tak tím zřejmě zaťížíme procesor na maximum, a zpomalení bude mnohonásobně větší kvůli přepínání kontextu. Jednotlivé tabulky pro malou topologii7.3 a rozsáhlejší topologii7.2 pak ukazují jednotlivé teoretické nároky na laborku, za předpokladu že zařízení nejsou nikdy ve stavu idle a paměť se alokuje ne virtuální.



Obrázek 7.3 Rozsáhlejší topologie Cisco SD-WAN v EVE-NG

Tabulka 7.3 Tabulka použitých zařízeni v OpenFlow SDN topologii

Zařízení	Počet	Název obrazu EVE-NG/GNS3	RAM GiB	vCPU	Ethernet
generic openflow switch	3	-	1	1	4/8
Docker client	3	-	1	1	1
Cisco OFM	1	Linux Ubuntu 18.04	4	4	1
IOSv switch	1	viosl2-adverterprisek9-m	0.5	1	8/16
Docker Jumpbox	1	-	1	1	1
Celkem			8.5	23/35	

7.2 Potřebné licence

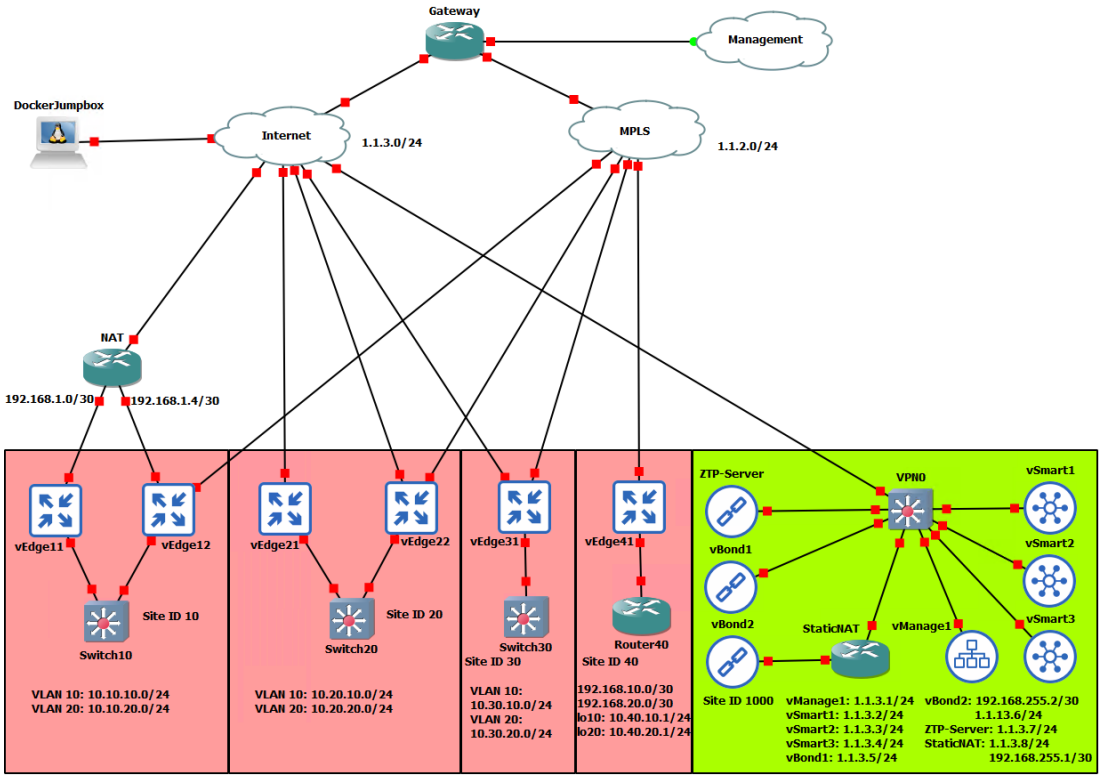
Hlavní motivací jak řešit implementaci SD-WAN je nutnost mít buďto licence zdarma nebo pokud možno využít prostředky, které nevyžadují licenci. Proto vše bylo implementováno na nástrojích dostupných zdarma, se softwarem dostupným zdarma. Software, který obecně zdarma není, byl získán legální cestou akademické spolupráce.

EVE-NG je zdarma dostupný nástroj pro simulaci síťových prvků a topologií, v podobě komunitní edice. Licence pro profesionální verzi byla dodána fakultou.

GNS3 je zdarma dostupný software pro smulaci síťových prvků a topologií s otevřeným zdrojovým kódem pod licencí GPLv3. Jeho přívětivost byla oceněna v rámci diplomové práce.

Obrazy zařízení , které byly dodány byly převážně ve formátech OVF, VMDK a QCOW2 jak jednotlivá zařízení, tak obrazy EVE-NG a GNS3-VM. V této práce jsou všechny virtualizéry virtualizovány.

VMWare ESXi byl dodán vedoucím práce Ing. Alexem Mouchou, Phd. Toto zařízení obstarává prostor pro veškeré simulace.



Obrázek 7.4 Rozsáhlejší topologie Cisco SD-WAN v GNS3

Klient telnetu na Windows OS je implicitně vybrán nástroj PuTTy, protože je součástí instalace klienta GNS3. Tento nástroj je zdarma dostupný. Linux sám o sobě velmi často s klientem telnetu přichází z výroby (podle distribuce). Co se týče MacOS, tak ti se taktéž mohou spokojit s PuTTy.

Wireshark je zdarma nástroj pro zachytávání paketů pomocí knihovny libpcap. Taktéž je součástí instalace klienta GNS3.

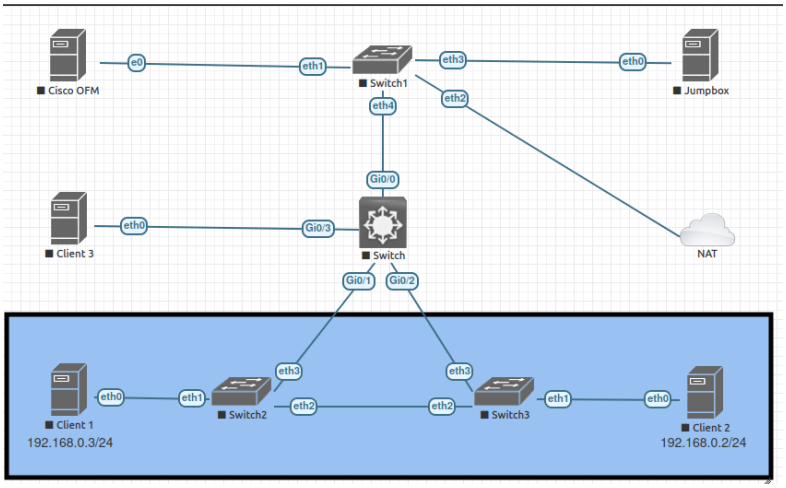
WAN-edge seriový soubor je zdarma pro všechny co mají Cisco Smart účet. Tyto soubory dle mých instrukcí byly dodány vedoucím práce ve spolupráci s cisco akademií. Tyto soubory jsou nezbytné pro začlenění virtuálních zařízení do SD-WAN.

7.3 Instalace EVE-NG

Protože původní instalace EVE-NG byla nestabilní, tak zde popíši jak probíhala nová instalace EVE-NG. Nejprve byla stažena OVF varianta EVE-NG[1], vhodná pro přímé nasazení virtuálního stroje. EVE-NG pak bylo dodáno 96GiB hlavní paměti, 16 virtuálních jader procesoru (vCPU), 600GiB vedlejší paměti pro uspokojení nároků virtuálních stroju a jejich obrazů na EVE-NG. Samozřejmě tento způsob instalace pomocí OVF je možný pouze při nasazení na hypervizor, pokud bycho chtěli přímo dedikovat hardware EVE-NG, potřebovali bycho instalační ISO obraz, jež je přímo dostupný z oficiálních stránek EVE-NG, spolu s podrobnými video návody jak EVE-NG zprovoznit. Minimální požadavky na virtuální serverový systém jsou: [44]

CPU: Alespoň 2x CPU Intel E5-2650v4 (48 logických procesorů) nebo lepší.

[1] Ke dni psaní této části diplomové práce již je dostupná pouze varianta ISO



Obrázek 7.5 Topologie OpenFlow SDN v EVE-NG

Tabulka 7.4 Tabulka použitých zařízeni v OpenFlow SDN topologii

Zařízení	Počet	Název obrazu EVE-NG/GNS3	RAM GiB	vCPU	Ethernet
generic openflow switch	0/2	-	1	1	4/8
Docker client	4	-	1	1	1
Cisco OFM	1	Linux Ubuntu 18.04	4	4	1
Mikrotik RouterOS	2	vtedge-20.5.1	1	1	4/6
IOSv switch	4/2	viosl2-adverterprisek9-m	0.5	1	8/16
IOSv router	1	vios-adverterprisek9-m	0.5	1	4
Docker Jumpbox	1	-	4	2	1
Celkem			40/71	33	

RAM: 128 Gb [2]

Uložiště: 2 Tb

Síť: LAN Ethernet.

Jedinou důležitou informací je, že během instalace musí být nastavena IP adresa. Webové rozhraní EVE-NG pak právě bude dostupné přes tuto IP adresu. Počáteční přihlašovací údaje jsou admin a heslo eve. Pokud všach chceme přistoupit přes vzdálenou konzoli, tak uživatel je root.

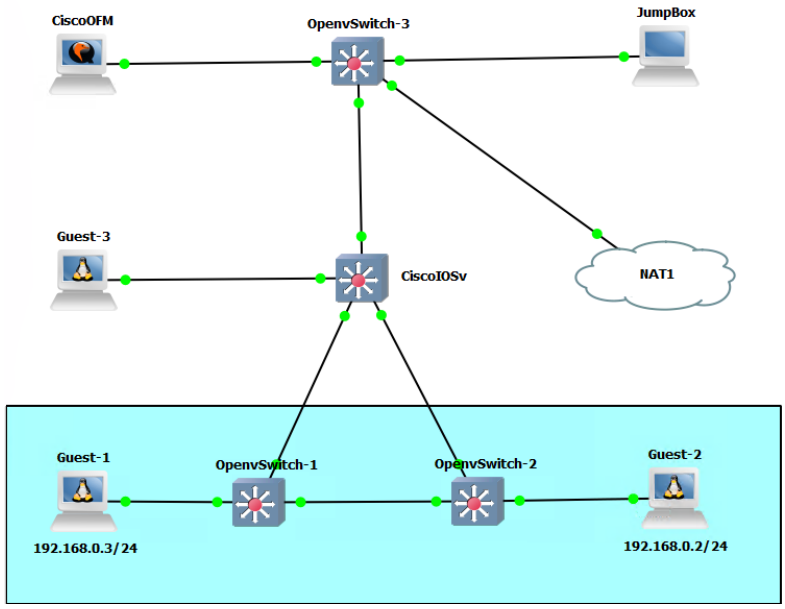
7.3.1 Dodatečný software

Veškerý dodatečný software pro verzi pro neni potřebný, protože veškery potrebný software jako je Wreshark a Telnet či VNC je emulován přímo v grafickém prostředí. Proto tato sekce bude obsáhlejší v případě GNS3.

7.3.2 Obrazy zařízení

Jako každý simulační nástroj, který umožňuje používání proprietářního softwaru, tak ho neposkytuji, jelikož by porušovali licenční smlouvy. Proto při čité instalaci nejsou dostupné k používání

[2] Malé b v jednotkách je záměrně jelikož takto je to psáno napříč celou dokumentací EVE-NG



Obrázek 7.6 Topologie OpenFlow SDN v GNS3

žádné uzly. Uživatel tudíž musí nějakým způsobem si je obstarat sám. Jak nasadit software do EVE-NG je popsané na jejich stránkách, kde případech známích zařízení poskytuji návod na vytvoření patřičné složky v podadresáři adresáře /opt/unetlab/addons/. Nejjednodužší způsob jak dodat zařízení do EVE-NG je přes SFTP komunikaci, která zaručuje šifrovaný a bezpečný proud dat.

Přes vzdálenou konzoli, realizovanou programem SSH, jsem do EVE-NG vložil tyto obrazy zařízení následujícími způsoby:

vManage potřebuje obraz s operačním systémem a disk s alespoň 100GiB paměti. Protože tento disk je součástí bootovacího procesu, tak musí být přidán.

```
root@eve-pe:~# mkdir /opt/unetlab/addons/qemu/vtmgmt-20.5.1
root@eve-pe:~# mv viptela-vmanage-20.5.1-genericx86-64.qcow2 \
               /opt/unetlab/addons/qemu/vtmgmt-20.5.1/virtioa.qcow2
root@eve-pe:~# qemu-img create -f QCOW2 \
               /opt/unetlab/addons/qemu/vtmgmt-20.5.1/virtiob.qcow2 \
               100G
```

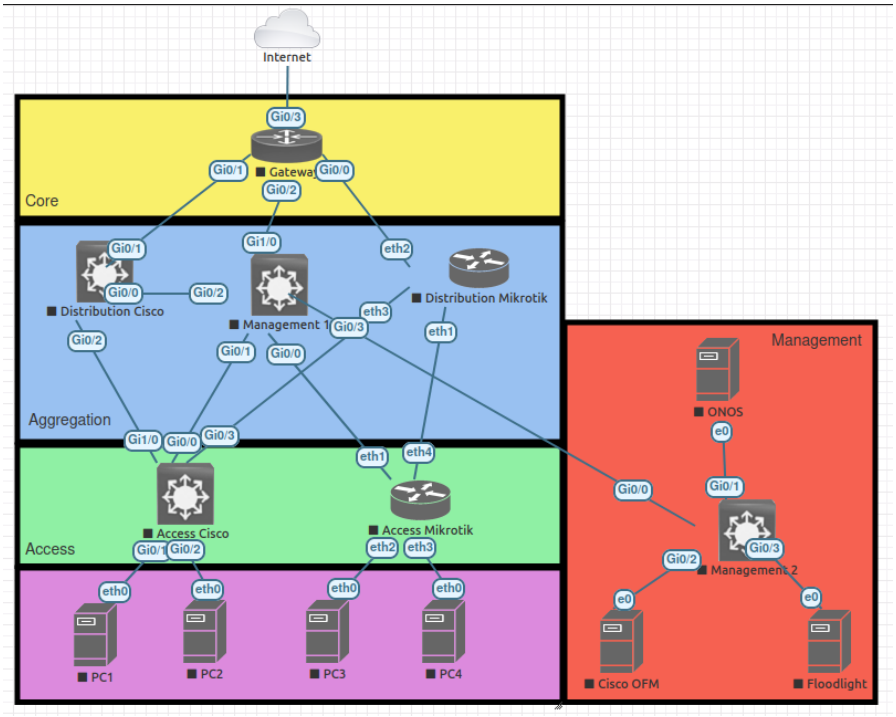
vSmart stačí pouze přejmenovat a vytvořit složku.

```
root@eve-pe:~# mkdir /opt/unetlab/addons/qemu/vtsmart-20.5.1
root@eve-pe:~# mv viptela-smart-20.5.1-genericx86-64.qcow2 \
               /opt/unetlab/addons/qemu/vtsmart-20.5.1/virtioa.qcow2
root@eve-pe:~# qemu-img create -f QCOW2 \
```

vEdge stačí pouze přejmenovat a vytvořit složku.

```
root@eve-pe:~# mkdir /opt/unetlab/addons/qemu/vtedge-20.5.1
root@eve-pe:~# mv viptela-edge-20.5.1-genericx86-64.qcow2 \
               /opt/unetlab/addons/qemu/vtbond-20.5.1/virtioa.qcow2
```

c8000v stačí pouze přejmenovat a vytvořit složku.



Obrázek 7.7 Multi-vendor topologie OpenFlow SDN v EVE-NG

```
root@eve-pe:~# mkdir /opt/unetlab/addons/qemu/c8000v-17.09.01
root@eve-pe:~# mv c8000v-17.09.01a.qcow2 \
                 /opt/unetlab/addons/qemu/c8000v-17.09.01/virtioa.qcow2
```

csr1000vng stačí pouze přejmenovat a vytvořit složku.

```
root@eve-pe:~# mkdir \
/opt/unetlab/addons/qemu/csr1000vng-ucmk9.16.12.3-sdwan
root@eve-pe:~# mv csr1000vng-ucmk9.16.12.3-sdwan.qcow2 \
/opt/unetlab/addons/qemu/csr1000vng-ucmk9.16.12.3-sdwan/virtioa.qcow2
```

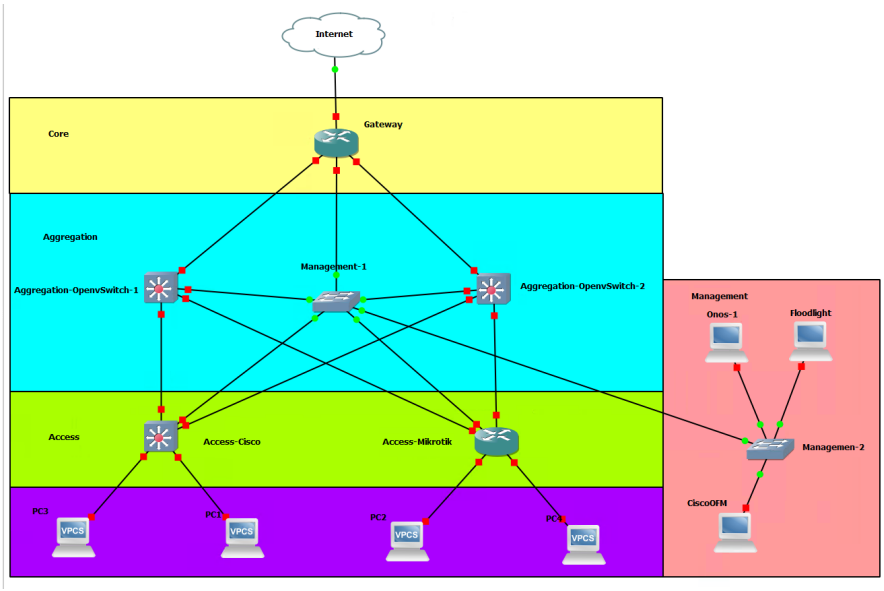
vBond stačí pouze přejmenovat a vytvořit složku.

```
root@eve-pe:~# mkdir /opt/unetlab/addons/qemu/vtbond-20.5.1
root@eve-pe:~# mv viptela-edge-20.5.1-genericx86-64.qcow2 \
                 /opt/unetlab/addons/qemu/vtbond-20.5.1/virtioa.qcow2
```

viosl2-adventerprisek9-m stačí pouze přejmenovat a vytvořit složku.

```
root@eve-pe:~# mkdir \
            /opt/unetlab/addons/qemu/viosl2-adventerprisek9-m.03.2017
root@eve-pe:~# mv viosl2-adventerprisek9-m.03.2017.qcow2 \
/opt/unetlab/addons/qemu/viosl2-adventerprisek9-m.03.2017/virtioa.qcow2
```

opnsense stačí pouze přesunout a vytvořit složku.



Obrázek 7.8 Multi-vendor topologie OpenFlow SDN v GNS3

```
root@eve-pe:~# mkdir /opt/unetlab/addons/qemu/opnsense-21.1
root@eve-pe:~# mv OPNsense-21.1-OpenSSL-vga-amd64.img \
                  /opt/unetlab/addons/qemu/opnsense-21.1/
```

vios-adventerprisek9-m vyžaduje kromě přesunu a vytvoření složky, také konverzi z VMDK na QCOW2.

```
root@eve-pe:~# mkdir \
     /opt/unetlab/addons/qemu/vios-adventerprisek9-m.SPA.156-1.T
root@eve-pe:~# qemu-img convert -cpf VMDK -O QCOW2 \
vios-adventerprisek9-m.SPA.156-1.T.vmdk \
/opt/unetlab/addons/qemu/vios-adventerprisek9-m.SPA.156-1.T/virtioa.qcow2
```

linux-onos zde bylo potřeba nejprve vytvořit virtuální stroj ve VirtualBoxu kde následně po zprovoznění virtuálního stroje, se překopíroval vmdk soubor do EVE-NG kde se konvertoval na QCOW2 formát a vytvořila se složka.

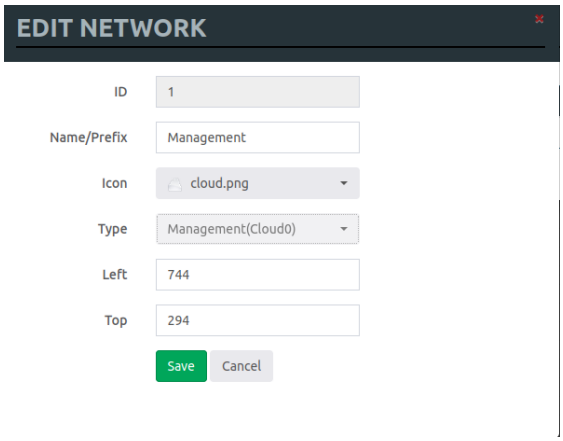
```
root@eve-pe:~# mkdir /opt/unetlab/addons/qemu/linux-onos-2.7.0
root@eve-pe:~# qemu-img convert -cpf VMDK -O QCOW2 linux-onos-2.7.0.vmdk \
                  /opt/unetlab/addons/qemu/linux-onos-2.7.0/virtioa.qcow2
```

linux-ubuntu

7.3.3 Přístup do vnější sítě

Jeden z cílů práce je začlenit fyzické zařízení typu WAN-edge do SD-WAN uvnitř EVE-NG simulátoru. Postup pro nastavování zařízení bude analogický s virtuálními až, na drobné změny v softwaru WAN-edge zařízení. Stejně jako přidání nového uzlu do stávající topologie, tak stejně se přidávají síťové objekty v EVE-NG pojmenované Cloud0-9, které jsou přpraveny se mapovan na jednotlivá rozhraní eth0-9 pokud existujou, s tím že máme zaručenou existenci eth0. V tomto

případě se jedná o přímé mapování virtuálního portu na fyzický. Jelikož WAN-edge zařízení je připojeno do stejné sítě jako je EVE-NG, není potřeba oddělené sítě.7.9 Kdyby čak byla potřeba oddělené sítě, tak konfigurační soubory se nachází v /etc/network/interfaces.



Obrázek 7.9 Nastavení EVE-NG pro vnější komunikaci

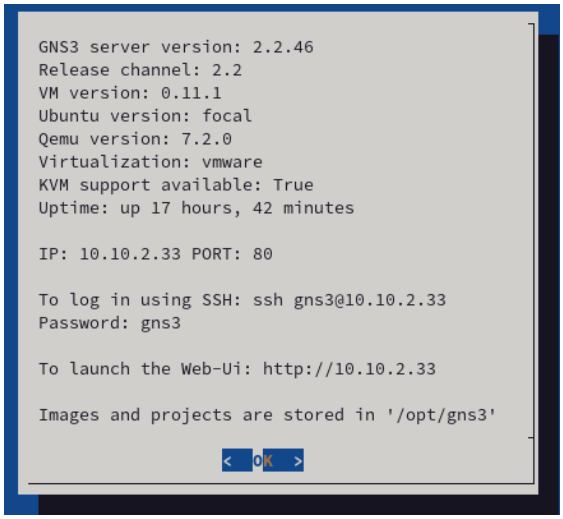
Všechny laboratoře zde popsány používají síť 10.10.0.0/16, což je více než jednotlivé laboratoře pořebují. Zároveň se dají sítě uvnitř laborek směrovat, tudíž konfigurace nestojí pouze na pomalém Jumpboxu. Připojení pak stálo pouze pár statických směrů a pomocí SSH tunelu je možno se dostat odkuvkoli na laboratoře.

7.4 Instalace GNS3

Instalace GNS3 spočívá v instalaci klienta a serveru.[45] Jelikož v našem prostředí je server a klient na oddělených strojích, bude potřeba instalovat klienta a server zvlášť. Nejprve nainstalujeme server tak, že z oficiálních stránek stáhneme připravený obraz pro VMWare ESXi. Ten nastavíme podle tak, aby obsáhl náročnost laborek, tudíž nastavení hardwaru budou pro účely testování podobné jako pro EVE-NG.

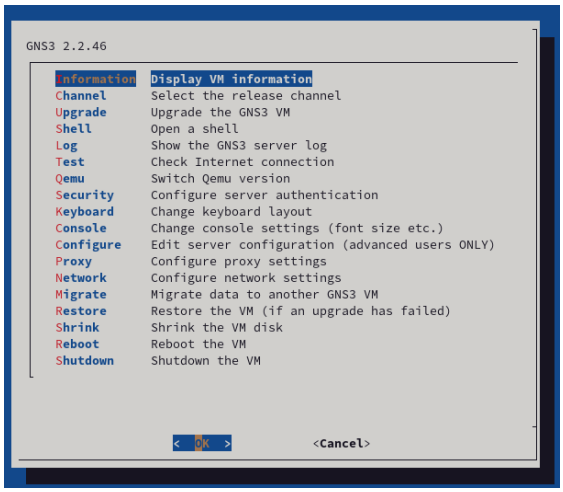
Veškeré laboratoře a obrazi použitých laboratoří se nacházejí ve virtuálním stroji, hardwarové požadavky pro virtuální počítač jsou 64 GiB RAM, 24vCPU a 256GiB úložiště. Při instalaci GNS3 na reálný hardware, je pořeba jej instalovat z obrazu zařízení, jelikož GNS3 oficiálně nepodporuje instalaci na baremetal. Zato soubor potřebný pro instalaci na hypervizor lze získat přímo z oficiálních stránek GNS3. Spolu s podrobným návodem, jak připravit obrazy softwaru pro podporovaná zařízení, a minimálními systémovými požadavky pro spuštění GNS3. Narozdíl od EVE-NG nejsme omezeni na pouze podporované obrazy zaříné, ale jsme schopni vytvářet vlastní obrazy zpusoby jako je QEMU, Dynamips, Docker a podobně. V současné době jsou požadavky následující. GNS3 je v současné době vydána jako soubory OVF, VMDK a další pro různé hypervizory a EXE, ELF jako binárky klienta. OVF je otevřený virtuální formát. Jedná se o otevřený virtualizační soubor pro virtuální počítače. GNS lze také přímo nainstalovat na fyzický hardware, s tím, že je nutno jej instalovat přes live CDs jiných operačních systémů. Vzhledem k tomu, že GNS3 používá mnoho hypervizorů, je vhodné pokud existuje možnost mít pro něj vyhrazený fyzický server bez jakéhokoli virtualizačního softwaru. 7.10 Mějte na paměti, že vnořená virtualizace není dobrá věc a může vést k nízkému výkonu. Protože se jedná svým způsobem o distribudci založené na debianu, můžeme optimalizovat software tak, že vypneme mitigation na procesu pomocí cmdline příkazu mitigations=off.

Jelikož je důležité, abychom se zvládly vždy připojit k serveru GNS3, je potřeba buďto nastavit statickou IP adresu nebo alespoň statické jméno v rámci domény. Stejně jako u EVE-NG, GNS3 má také webové rozhraní, pro zpříjemnění práce bude však použit GNS3 klient. Ve



Obrázek 7.10 GNS3 VM přehled.

výchozím stavu se GNS3 neptá na heslo a jméno, pokud bycho to ale chtěli, je nutné toto nastavit v příslušných konfiguračních souborech, kde jméno a heslo nemusí odpovídat uživatelským účtům virtuální mašiny.



Obrázek 7.11 GNS3 VM nastavení.

▶ Poznámka 7.1. Obraz má v základu jeden virtuální NIC, k němuž jsme nastavily statickou IP adresu 10.10.2.33/16 v rámci hypervizoru. Zařízení si pak dynamicky pomocí DHCP nastaví námi řečenou adresu. GNS3 umožňuje vytvářet spojení s vnější sítí, tudíž neí potřeba mít různých rozhraní pro segmentaci sítě. Důsledkem je, že zařízení mohou vidět do internetu a zařízení z vnitřní sítě mohou vidět dovnitř.

7.4.1 Dodatečný software

Veškerý dodatečný sofware instalovaného klienta GNS3 zahrnuje nástroje na analýzu datového toku Wireshark, pkud jsme na Windows tak telnet klienta PuTTY, a lokální instanci GNS3 serveru.

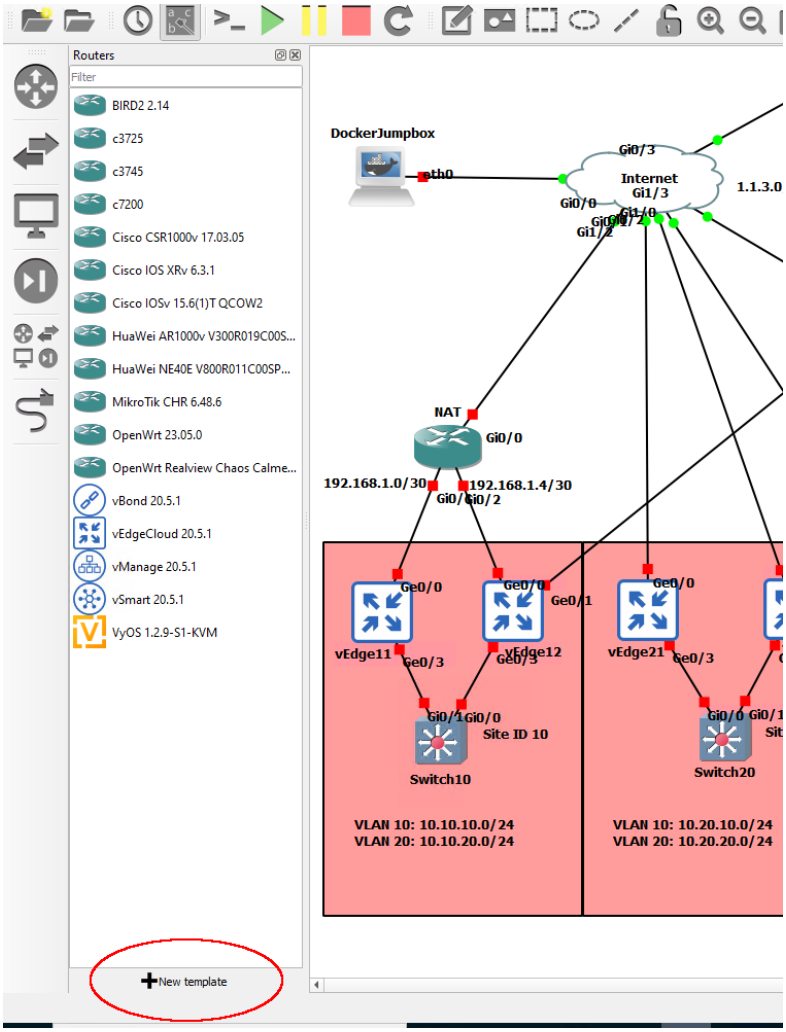
7.4.2 Obrazy zařízení

Stejně jako v případě EVE-NG, GNS3 neposkytuje kvůli licenčním podmínkám přístup ke komerčním obrazům zařízení, a tudíž je uživatel opět odkázán na sebe, a musí dodat obrazy zařízení do GNS3. Nejjednožší způsob instalace nových softwarových obrazů je právě přes klienta, který se připojí do serveru, který nahraje a nastaví softwarový obraz zařízení za nás. Toto je zásadní změna oproti EVE-NG, kde jsme musely pro každý obraz vytvářet složku a vkládat obraz ve specifickém formátu a jména.

▶ Poznámka 7.2. Protože komunikace s GNS3 strojem je realizována přes http, tak s velkou pravděpodobností se obrazy zařízení nahrávají do GNS3 přes WebDAV.

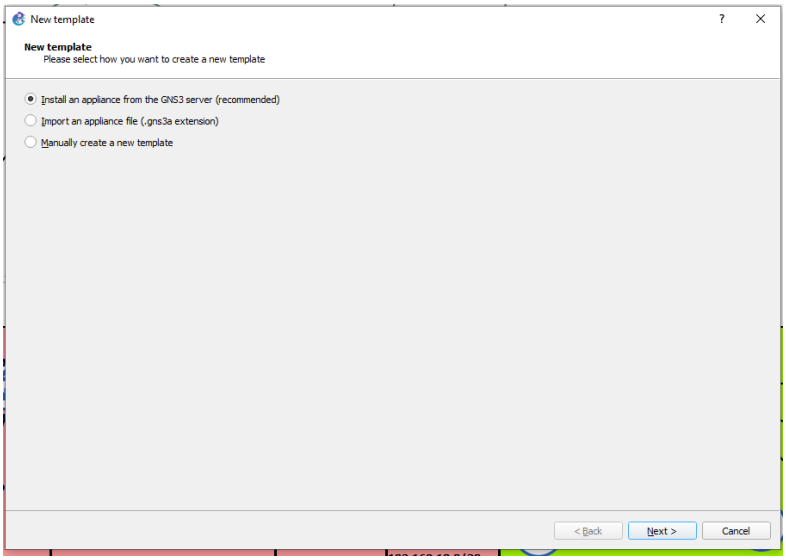
Následuje podrobný postup nahrání softwaru do GNS3 VM z klienta.

- Nejprve vyberem možnost přidat novézařízení do GNS3.7.12



Obrázek 7.12 GNS3 Vytvoření nového obrazu zařízení.

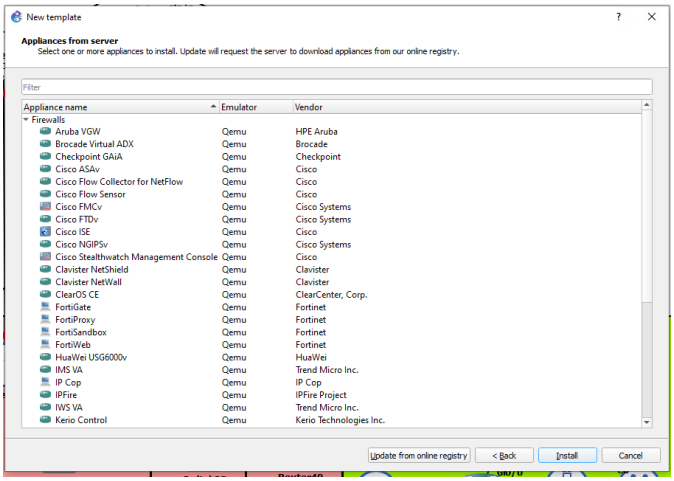
- Následně vybere, jakým způsobem chceme vytvořit novou předlohu zařízení.7.13
- V předchozím bodě jsme vybrali možnost z referenčních předloh. Proto teďka stačí vybrat konkrétní žádanou předlohu obrazu.7.14



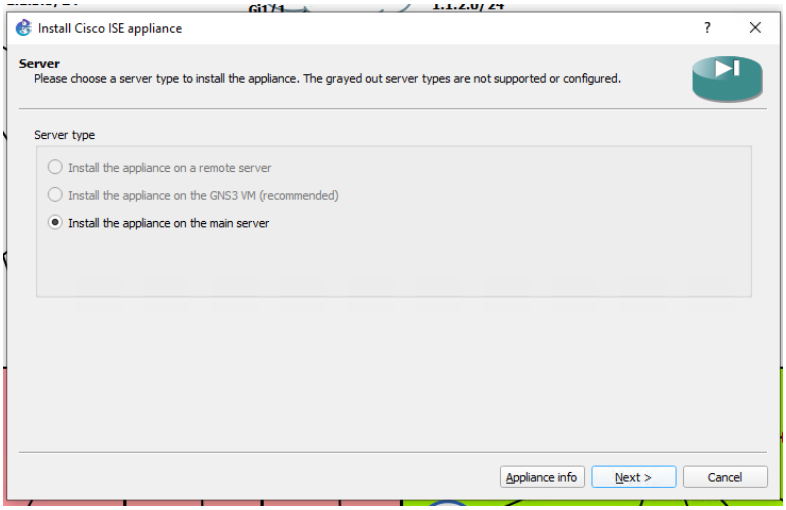
Obrázek 7.13 GNS3 výběr způsobu vytvoření obrazu zařízení.

- Nyní musíme vybrat místo, kam chceme uložit obraz zařízení.7.15
- Následně vybere verzi emulačního nástroje pro nové zařízení.7.16
- Poté se GNS3 klient snaží podívat do downloads, jestli se tam nachází soubor daných jmen, pokud ano tak provede kontrolní souče a pokud sedí tak ho nabídne jako instalovatelnou verzi.7.17
- Nakonec se zařízení nahraje na GNS3 VM.

V okamžik kdy se nahraje můžeme ihned začít používat obraz, či nastavit parametry předlohy zařízení, pokud je to třeba zařízení se nehce rozeběhnout nebo se zasekne. Samozřejmě pro případ mazání a modifikací systému virtuálního stroje GNS3 nabízí SSH s přihlašovacími údaji gns3:gns3 (jméno:heslo), kde jako úvodní obrazovka je přehled stavu systému, poté následuje krátké menu s možnostmi upravit GNS3 VM vlastnosti.



Obrázek 7.14 GNS3 výběr předlohy obrazu zařízení.

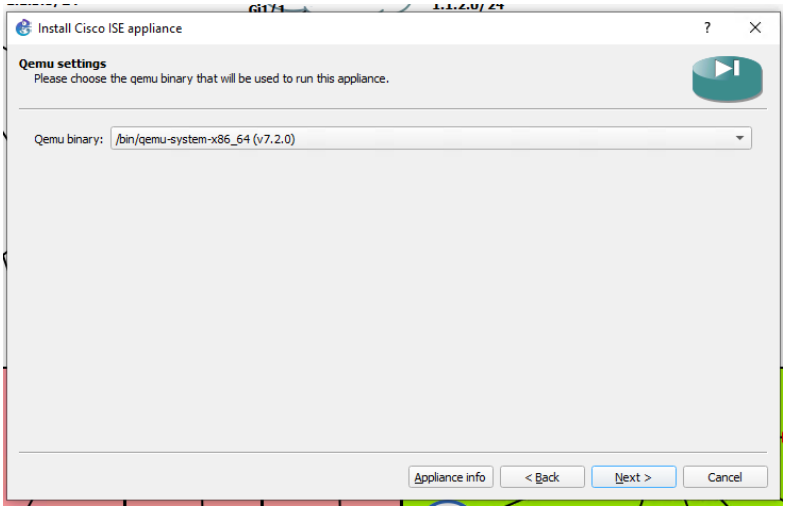


Obrázek 7.15 GNS3 výběr místa, kam se má obraz zařízení vložit.

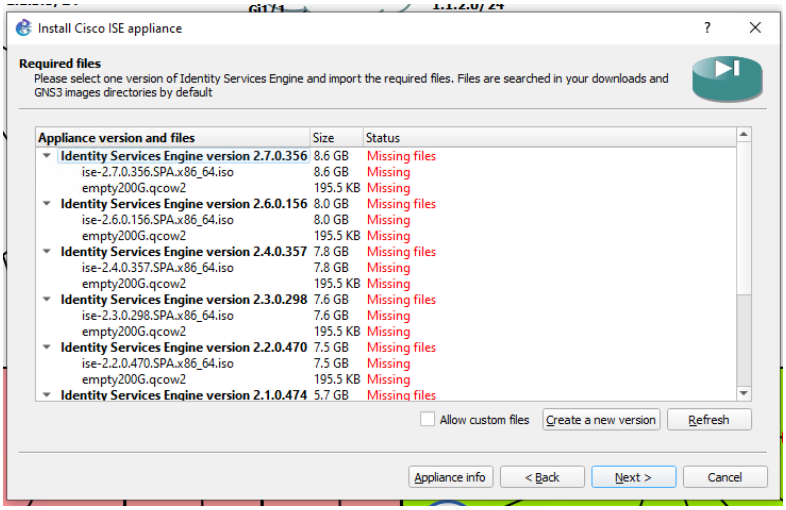
7.4.3 Přístup do vnější sítě

Jedním z cílů je propojení simulátoru GNS3 se sítěmi mimo něj, ať už při spuštění GNS3 jako virtuálního počítače nebo jako hostitelského operačního systému. Přístup k tomu zůstává stejný pro obě možnosti nasazení. Stejně jako přidávání uzlů existuje možnost přidat síťový objekt GNS3 s názvem Cloud nebo NAT. Obr.7.19 Výchozí mapování těchto cloudů je nastaveno tak, že Cloud je připojení na námi řečené rozhraní GNS33. Tyto cloudy jsou přemostěny 1:1 k jednotlivým námi řečeným rozhraním NIC. Druhé rozhraní typu NAT dělá to samé, až na to že také poskytuje DHCP server a NAT na daném rozhraní GNS3 VM.

V laboratořích popsaných v následující kapitole využívám připojení simulovaného se sítí Ing. Alexe Mouchy, Phd., abych měl přístup k zařízením a mohl je konfigurovat z jiných zařízení, než abych musel pro stejný účel nasazovat virtuální počítač Linux v simulátoru, to je v topologii Jumpbox. K tomu jsem použil připojení ze směrovače v GNS3 s názvem Gateway připojeného ke Cloudu pojmenovaném Management, který přemostilo připojení k druhé virtuální síťové kartě virtuálního počítače EVE-NG spuštěného v systému ESXi, který byl připojeno přes ESXi vSwitch. Instalace7.18 a konfigurace7.19 mapování rozhraní do cloudu figurace mimo server je



Obrázek 7.16 GNS3 emulačního nástroje.

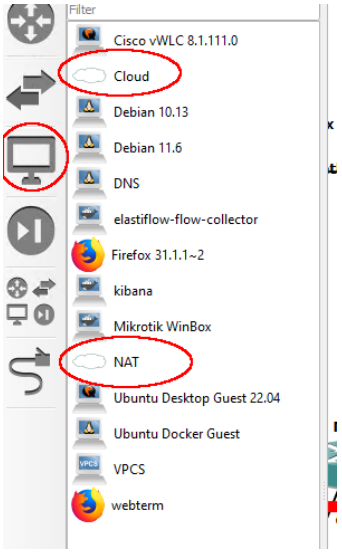


Obrázek 7.17 GNS3 výběr verze obrazu zařízení.

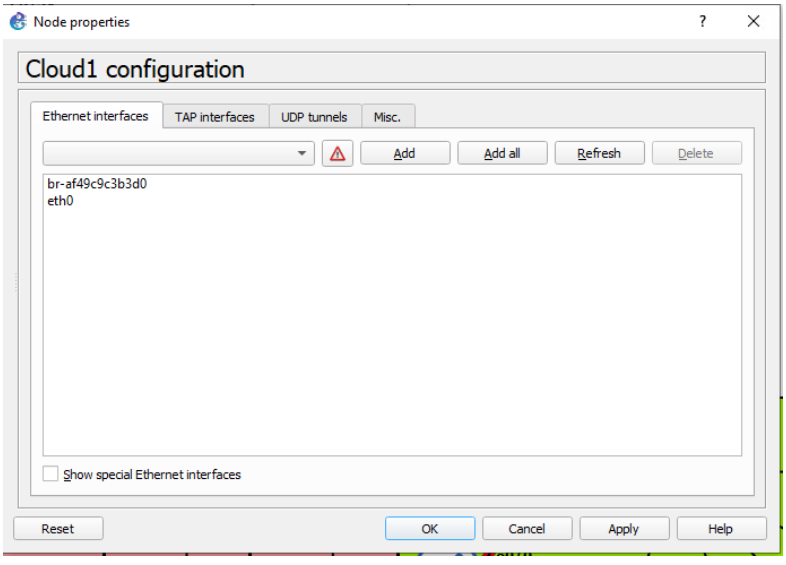
také velmi jednoduchá. Stačí použít stejný postup a umístit síťovou kartu serveru ESXi a GNS3. vNIC na stejný vSwitch a zároveň připojte hardwarový směrovač ke zmíněné serverové NIC.

Souhrn kapitoly přípravy implementací

V této kapitole jsme zvládli nasadit virtuální prostředí EVE-NG a GNS3, nahrát do nich protřebné obrzy zařízení a vytvořit uvnitř nich několik topologií softwarově definovaných sítí pro danou problematiku.



Obrázek 7.18 GNS3 vytvoření přítupu ven.



Obrázek 7.19 GNS3 nastavení přístupu ven.

Kapitola 8
Cisco Laboratoře

V této kapitole se zaměřím na podrobný popis, jak jednotlivé laboratoře SD-WAN implementovat v rámci simulátorů EVE-NG a GNS3. Zmíněné funcionality z předchozích kapitol budou postupně implementovány v rámci schopností a možnostmi zajištěných zařízení, kde některé implementace jsou, jako Huawei SD-WAN je nemožné implementovat. Kapitola začne zmíněním použité technologie, nastavováním dílčích zařízení a v neposlední řadě spuštění mechanizmů pro začleňování zařízení a zmínění některých vlastností SD-WAN.

Jako příklad, tak na obrázku 7.3 jsou vidět použité IP adresy, jak jsou zařízení rozlořena v sítí, jejich spojení a použitá rozhraní. Kvůli nemožnosti kopírovat laboratoře, aniž by se rozbila konfigurace, budou dodány postupy nastavení jednotlivých zařízení.

8.1 Příprava fyzické sítě

V této sekci se zaměřím na konfiguraci zařízení fyzické vrstvi, jako jsou směrovače a přepínače. Jinými slovy zde konfiguji zařízení, která neumí SD-WAN.

Jelikož na některých místech používám zařízení vIOS Adverterprise 9k, která neumí uložit stav jednotlivých rozhraní, je potřeba, abe v jednotlivých smerovačích byly na začatku laboratoře zapnuty všechny jejich rozhraní. Toho se dá docílit zhruba takto.

```
Router> enable
Router# configure terminal
Router(config)# interface range GigabitEthernet0/0-3
Router(config-range)# no shutdown
```

8.1.1 WAN okruhové přepínače

Funkce fyzické sítě je dodat jednotlivým zařízením IP konektivitu. Ta se dá docílit spoustou mezi sebou propojenými směrovači, téže známí jako Internet. Jiné WAN okruhy, jako například MPLS či LTE, existují s podobnými funcionalitami, poskytují IP konektivitu mezi jednotlivými WAN-edge zařízeními. Zaměřením, které se právě zabívá tato laboratoř, je SD-WAN, tudíž není potřeba zahlcovat síť zbytečně realistickou sítí se směrovači implementující BGP, ale stačí celý problém zaobalit do několika přepínačů. Stejně jako řešení SD-WAN tímto šetříme drahocené prostředky, kterých není nazbyt. Jednou z hlavních výhod přepínačů je, že se nemusí konfigurovat proto, aby fungovaly, nýbrž proto aby implementovali nějakou funkcionalitu jako jsou VLANy a podobně. Tato vlastno fyzických přepínačů se však ne vždy podaří promínou do simulačních prostředí.

Výpis kódu 8.1 WAN přepínač

```
switch> enable
switch# configure terminal
switch(config)# hostname INET_switch
INET_switch(config)# interface range gigabitEthernet 0/0 - 3
INET_switch(config-if-range)# no shutdown
INET_switch(config-if-range)# switchport mode access
INET_switch(config-if-range)# switchport access VLAN 10
INET_switch(config-if-range)# exit
INET_switch(config)# interface range gigabitEthernet 1/0 - 3
INET_switch(config-if-range)# no shutdown
INET_switch(config-if-range)# switchport mode access
INET_switch(config-if-range)# switchport access VLAN 10
INET_switch(config-if-range)# exit
INET_switch(config)# vlan 10
INET_switch(config-vlan)# name INET
INET_switch(config-vlan)# exit
INET_switch(config)# int vlan 10
INET_switch(config-if)# ip add 1.1.1.221 255.255.255.224
INET_switch(config-if)# no shutdown
INET_switch(config-if)# exit
INET_switch(config)# ip route 0.0.0.0 0.0.0.0 1.1.1.222
INET_switch(config)# ip domain-name sdwan-fit-lab.org
INET_switch(config)# crypto key generate rsa modulus 2048
INET_switch(config)# username admin privilege 15 password 0 admin
INET_switch(config)# line vty 0 4
INET_switch(config-line)# login local
INET_switch(config-line)# transport input ssh
INET_switch(config-line)# end
```

▶ Poznámka 8.1. Přepínače, které právě budou zastupovat Internet, jsou v obou simulátorech zobrazeny jako obláček. Proto se může plést obláček jako přepínač s obláčkem jako bod úniku z laboratoře. Proto switche jako jediné budou u sebe mít staticky přidělený adresní rozsah a úniky ze sítě budou brány jako něco proměnlivého o čem nevíme jak se bude chovat.

Následující konfigurace je právě konfigurace přepínače tvořící WAN okruh. 8.1 Toto je minimální konfigurace potřebná proto, aby 8 portů přepínače přepínalo pakety mezi sebou, s přidanou funkcionalitou SSH, aby byl přepínač dosptunpný z okolí, pokud by bylo potřeba řešit problémy, či změnit konfiguraci zařízení za podmínky, že si neutrhneme aktivní spojení SSH. Samozřejmě toto není potřeba dělat, jelikož oboje simulační prostředí nabízí způsoby jak se do zařízení dostat, avšak používání SSH je důležité pro kopírování textu z Jumpboxu do zařízením, jelikož Jumpbox neumožňuje kopírovat text vně zařízení. Další možná varianta jak si zpříjemnit život by bylo vypnout úvodní text. Naneštěstí se do zařízení nepřihlašujem tolikrát aby psaní těchto příkazů bylo ospravedlněno. Tyto texty jsou ze začátku přednastavené na přepínači, a přestože tyto texty jsou docela dlouhé, tak smazány nebudou.

8.1.2 Výchozí brána

Další samozřejmou součástí fyzické sítě je směrovač, zde konkrétně Cisco IOSv směrovač poskytující laboratoři nezbytné funkce. Tyto funkce jsou:

DHCP server potřebný pro začleňování vEdge směrovačů pomocí ZTP procesu.

DNS server potřebný též pro ZTP proces a pro nasazování vícero orchestračních vBond kon-

Výpis kódu 8.2 Gateway směrovač

```
Router> enable
Router# configure terminal
Router(config)# hostname Gateway_router
Gateway_router(config)# ip dhcp excluded-address 1.1.1.193 1.1.1.210
Gateway_router(config)# ip dhcp pool POOL1
Gateway_router(dhcp-config)# network 1.1.1.192 255.255.255.224
Gateway_router(dhcp-config)# default-router 1.1.1.222
Gateway_router(dhcp-config)# dns-server 1.1.1.222
Gateway_router(dhcp-config)# exit
Gateway_router(config)# ip domain round-robin
Gateway_router(config)# ip domain name sdwan-fit-lab.org
Gateway_router(config)# ip host vbond.sdwan-fit-lab.org 1.1.1.7 1.1.1.8
Gateway_router(config)# ip host ztp.viptela.com 1.1.1.7
Gateway_router(config)# ip name-server 8.8.8.8
Gateway_router(config)# username admin privilege 15 password 0 admin
Gateway_router(config)# interface Loopback1000
Gateway_router(config-if)# ip address 1.1.255.1 255.255.255.255
Gateway_router(config-if)# interface GigabitEthernet0/0
Gateway_router(config-if)# description LAB_GW
Gateway_router(config-if)# ip address 1.1.1.222 255.255.255.224
Gateway_router(config-if)# no shutdown
Gateway_router(config-if)# interface GigabitEthernet0/1
Gateway_router(config-if)# description LAB_MPLS_GW
Gateway_router(config-if)# ip address 1.1.2.100 255.255.255.0
Gateway_router(config-if)# no shutdown
Gateway_router(config-if)# interface GigabitEthernet0/2
Gateway_router(config-if)# description LAB_outside_INET
Gateway_router(config-if)# ip address dhcp
Gateway_router(config-if)# no shutdown
Gateway_router(config-if)# exit
Gateway_router(config)# ip dns server
Gateway_router(config)# crypto key generate rsa modulus 2048
Gateway_router(config)# ntp source Loopback 1000
Gateway_router(config)# line vty 0 4
Gateway_router(config-line)# login local
Gateway_router(config-line)# transport input ssh
Gateway_router(config-line)# end
```

trolérů

NTP server je zde jen jko jistota pro synchronizaci času SD-WAN zařízení kvůli automatickém procesu certifikace zařízení

Výchozí brána pro možnou komunikaci vnějška s vnitřkem laboratoře a pro začlenění externích cEdge zařízení

Zde následuje konfigurace směrovače realizující výchozí bránu simulace. 8.2

▶ Poznámka 8.2. Ačkoliv nepoužívám v obou simulátorech, pro tento typ laboratoře, NAT pro vnějšek nýbrž přímé spojení, tak se spoléhám na existenci a funkčnost DHCP serveru vnější sítě pro zajištění adres. Pro případ výpadku, či nutnosti mít statickou ip adresu, tak pro tento případ byl vyhrazen rozsah address 10.10.77.0-10.10.77.255/16.

8.1.3 Docker Jumpbox

Aby možnost připojit se do laborky z venku byla čistě volitelná záležitost, je nasazen do laboratoře prvek Jumpbox sloužící jako webový terminál pro vManage a další zařízení obsahující webové rozhraní. Jelikož výpočetního výkonu není moc, tak je explicitně volena kontejnerizace namísto virtualizace, která umožňuje dosáhnutí stejných výsledků s využitím menšího množství prostředků. Jak jsem zmínil tak bohužel je potřeba nějaká forma HTTP/HTTPS komunikace pro přístup do GUI vManage. Toho se dá docílit buďto z vnějšku přímím routováním laboratorní sítě, nebo právě již zmíněným Jumpbox zařízením. Pro účely práce jsem volil vždy volbu menšího odpory, kdy když jsem chtěl dostat sériový soubor do laboratoře tak bylo jednodužší použít externí počítač. V jiných případech jako například podepisování certifikátů bylo jednodužší z Jumpboxu jelikož má v sobě program openssl. Jak docílit toho, aby bylo možné směrování do laboratoní sítě z vnějšku bylo částečně vysvětleno v předchozích kapitolách. 7

Co se týče nastavení Jumpboxu tak buď uvnitř nebo v konfiguračních souborech stačí povolit DHCP klienta, nebo napřímo nastavit konfiguraci zařízení v konzoli jako každý jiný Linux příkazy ip z balíčku iproute2.

▶ Poznámka 8.3. V případě potřeby může tento Jumpbox sloužit jako manuální certifikační autorita SD-WAN, ale v rámci laboratoře zde představené kódy předpokládají výkon operací přímo na zařízeních.

8.1.4 VPN přepínač

Tento přepínač simuluje funkce v segmentu LAN za WAN-edge směrovači. Nakonfigurovaný s více síťovými segmenty simuluji různá oddělení konkrétní pobočky. Účel tohoto přepínače bude jasnější v následujících částech, kdy ukážu směrování a VPN segmentace napříč logickou sítí SD-WAN. Přesně tuto stejnou konfiguraci používají i přepínače Switch10, Switch20 a Switch30 8.3 s jedniným rozdílem, že Switch30 má pouze jeden spoj. To však nemá na konfiguraci vliv. Posledním rozdílem je změna hodnoty ve třetím oktetu IP adres a názvu zařízení.

8.1.5 VPN směrovač

Toto zařízení slouží ke stejnému účelu jako přepínače VPN, ale využívá k tomu loopback rozhraní rozhraním simulujícím koncové sítě. Toto dělám kvůli tomu, aby existoval aspoň jeden skok na třetí vrstvě mezi zařízením na straně jedné (VPN směrovače) a vEdge na straně druhé. Zde je opět sepsána konfigurace směrovače. 8.4

8.1.6 Směrovač s NATem

Tento směrovač simuluje typický podnik připojen do Internetu přes síť poskytovatele síťových služeb, který zde konkretně poskytuje Internetové připojení lokalitě 10 a patřičním vEdge směrovačům. S použitím NATu na obou zařízení vEdge11 a vEdge12 jsem docílil míti jednu veřejnou adresu. Konfikurace 8.5 ukazuje nastavení typicého PATu, kdy se přetěžuje adresa pro více překládaných adres. Účel tohoto zařízení je znázornit funkcionalitu SD-WAN NAT Traversal, kdy vEdge zařízení nepotřebují mít veřejnou adresu aby se dokázaly domluvit.

▶ Poznámka 8.4. Jeden by mohl namítnout, že se jedná o NAT 1:N a nebyl by na omylu.

8.1.7 Směrovač se Statickým NATem

Účel tohoto směrovače je ukázat, že pokud má vBond překlad adres 1:1, tak také může být za NATem. Tento směrovač překládá konkrétní veřejnou IP adresu podle zadání laboratoře, na

Výpis kódu 8.3 VPN přepínač

```
Switch> enable
Switch# configure terminal
Switch(config)# hostname Switch20
Switch20(config)# VLAN 10
Switch20(config-vlan)# name Enterprise
Switch20(config-vlan)# exit
Switch20(config)# interface VLAN 10
Switch20(config-if)# ip address 10.10.20.1 255.255.255.0
Switch20(config-if)# no shutdown
Switch20(config-if)# exit
Switch20(config)# VLAN 20
Switch20(config-vlan)# name Guest
Switch20(config-vlan)# exit
Switch20(config)# interface VLAN 20
Switch20(config-if)# ip address 10.10.20.1 255.255.255.0
Switch20(config-if)# no shutdown
Switch20(config-if)# interface range gigabitEthernet 0/0 - 1
Switch20(config-if-range)# switchport
Switch20(config-if-range)# switchport trunk encapsulation dot1q
Switch20(config-if-range)# switchport mode trunk
Switch20(config-if-range)# switchport trunk allowed VLAN 10,20
Switch20(config-if-range)# no shutdown
Switch20(config-if-range)# end
```

Výpis kódu 8.4 VPN směrovač

```
Router> enable
Router# configure terminal
Router(config)# hostname Router40
Router40(config)# interface loopback 10
Router40(config-if)# ip address 10.10.40.1 255.255.255.0
Router40(config-if)# no shutdown
Router40(config-if)# interface loopback 20
Router40(config-if)# ip address 10.20.40.1 255.255.255.0
Router40(config-if)# no shutdown
Router40(config-if)# interface GigabitEthernet0/0
Router40(config-if)# no shutdown
Router40(config-if)# interface GigabitEthernet0/0.10
Router40(config-subif)# encapsulation dot1Q 10
Router40(config-subif)# ip address 192.168.10.2 255.255.255.252
Router40(config-subif)# no shutdown
Router40(config-subif)# interface GigabitEthernet0 /0.20
Router40(config-subif)# encapsulation dot1Q 10
Router40(config-subif)# ip address 192.168.20.2 255.255.255.252
Router40(config-subif)# no shutdown
Router40(config-subif)# exit
Router40(config)# ip route 0.0.0.0 0.0.0.0 192.168.1.5
Router40(config)# end
```

Výpis kódu 8.5 NAT směrovač

```
Router> enable
Router# configure terminal
Router(config)# hostname NAT
NAT(config)# ip dhcp excluded-address 192.168.1.1
NAT(config)# ip dhcp excluded-address 192.168.1.5
NAT(config)# ip dhcp pool LAN1
NAT(dhcp-config)# network 192.168.1.0 255.255.255.252
NAT(dhcp-config)# default-router 192.168.1.1
NAT(dhcp-config)# dns-server 1.1.1.222
NAT(dhcp-config)# exit
NAT(config)# ip dhcp pool LAN2
NAT(dhcp-config)# network 192.168.1.4 255.255.255.252
NAT(dhcp-config)# default-router 192.168.1.5
NAT(dhcp-config)# dns-server 1.1.1.222
NAT(dhcp-config)# exit
NAT(config)# ip domain name sdwan-fit-lab.org
NAT(config)# ip name-server 1.1.1.222
NAT(config)# username admin privilege 15 password 0 admin
NAT(config)# crypto key generate rsa modulus 2048
NAT(config)# interface GigabitEthernet0/0
NAT(config-if)# ip address 192.168.1.1 255.255.255.252
NAT(config-if)# ip nat inside
NAT(config-if)# no shutdown
NAT(config-if)# interface GigabitEthernet0/1
NAT(config-if)# ip address 192.168.1.5 255.255.255.252
NAT(config-if)# ip nat inside
NAT(config-if)# no shutdown
NAT(config-if)# interface GigabitEthernet0/2
NAT(config-if)# ip address 1.1.1.220 255.255.255.224
NAT(config-if)# ip nat outside
NAT(config-if)# no shutdown
NAT(config-if)# exit
NAT(config)# ip nat pool OUT 1.1.1.220 1.1.1.220 prefix-length 27
NAT(config)# ip nat inside source list 1 pool OUT overload
NAT(config)# ip route 0.0.0.0 0.0.0.0 1.1.1.222
NAT(config)# access-list 1 permit 192.168.1.0 0.0.0.3
NAT(config)# access-list 1 permit 192.168.1.4 0.0.0.3
NAT(config)# line vty 0 4
NAT(config-line)# login local
NAT(config-line)# transport input ssh
NAT(config-line)# end
```

Výpis kódu 8.6 Směrovač realzující statický NAT

```
Router> enable
Router# configure terminal
Router(config)# hostname staticNAT
staticNAT(config)# ip domain name sdwan-fit-lab.org
staticNAT(config)# username admin privilege 15 password 0 admin
staticNAT(config)# crypto key generate rsa modulus 2048
staticNAT(config)# interface GigabitEthernet0/3
staticNAT(config-if)# ip address 1.1.1.201 255.255.255.224
staticNAT(config-if)# ip nat outside
staticNAT(config-if)# no shutdown
staticNAT(config-if)# interface GigabitEthernet0/0
staticNAT(config-if)# ip address 192.168.255.1 255.255.255.252
staticNAT(config-if)# ip nat inside
staticNAT(config-if)# no shutdown
staticNAT(config-if)# exit
staticNAT(config)# ip nat inside source static 192.168.255.2 1.1.1.198
staticNAT(config)# ip route 0.0.0.0 0.0.0.0 1.1.1.222
staticNAT(config)# ip route 1.1.1.198 255.255.255.255 GigabitEthernet0/0
staticNAT(config)# line vty 0 4
staticNAT(config-line)# login local
staticNAT(config-line)# transport input ssh
staticNAT(config-line)# end
```

neveřejnou adresu 192.168.255.2, která zůstala napříč laboratořemi stejná. Tato neveřejná adresa je adresa zařízení vBond2 kontroléru. Naslující ukázka 8.6 opět popisuje konfiguraci zmíněného zařízeni.

8.2 Správa certifikátů a začleňování směrovačů

V této části kapitoli je krátce vysvětleno, jak konfigurovat kontroléry SD-WAN, spravovat certifikáty a vytvářet řídicí spojení mezi kontroléry. Jinými slovy se jedná o nastavení řídící, kontrolní a orchestrační vrstvy Cisco SD-WAN řešení. Minimální konfigurace pro začlenění kontrolérů se skládá z následujících informací.

System-ip je unikátní SD-WAN identifikátor pro každé zařízení. Směrování není založeno na této IP adrese, která sloučí čistě k identifikaci v rámcí SD-WAN.

Site-id určuje v jaké lokalitě se směrovač nachází.

Organization-name nebo-li jméno organizace do které SD-WAN síť spadá. Jméno organizace je důležitá informace, která musí být napříč všemi zařízeními jedné SD-WAN stejná a je case-sensitivní.

vBond IP je použito pokud není dostupný DNS server pro dosažení vBond zařízení a je to nutná informace pro funkci orchestrační vrstvy. Může být i doménové jméno.

IP konfigurace konfiguruje přenost po VPN0 přes IP. Může být za NATem do té doby dokud je vBond dostupný. Jedná se o bod připojení do fyzické části WAN.

Výchozí přihlašovací údaje jsou vždy admin s heslem admin, ale jelikož po prvotní přihlášením zařízení vyžaduje změnu hesla a změna na původní je zakázáno, tak nové heslo je vždy administrator. Tento fakt platí pro všechy zařízen Cisco SD-WAN.

Výpis kódu 8.7 Konfigurace vManage

```
vmanage# config
vmanage(config)# system
vmanage(config-system)# host-name vManage1
vmanage(config-system)# system-ip 100.100.100.100
vmanage(config-system)# site-id 1000
vmanage(config-system)# organization-name SDWAN-FIT-LAB
vmanage(config-system)# vbond vbond.sdwan-fit-lab.org
vmanage(config-system)# vpn 0
vmanage(config-vpn-0)# ip route 0.0.0.0/0 1.1.1.222
vmanage(config-vpn-0)# dns 1.1.1.222
vmanage(config-interface-eth0)# interface eth0
vmanage(config-interface-eth0)# ip address 1.1.1.193/24
vmanage(config-interface-eth0)# no shutdown
vmanage(config-interface-eth0)# tunnel-interface
vmanage(config-tunnel-interface)# allow-service all
vmanage(config-tunnel-interface)# commit and-quit
```

▶ Poznámka 8.5. Možná se ptáte, jak se zadává daná IP adresa vBond. Systémové příkazy vbond bere v potaz plnohodnotná doménová jména. Pokud je směrovač výchozí brány nastaven tak, jak je uvedeno na obrázku, s položkami hostů odpovídajícími IP adresám kotroléru vBond, použije se doménové jméno namísto IP adresy když se zadá jméno vBondu do konfigurace systémového příkazu vbond. Příkaz v bond navíc umožňuje mít více redundantní záznamů.

▶ Poznámka 8.6. Vy výchozím stavu jsou na SD-WAN zařízeních nastavena dvě VPN tunely, VPN0 pro transport a VPN512 pro správu. Jelikož vše se dá zvládnout udělat v rámci jedné VPN a to transportní VPN0. Zde proto VPN512 nebude použita

8.2.1 vManage

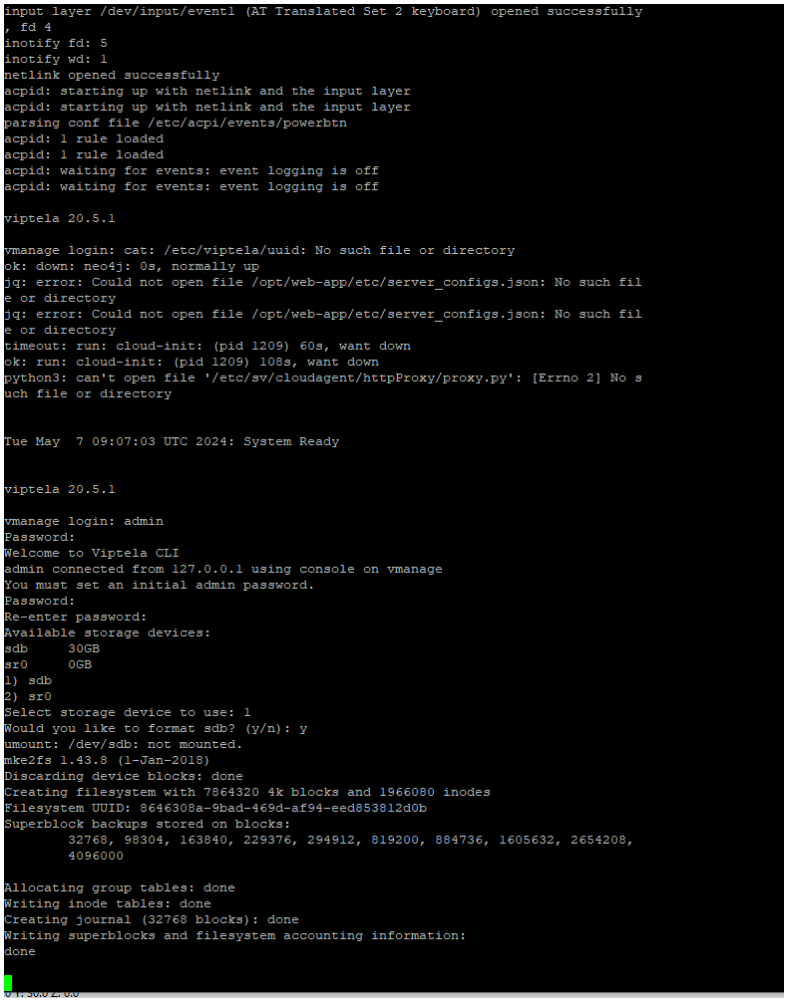
Když se poprvé spustí vManage kontrolér, budete vyzvání zadání přihlašovacích údajů. Z výroby je tam dané přihlašovací údaje uživatelské jméno admin s heslem admin. Po zadání přihlašovacích údajů budete opět vyzváni k zadání nového hesla.

Nato budete vyzváni k vybrání disku k formátování, s tím že na výběr je z prázdné virtuální cd sr0 mechaniky a virtuálního disku vdb. Zde vybereme samozřejmě virtuální disk a potvrdíme formátování disku. Po dokončení formátování disku se systém opět restartuje, takže opět čekáme než se objeví systémová hláška: „System ready“. Opět se přihlásíme a systém je připraven k použití. Na obrázku 8.1 je vidět jak proces inicialize systému vypadá na poprvé před druhým restartováním.

Následující kód 8.7 ukazuje jak jsem nastavoval vManage ve své laboratoři, s tím že tady je důležité vyplnit organisation-name SDWAN-FIT-ORG přesně takto. Proč se dozvíte v podsekci 8.2.4.1

8.2.2 vSmart

Konfigurace kontroléru vSmart je téměř totožná s vManage, jediné co se však mění je IP, system-ip a název zařízení. Zato se nemusí čekat na formátování disku, ale zadávat nové heslo musíme všude a tudíž vSmart není vyjímkou. Konfigurace 8.8 je pro uzly vSmart2 a vSmart3 zcela totožná až na ty samé informace jako byly u vSmart1 vůči vManage. Ohledně konkrétních adres se odkazuji na obrázek 7.3 aby byla držena konzistence kódu s popisem.



Obrázek 8.1 Nastavení a formátování disku při spuštění vManage.

8.2.3 vBond

Minimální konfigurace 8.9 pro vBond se drobně liší oproti konfiguracím vManage a vSmart na několika místech. Klíčovým rozdílem je vložení u příkazu vbond IP adresu vbondu a klíčové slovo local. Tento příkaz je klíčový pro funkčnost vBondu jako vBond, jelikosž toto klíčové slovo local efektivně mění funcionalitu zařízení z vEdge na vBond, jelikož mají stejné obrazy softwaru. Dalším malím rozdílem je, že namísto eth0 je zde použito rozhraní ge0/0. Poslední věci, která je mírně odlišná je, že v sekci tunnel-interface je přidán přikaz encapsulation ipsec. Tyto drobné změny jsou částečně zapřičiněny tím, že se jedná o rozdílný softwarový obraz oproti vManage a vSmart. vBond2 bude mít svoji dedikovanou sekci v sekci NAT traversal.

Výpis kódu 8.8 Konfigurace vSmart

```
vsmart# config
vsmart(system)# system
vsmart(system-system)# host-name vSmart1
vsmart(system-system)# system-ip 100.100.100.101
vsmart(system-system)# site-id 1000
vsmart(system-system)# organization-name SDWAN-FIT-LAB
vsmart(system-system)# vbond vbond.sdwan-fit-lab.org
vsmart(system-system)# vpn 0
vsmart(system-vpn-0)# ip route 0.0.0.0/0 1.1.1.222
vsmart(system-vpn-0)# dns 1.1.1.222
vsmart(system-vpn-0)# interface eth0
vsmart(system-interface-eth0)# ip address 1.1.1.194/24
vsmart(system-interface-eth0)# no shutdown
vsmart(system-interface-eth0)# tunnel-interface
vsmart(system-tunnel-interface)# allow-service all
vsmart(system-tunnel-interface)# commit and-quit
```

Výpis kódu 8.9 Konfigurace vBond

```
vedge# config
vedge(config)# system
vedge(config-system)# host-name vBond1
vedge(config-system)# system-ip 100.100.100.104
vedge(config-system)# site-id 1000
vedge(config-system)# organization-name SDWAN-FIT-LAB
vedge(config-system)# vbond 1.1.1.197 local
vedge(config-system)# vpn 0
vedge(config-vpn-0)# ip route 0.0.0.0/0 1.1.1.222
vedge(config-vpn-0)# dns 1.1.1.222
vedge(config-vpn-0)# interface ge0/0
vedge(config-interface-ge0/0)# ip address 1.1.1.197/24
vedge(config-interface-ge0/0)# no shutdown
vedge(config-interface-ge0/0)# tunnel-interface
vedge(config-tunnel-interface)# encapsulation ipsec
vedge(config-tunnel-interface)# allow-service all
vedge(config-tunnel-interface)# commit and-quit
```

Výpis kódu 8.10 Generování vlastních kořenových certifikátů

```
vmanage1# vshell
vmanage1$ openssl genrsa -out CA.key 2048
vmanage1$ openssl req -new -x509 -days 365 -key CA.key -out CA.pem \
> -subj '/C=CZ/ST=Czech␣Republic/L=Prague/O=SDWAN-FIT-LAB/CN=SDWAN-FIT-LAB
```

Výpis kódu 8.11 Možná distribude kořenových certifikátů do kontrolérů

```
vmanage1# vshell
vmanage1$ for i in {194..197}; do scp CA.pem admin@1.1.1.${i}:; done
```

8.2.4 Certifikáty

Jak bylo vysvětleno dříve, certifikáty jsou potřeba pro chod SD-WAN, jelikož pomocí něho se jednotlivá zařízení identifikuí a autentikují mezi ostatními zařízeními, tak postupně vybudovávají řetěz důvěry. Spolu s vysvětlením, proč byla zvolena cesta Enterprise CA, a proč se právě hodí do těchto laboratoří nejvíce. Následující kroky popisují postup vytvoření a instalaci kořenového certifikátu na jednotlivé kontroléry. Co se týče certifikační autority (CA), kdokoli může tuto funkci zastoupit pokud má možnost provozovat příkazy openssl a ssh. Zpousta online zdrojů je právě založena na Linuxových serverech pro tyto případy. Nicméně možnost výběru CA je v tomto prostředí nejlogičtější vybrat vManage. Viptela OS má vestavěnou podporu terminálu pro tyto případy včetně předinstalovaných potřebných softwarů. Následuje zde ukázka kódu generování certifikátu příkazy 8.10 a ukázka výstupu 8.2. Většinu parametrů, když generujeme kořenové certifikáty, je volitelná. Nejdůležitější je vyplnit jméno organizace a běžné jméno tak aby bylo stejné jako při konfiguraci uvnitř zařízení.

```
vmanage:~$ openssl genrsa -out CA.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
..............
.............
e is 65537 (0x010001)
vmanage:~$ openssl req -x509 -new -days 365 -key CA.key -out CA.pem -subj '/C=CZ
/ST=Czech Republic/L=Prague/O=SDWAN-FIT-LAB/CN=SDWAN-FIT-LAB'
vmanage:~$
```

Obrázek 8.2 Generování kořenové certifikační autority.

Poté co sme úspěšně vygenerovali nový certifikát, tak je potřeba je nakopírovat do jednotlivých kontrolérů. Toho docílíme pomocí jednoduchého bashovského scriptu, abychom nemusely psát jeden příkaz pro každou IP adresu zvlášt.8.11 Následující obrázek8.3 ukazuje očekávané chování.

Poté se přihlásíme do jednotlivých kontrolérů ve kterých je potřeba zkopírováný kořenový certifikát naisntalovat. 8.12 Tento proces přepíše původní Viptela certifikát, který byl předinstalován. Toto by měla být poslední manuální příkaz, který se zadávádo jednotlivých kontrolérů. Odkažte se kdyžtak na obrázek 8.4 pro případ očekávaného výstupu.

Co se týče generování certifikátů a jejich následná distribuce a instalace, tak to není proces, kde by se dalo udělat mnoho chyb. Pokud by však nastala chyba, která by byla potřeba vyřešit, je vždy dobré otestovat konektivitu mezi zařízeními jako první, jelikož je to jediné momentální místo, kde by se reálně dala udělat chyba. Druhou možností je, že zařízení blokuje některou z komunikací, buď SSH nebo ICMP.

Zbytek procesu začleňování kontrolérů se provede z grafického rozhraní, jelikož se dělo, že konfigurace CLI se tloukla s konfigurací GUI. Zde je představen postup:

```
vmanage:~$ for i in {2..8}; do scp CA.pem admin@1.1.3.${i}:; done
The authenticity of host '1.1.3.2 (1.1.3.2)' can't be established.
ECDSA key fingerprint is SHA256:YM6IdU+yuIi48bcldiAW1OlcsudnNk3GprYIhEmCme0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '1.1.3.2' (ECDSA) to the list of known hosts.
viptela 20.5.1

Password:
CA.pem                                        100% 1338    12.3KB/s   00:00
```

Obrázek 8.3 Distribuce certifikátu kontrolérům.

Výpis kódu 8.12 Instalace kořenového certifikátu

```
vsmart1# request root-cert-chain install /home/admin/CA.pem
```

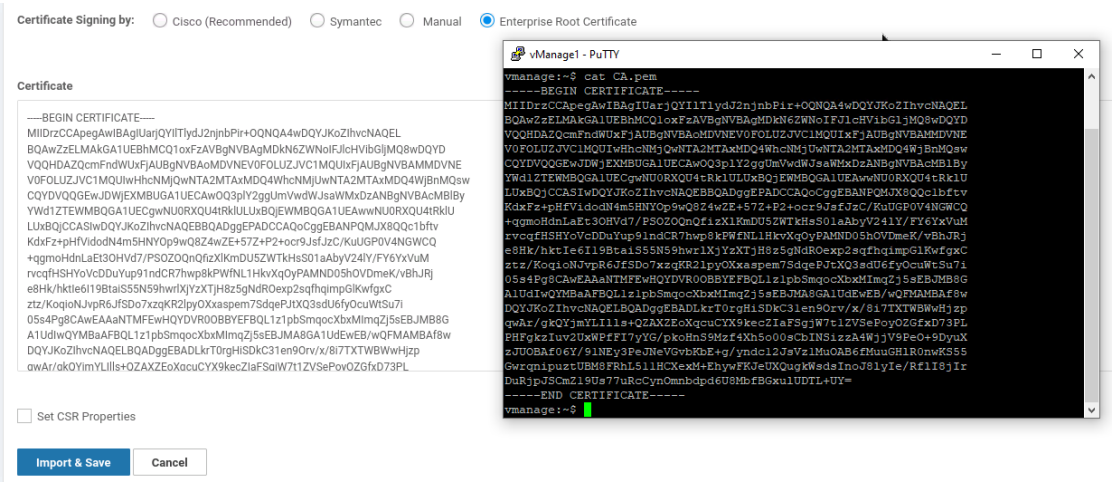
1. Připojte se na vManage GUI skrze webové rozhraní na adresu https://1.1.1.193.
2. Přihlašte se do vManage přihlašovacími údaji stejnými jako do CLI.
3. Jděte do sekce Administration→Settings.
   a. Zkontrolujte, že jméno organizace je správně vyplněno.
   b. Vyplňte informaci o vBond.
   c. Změňte Controller Certificate Authority na Entreprise Root Cerificate a vložte do něj obsah našeho kořenového certifikátu. 8.5
4. Přesuňte se do sekce Configuration→Devices→Controllers.
5. Přidejte všechny kontroléry do topologie.
   a. Vyplňte potřebné informace pro přidání zařízení.
   b. Nyní negenerujte CSR.
6. Nyní se přesuňte do sekce Configuration→Certificates→Controllers, všechny přidané kontroléry by se měly zde ukázat.
   a. Na konci každé řádky se nachází 3 svislé tečky, na ty poklikejte, měla by se objevit nabídka vygenerovat CSR. Tak jej vygenerujte.
   b. Nyní se objeví okno vygenerovaným CSR, obsah tohoto okna vložte do patřičného souboru uvnitř certifikační autority (to je v našem případě vManage). vManage pro modifikaci souborů má editor vim. 8.6
   c. Podepište jednotlivé certifikáty následujícím skriptem. Tento skript automaticky vyhledá všechny soubory s připonou csr, podepíše je a uloží do souboru pem.
   d. Uvnitř vManage GUI v tom samém místě kde jsme nechali generovat jednotlivá csr, tak v horním rohu je možnost uploadovat certifikát.

Výpis kódu 8.13 Podepisování certifikátů

```
vmanage1# vshell
vmanage1$ for i in *.csr; do openssl x509 -req -in "${i}" \
> -CA CA.pem -CAkey CA.key -CAcreateserial -out "${i/.csr/.pem}" \
> -days 365 -sha256; done
```

```
vbondl# request root-cert-chain install /home/admin/CA.pem
Uploading root-ca-cert-chain via VPN 0
Copying ... /home/admin/CA.pem via VPN 0
Updating the root certificate chain..
Successfully installed the root certificate chain
```

Obrázek 8.4 Instalace certifikátu na kontrolérech.



Obrázek 8.5 Instalace Enterprise CA na vManage.

e. Vložte obsah jednotlivých souborů do pole. Pouze jeden v daný okamžik.8.7

f. V okamžik kliku instalovat, vManage GUI vás přeměruje do přehledu úloh. Po krátkém času by se měla objevit fajvka v zeleném kruhu a nápis „Success“.

g. Nezapomeňte toto provést pro všechny zařízení v sekci kontroléry.

7. Zkontrolujte, že jednotlivé kontroléry naběhli v dashboardu ve vManage GUI.

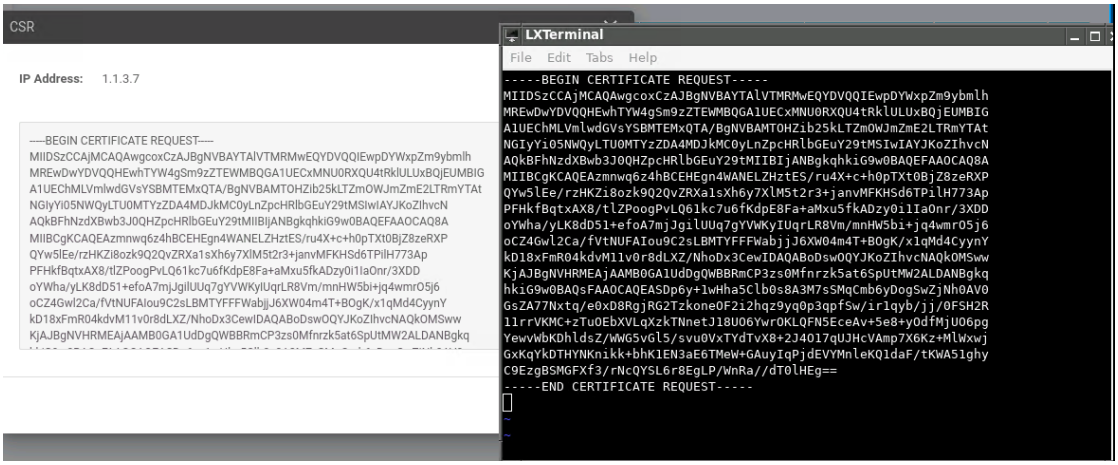
Alternativně můžete generovat jednotlivé csr soubory pomocí příkazové řádky a jak bylo viděno dříve tak, překopírování těchto souborů se dělá pomocí příkazu scp.

8.2.4.1 Cisco Smart účet

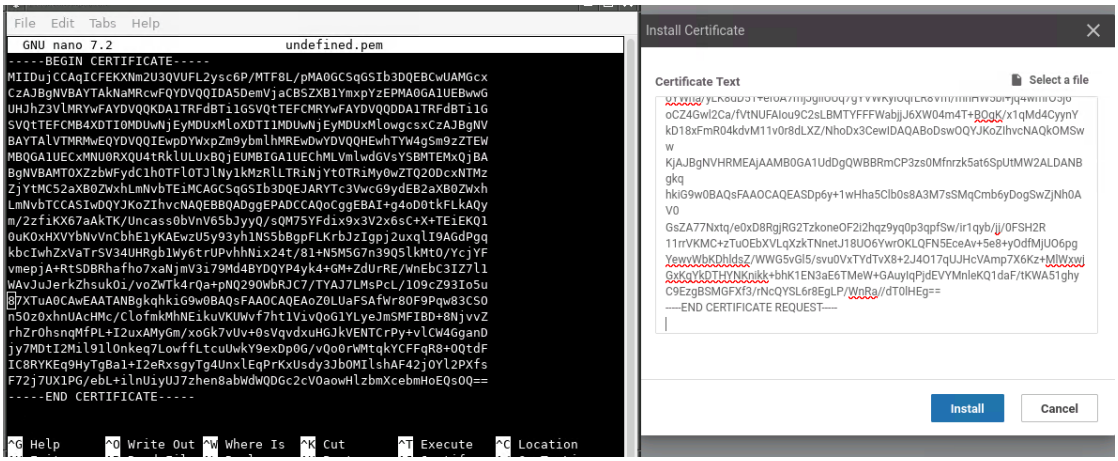
Před tím, než budeme moci pracovat s zařízeními, je potřeba pro vrtuální zařízení vytvořit Cisco Smart účet. Tento účetje dostupným všem, kdo mají u Cisca účet a jsou součástí podniku. Pokud nejste součástí podniku, Cisco vám neumožní vytvořit Smart účet a tudíž nemůžete ani vytvořit laboratoř. Naneštěstí tento soubor byl dodán vedoucím práce, který zdokumentoval celý proces vytvoření Cisco viptela souboru. Pro fyzická zařízení tento postup není potřeba, jelikož sériové číslo a číslo šasi je přimo vypáleno do hardwaru od výrobce. Pro účely simulace, jelikož není zařízení nějak extra odlišné od vEdge krom OS, který je Cisco IOS XE se tyto typy zařízení nepoužívají.

1. Nejprve se přesuneme do správy Cisco Smart účtu.8.8

```
vsmart1# request csr upload /home/admin/vsmart1.csr
```

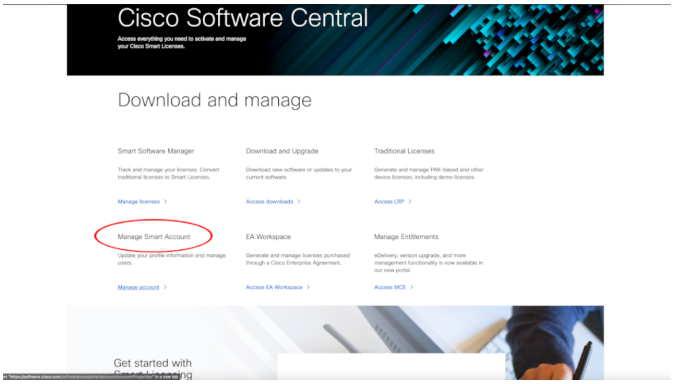


Obrázek 8.6 Generování CSR souborů za každého kontroléra včetně vManage.

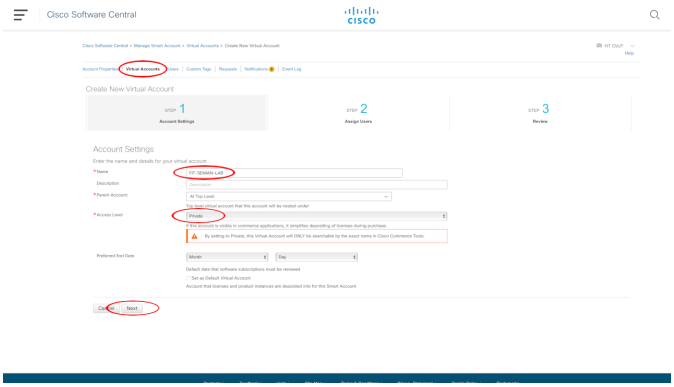


Obrázek 8.7 Nahrání podepsaných certifikátů do vManage

2. Zde vytvoříme nový virtuální účet pro naši společnost v laboratoři, tato společnost musí být jednotná napříč všemi zažízeními a kořenovými certifikáty. Zde volené jméno je volitelné.8.9
3. Na další stránce přiřadíme náš nově vytvořený virtuální účet sobě a dáme další.8.10
4. Nakonec potvrdíme vytvoření virtuálního účtu.8.11
5. Dalčím krokem bude vytvořit předlohu zařízení pro vManage v sekci Plug'n'Play.8.12
6. Zde v sekci profilů kontrolérů vytvoříme nový profil typu vBond.8.13
7. Poté co vybereme z výberu vBond tak se přesuneme k nastavení parametrů.8.14
8. Zde vyplníme jméno a systémové IP hlavního kontrolérů v síti. Nejdůležitějším údajem pro všechny laboratoře je právě údaj o jménu organizace, který musí souhlasit s kořenovým certifikátem laboratoří. Pokud by nesouhlasil je potřeba vygenerovat nový certifikát, tak aby souhlasil protože osobně nemám přístup do Cisco Smart účtu. IP nakonec zde nehraje roli ale tento údaj musí být vyplněn.8.15
9. Nyní jsme úspěšně vytvořily nový profil kontroléru. Tudíž je na čase ho populovat zařízeními.8.16



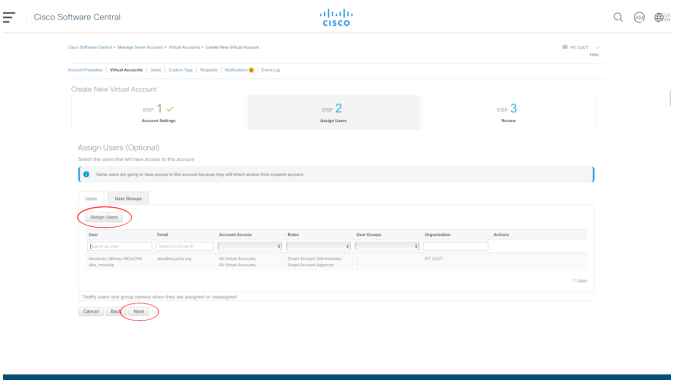
Obrázek 8.8 Úvodní strana portálu Cisca.



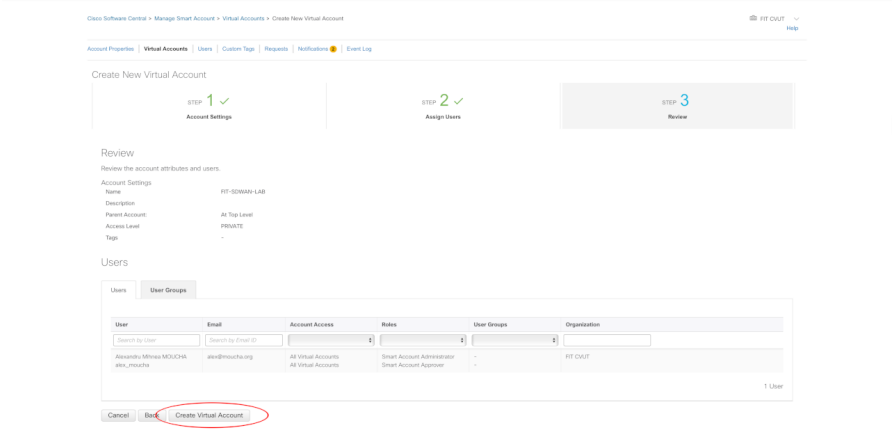
Obrázek 8.9 Postup vytvoření virtuálního účtu pro sériový soubor v Cisco Smart účtu.

10. V sekci zařízení je potřeba přidáme nová zařízení kde vybereme model zařízení a počet. Model je zde vEdge-Cloud-DNA.8.17
11. Následně pridáme 20 virtuálních ISR.8.18
12. A 20 virtuálních CSR1K8.19
13. Další zařízení které přidáme je virtuální směrovač C8000.8.20
14. Počkáme až Cisco vygeneruje sériová čísla zařízení.8.21
15. V tento okamžik jsou vygenerované virtuální sériová čísla zařízení.8.22
16. Protože máme k dispozici vygenerovaná zařízení, tak můžeme stáhnou soubor serialFile.viptela ze sekce profilů kontroléru.8.23
17. Následně vybereme verzi zařízení vManage, do kterého viptela soubor nahrajeme.8.24

V tento okamžik máme úspšně vytvořenou společnost SDWAN-FIT-LAB, s 80 virtuálními zařízeními, který se dá použít jednotně napříč všemi laboratořemi, jelikož tyto zařízení se obecně nepřipojují na devicehelper.cisco.com ale ztp.viptela.com, která je standartně konfigurovaná lokálně.



Obrázek 8.10 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.



Obrázek 8.11 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.

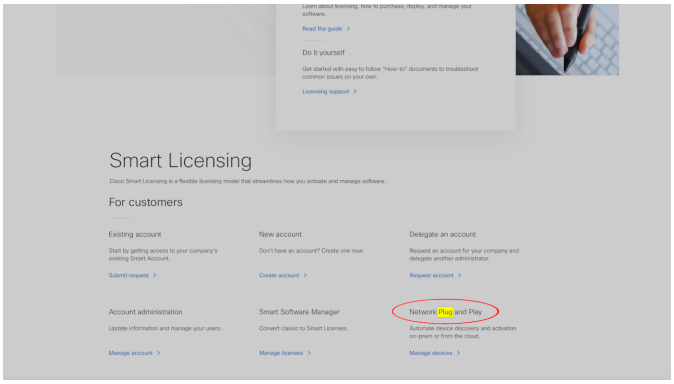
8.3 vEdge a jejich začleňování

Jak bylo již zmíněno v teoretické části, existují 3 praktické způsoby jak na tento problém. Těmi jsou manuálně přes CLI, pomocí Boostrap souboru a přes ZTP. Jelikož se konfigurace pomocí CLI a konfigurace pomocí Bootstrapu v mnohém neliší, tak se dá říct, že oba jsou stejně neefektiví. Proto do hloubky bude probrána pouze konfigurace pomocí příkazové řádky a Bootstrap bude popsán rychleji. Proces ZTP však má svojí vyhrazenou sekci, jelikož vyžaduje znalost konfigurace ZTP serveru a tvorby předloh. WAN-edge sériový list je třeba získat z Cisco Smart účtu, tomuto procesu byla vyhrazena celá jedna podkapitola8.2.4.1. Tento soubor obsahuje námi vybrané typy WAN-edge směrovačů, včetně jejich ID a autorizačního tokenu.

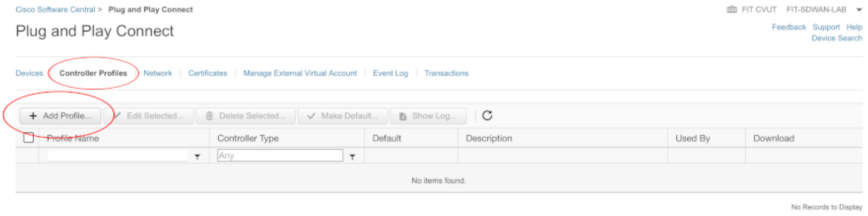
Přejděte do sekce Configuration→Devices v GUI vManage a importujte serialFile.viptela. Je důležité také zvolit možnost ověření a nahrání vEdge listu do kontrolérů. 8.25 Alternativně lze později dodat zařízení tento seznam ze sekce Configuration→Certificate, kde je monost „Send to controllers“. 8.26

8.3.1 Manuální konfigurace

Tato podkapitola ukazuje jak manuálně nakonfigurovat směrovač vEdge21.



Obrázek 8.12 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.



Obrázek 8.13 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.

- Nutná minimální konfigurace, postup replikuje postup jako u vManage a vSmart.
- Jako v předchozích částech nakopírujte CA.pem soubor do vEdge21. 8.3
- Instalace kořenového certifikátu.

```
vedge21# request root-cert-chain install /home/admin/CA.pem
```

- Aktivace vEdge21 s použitím libovolného nahraného zařízení z serialFile.viptela8.27

```
vedge21# request vedge-cloud activate chassis-number [UUID] token [OTP]
```

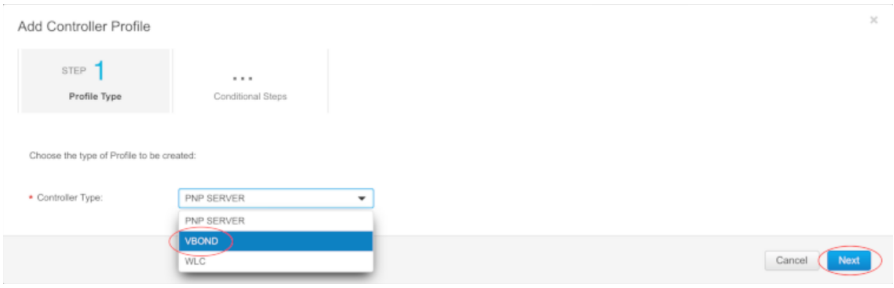
- V okamžik vložení tohoto příkazu tak vEdge se začne snažit vytvoři DTLS kontrolní spojení s vBond, autentikuje se a vytvoří DTLS spojení ke zbylím ostatním kontrolérům.

▶ Poznámka 8.7. Všimněte si toho, že CSR generování, podepisování a instalování probíhá zcela automaticky pro vManage, mimo ostatních kontrolérů. Toto nastavení se dá změnit v GUI vManage v sekci Administration→Settings.

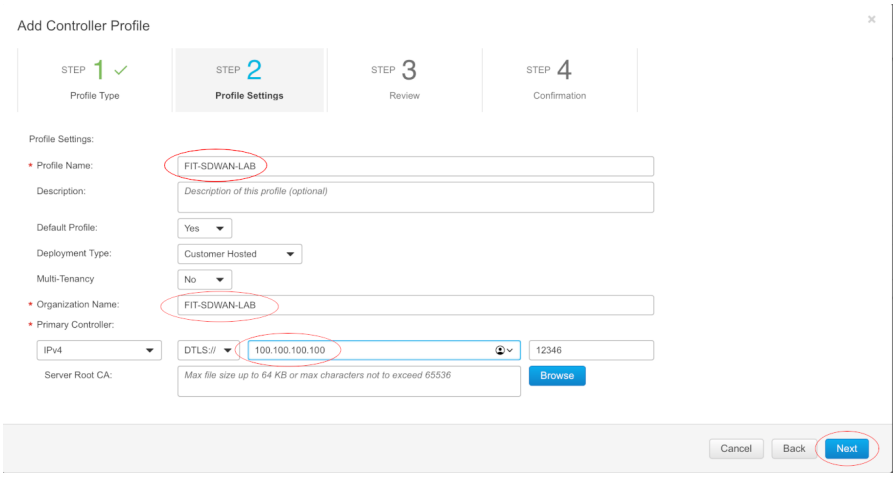
8.4 cEdge

Protože hlavní součástí práce jsou vEdge zařízení a toto zařízení je pouze jedno, bude následující podkapitola stručne vyprávět o konfiguraci daného zařízení. Toto zařízení bylo dodáno vedoucím práce a je přístupno přes sériovou konsoli.

Následující konfigurace představuje minimální konfiguraci cEdge zařízení8.4, aby bylo ve stavu stejném jako v případě vEdge. Před tím než je možno zařízení takto konfigurovat je nutno zadat příkaz controller-mode enable a následně restartovat zařízení. Tento krok by nám měl umožnit používat příkaz, config-transaction, který logikou ovládání je mišmaš vEdge a klasických routerech značky Cisco.

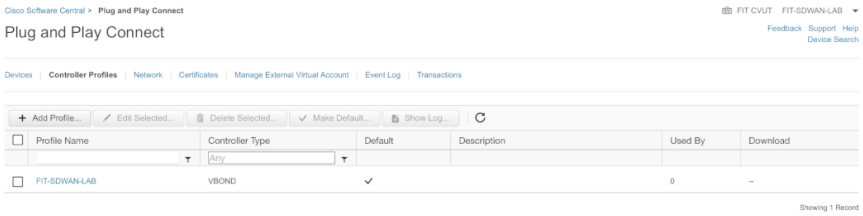


Obrázek 8.14 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.

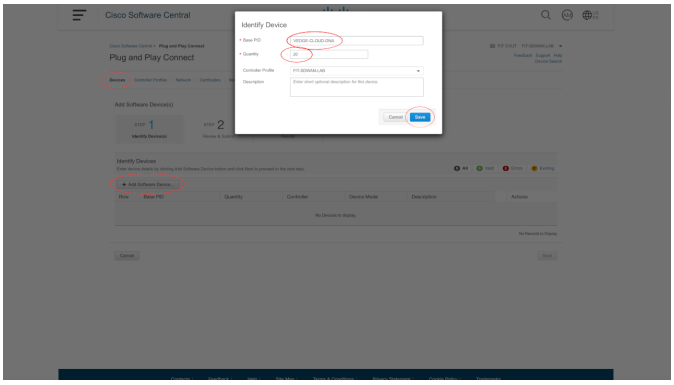


Obrázek 8.15 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.

```
Router# config-transaction
Router(config)# hostname cedge51
Router(config)# ip route 0.0.0.0 0.0.0.0 10.10.0.238
Router(config)# ip name-server 10.10.0.238
Router(config)# ip domain-name sdwan-fit-lab.org
Router(config)# system
Router(config-system)# site-id 50
Router(config-system)# system-ip 100.100.100.151
Router(config-system)# organization-name SDWAN-FIT-LAB
Router(config-system)# vbond vbond.sdwan-fit-lab.com
Router(config-system)# exit
Router(config)# interface GigabitEthernet 0/0/0
Router(config-if)# ip address 10.10.77.151 255.255.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# sdwan
Router(config-sdwan)# interface GigabitEthernet 0/0/0
Router(config-interface-GigabitEthernet0/0/0)# interface GigabitEthernet0/0/0
Router(config-interface-GigabitEthernet0/0/0)# tunnel-interface
Router(config-tunnel-interface)# encapsulation ipsec
Router(config-tunnel-interface)# exit
Router(config-interface-GigabitEthernet0/0/0)# exit
Router(config-sdwan)# interface Tunnel 0
Router(config-if)# ip unnumbered GigabitEthernet0/0/0
```



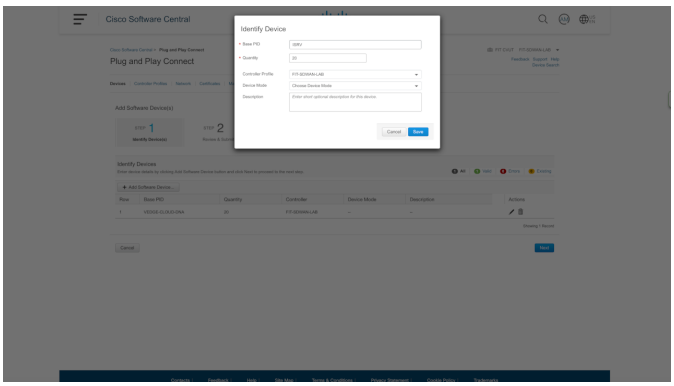
Obrázek 8.16 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.



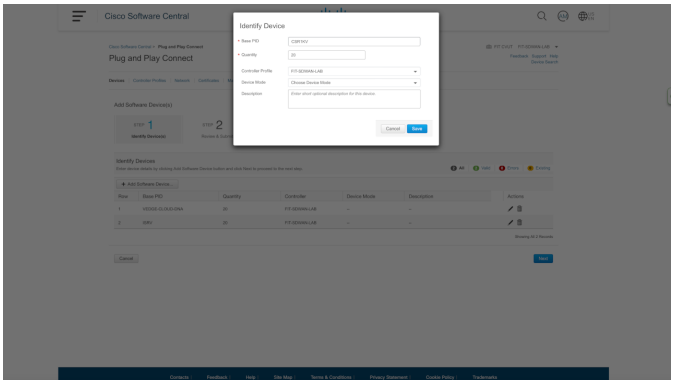
Obrázek 8.17 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.

```
Router(config-if)# tunnel source GigabitEthernet0/0/0
Router(config-if)# tunnel mode sdwan
Router(config-if)# commit
Router(config-if)# end
cedge51# copy sftp: bootflash:
Address or name of remote host []? 1.1.3.1
Source username [cedge51]? admin
Source filename []? CA.pem
Destination filename [CA.pem]?
cedge51# request platform software sdwan root-cert-chain install bootflash:CA.pem
cedge51# show sdwan certificate serial
```

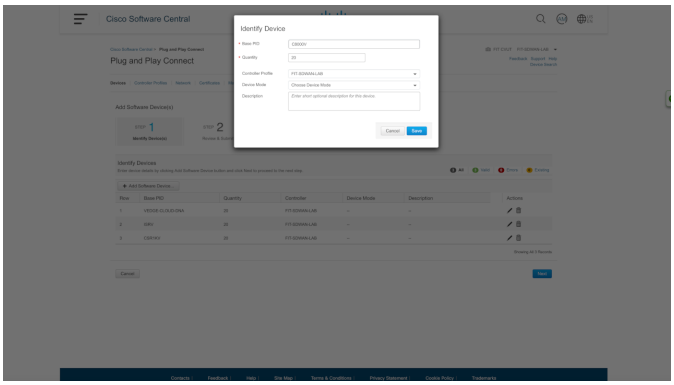
Poté je nutné importovat výstup posledního příkazu do vManage a čekat až se zařízení do-



Obrázek 8.18 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.

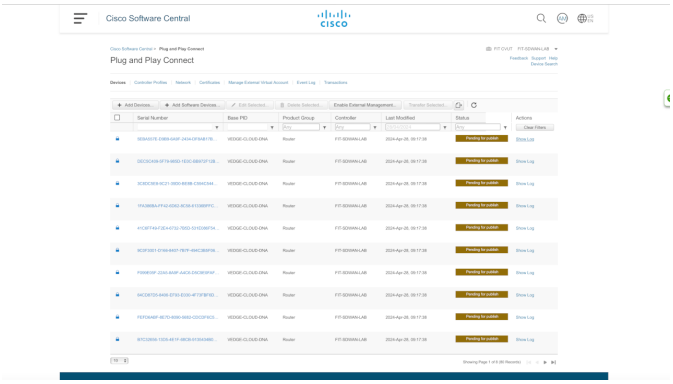


Obrázek 8.19 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.

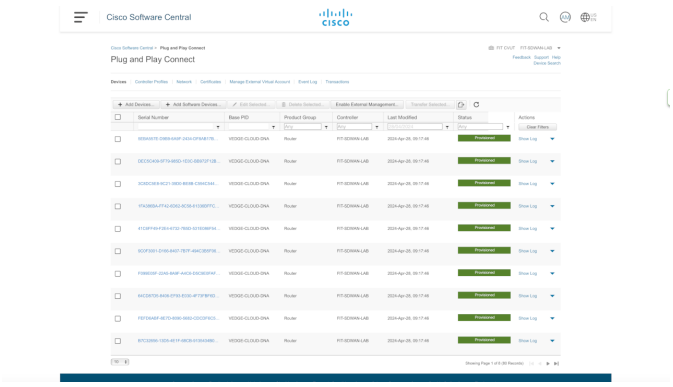


Obrázek 8.20 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.

mluví. Jiny způsob by byl nastavit konkrétní nový sériový číslo z portálu Smart účtu Cisca. V takovémto případě je nutné zadat příkaz „request platform software sdwan vedge_cloud activate chassis-number [UUID] token [OTP]“



Obrázek 8.21 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.



Obrázek 8.22 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.

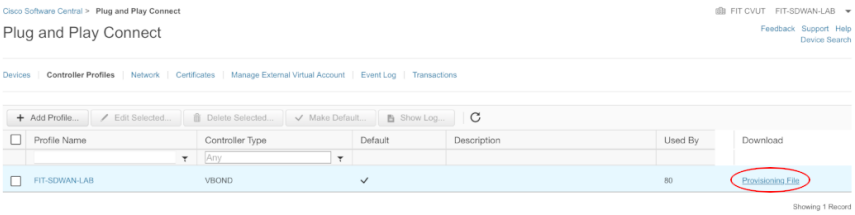
8.5 Předlohy

Předlohy hrají velkou roli v automatizaci SD-WAN. Definované ve vManage a distribuované napříč celou SD-WAN přes kontrolní vrstvu pomocí OMP aktualizací. Dva typy předloh existují.

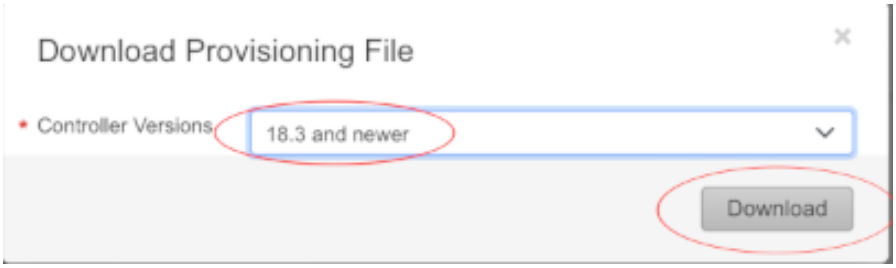
8.5.1 Předlohy příkazové řádky

Jak zde název napovídá, tak toto je čistě textový soubor obsahující running-configuration. Není až tak všestranný jako jsou Předlohy vlastností. Vyžadují, aby administrátor znal všechny možné příkazy na zařízení a díky tomu je tento způsob náchylný na chyby. Všechny tyto důvody jsou proč je toto ta méně používaná varianta předlohy. Jediné místo, kde se stímto typem předlohy se dá setkat případ konfigurace kontroléru jako je vSmart. vSmart potřebuje být řízen vManage, aby mohl plně využít svůj potenciál aplikování politik. Na obrázku 8.28 je vidět jak by mohla taková předloha vypadat.

1. Jděte do sekce Configuration→Templates→Device
2. Vytvořte novou předlohu příkazové řádky.
3. Jelikož se jedná o již nastavený vSmart v naší síti, tak nahrajem již běžící running-configuration z dostupných zařízení.
4. Nahraďte měnící se informace v předloze za proměné. System IP s System IP, host-name s Hostname a eth0 IP adresu za Eth0 IP.8.28



Obrázek 8.23 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.



Obrázek 8.24 Postup vytvoření sériového souboru SD-WAN v Cisco Smart účtu.

5. Přesuňme se do sekce Configuration→Templates→Device
6. Klikněte na 3 tečky vedle vytvořené CLI předlohy a vyberte Přiřadit zařízení
7. Vybere všechny 3 vSmart kontroléry a přiřaďte jim příslušnou předlohu tak že je z levého sloupce přesunete do pravého. 8.29
8. Vyplňte specifické parametry zařízení do námi definovaných proměných. Do této sekce se opět dostanete přes 3 svislé tečky. 8.30
9. Vyberte další zařízení a nakonfigurujte jej. Pokud poslaná konfigurace vyústí ve ztrátu spojení, tak vManage automaticky odebere změny, čili konfigurace se nijak nepropíše a nic se nestane.
10. Abyste si ověřili, že opravdu zařízení mají novou konfiguraci řízenou předlohami, tak se přesuňme do Configuration→Device→Controllers[1] a zkontrolujte sloupec Mode, kde by se měl nově ukázat vManage místo původního CLI. 8.31

8.5.2 Předlohy vlastností

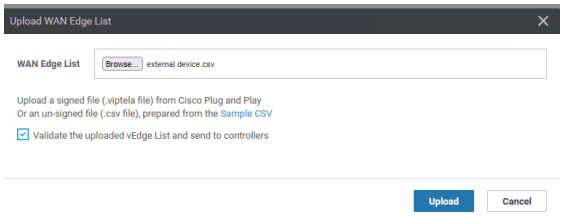
Předlohy vlastností fungují trochu jinak než předlohy příkazové řádky. Tento druh předloh přímo rozděluje konfiguraci do bloků vlastností a kobinací daných bloků vytváří konfigurace zařízení.

▶ **Příklad 8.8.** Příkladem tohoto postupu je, když chceme přidat do zařízení nějakou vlastnost, jako například BGP. Místo vyhledávání si kdejakých příkazů pro konfiguraci BGP, tak jen přidáme předlohu vlastnosti BGP a vyplníme patřičné proměné.

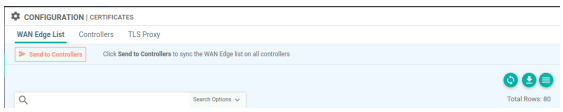
Existují 3 typy proměných v předlohách vlastností. Těmi jsou:

- Výchozí
- Globální

[1] Důvod proč píši tyto cesty anglicky je proto, aby se držela konzistence s grafickým prostředím, které není česky.



Obrázek 8.25 Nahrání Viptela sériového souboru zařízení.



Obrázek 8.26 Nahrání souborů zařízení do kontrolérů.

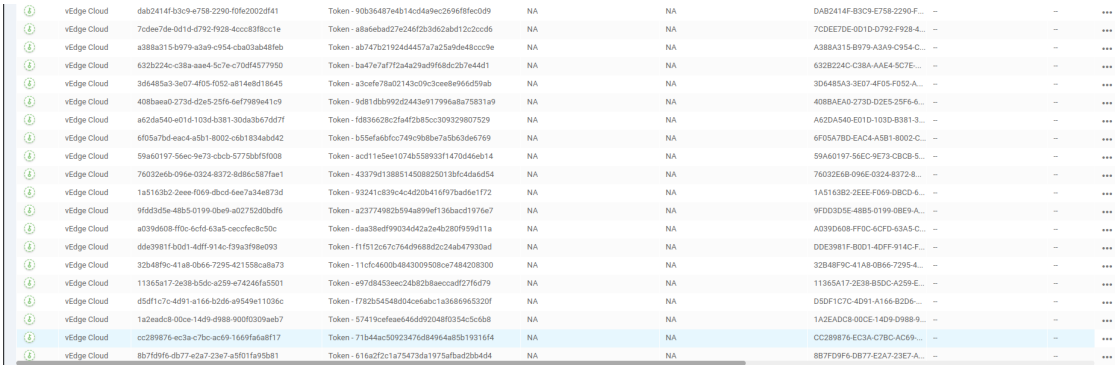
Specifický pro zařízení

Když vytváříme předlohy příkazové řádky tak mi definujeme proměnné, které se nakonec promítnou do konfigurace zařízení, tak tomu bycho říkali proměné specifické pro zařízení. Kde věci jako organisation-name se nemění napříč zařízeními v síti, tomu by se říkalo Globální proměná. Výchozí proměné jsou také vlastně globální, až na to že jsme je nemodifikovali my. Využívat proměné specifické pro zařízení tam, kde jsou změny potřeba má smysl dělat jako předlohu když víme že stejná konfigurace půjde ještě na další zařízení. Vytvoříme zařízením předlohy vlastností, které by byly použity na všechny vEdge zařízení v topologii.

1. Začnem vytvořením individuálních vlastností, které chceme přidat do předlohy.
2. Přesunem se do Configuration→Templates→Feature a přídáme předlohu
3. Z levé části obrazovky vyberem zařízení, na které tato předloha bude použita. Pro naše potřeby budou pouze potřeba vEdge Cloud zařízení. V případě, že bychom vybrali zařízení více, tak systém nám nabídne modifikovat pouze společnou skupinu proměných. Toto je opět velká výhoda využívání předloh vlastností. 8.32
4. Basic Information→System je první vlastnost, kterou vytvoříme.
   a. Vyplňte název vytvářené vlastnosti libovolným názvem, a vyplňte její popis.
   b. Výchozí hodnoty by měli být id lokace, systémové ip a název zařízení. Všechny tyto vyjmenované parametry by měly být specifické zařízení. (site-id, system-ip, host-name) 8.33
   c. Konsoli změňte z původně globální proměný na proměnu výchozí. Poté uložte vlastnost.
5. Další vlastnost na řadě je VPN - VPN
   a. Následující obrázky podrobně popisují nezbytnou konfiguraci předlohy vlastnosti. Vše co není zobrazeno v obrázku, bylo ponecháno výchozímu stavu. Některé proměné, jako je výchozí brána, nexthop a DNS jsou nastaveny jako proměné specifické zařízení i přesto že by mohli zde být jako globální proměné. Důvod je ten, že takto nastavená předloha umožňuje větší flexibilitu nastavení.
   b. Sekce Basic Configuration.8.34
   c. V sekci IPv4 Route nastavujeme výchozí bránu.8.35
   d. Nexthop výchozí brány. 8.36

Výpis kódu 8.14 Minimální manuální konfigurace vEdge

```
vedge# config
vedge(system)# system
vedge(system-system)# host-name vEdge21
vedge(system-system)# system-ip 100.100.100.111
vedge(system-system)# site-id 10
vedge(system-system)# organization-name SDWAN-FIT-LAB
vedge(system-system)# vbond vbond.sdwan-fit-lab.org
vedge(system-system)# vpn 0
vedge(system-vpn-0)# ip route 0.0.0.0/0 1.1.1.222
vedge(system-vpn-0)# dns 1.1.1.222
vedge(system-vpn-0)# interface ge0/0
vedge(system-interface-ge0/0)# ip address 1.1.1.202/24
vedge(system-interface-ge0/0)# no shutdown
vedge(system-interface-ge0/0)# tunnel-interface
vedge(system-tunnel-interface)# allow-service all
vedge(system-tunnel-interface)# encapsulation ipsec
vedge(system-tunnel-interface)# commit and-quit
```



Obrázek 8.27 Přehled nahraných kontrolérů.

e. Konfigurace DNS.8.37

6. VPN - VPN Interface Ethernet je vlastnost, která je použita pro připojení rozhraní do Internetu.

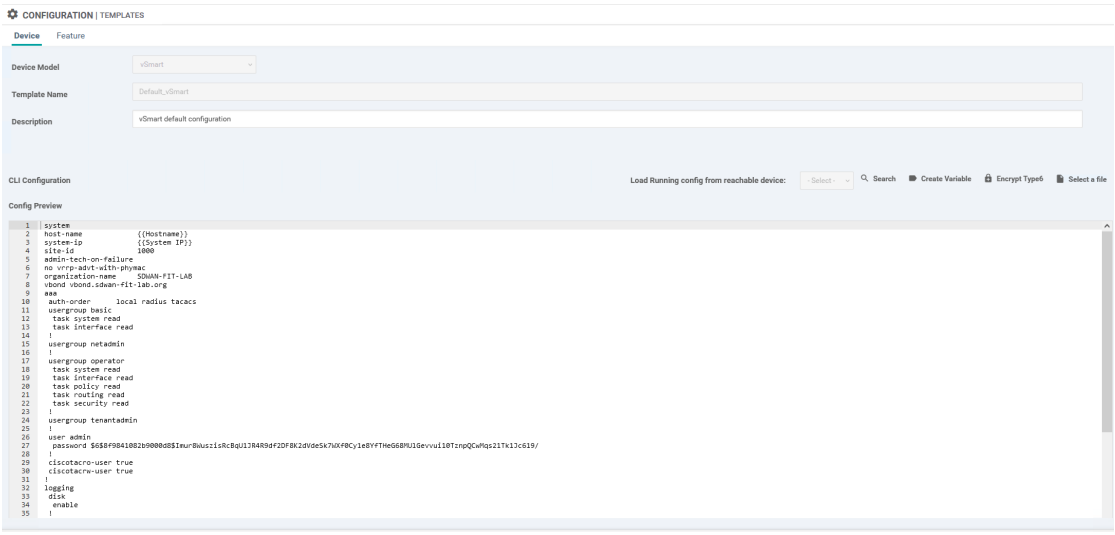
   a. V naší topologii veškerá připojení směrem do Internetu v zařízeních vEdge je realizováno na rozhraní ge0/0. Globální hodnota tedy bude toto rozhraní pro všechny vEdge zařízení.8.38

   b. Dále povolíme tunelové rozhraní a nastavíme barvu rozhraní na public-internet, jelikož Internet je to kam toto rozhraní náleží. Sice to není vidět na obrázku8.39, ale je ještě potřeba povolit v části nastavení Allow Service all na povoleno globálně.

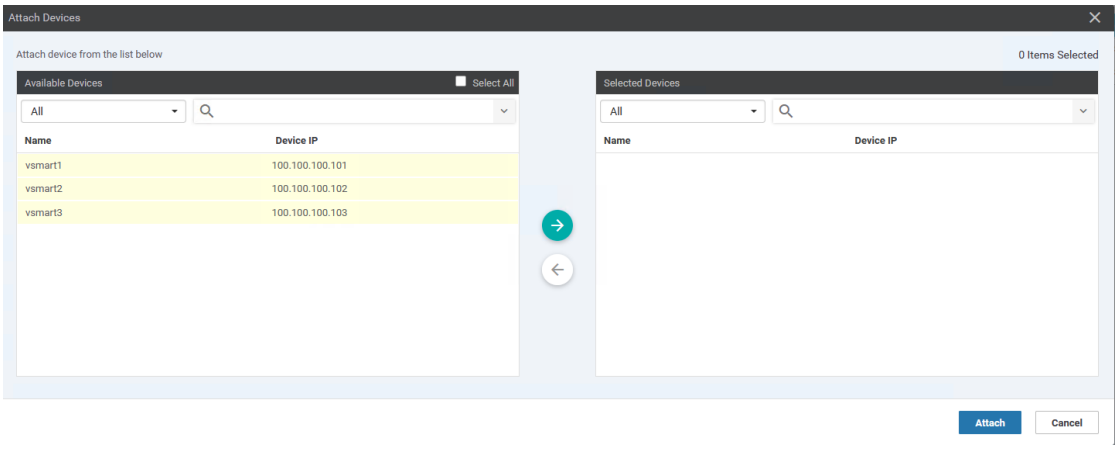
7. Přidejte druhou vlastnost stejného typu, tedy VPN - VPN Interface Ethernet pro nastavení MPLS komunikace.

   a. Veškerá MPLS konektivita na zařízeních vEdge budou připojena na rozhraní ge0/1. Stejné jako v případě s rozhraním ge0/0, tak tato předloha vlastnosti bude mít zcela totožnou konfiguraci až na pár vyjímek. 8.40

   b. Povolte a nastavte rozhraní stejně jako pro případ ge0/0 s tím, že bude vybrána barva MPLS mpls. 8.41



Obrázek 8.28 Nastavení GNS3 pro vnější komunikaci.



Obrázek 8.29 Nastavení GNS3 pro vnější komunikaci.

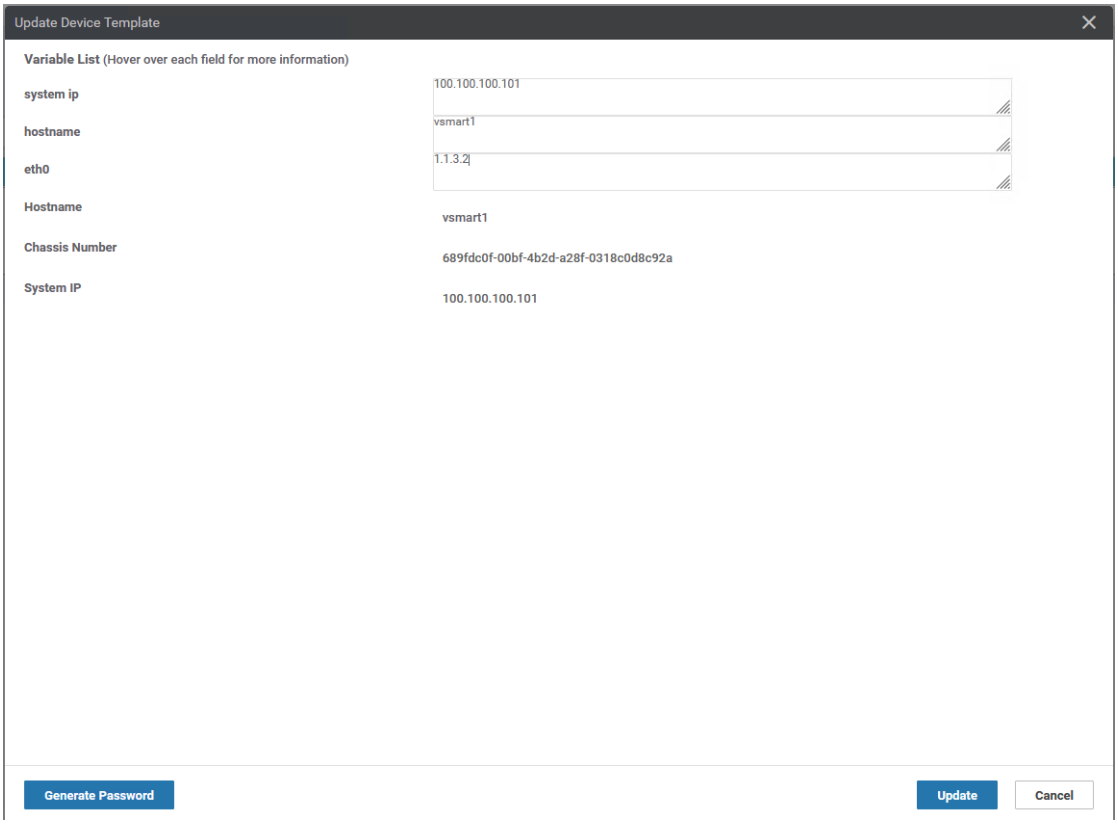
8. Rozhraní eth0 je třeba také nakonfigurovat kvůli řídící VPN512. Toto nastavení vlastnosti musí být přítomno i přestože VPN512 není reálně používána. Jedná se o nevýhodu, která je obsažena ve výchozích předlohách zařízení vManage. Výchozí předloha nezahrnuje přítomnost rozhraní eth0 a kvůli tomu vManage zkolabuje při pokusu synchronizace předlohy se zařízením, efektivně se pokoušející odebrat eth0 z původní VPN512. Jakákoliv konfigurace vystačí. Důležité je dát si pozor na to, aby rozhraní mělo globální proměnou rozhraní nastavenou na eth0.

9. Se všemy připravenými nezbytnými vlastnostmy je možno začí budovat předlohu vlastností zařízení. Přesuňme se tedy do Configuration→Templates→Device, kde začnem vytvářet předlohu.

10. Cílem bude, na základě topologie, vytvořit minimálně 3 předlohy chování pro 3 různé typy nasazení zařízení vEdge. Všechny 3 předlohy mají stejnou konfigurace v sekci Basic Information. 8.42

    a. Implementace vEdge s Internetovou a MPLS síťovou přípojkou. 8.43

    b. Implementace vEdge s pouze Internetovou síťovou přípojkou. 8.44



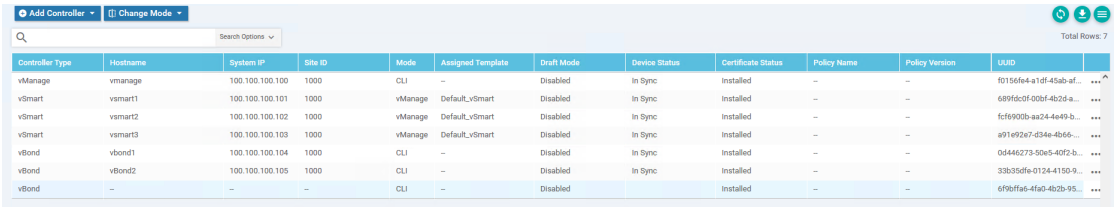
Obrázek 8.30 Nastavení GNS3 pro vnější komunikaci.

c. Implementace vEdge s pouze MPLS síťovou přípojkou. 8.45

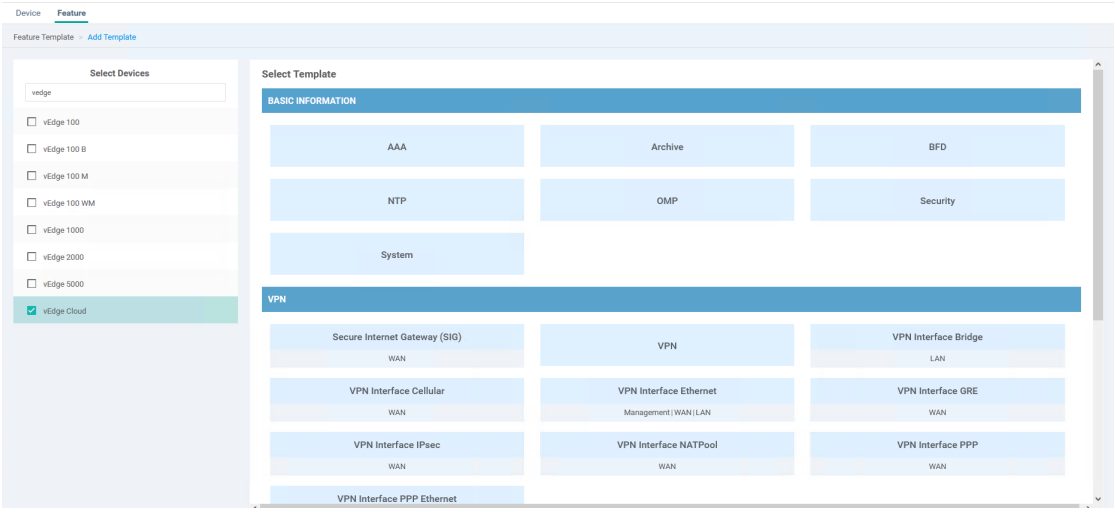
Jelikož máme vytvořeny veškeré potřebné předlohy chování, tak je čas k nim přiřadit jednotlivá zařízení, podobným způsobem jako se vyplňovaly předlohy příkazové řádky. Takto nakonfigurujeme zařízení vEdge11 a vEdge21 aby používaly předlohu čistě internetovou. Zařízení vEdge12, vEdge22 a vEdge31 používali předlohu pro obojí komunikaci a nakonec vEdge41 aby používalo čistě předlohu MPLS. Jelikož je vEdge41 přímo připojen pouze s MPLS sítí, tak se jeho konfigurace drobně liší. Jelikož MPLS neposkytuje konektivitu do kontrolérů, připojení bude muset jít skrze Výchozí bránu laboratoře. Což znamená, že výchozí brána vEdge41 je adresa 1.1.2.100/24.

```
vedge41# conf
vedge41(config)# vpn 0
vedge41(config-vpn-0)# dns 1.1.1.222 primary
vedge41(config-vpn-0)# ip route 0.0.0.0/0 1.1.2.100
vedge41(config-vpn-0)# interface ge0/1
vedge41(config-interface-ge0/1)# ip address 1.1.2.41/24
vedge41(config-interface-ge0/1)# no shutdown
vedge41(config-interface-ge0/1)# tunnel-interface
vedge41(config-tunnel-interface)# encapsulation ipsec
vedge41(config-tunnel-interface)# allow-service all
vedge41(config-tunnel-interface)# commit and-quit
```

▶ Poznámka 8.9. Při konfiguraci specifických proměných smerovaču vEdge11 a vEdge12 je potřeba si dát pozor na to, že běží za NATem a jejich brána je jinačí než brána ostatních směrovačů v topologii.



Obrázek 8.31 Nastavení GNS3 pro vnější komunikaci.



Obrázek 8.32 Nastavení GNS3 pro vnější komunikaci.

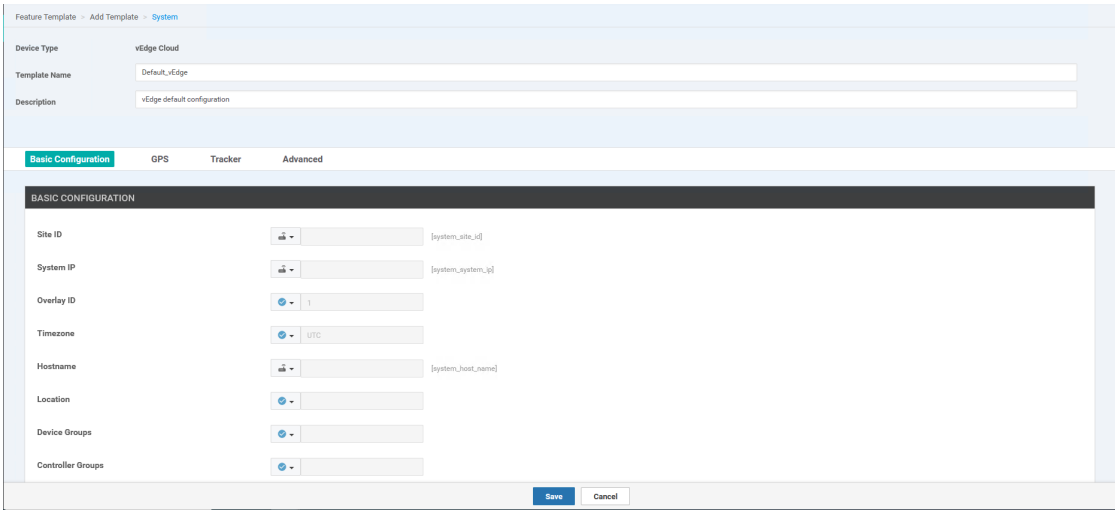
Přestože směrovače vEdge nejsou pořád v tento okamžik součástí SD-WAN, konfigurační předlohy zůstanou ve frontě čekající, nežli se jednotlivé vEdge nestanou součástí SD-WAN. Tento proces vytváření předloh je přesdtupem nasazení ZTP procesu, popsaného v následující sekci. Předlohy mohou být měněny kdykoliv s tím, že nám zajišťují propsání nové konfigurace do připojených zařízení. Připravené předlohy proto budou muset být ještě rozšířeny v následujícící sekci.

8.6 ZTP

Zero touch provisioning je proces, který již byl vysvětlen v teoretické části této diplomové3.3.3, tato podkapitola se zaměřuji na implementaci ZTP do větších detailů. Zbývající vEdge zařízení mohou být začleněny do SD-WAN přes proces ZTP, za použití předloh vlastností zařízení, popsané v předchozí sekci předloh.

Celý proces ZTP by se dal shrnout do 3 kroků.

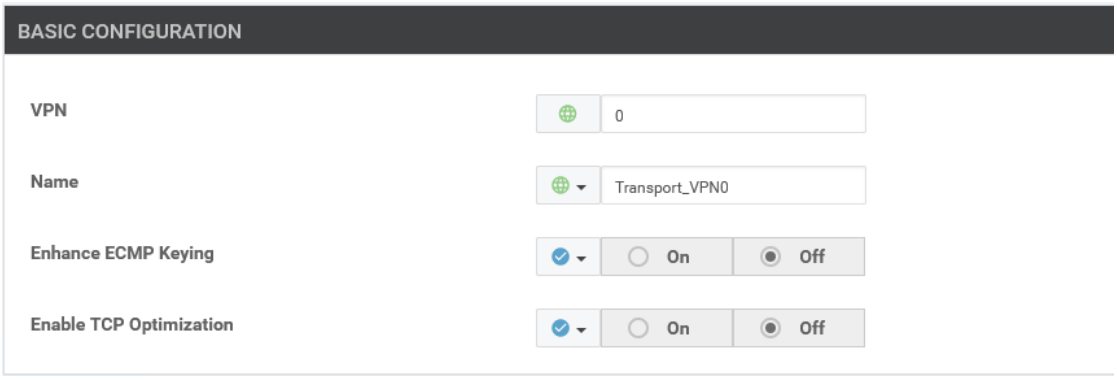
1. Nasazení ZTP serveru
   a. ZTP serve používá vBond zařízení ke své realizaci a jeho konfigurace je velmi podobná. Jako napovídajícím faktorem může být srovnání konfigurae ZTP serveru s vBond1. Bystré oko s všimne, že ZTP je rozšířen o klíčové slovo ztp-server v sytémovém příkazu vbond.
   b. Stejně jako v předchozích kapitolách, překopírujeme CA.pem soubor na ZTP server.
   c. Přidáme ZTP server jako vBond kontrolér ve vManage GUI.
   d. Vygenerujeme CSR, pak jej následně podepíšeme a dodáme na ZTP server.
   e. V tento okamži by měl být ZTP součástí SD-WAN.



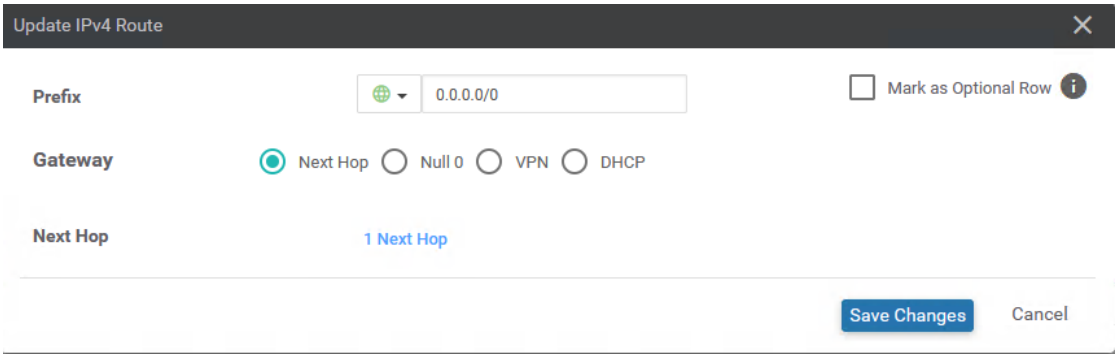
Obrázek 8.33 Nastavení GNS3 pro vnější komunikaci.

Výpis kódu 8.15 Konfigurace ZTP serveru

```
vedge# config
vedge(system)# system
vedge(system-system)# host-name ZTP-Server
vedge(system-system)# system-ip 100.100.100.106
vedge(system-system)# site-id 1000
vedge(system-system)# organization-name SDWAN-FIT-LAB
vedge(system-system)# vbond 1.1.1.199 local ztp-server
vedge(system-system)# vpn 0
vedge(system-vpn-0)# ip route 0.0.0.0/0 1.1.1.222
vedge(system-vpn-0)# dns 1.1.1.222
vedge(system-vpn-0)# interface ge0/0
vedge(system-interface-ge0/0)# ip address 1.1.1.199/24
vedge(system-interface-ge0/0)# no shutdown
vedge(system-interface-ge0/0)# tunnel-interface
vedge(system-tunnel-interface)# encapsulation ipsec
vedge(system-tunnel-interface)# allow-service all
vedge(system-tunnel-interface)# commit and-quit
```



Obrázek 8.34 Nastavení GNS3 pro vnější komunikaci.



Obrázek 8.35 Nastavení GNS3 pro vnější komunikaci.

f. Vybraná zařízení vložíme, pomocí těchto příkazů, do ZTP serveru aby se mohla účastnit ZTP procesu.

```
ZTP-Server# request device add chassis-number [UUID] serial-number \
 [OTP] validity valid vbond vbond.sdwan-fit-lab.org org-name \
 SDWAN-FIT-LAB enterprise-root-ca /home/admin/CA.pem
```

2. Začlenění vEdge užitím ZTP procesu.

a. Nejprve ověříme, že začleňované zařízení má k dispozici připravenou předlohu.

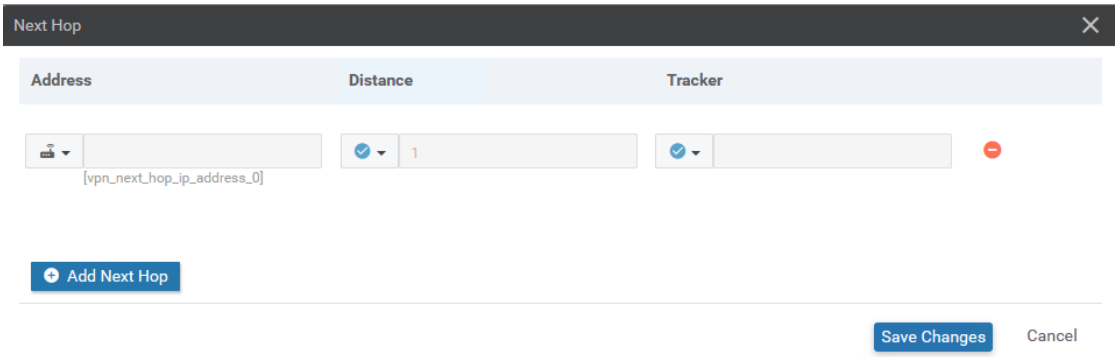
b. Pomoc VPN 0 se připojíme do vEdge přes VPN0.

c. Při zkoušení procesu ZTP s vEdge cloud a Enterprise CA je stále potřeba přistupovat ke směrovači pomocí konzole. Konfigurace zařízení se správným kořenovým certifikátem a také se správným sériovým číslem. To je na hardwarových směrovačích vEdge předkonfigurováno, kde proces ZTP skutečně zazáří. I přestože musíme manuálně přistupovat k vEdge a dodávat vlastní konfiguraci je stále rychlejší začlenit vEdges pomocí ZTP než celková manuální konfigurace.

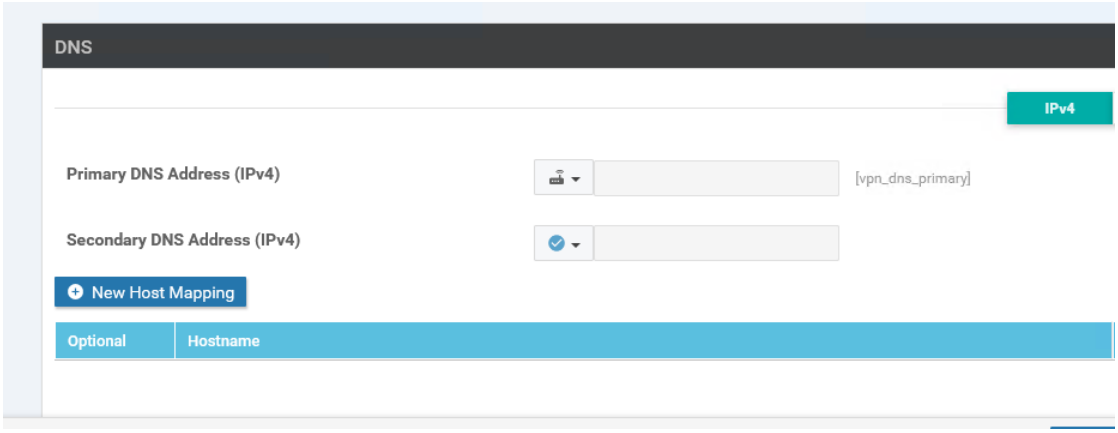
d. Ověřte, zda zařízení vEdge získalo informace o síti pomocí protokolu DHCP z brány. Jako ověření pošleme ICMP a DNS dotaz na název domény ztp.viptela.com, který by měl být přeložen a zároveň být dosažitelný.

e. Povolíme službu SSH na rozhraní ge0/0 VPN0 tunelového rozhraní.

```
vedge# config
vedge(config)# vpn 0
```



Obrázek 8.36 Nastavení GNS3 pro vnější komunikaci.



Obrázek 8.37 Nastavení GNS3 pro vnější komunikaci.

```
vedge(config-vpn-0)# interface ge0/0
vedge(config-interface-ge0/0)# tunnel-interface
vedge(config-tunnel-interface)# allow-service sshd
vedge(config-tunnel-interface)# commit and-quit
```

f. Zkopírujem do vEdge kořenový certifikát.

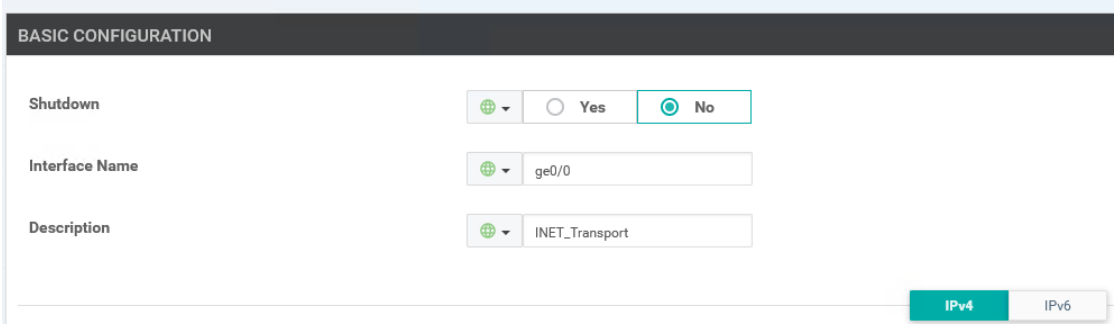
```
vedge21# request vedge-cloud activate chassis-number [UUID] token [OTP]
```

g. Aktivujeme vEdge použitím chassis-number a tokenu, získané z vManage GUI v sekci zařízení.8.27 Ujistěte se předtím, že zařízení není již manuálně začleněno zařízení vEdge21 a že záznam o zařízení se nachází uvnitř ZTP serveru.8.27

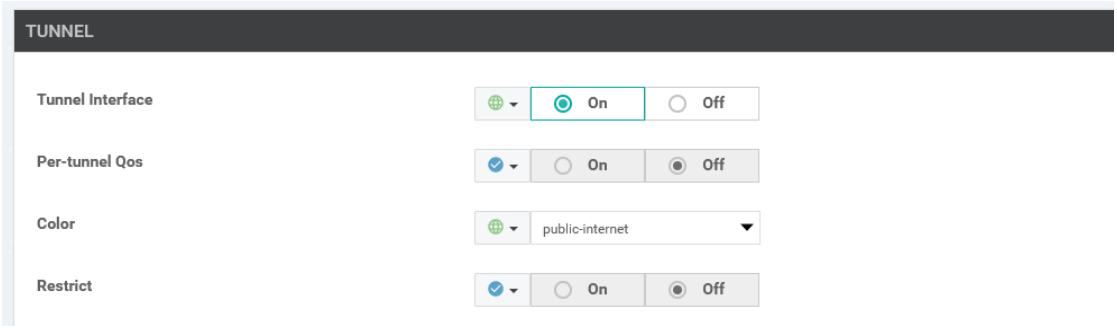
h. Po krátké době se vEdge stane součástí SD-WAN a začne si natahovat konfiguraci z vManage.

3. Opakuj proces začleňování pro zbylá vEdge zařízení. Krom zařízení za NATem, tedy zařízení vEdge11 a vEdge12, tyto budou začleněny až v sekci NAT traversal (průchod NATem). To bude názorná ukázka toho, že nemusí být zařízení přímo připojena k internetu s veřejnými adresami, a jediné na čam záleží je dostupnost.

Obrázek 3.6 znázorňuje jak proces ZTP funguje.



Obrázek 8.38 Nastavení GNS3 pro vnější komunikaci.



Obrázek 8.39 Nastavení GNS3 pro vnější komunikaci.

8.7 Procházení NATem

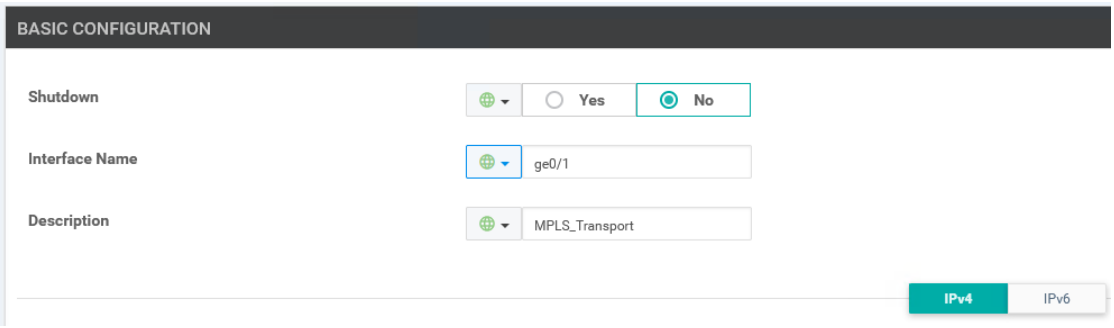
Kontrolér vBond poskytuje funkce pro procházením skrze NAT. Dokonce i samotný vBond může být za NATem, ale musí se jednat o NAT 1:1 na veřejnou adresu. Toto je praktický případ získání veřejné adresy od poskytovatele síťových služeb. Topologie obsahuje dva směrovače, které jsou nakonfigurovány s NATem. První z nich s názvem NAT překládá adresy pro vEdges v lokalitě 10 a druhý s názvem StaticNAT, který právě realizuje statický překlad 1:1 pro druhý kontrolér vBond. Oba směrovače NAT jsou nakonfigurovány v sekci fyzické sítě této kapitoly.

1. Začleňte vBond2 do sítě tradičním, již známím postupem. Všechny potřebné postupy k tomu již byly ukázány v předchozích kapitolách. Jen se ujistěte, že používáte správné adresy, ty jsou v tomto případě neveřejné. vBond2 brána je 192.168.255.1, IP adresa ge0/0 je 192.168.255.2/30 a systémová IP je 100.100.100.6.

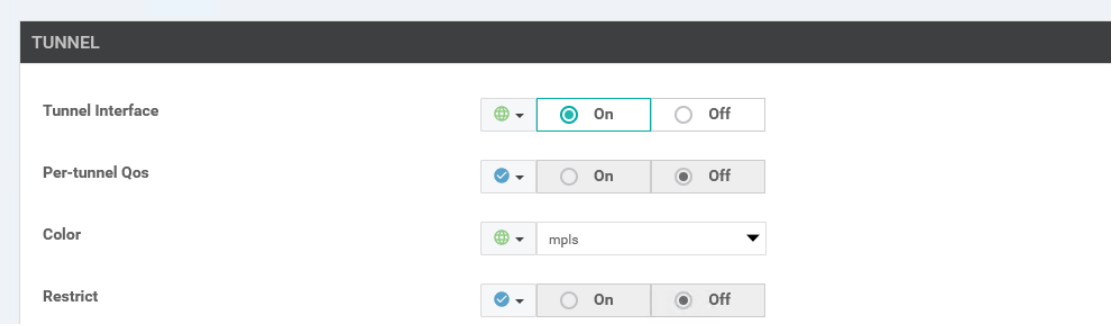
2. Dočasně změňte příkaz ip host vbond.sdwan-fit-lab.org 1.1.1.6 na směrovači brány aby byl dotaz DNS na vbond.sdwan-fit-lab.org správně vyřešen v rámci fyzické sítě. Toto není nutné jelikož zátěž se mezi jednotlivé vBond kontroléry rozděluje.

3. Chcete-li otestovat procházením NATem, dočasně vypněte zařízení vBond1 a poté zkuste připojit zařízení vEdge31. a vEdge32 do sítě. Také však můžete poslat ICMP dotaz, který by se měl také vyřešit správně. To lze provést pomocí ZTP uvedeného v předchozí kapitole. Jednoduše povolte službu SSH na tunelovém rozhraní. Jeden malý rozdíl při začleňování těchto směrovačů do SD-WAN je, že přenos kořenového certifikátu SCP musíte iniciovat ze směrovačů vEdge, protože je třeba vytvořit mapování portů na směrovači NAT. Poté pokračujte s instalací kořenového certifikátu a aktivujte směrovače vEdge pomocí ZTP.

```
vedge11# vshell
vedge11$ scp admin@1.1.1.191:CA.pem .
```



Obrázek 8.40 Nastavení GNS3 pro vnější komunikaci.



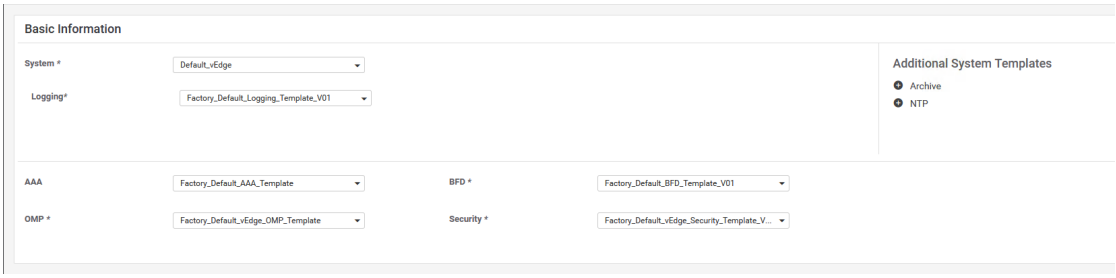
Obrázek 8.41 Nastavení GNS3 pro vnější komunikaci.

4. Obrázek 8.47 ukazuje všechny zahájené kontrolní komunikace z nově začleněného vEdge12.
5. V NATovací tabulce na směrovači NAT je krásně vidět, jaká komunikace byla zahájena a ským během procesu začlenění. Z toho můžeme vyvodit, že orchestrátor vBond správně zprostředkovává procházení NATem. 8.48

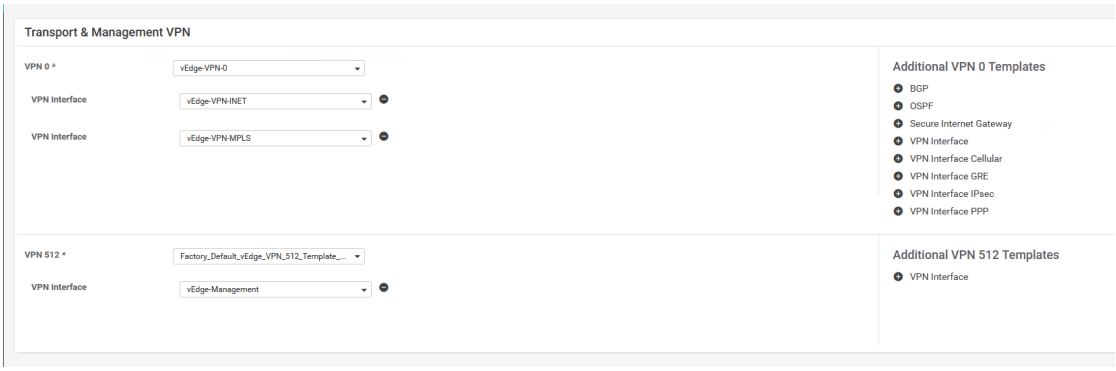
8.8 Směrování

V této poslední sekci Cisco laboratoře se zaměřím na konfiguraci statických směrů a připojený OMP route advertisement v kontrolní vrstvě. Obrázek 8.49 zobrazuje výstup z příkazu „show omp peers“ a objasňuje jak OMP relace jsou vytvářeny. Lokalita 40 bude využívat inzerování statických tras a zbytek si vystačí pouze s připojenými trasami, protože mezi nimi nejsou žádné skoky na třetí vrstvě. Sítě, které se mají inzerovat, jsou na straně služeb směrovačů vEdge. Dvě sítě VLAN 10 a 20 s odpovídajícími servisními sítěmi VPN 10 a 20. Cílem je mít plnou dosažitelnost těchto sítí na straně služeb přes WAN.

1. Je třeba přidat další konfiguraci směrovačům vEdge. Toho se dosáhne přidáním na nakonfigurované předlohy vlastností zařízení. To znamená, že všechny vEdge by měly být řízené vManaged a mít připojenou šablonu. Pokud tomu není, tak se podívejte na předchozí sekce a opravte to.
2. Začněte vytvořením nové předlohy vlastností VPN-VPN použité pro VPN10.
   a. V základní konfigurační části šablony není vyžadováno nic zvláštního. Jen několik zřejmých globálních proměnných pro ID VPN a popis. 8.50



Obrázek 8.42 Nastavení GNS3 pro vnější komunikaci.



Obrázek 8.43 Nastavení GNS3 pro vnější komunikaci.

   b. Důležitou částí předlohy je část pro inzerci OMP. Výběrem možnosti inzerovat statické a připojené směry způsobí, že připojené a statické trasy naučené zařízením vEdge ve službě VPN10 budou inzerovány přes řídicí vrstvu.8.51
   c. Pro lokalitu 40 bude zapotřebí statická trasa. Pro tento účel vytvoříme 2 volitelné proměnné specifické pro zařízení v části IPv4 směry šablony vlastností VPN10, jak je znázorněno na obrázku. 8.52

3. Stejný postup aplikujte pro konfiguraci VPN20. Namísto vytváření nových předloh, stačí zkopírovat vytvořenou předlohu pro VPN10 a editovat globální proměné a přejmenovat parametry specifické pro zařízení.

4. 8.53

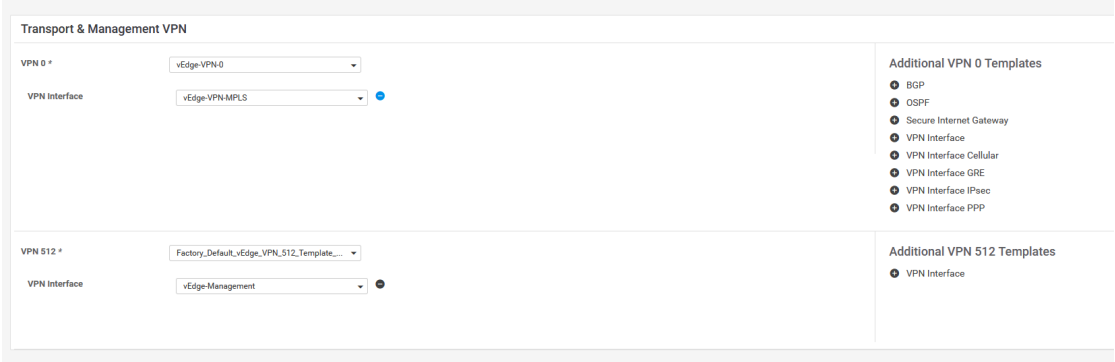
5. Dále vytvoříme další předlohu VPN-VPN Ethernet rozhraní pro dílčí rozhraní ge0/2.10 sloužící jako koncový bod pro VLAN 10. 8.54

6. V části pro pokročilé je třeba nastavit IP MTU jako globální proměnnou a změnit její hodnotu na 1496, aby bylo možné použít 4-bytovou hlavičku dot1Q.

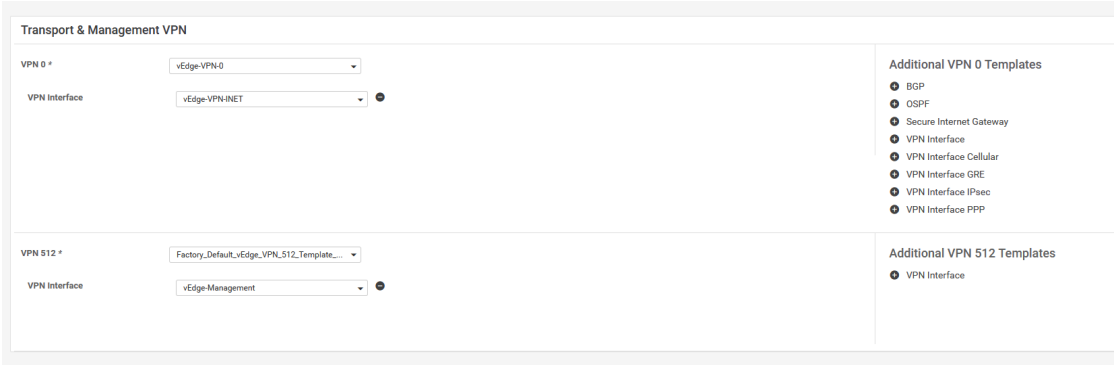
7. Toto zopakujte i pro podrozhraní ge0/2.20

8. Vytvořené předloze vlastností přidejte k již existujícím předlohá vlastnosti zařízení tímto způsobem.

   a. Přidejte rozhraní ge0/2 VPN Interface do VPN0
   b. Přidejte předlohu služby VPN10 a do ní vložte VPN předlohu rozhraní pro rozhraní ge0/2.10.



Obrázek 8.44 Nastavení GNS3 pro vnější komunikaci.



Obrázek 8.45 Nastavení GNS3 pro vnější komunikaci.

c. Přidejte předlohu služby VPN20 a do ní vložte VPN předlohu rozhraní pro rozhraní ge0/2.20.

d. Aktualizujte předlohu.

e. Podle topologie vyplňte požadované proměnné specifické pro zařízení.8.55 Nepovinné proměnné jsou nutné pouze pro vEdge41,8.56 protože se jedná o zařízení, které má mezi inzerovanými sítěmi skok na třetí vrstvě.

f. Propište konfigurace do zařízení vEdge.

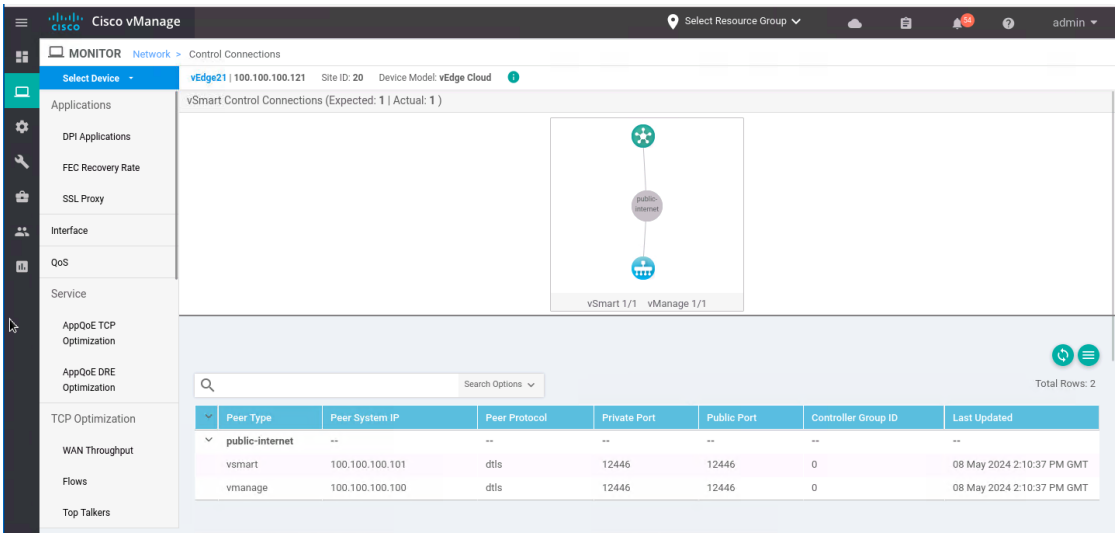
9. Po dokončení budou směrovače vEdge inzerovat všechny místní statické a připojené trasy všem ostatním partnerům v různých lokalitách. Výsledkem je celkem 70 inzerátů sítě. Obrázek 8.57 ukazuje část výstupu inzerovaných tras OMP v řídicí vstvě.

Lze také pozorovat segmentaci výchozí VPN v logické síti. Zařízení na straně různých stran VPN mohou komunikovat pouze se zařízeními na stejné straně VPN, přestože se reálně nacházejí v jiné lokalitě. 8.58

```
INDEX  CHASSIS NUMBER                          SERIAL NUMBER                          VALIDITY  VBOND IP                VBOND PORT  ORGANIZATION NAME  ROOT CERT PATH
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
1      32B48F9C-41A8-0B66-7295-421558CA8A73    122F3404788F41E48B5AD264F01D123C  valid     vbond.sdwan-fit-lab.org  12346       SDWAN-FIT-LAB      /home/admin/CA.pem
2      DDE3981F-B0D1-4DFF-914C-F39A3F98E093    646DC7316A314445956AD8BB534DFE85  valid     vbond.sdwan-fit-lab.org  12346       SDWAN-FIT-LAB      /home/admin/CA.pem
3      A039D608-FF0C-6CFD-63A5-CECCFEC8C50C    D2A74940ECE4460492B582D0E45BE99C  valid     vbond.sdwan-fit-lab.org  12346       SDWAN-FIT-LAB      /home/admin/CA.pem
4      9FDD3D5E-48B5-0199-0BE9-A02752D0BDF6    711D79AE87114961B65F1312CE227AA2  valid     vbond.sdwan-fit-lab.org  12346       SDWAN-FIT-LAB      /home/admin/CA.pem
5      1A5163B2-2EEE-F069-DBCD-6EE7A34E873D    1FF7F13772BA411CA922B69E448370C4  valid     vbond.sdwan-fit-lab.org  12346       SDWAN-FIT-LAB      /home/admin/CA.pem
6      76032E6B-096E-0324-8372-8D86C587FAE1    444D8BF056D940DCA8053F63E36488D3  valid     vbond.sdwan-fit-lab.org  12346       SDWAN-FIT-LAB      /home/admin/CA.pem

ZTP-Server#
```

Obrázek 8.46 Validní zařízení v ZTP.



Obrázek 8.47 Propojení vEdge se zbytkem sítě ZTP.

Souhrn kapitoly Cisco laboratoře

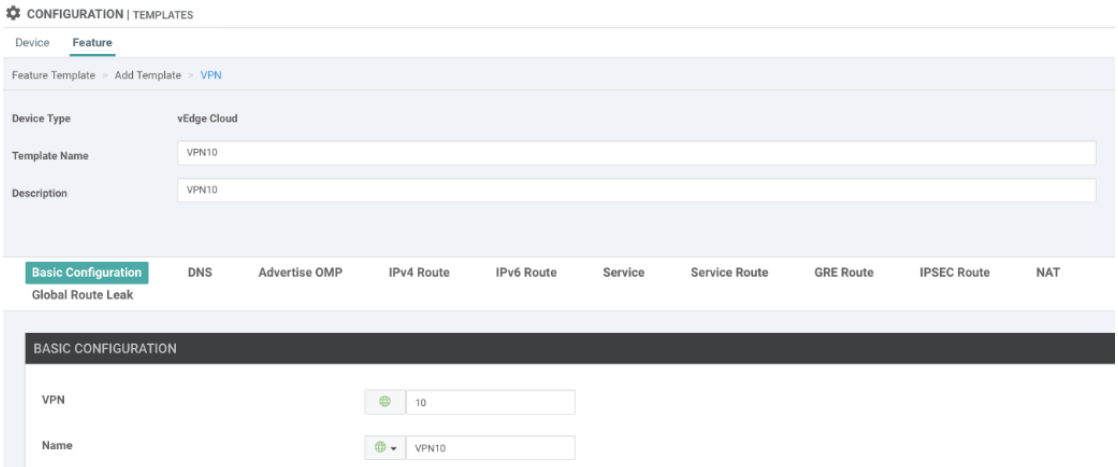
V této kapitole jsme v námi vybrané topologii nasadily Cisco řešení problému SD-WAN. Nejprve bylo nutné nakonfigurovat jednotlivé kontroléry sítě minimální nutnou konfigurací a zařídit kořenové certfikáty napříč sítí. Poté jsme ručně začleníily jednotlivé kontroléry do sítě SD-WAN a přidali sériová čísla zařízení do vManage. Následně jsme postupně různými způsoby začaly začleňovat jednotlivé WAN-edge zařízení do sítě SD-WAN. Nakone jsme vyzkoušeli funkce směrování, segmentace a průchod NATem.

```
udp 1.1.3.251:1038    192.168.1.2:12426  1.1.3.1.12346      1.1.3.1:12346
udp 1.1.3.251:1037    192.168.1.2:12426  1.1.3.2.12346      1.1.3.2:12346
udp 1.1.3.251:1036    192.168.1.2:12426  1.1.3.4.12346      1.1.3.4:12346
udp 1.1.3.251:1035    192.168.1.2:12426  1.1.3.5.12346      1.1.3.5:12346
udp 1.1.3.251:1039    192.168.1.2:12426  1.1.3.21.12346     1.1.3.21:12346
udp 1.1.3.251:1040    192.168.1.2:12426  1.1.3.22.12346     1.1.3.22:12346
udp 1.1.3.251:12426   192.168.1.2:12426  1.1.3.1.12346      1.1.3.1:12346
udp 1.1.3.251:12426   192.168.1.6:12426  1.1.3.2.12346      1.1.3.2:12346
udp 1.1.3.251:12426   192.168.1.6:12426  1.1.3.3.12346      1.1.3.3:12346
udp 1.1.3.251:12426   192.168.1.6:12426  1.1.3.5.12346      1.1.3.5:12346
udp 1.1.3.251:12426   192.168.1.6:12426  1.1.3.21.12346     1.1.3.21:12346
udp 1.1.3.251:12426   192.168.1.6:12426  1.1.3.22.12346     1.1.3.22:12346
```

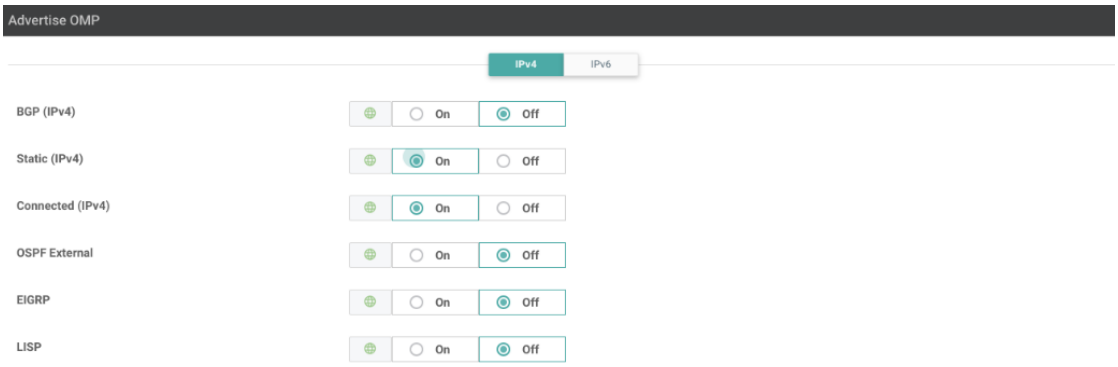
Obrázek 8.48 Průchod natem

```
                                DOMAIN     OVERLAY    SITE
PEER                 TYPE       ID         ID         ID          STATE
------------------------------------------------------------------------
100.100.100.121      vedge      1          1          20          up
100.100.100.122      vedge      1          1          20          up
```

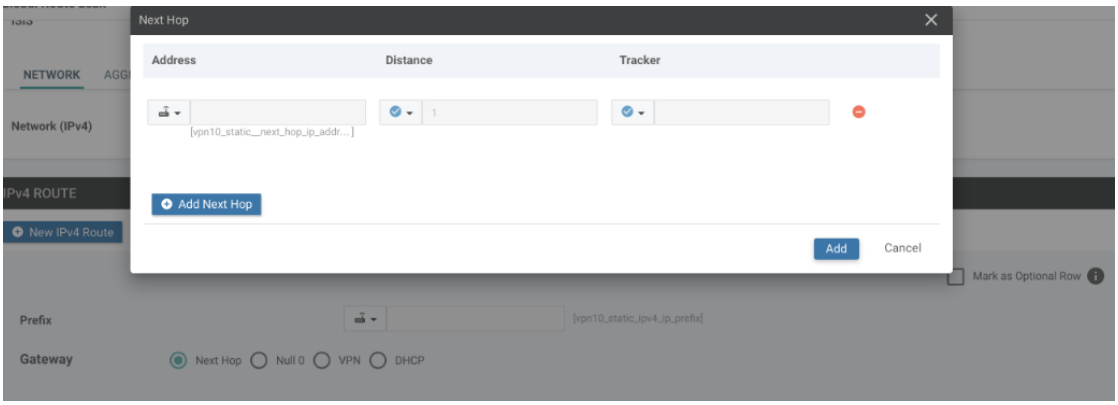
Obrázek 8.49 OMP spojení na vSmart1



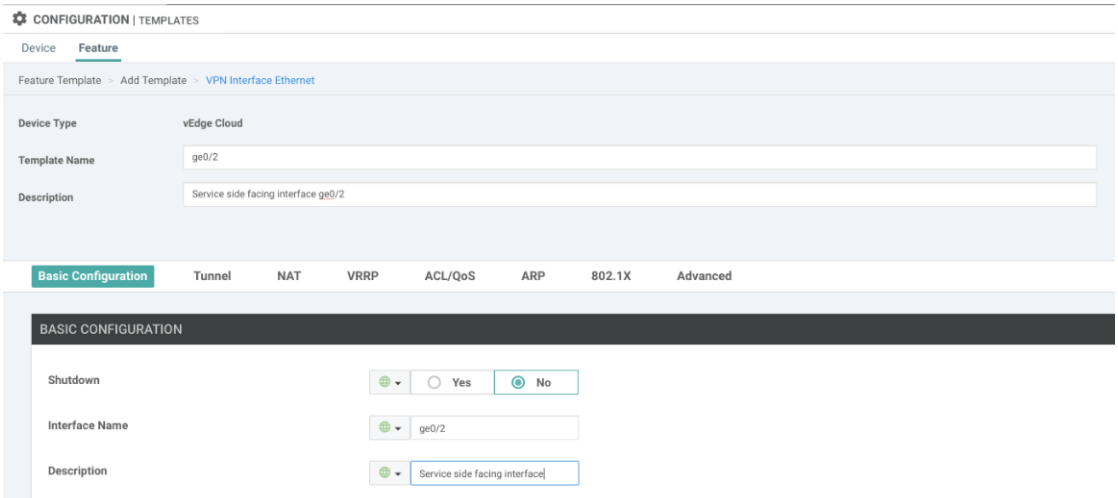
Obrázek 8.50 VPN10 základní konfigurace



Obrázek 8.51 VPN10 inzeruje OMP

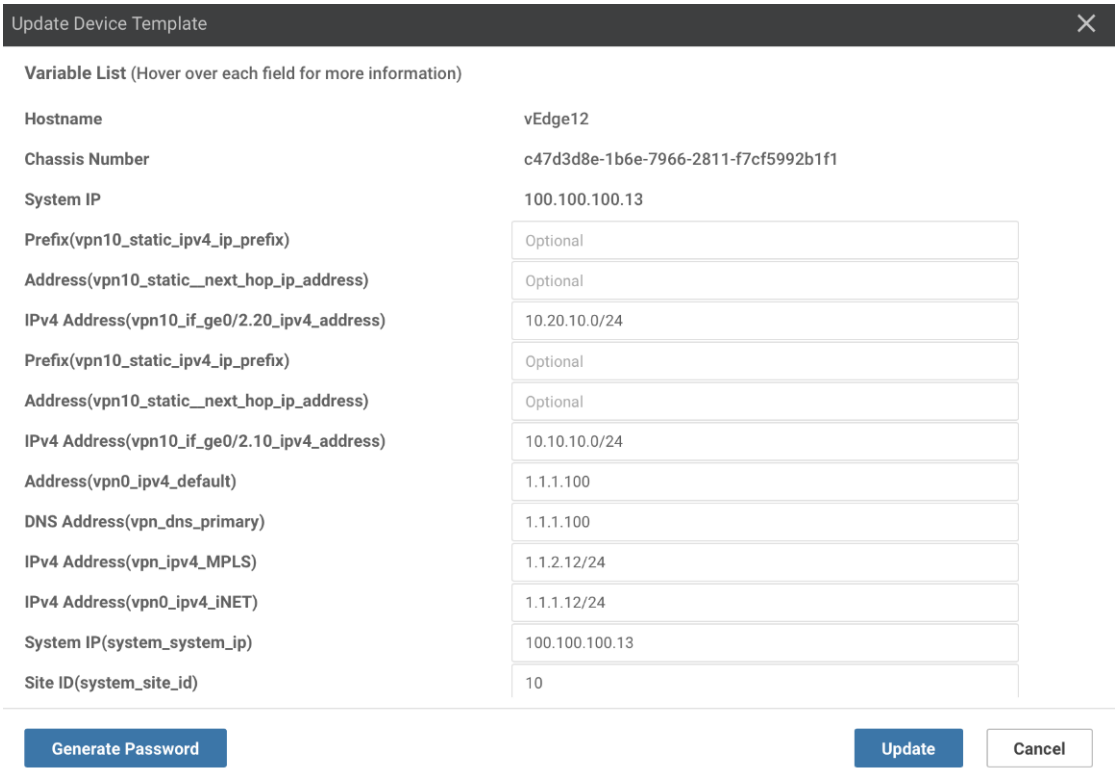


Obrázek 8.52 VPN10 IPv4

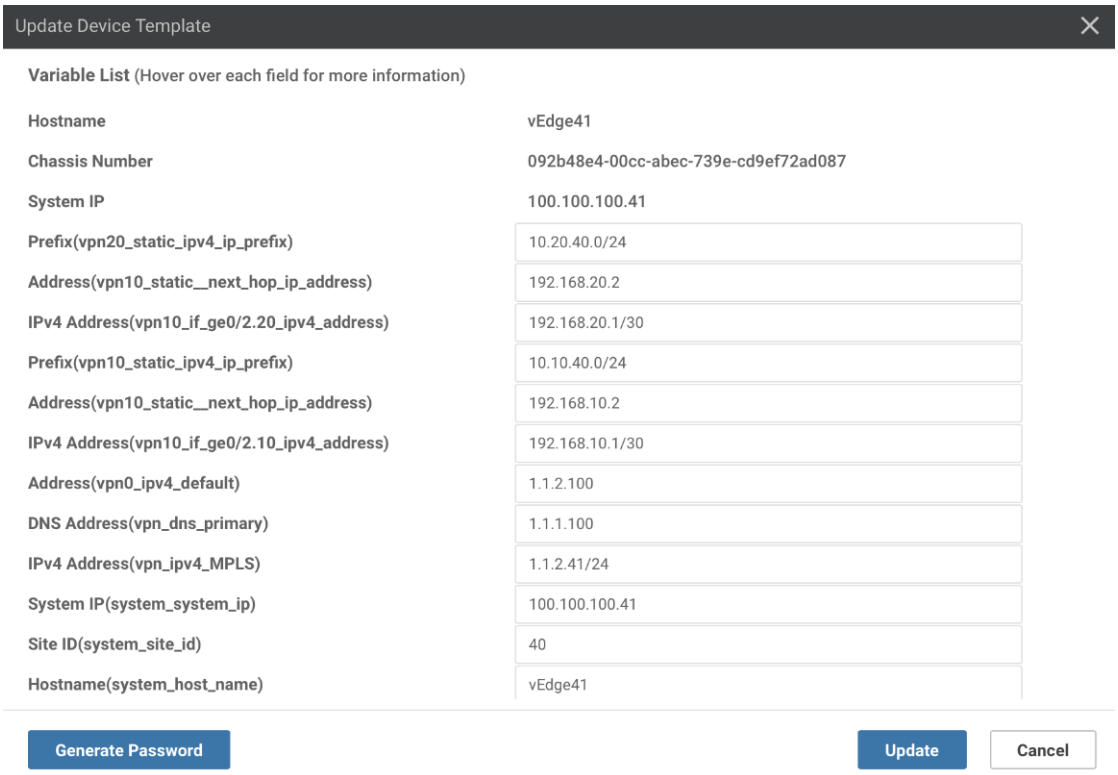


Obrázek 8.53 Základní konfigurace rozhraní ge0/2

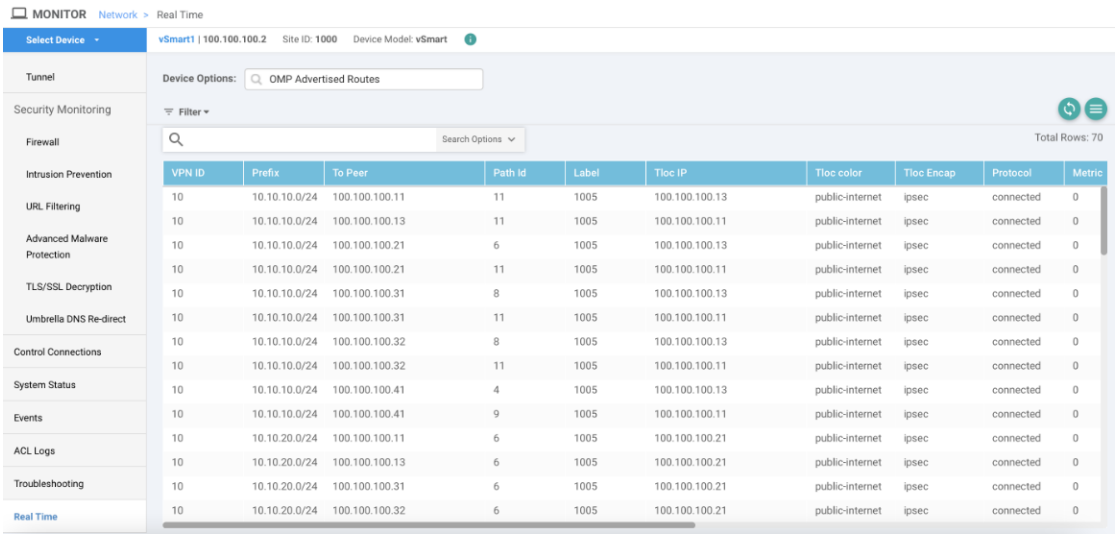
Obrázek 8.54 Základní konfigurace podrozhraní ge0/2.10



Obrázek 8.55 Příklad hodnot specifických pro zařízení na straně připojené služby na zařízení vEdge12



Obrázek 8.56 Příklad hodnot specifických pro zařízení na straně statické služby v zařízení vEdge41



Obrázek 8.57 OMP inzeroval směry

```
vEdge12# ping vpn 10 10.30.10.254
Ping in VPN 10
PING 10.30.10.254 (10.30.10.254) 56(84) bytes of data.
64 byts from 10.30.10.254: icmp_seq=1 ttl=64 time=28.25.ms
64 byts from 10.30.10.254: icmp_seq=2 ttl=64 time=39.47ms
^C
--- 10.30.20.254 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
vEdge12#
vEdge12# ping vpn 10 10.30.20.254
Ping in VPN 10
PING 10.30.20.254 (10.30.20.254) 56(84) bytes of data.
From 127.1.0.2 icmp_seq=1 Destination Net Unreachable
From 127.1.0.2 icmp_seq=2 Destination Net Unreachable
^C
--- 10.30.20.254 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1057ms
```

Obrázek 8.58 VPN segmentace

Kapitola 9
FOSS Laboratoře

Závěrečná kapitola implementace síťové laboratoře pro práci s OpenFlow na zařízení Mikrotik, Cisco a Open vSwitch (OVS). V první části se zaměříme na představení různých kontrolérů a vyzkoušíme jejich funčnost. Dále přstavíme způsoby připojení jednotlivých aktivních prvků sítě do sítě řízené kontrolérem OpenFlow. Nakonec prozkoumáme možsnosti segmentace a různých vlastností OpenFlow řízené topologie.

9.1 Kontroléry OpenFlow

V této sekci se zaměřím na ukázku rozhraní a popis jednotlivých, velmi známích, OpenFlow kontrolérů. Kontroléry které představím jsou OpenDaylight, kontrolér pod záštitou Linux Foundation, Floodligh FOSS kontolér, který není postaven na Apache Karaf a poslední ONOS, také sice s otevřeným zdrojovým kódem ale společnost která ho vyvíjela od něj opustili prospěch síťového programovacího jazyka P4 (Programming Protocol-independent Packet Processors).

9.1.1 OpenDaylight

OpenDaylight je modulární síťový kontrolér využívající OpenFlow a je stále vyvíjen do dnešního dne, pod záštitou Linux Foundation, jako backend pro aplikace. Poslední verze která měla grafické rozhraní dlux, je verze 8.4.

Protože se jedná o aplikaci kontroléru je protřeba ji nainstalovat, včetně jejích závislostí. V sekci kódu je vidět co všechno bylo třeba nainstalovat pro běh aplikace. Následujícíinstlace proběhla na operačním systému Ubuntu 18.04.9.1

Pokud instalace proběhla dobře tak bychom měli vidět na adrese http://localhost:8181/index.html přihlašovací okno.9.1 Výchozí jméno a heslo je admin:admin.

9.1.1.1 Cisco OpenFlow Management App (OFM)

Do roku 2017 byla aktivně vyvíjena nástavba pro OpenDaylight od společnosti Cisco se jménem OpenFlow Management. Poté Cisco přešlo svoji implementaci Cisco Open SDN Controller. Ten však oficiálně roku 2020 zanikl a zda se, že přímí Cisco kontrolér už neexistuje. Naneštěstí Cisco OFM má své zdrojové kódy a aplikace je pořád dostupné, tuďíž je použita v této práci. Vzhledem k tomu, že aplikace slouží jako grafický frontend pro kontrolér OpenDaylight, tak ho budu používat namísto výchozího Opendaylight GUI.9.2

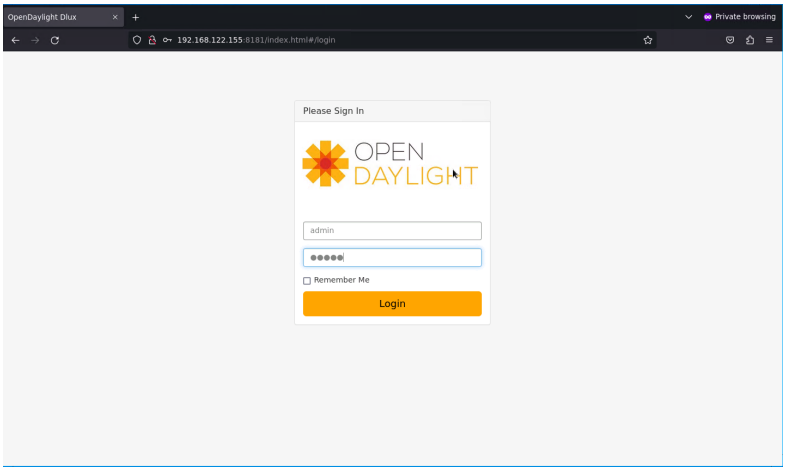
Následující postup 9.2 umožňuje instalaci nástavby OpenDaylight. Výsledkem by měla být spuštěná aplikace na portu 9000. Jedná se o přímé pokračování předchozí sekce, proto ve finále

Výpis kódu 9.1 Instalace Open Daylght

```
apt-get update
apt-get install -y bash-completion software-properties-common \
                      python-software-properties sudo curl ssh git
apt-get install -y mininet tmux wget openjdk-8-jdk npm
nano /etc/ssh/sshd_config  ###(zde zmente PermitRootLogin z \
 PasswordProhibited na yes)
service ssh start
ssh-keygen -t rsa -P ""
echo 'JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64"' >> /etc/environment
. /etc/environment
wget https://nexus.opendaylight.org/content/groups/public/org/ \
opendaylight/integration/distribution-karaf/0.3.0-Lithium/ \
distribution-karaf-0.3.0-Lithium.tar.gz
tar zxvf distribution-karaf-0.3.0-Lithium.tar.gz
curl -sL https://deb.nodesource.com/setup_4.x | sudo -E bash -
apt-get install nodejs
npm install -g grunt-cli

nasleduje cast provadena uvnitr tmux1 {
cd distribution-karaf-0.3.0-Lithium
./bin/karaf
feature:install odl-restconf-all odl-openflowplugin-all \
 odl-l2switch-all odl-mdsal-all odl-yangtools-common webconsole
}

tmux2 {
mn --topo=tree --controller=remote,ip=127.0.0.1,port=6653 \
--switch=ovsk,protocols=OpenFlow13
}
```



Obrázek 9.1 OpenDaylight příhlašovací obrazovka.

Výpis kódu 9.2 Instalace Cisco OFM

```
git clone https://github.com/CiscoDevNet/OpenDaylight-Openflow-App.git
sed -i 's/localhost/<aktualni adresa zarizeni>/g' \
./OpenDaylight-Openflow-App/ofm/src/common/config/env.module.js \

tmux3 {
cd OpenDaylight-Openflow-App
grunt
}
```

budem mít zapnuté aplikace na pozadí, v prostředí tmux, tři.

9.1.2 ONOS

Jak již bylo zmíněno, vývoj síťového operačního systému ONOS byl zrušen ve prospě P4, ale kód zůstává prořád dostupný a je volně k dispozici pro užití. Instalace tohoto systému bylo nejtěžší, jelikož na novějších operačních systémech než je Ubuntu 18.04 se teto software nezpustí, i když instalace zcela totožná. Následuje popis postupu instalace ONOSu. 9.3

Pokud vše proběhlo bez problémů tak by mělo být dostupné grafické rozhraní na adrese http://localhost:8181/onos/ui/index.html. Výchozý přihlašovací údaje jsou onos:rocks. Úvodní obrazovka operaního ssytému vypadá takto.9.3

▶ Poznámka 9.1. Není náhoda, že ONOS i OpenDaylight poslouchají na portu 8181. Je to totiž zapříčiněno tím, že oba softwary jsou postaveny na Apache Karaf.

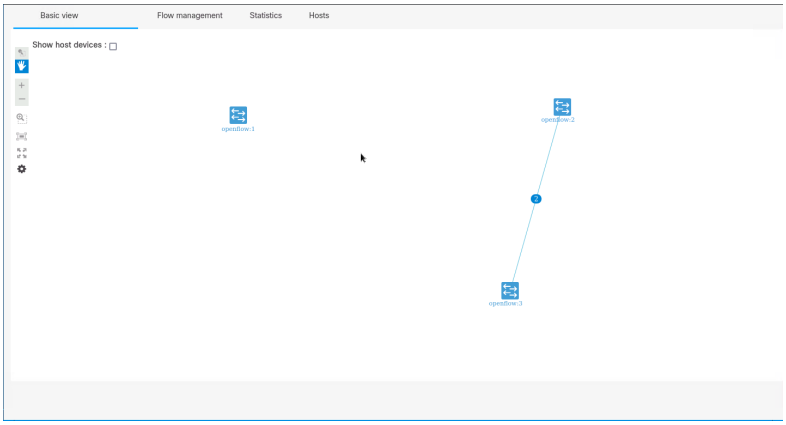
9.1.3 Floodlight

Posledním zde testovaným kontrolérem je kontrolér Floodlight, který, jak bylo zmíněno výše, není postaven na Apache Karaf. Sice není založen na Apache Karaf, zato ale je založen na Javě, stejně jako ostatní. Protože stejně jako ONOS je už několik let nevyvíje, je opět jeho instalace obtížnější. Zde opět bude podrobný návod na jeho nasazení. Zvolený operační systém pro nasazení Floodlight bylo zvoleno Ubuntu 22.04.

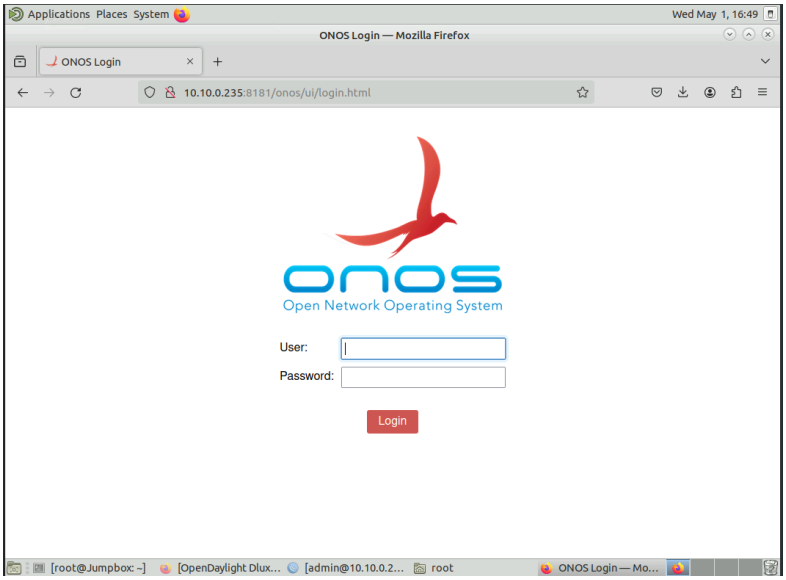
Dalším zásadním rozdílem je, protože tento nástroj není naimplementovaný přes Apache Karaf, že neposlouchá tradičně na portu 8181 prowebové rozhraní, ale na portu 8080. Do webového

Výpis kódu 9.3 Instalace ONOS

```
git clone https://github.com/CiscoDevNet/OpenDaylight-Openflow-App.git
sudo adduser sdn --system --group
sudo apt install openjdk-11-jdk
sudo su
cat >> /etc/environment <<EOL
JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
JRE_HOME=/usr/lib/jvm/java-11-openjdk-amd64/jre
EOL
sudo apt-get install curl
sudo mkdir /opt
cd /opt
export ONOS_VERSION=2.7.0
sudo wget -c https://repo1.maven.org/maven2/org/onosproject/\
onos-releases/$ONOS_VERSION/onos-$ONOS_VERSION.tar.gz
sudo tar xzf onos-$ONOS_VERSION.tar.gz
sudo mv onos-$ONOS_VERSION onos
#otestujeme funcnost instalace pomoci nasledujiciho prikazu
/opt/onos/bin/onos-service start
sudo cp /opt/onos/init/onos.initd /etc/init.d/onos
sudo cp /opt/onos/init/onos.conf /etc/init/onos.conf
sudo cp /opt/onos/init/onos.service /etc/systemd/system/
sudo systemctl daemon-reload
sudo systemctl enable onos
cat >> /opt/onos/options <<EOL
ONOS_USER=sdn
# Optional: add any apps here that you wish to activate by default
ONOS_APPS=
EOL
sudo systemctl {start|stop|status|restart} onos.service
```



Obrázek 9.2 Cisco Openflow manažer.



Obrázek 9.3 Úvodní obrazovka ONOS.

rozhraní se dostaneme tedy přes http://localhost:8080/ui/pages/index.html. Toto webové rozhraní nepoužívá ve výchozímstavu přihlášení podobně jako u případu Cisco OFM.

9.2 Zařízení OpenFlow

Tato sekce se věnuje nastavením jednotlivých zařízení tak, aby byly schopny provozovat OpenFlow v simulačním prostředí. Tyto zařízení Zpracovávají data pro jednotlivých paketech, na které aplikují politiky podle tabulky toku, která se distribuje do sítě z centálního bodu či klastru bodů. Následující obrázek 9.5 znázorňuje, jak by měla vypadat logika zařízení v rámci OpenFlow SDN sítě.

Zařízením, kterým se v této sekci budem věnovat jsou zařízení od společnosti Cisco, Mikrotik a ryze virtuální přepínač s otevřeným zdrojovým kódem Open vSwitch. Těmto zařízením se věnuji z důvodů lehké a levné dostupnosti. Ze zde nezobrazovaných měření mohu potvrdit, že zařízení drží standard a posílají zprávy kontroléru v pravidelný časových okamžicích.

Výpis kódu 9.4 Instalace a nasazení Floodlight

```
sudo -i
apt install git openjdk-8-jdk python2-dev build-essential mininet ant tmux -y
git clone https://github.com/floodlight/floodlight.git
cd floodlight/
git submodule init
git submodule update
ant clean
cd lib
rm -f netty-all-4.0.31.Final.jar libthrift-0.9.0.jar
wget "https://repo1.maven.org/maven2/io/netty/netty-all/4.1.66.Final/
netty-all-4.1.66.Final.jar"
wget "https://repo1.maven.org/maven2/org/apache/thrift/
libthrift/0.14.1/libthrift-0.14.1.jar"
cd -
sed -i -- 's/libthrift-0\.9\.0\.jar/libthrift-0\.14\.1\.jar/;
s/netty-all-4\.0\.31\.Final\.jar/netty-all-4\.1\.66\.Final\.jar/' build.xml
ant clean && ant
git submodule init
git submodule update
tmux1 {
java -jar target/floodlight.jar
# ./floodlight.sh pokud ho upravime k funkcnosti muzem pouzit i tento skrypt.
}
tmux2 {
mn --topo=single,5 --controller=remote,ip=localhost,port=6653
}
```

9.2.1 Mikrotik

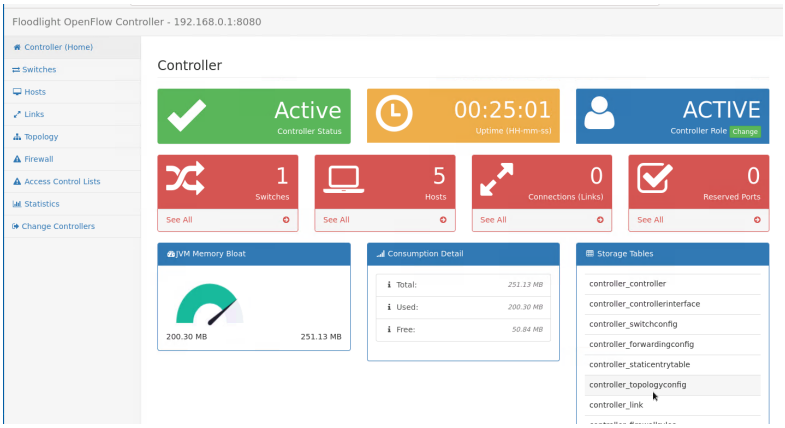
Jedíný představtel zařízení s grafickým rozhraním, jetsli nepočítáme kontroléry. Každé zařízení od Mikrotiku nativně nemá předinstalovaný balíček pro implementaci OpenFlow. Je ale možne ho doinstalovat v rámci extra balíčků z oficiálních stránek Mikrotiku.9.6 Protože obecně virtuální zařízení vrtuálního prostředí nemají předinstalovaný program WinBox pro správu Mikrotik zařízení, je tedy nutno vždy před začátkem laboratoře si vytvořit virtuální stroj, který se připojí vně simulátoru, aby si mohl stáhnout potřebné nástroje pro práci.

Protože podporované obrazy mikrotiku nemusí souhlasit s aktuální LTS verzí je potřeba aktualizovat Mikrotik na nejnovější verzi, buďto Stable nebo Long Term. 9.7 Toto je nutné udělat kvůli kompatibilitě zařízení, jinak balíčky jsme stáhli zbytečně, protože by nebyly kompatibilní s verzí zařízení.

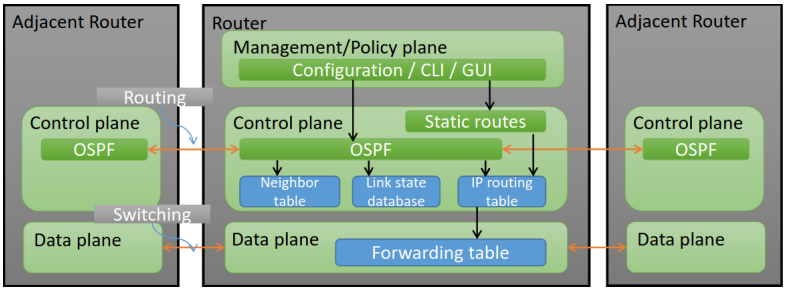
Po úspěčné aktualizaci nás směrovač vyzve k restartování zařízení. V tento okamžik ještě nebudeme restartovat zařízení, ale nahrajeme patřičný softwarový balíček do směrovač. Teprve pak můžeme bez problému restartovat zařízení. Tento postup volíme proto, jelikož kdybychom to neudělaly tak tento proces restartování bychom museli, po uploadu softwarového balíčku, opakovat. 9.8

Necháme zařízení restartovat. Po nějaké době naběhne, připraven k použití v laboratoři. 9.9

Nyní máme aktualizovaný virtuální hardware a nainstalovaný softwarový balíček OpenFlow. Ověříme to tak, že v levé liště možností by se měla objevit ikonka pro sekci dedikované OpenFlow. V této verzi realizuje směrovač OpenFlow přepínač, který zpracovává komunikaci podle tabulky toků.



Obrázek 9.4 Úvodní obrazovka Floodlight.



Obrázek 9.5 Interní logika Openflow zařízení v rámci OpenFlow SDN.

9.2.1.1 Konfigurace OpenFlow

Poté, co jsme úspěšně zvládli nainstalovat softwarový balíček pro funkci OpenFlow, můžeme začít nastavovat OpenFlow. Protokol jako takový nastavit je velmi lehké, stačí vytvořit nový OpenFlow virtuální přepínač, přiřadit mu fyzická rozhraní a na závěr mu nastavit IP adresu kontroléru. V tento okamžik sebude snažit přes konfigurovaná rozhraní najít kontrolér, pokud se kontrolé nepodaří nají tak bude hledatdo nekonečna, v opačné případě se úspěšně připojí k kontroléru. Kontroléru poví informace o sobě a o naučených sousedech a kontrolér následně z těchto informací sestavý pohled topogie.

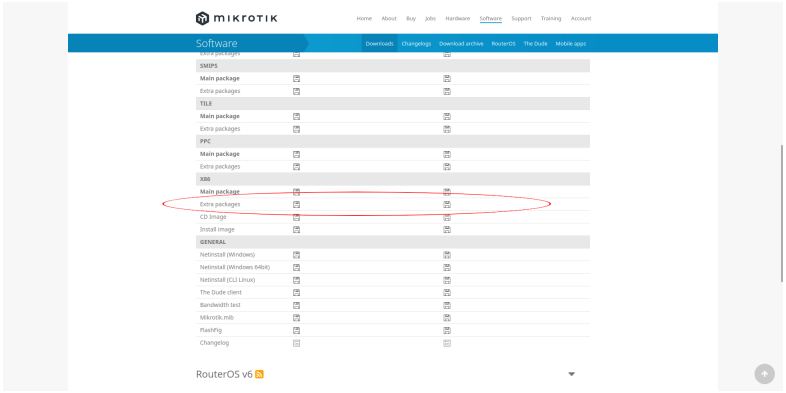
Na následujícím obrázku je příklad toho, jak vypadá směrovač populovanej o toky v tabulce toků. 9.11

9.2.2 Cisco

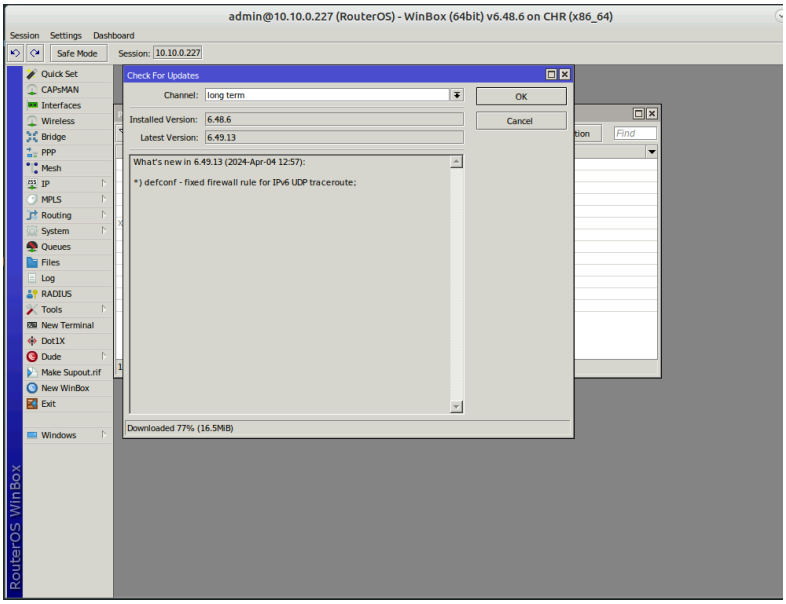
Virtuální přepínač Cisco IOSv, je schopno používat protokol OpenFlow. Protože konfigurace je v mnoha částech stejná jako v případě Mikrotiku, představím jednotlivé příkazy pro konfiguraci OpenFlow přepínače zde. Zároveň tím doplním některé vizuální mezery, které se v sekci Mikrotiku nacházejí.

Následující konfigurace 9.5 popisuje minímlní konfiguraci zařízení Cisco se systémem IOS. Aby bylo možné použít OpenFlow v Ciscu, tak musí existovat možnost tuto vlastnost spustit. Toto se dělá príkazem „feature openflow“.

Je možné, že po konfiguraci bude Cisco hlásit, že se mu nepodaří připojit do zařízení kontroléru. V tento okamžik je dobré se ujistit, že je zprovozněná ipv4 komunikace s kontrolérem. Pokud není je potřeba buďto vyhradit port pro připojení do sítě, nebo OpenFlow switch připojit



Obrázek 9.6 Stažení potřebných balíčků pro zprovoznění OpenFlow.



Obrázek 9.7 Aktualizace Mikrotiku na aktuální verzi.

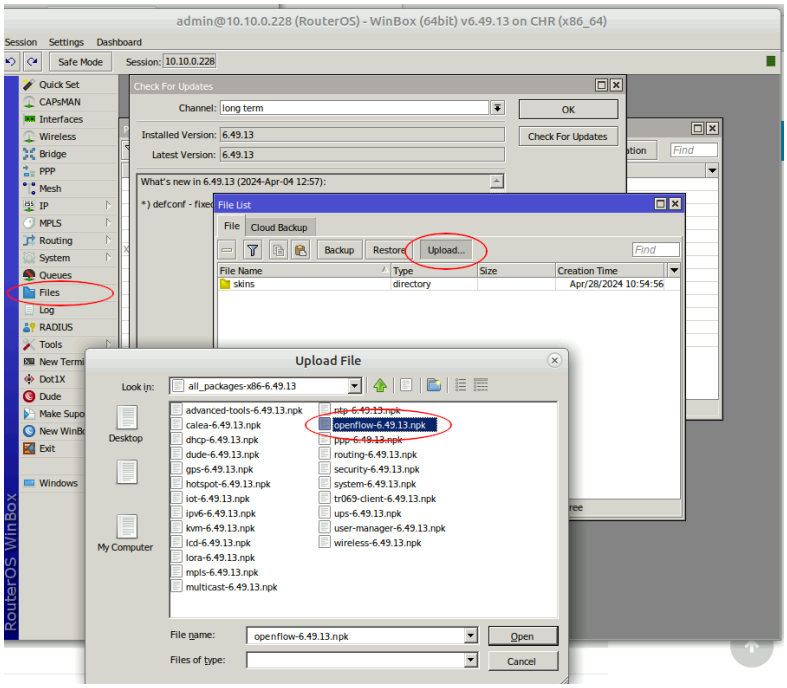
do již existujícího VRF (pomocí parametru vrf v příkazu controller ipv4 ...).

Po úšěšném nastavení konektivity IP s kontrolérem se zařízení Cisco objeví mezi zařízeními topologie.

9.2.3 Open vSwitch

Virtuální platforma vSwitch byla a pořád platforma určená pro datacentra, kde namísto konfigurace segmentací a řízení jednotlivých toků by na fyzickém hardwaru bylo těžko udržitelné, se tento proble přesouvá do softwarové abstrakce pomocí řešení Open vSwitch. V rámci simulačního prostředí, neexistuje jednoduchý způsob, jak dostat dockerovský obraz do EVE-NG, a dělat doporučovaný docker v dockeru je hardwarově náročná úloha.

Nastavení Open vSwitche je ze všech zařízení nejjednodužší na nastavení. Je to tak jednoduché, že je to doslova jen jeden příkaz. . V příkaze br0 reprezentuje rozhraní které má mít kontrolér pod kontrolou. Pokud v tento okamžik existuje konektivita mezi kontrolérem a zařízením, tak v tento okamžik je zařízení úspěšně začleněno do sítě SDN.



Obrázek 9.8 Nahrání balíčku OpenFlow do Mikrotiku.

9.3 Funkcionality OpenFlow SDN

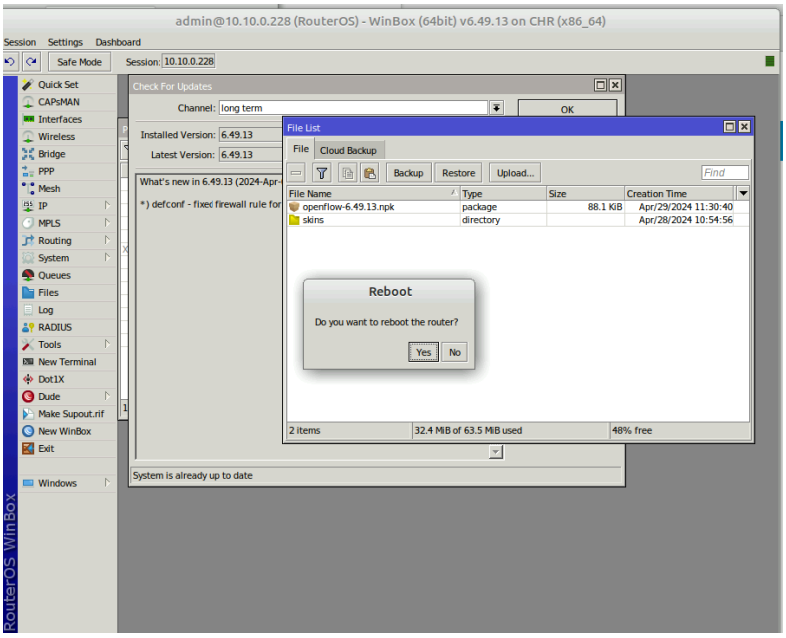
V předchozích sekcí jsme nasadily jednotlivé kontroléry a programovatelné přepínače. Samotné nasazení však není jediný, či dokonce hlavní účel těcto zařízení. V této sekci se zaměříme na implementaci několika vlastností, které nám dodává OpenFlow SDN.

Všechny konfigurace zařízení se do zařízení dostávají přes konfigurační tunely s kontrolérem, ale samotný kotrolér nemusí vědět o tom co a jak se to nastavuje. K tomu právě slouží skrypty v jazyce YANG,9.12

Protože se v mnoha místech často přechází na komerční způsoby realizace softwarově definovaných sítí, a mnoho podniků aktivně bojuje s vývojem těchto technologií, tak tyto technologie kvůli tomu postupem času zaostávají s dobou a tím se snižje jejich podíl na dění věcí ve světe. Proto v této kapitole nepůjdu do takové hloubky jako v kapitole ohledně Cisco SD-WAN sítí a jen vyjmenuji klíčové vlastnosti, ve kterých sítě řízené technologií OpenFlow vynikaly.

Těmito vlastnostmi jsou:

- Odolnost vůči DDoS útoků.
- Skupinové tabulky.
- Vysoká dostupnost služeb.
- Možnost limtování hardwarových prostředů zařízení.
- OpenFlow fyzické, logické a rezervované porty zařízení (NIC a podobně.).

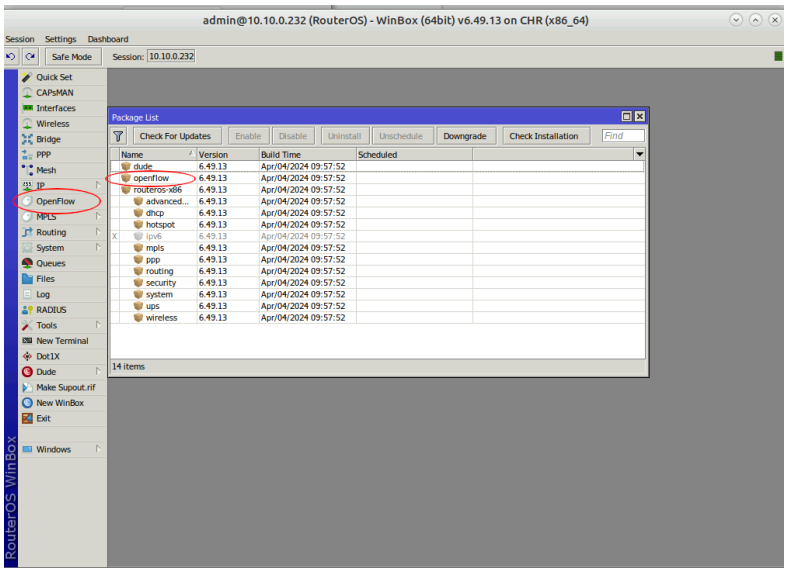


Obrázek 9.9 Restartování Mikrotiku, aby začlenil nově instalovaný balíček.

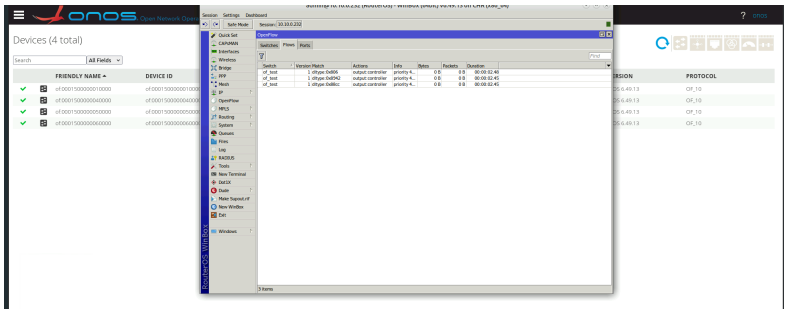
Souhrn kapitoly FOSS laboratoře

V této kapitole jsme prošli nastavení vybraných kontrolérů implementující standard OpenFlow, otevřeného standardu pro implementaci softwarově definovaných sítí. Následně jsme nastavily zařízení od několika různých výrobců a začenily je do sítě OpenFlow SDN. Nakonec jsme prošli možnosti nastavování zařízení přes kontroléry. Bohužel tato technologie pomalu mizí z míst, kde se nedá virtualizovat a tuďíž by musel existovat hardware, který je nativně kompatibilní s OpenFlow standartem. Tento harware se ukazuje v mnoha místech dražší jak na nákup, tak na provoz.[1]

[1] Mé osobní podezření je na vině implementace OpenFlow pipeliny.



Obrázek 9.10 Kontrola instalace OpenFlow.



Obrázek 9.11 Mikrotik v OpenFlow kontroléru ONOS.

Výpis kódu 9.5 Minimální konfigurace OpenFlow na Cisco IOS zařízení

```
Switch> enable
Switch# configure terminal
Switch(config)# feature openflow
Switch(config)# openflow
Switch(config-openflow)# switch 1 pipeline 1
Switch(config-openflow-switch)# controller ipv4 [Controller address]
                                port 6653 security none # volitelne
                                vrf [jmeno vrf]
Switch(config-openflow-switch)# of-port interface [GigabitEthernet0/1]
                                # Dulezite nepsat zkratky
Switch(config-openflow-switch)# of-port interface ...
                          # pokud chceme do switche pridat dalsi porty
Switch(config-openflow-switch)# default-miss controller
Switch(config-openflow-switch)# protocol-version negotiate
Switch(config-openflow-switch)# datapath-id
                                [64-bitova hodnota v hexa zapisu]
Switch(config-openflow-switch)# statistics collection-interval 7
Switch(config-openflow-switch)# end
```

```
/ # ovs-vsctl set controller br0 tcp:[ip adresa kontroleru]:6653
```

```
"flow": [
    {
        "table_id": "0",
        "id": "12345",
        "priority": "1000",
        "hard-timeout": "60",
        "match": {
            "in-port": "openflow:3:1"
        },
        "instructions": {
            "instruction": [
                {
                    "order": 0,
                    "apply-actions": {
                        "action": [
                            {
                                "order": 0,
                                "drop-action": {}
```

Obrázek 9.12 Ukázka jazyka YANG.

Závěr

Tato práce měla za cíl zaměření se na simulaci a nasazením softwarové-definovaných sítí ve virtuálních prostředí a srovnáním jednotlivých řešení softwarově-definovaných sítí od různých výrobců, jako jsou Cisco, Huawei a Mikrotik. V prvních kapitolách byly představeny a podrobně popsány jednotlivá řešení a bylo zjištěno, zdali by se daly implementovat jednotlivá řešení v simulovaných prostředích. Jak se pak nakonec zjistilo v případě Huawei bylo klíčovým faktorem nedostupnost funkčních zařízení jako důvod, proč se toto řešení nedalo implementovat. Jelikož se jednalo o analýzu ve virtuálních prostředí, práce se zaměřila na implementaci a analýzu především na SD-WAN v různých simulátorech. Jelikož v komerčních prostředích se ne vždy můžeme setkat pouze s jedním výrobcem řešení softwarově-definovaných sítí, je tudíž potřeba, aby jednotlivé stroje, ne nutně stejného výrobce, fungovali pod jedním ekosystémem. Nakonec se zjistilo, že protokoly OpenFlow sice umí nastavovat zařízení a přímo ovliňovat datový tok, způsob jakým ho modifikuje je neslučitelný s Cisco řešení SD-WAN. v Práci byly prodrobně vysvětleny nasazení softwarově-definovaných sítí ve virtuálních prostředí různých řešní a bylo zjištěno, zdali existuje možná interoperabilita. Protože Cisco SD-WAN a OpenFlow řeší takřka jinou úlohu, i když by se dala jedna technologie přizpusobit druhé, cílový výsledek, tudíž existence dvou různých kontrolérů na různé problémy by do sítě přineslo problémi. Nakonec v každé laboratoři byli implementovány nějaké vlastnosti dané technologie. Všechny postupy zde jsou detailně popsány pro případnou replikaci řešení pro studijní či osobní účely.

Práce stále nabízí místa, se dá podívat do větší hloubky, jako je možnost nasazení a začlenění prvů jako je CUCM do sítě SDN Cisca, nebo prozkoumat způsoby generování potřebných sériových čisel k virtuální edge zařízením pro účely výuky.

Bibliografie

1. LANČA, Matěj. *Analýza a implementace simulovaného prostředí pro SDN*. Bakalářská práce FIT ČVUT, 2022.
2. EVE-NG [online]. [B.r.]. [cit. 2023-03-04]. Dostupné z: https://www.eve-ng.net.
3. SHAH, Nirav. *Security-driven networking* [online]. [B.r.]. [cit. 2023-03-05]. Dostupné z: https://www.networkworld.com/article/969930/sd-wan-enables-secure-seamless-and-superior-user-experience-for-the-cloud-on-ramp.html.
4. BHARDWAJ, Rashmi. *Introduction to Eve-NG, GNS3 and VIRL* [online]. [B.r.]. Dostupné také z: https://ipwithease.com/gns3-vs-eve-ng-vs-virl/.
5. *Cisco Recommended SD-WAN Software Versions for Controllers and WAN Edge Routers* [online]. [B.r.]. Dostupné také z: https://www.cisco.com/c/en/us/support/docs/routers/sd-wan/215676-cisco-tac-and-bu-recommended-sd-wan-soft.html.
6. GOOLEY, Jason; YANCH, Dana; SCHUEMANN, Dustin; CURRAN, John. *Cisco Software-Defined Wide Area Networks: Designing, Deploying and Securing Your Next Generation WAN with Cisco SD-WAN*. Cisco Press, 2020. ISBN 978-0-13-653317-7.
7. *Cisco DNA Center 2.3.7.0 on ESXi Deployment Guide* [online]. [B.r.]. Dostupné také z: https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/dna-center/dna-center-va/esxi/2-3-7/deployment-guide/b_cisco_dna_center_2370_on_esxi_deployment_guide.html.
8. *Cisco Identity Services Engine Installation Guide, Release 3.3* [online]. [B.r.]. Dostupné také z: https://www.cisco.com/c/en/us/td/docs/security/ise/3-3/install_guide/b_ise_installationGuide33/b_ise_InstallationGuide33_chapter_2.html.
9. *Cisco DNA Center Data Sheet* [online]. [B.r.]. Dostupné také z: https://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/dna-center/nb-06-dna-center-data-sheet-cte-en.html.
10. *Cisco Catalyst 9800-CL Wireless Controller for Cloud Data Sheet* [online]. [B.r.]. Dostupné také z: https://www.cisco.com/c/en/us/products/collateral/wireless/catalyst-9800-cl-wireless-controller-cloud/nb-06-cat9800-cl-cloud-wirel-data-sheet-ctp-en.html.
11. *SD-WAN* [online]. [B.r.]. Dostupné také z: https://support.huawei.com/enterprise/en/enterprise-network-solution/sd-wan-pid-22584369?category=product-documentation-sets.
12. *Huawei CloudCampus Solution vs. Cisco DNA Solution* [online]. [B.r.]. Dostupné také z: https://e.huawei.com/sa/material/networking/633bc208c21d493ea7167b59c1d67347.

13. MCKEOWN, Nick; ANDERSON, Tom; BALAKRISHNAN, Hari; PARULKAR, Guru; PETERSON, Larry; REXFORD, Jennifer; SHENKER, Scott; TURNER, Jonathan. OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.* 2008, roč. 38, č. 2, s. 69–74. ISSN 0146-4833. Dostupné z DOI: 10.1145/1355734.1355746.

14. *OpenFlow Switch Specification* [online]. [B.r.]. Dostupné také z: https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf.

15. NUNES, Bruno Astuto A.; MENDONCA, Marc; NGUYEN, Xuan-Nam; OBRACZKA, Katia; TURLETTI, Thierry. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. *IEEE Communications Surveys and Tutorials*. 2014, roč. 16, č. 3, s. 1617–1634. Dostupné z DOI: 10.1109/SURV.2014.012214.00180.

16. XIA, Wenfeng; WEN, Yonggang; FOH, Chuan Heng; NIYATO, Dusit; XIE, Haiyong. A Survey on Software-Defined Networking. *IEEE Communications Surveys and Tutorials*. 2015, roč. 17, č. 1, s. 27–51. Dostupné z DOI: 10.1109/COMST.2014.2330903.

17. PFAFF, Ben; PETTIT, Justin; KOPONEN, Teemu; JACKSON, Ethan; ZHOU, Andy; RAJAHALME, Jarno; GROSS, Jesse; WANG, Alex; STRINGER, Joe; SHELAR, Pravin; AMIDON, Keith; CASADO, Martin. The Design and Implementation of Open vSwitch. In: *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*. Oakland, CA: USENIX Association, 2015, s. 117–130. ISBN 978-1-931971-218. Dostupné také z: https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/pfaff.

18. GUDE, Natasha; KOPONEN, Teemu; PETTIT, Justin; PFAFF, Ben; CASADO, Martín; MCKEOWN, Nick; SHENKER, Scott. NOX: towards an operating system for networks. *ACM SIGCOMM computer communication review*. 2008, roč. 38, č. 3, s. 105–110.

19. KAUR, Sukhveer; SINGH, Japinder; GHUMMAN, Navtej Singh. Network programmability using POX controller. In: *ICCCS International conference on communication, computing & systems, IEEE*. sn, 2014, sv. 138, s. 70.

20. SHIMONISHI, Hideyuki; TAKAMIYA, Yasuhito; CHIBA, Yasunobu; SUGYO, Kazushi; HATANO, Youichi; SONODA, Kentaro; SUZUKI, Kazuya; KOTANI, Daisuke; AKIYOSHI, Ippei. Programmable network using OpenFlow for network researches and experiments. In: *Proc. 6th International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2012)*. 2012, s. 164–171.

21. *Ryu: component-based software defined networking framework.* [Online]. [B.r.]. Dostupné také z: https://github.com/faucetsdn/ryu.

22. *What Is OpenFlow?* [Online]. [B.r.]. Dostupné také z: https://info.support.huawei.com/info-finder/encyclopedia/en/OpenFlow.html.

23. *Nante-WAN: SD-WAN just for fun.* [Online]. [B.r.]. Dostupné také z: https://github.com/upa/nante-wan.

24. *Simulating Inter Branch Offices Networking Using FRRouting and Docker Containers.* [Online]. [B.r.]. Dostupné také z: https://barrihs.medium.com/simulating-inter-branch-offices-networking-using-frrouting-and-docker-containers-46f6e4bca935.

25. *RouteFlow: an open source project to provide virtualized IP routing services over OpenFlow enabled hardware.* [Online]. [B.r.]. Dostupné také z: https://routeflow.github.io/RouteFlow/.

26. *What Is Open vSwitch?* [Online]. [B.r.]. Dostupné také z: https://docs.openvswitch.org/en/latest/intro/what-is-ovs/.

27. *VyOS* [online]. [B.r.]. Dostupné také z: https://distrowatch.com/table.php?distribution=vyos.

28. *Review: 6 slick open source routers* [online]. [B.r.]. Dostupné také z: https://www.infoworld.com/article/3106865/review-6-slick-open-source-routers.html.
29. *VyOS official website* [online]. [B.r.]. Dostupné také z: https://vyos.io/vyos-platform.
30. *OpenWrt Project* [online]. [B.r.]. Dostupné také z: https://openwrt.org/.
31. *OpenDaylight Controller Overview* [online]. [B.r.]. Dostupné také z: https://docs.opendaylight.org/en/latest/user-guide/opendaylight-controller-overview.html.
32. *ONOS (Open Network Operating System)* [online]. [B.r.]. Dostupné také z: https://www.techtarget.com/searchnetworking/definition/ONOS-Open-Network-Operating-System.
33. *What Is a Floodlight Controller?* [Online]. [B.r.]. Dostupné také z: https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/what-is-sdn-controller/openflow-controller/what-is-floodlight-controller/.
34. *Cisco SD-WAN Design Guide* [online]. [B.r.]. Dostupné také z: https://www.cisco.com/c/en/us/td/docs/solutions/CVD/SDWAN/cisco-sdwan-design-guide.html.
35. *Cisco SD-WAN vEdge Routers Data Sheet* [online]. [B.r.]. Dostupné také z: https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/sd-wan/nb-07-vedge-routers-data-sheet-cte-en.html.
36. MATTHEWS, Philip; ROSENBERG, Jonathan; WING, Dan; MAHY, Rohan. *Session Traversal Utilities for NAT (STUN)* [RFC 5389]. RFC Editor, 2008. Request for Comments, č. 5389. Dostupné z DOI: 10.17487/RFC5389.
37. *What is the Huawei SD-WAN Approach?* [Online]. [B.r.]. Dostupné také z: https://www.sdxcentral.com/networking/sd-wan/definitions/software-defined-sdn-wan/sd-wan-vendors-overview/huawei-sd-wan/.
38. *What Is Overlay Network?* [Online]. [B.r.]. Dostupné také z: https://info.support.huawei.com/info-finder/encyclopedia/en/Overlay+network.html.
39. *Huawei Overlay Network design* [online]. [B.r.]. Dostupné také z: https://support.huawei.com/hedex/hdx.do?docid=EDOC1100214809&id=EN-US_TOPIC_0000001160545107.
40. *What is OpenFlow?* [Online]. [B.r.]. Dostupné také z: https://network-insight.net/2014/09/25/what-is-openflow/.
41. *Wireshark: The world's most popular network protocol analyzer* [online]. [B.r.]. Dostupné také z: https://www.wireshark.org/.
42. *Grafolean: Easy to use monitoring system* [online]. [B.r.]. Dostupné také z: https://github.com/grafolean/grafolean.
43. *elastiflow: Observability and Security Analytics for Modern Networks* [online]. [B.r.]. Dostupné také z: https://www.elastiflow.com/.
44. *EVE-NG PE Professional Edition Cookbook* [online]. [B.r.]. Dostupné také z: https://www.eve-ng.net/index.php/documentation/professional-cookbook/.
45. *gns3* [online]. [B.r.]. Dostupné také z: https://docs.gns3.com/docs/.

Obsah příloh

- `readme.txt` ..... stručný popis obsahu média
- `labs.tar.zst` ..... archív s exportama laboratoří
- `images.tar.zst` ..... archív s obrazy zařzení pro EVE-NG a GNS3
- `thesis-src.tar.zst` ..... archív zdrojových souborů práce
- `thesis.pdf` ..... text práce ve formátu PDF