Univerzita Karlova

Pedagogická fakulta

Katedra informačních technologií a technické výchovy

BAKALÁŘSKÁ PRÁCE

Nasazení, správa a monitoring rozsáhlých přístupových Wi-Fi sítí s využitím open-source technologií

Deployment, management and monitoring of large-scale access Wi-Fi networks using open-source technologies

David Zálešák

Vedoucí bakalářské práce:	PhDr. Martin Stejskal
Studijní program:	Specializace v pedagogice
Studijní obor:	Informační technologie se zaměřením na vzdělávání

Praha 2024

Odevzdáním této bakalářské práce na téma Nasazení, správa a monitoring rozsáhlých přístupových Wi-Fi sítí s využitím open-source technologií potvrzuji, že jsem ji vypracoval pod vedením vedoucího práce samostatně za použití v práci uvedených pramenů a literatury. Dále potvrzuji, že tato práce nebyla využita k získání jiného nebo stejného titulu.

Praha, 2. prosince 2024 Podpis autora

Děkuji především svému vedoucímu, PhDr. Martinu Stejskalovi, za vstřícnost, odborné vedení a trpělivost, které mi poskytoval během zpracování této práce.

Rovněž bych chtěl poděkovat Bc. Emilu Milerovi za cenné náměty, podněty k vylepšení, korektury a konstruktivní připomínky.

Velké díky patří také mému otci za jeho neustálou podporu, povzbuzení a pomoc, která mi byla oporou po celou dobu psaní této práce.

Abstrakt

Tato bakalářská práce se zabývá automatizovaným nasazením, správou a monitoringem rozsáhlých Wi-Fi sítí. Uvádí základní pojmy z počítačových sítí v kontextu autorizace uživatelů a přístupových Wi-Fi sítí. Popisuje sadu standardů Wi-Fi, fungování těchto standardů a autorizační metody. Následně je pomocí vybraného open-source softwaru implementována modelová přístupová Wi-Fi síť na základě deklarovaných požadavků. Tato modelová implementace je v závěru vyhodnocena.

Klíčová slova

802.1x, eduroam, Wi-Fi, automatizace, správa, monitoring, open-source

Abstract

This bachelor thesis deals with the automated deployment, management and monitoring of large-scale Wi-Fi networks. It introduces basic concepts from computer networks in the context of user authorization and Wi-Fi access networks. It describes a set of Wi-Fi standards, the functioning of these standards and the authorization methods. Subsequently, a model Wi-Fi access network is implemented using selected open-source software based on the declared requirements. This model implementation is evaluated at the end.

Keywords

802.1x, eduroam, Wi-Fi, automation, management, monitoring, open-source

Obsah

Úvod

Rostoucím počtem mobilních zařízení a jejich integrací do každodenního života vznikají stále větší nároky na přístupové Wi-Fi sítě. Uživatelé požadují nejen rychlé a stabilní připojení, ale také široké pokrytí a vysokou úroveň zabezpečení.(Rodrigues, 2024)

Větší pokrytí vyžaduje větší počet přístupových bodů, což představuje nové výzvy pro správu sítí. Administrátoři musí čelit stále většímu objemu práce při správě těchto zařízení a i přes rosoucí počet připojených zařízení musí garantovat bezpečnost sítě. Správa sítě se tímto stáva stále složitější, roste potřeba automatizace a získání přehledu nad infrastrukturou.

Tyto rostoucí požadavky a výzvy kladou důraz na moderní přístupy k ověřování uživatelů sítě, automatizaci a efektivitě správy síťové infrastruktury.

Cíle práce

Hlavním cílem této práce je navrhnout a implementovat modelovou Wi-Fi síť, která splňuje předem definované požadavky. Dílčím cílem je zajistit efektivní monitorování provozu této sítě. Pro dosažení optimální implementace je provedena analýza dostupného softwaru, na jejímž základě je vybráno nejvhodnější řešení.

Struktura práce

Tato práce je strukturována do několika částí, které postupně rozvíjejí problematiku návrhu, implementace a hodnocení modelové Wi-Fi sítě.

První část je zaměřena na teoretické základy počítačových sítí. Popisuje základní principy jejich fungování, síťovou architekturu a současné technologie používané v oblasti síťových zařízení, čímž vytváří potřebný kontext pro další kapitoly.

Druhá část práce se zabývá technologiemi Wi-Fi. Představuje standardy Wi-Fi, jejich fungování a principy zabezpečení, které jsou důležité pro bezpečný provoz moderních bezdrátových sítí.

Ve třetí části jsou formulovány požadavky na modelovou implementaci rozsáhlé Wi-Fi sítě v rámci projektu `eduroam`. Tato část se věnuje specifikaci funkcionalit a technických kritérií, která musí navrhovaná síť splňovat.

Čtvrtá část práce se zaměřuje na výběr vhodných technologií pro referenční implementaci. Zkoumá

dostupné nástroje a platformy a zdůvodňuje volbu konkrétních komponent a systémů.

Následující část popisuje samotnou implementaci Wi-Fi sítě podle stanovených požadavků. Obsahuje příklady konfiguračních souborů, detailní postup implementace a řešení pro monitoring a správu infrastruktury, čímž poskytuje praktický návod k realizaci podobného projektu.

Poslední část práce hodnotí výsledky modelové implementace Wi-Fi sítě. Identifikuje její přínosy i slabiny a předkládá návrhy na možná vylepšení pro budoucí iterace nebo rozšíření systému.

Kapitola 1

Počítačové sítě

Kapitola Počítačové sítě se zaměřuje na technologie a principy používané v počítačových sítích. Popisuje Ethernet jako základní síťovou technologii, jeho vývoj a možnost logického oddělení sítí pomocí VLAN. Kapitola také představuje standard IEEE 802.1X, který umožňuje zabezpečený přístup do sítě prostřednictvím autentizace uživatelů.

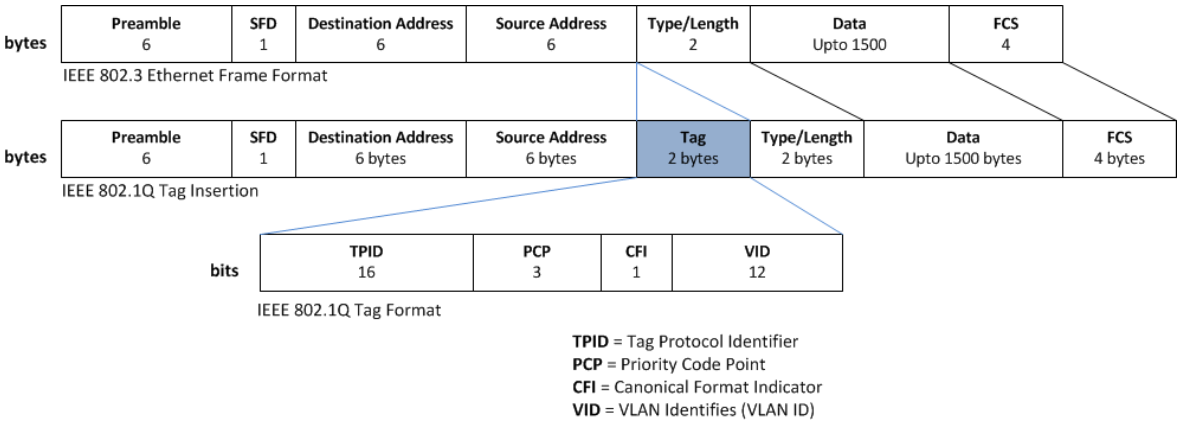
1.1 Ethernet

Ethernet je řada drátových síťových technologií, která je standardizována v IEEE[1] 802.3. Ethernet umožňuje komunikaci zařízení v lokální síti pomocí MAC adres. Díky své jednoduchosti a nízké ceně se jedná o nejrozšířenější technologii používanou v lokálních sítích (LAN). (ijs2.8u.cz, 2024) V kontextu síťového modelu ISO/OSI pracuje na linkové vrstvě a fyzické vrstvě.

Na linkové vrstvě specifikuje formát rámců, detekci chyb a způsob adresování. Dříve ethernet využíval sdílené přenosové média, ke kterým linková vrstva řídila přístup (předcházela kolizím a detekovala je). Dnešní ethernet je přepínaný (používají se přepínače) a v přístupových sítích se používá topologie sítě strom. (Bouška, 2007b) Vymizelo sdílení přenosového média, to má za výhodu odstranění kolizí a zvýšení rychlosti. (Bouška, 2007a)

Na fyzické vrstvě IEEE 802.3 specifikuje typy přenosových médií (koaxiální kabel, kroucená dvojlinka, optické vlákno), použité konektory, přenosové rychlosti a signálové kódování.

[1] Institute of Electrical and Electronics Engineers



Obrázek 1.1: Standardní ethernetový rámec a rozšířený rámec o 802.1Q. (FreeCCNAStudyGuide.com, 2024)

V horní části na obrázku 1.1 je zobrazen znázorněn standardní ethernetový rámec. Je složen z:

- **Preambule**: Prvních 48 bitů slouží k rozpoznání příchozího rámce na straně přijímacího zařízení. Obsahuje tuto sekvenci:
  `10101010 10101010 10101010 10101010`
  `10101010 10101010 10101010 10101011`. (IEEE, 2018)
- **SFD**: 8 bitů označuje konec preambule nebo také začátek samotného rámce.
  Obsahuje: `10101011`. (IEEE, 2018)
- **Cílová adresa**: MAC adresa cílového zařízení.
- **Zdrojová adresa**: MAC adresa odesílajícího zařízení.
- **Délka/Typ**: Pokud je hodnota v decimální soustavě menši než 1500 označuje délku přenášených dat, pokud je větší než 1500 označuje jaký protokol je přenášen na vyšší vrstvě (EtherType). Seznam hodnot a protokolů spravuje IEEE.
- **Data**: Obsahuje data z vyšší síťové vrstvy.
- **FCS**: Kontrolní CRC součet k ověření správnosti přenosu rámce. Poškozené rámce jsou zahozeny.

1.1.1 IEEE 802.1Q VLAN

VLAN, neboli Virtual LAN slouží k logickému rozdělení fyzické sítě. To zjednodušuje správu sítě. Díky možnosti vytvářet směrovací politiky pro různé skupiny zařízení a možnosti oddělení speciálního provozu (management sítě) se zvyšuje bezpečnost sítě. (Bouška, 2007c) VLAN je u Ethernetu umožněno prostřednictvím standardu IEEE 802.1Q, který rozšiřuje standardní Ethernetový rámec o další hodnoty.

V dolní části obrázku 1.1 je znázorněn ethernetový rámec dle IEEE 802.1Q rozšířený v hlavičce o 32 bitů. Za zdrojovou adresu jsou vloženy následující hodnoty:

- **TPID**: Identifikátor použitého VLAN protokolu - v případě IEEE 802.1Q obsahuje `0x8100`. Je na stejné pozici jako Délka/Typ u standardního rámce.
- **PCP**: Označuje prioritu rámce podle IEEE 802.1P.
- **CFI**: Říká jakým způsobem je přenášen rámec. Bud od bitu s nejnižší hodnotou k bitu s nejvyšší nebo naopak. Pro ethernet `0`. (Fairhurst, 2012)
- **VID**: Identifikátor (tag) konkrétní VLAN.

Aby bylo možné jednotlivé zařízení přiřazovat do různých logických sítí je nutné, aby IEEE 802.1Q podporovala zařízení ke kterým jsou připojeny. Na zařízeních s podporou IEEE 802.1Q lze nakonfigurovat do které logické sítě konkrétní port spadá a zda na daném portu má být použít tag či nikoli. Přepínače bez podpory IEEE 802.1Q přistupují k tagovanému provozu stejně jako k běžným rámcům. V tomto kontextu může port zahrnovat fyzický port i logické rozhraní, například síť Wi-Fi nebo konkrétní Wi-Fi klienty.

Trunk port je port, prostřednictvím kterého jsou přenášeny VLAN tagované rámce více logických sítí. Takto bývají propojeny směrovače s přepínači, přepínače mezi sebou nebo přístupové body nabízejí-li více Wi-Fi sítí. (documentation.meraki.com, 2024)

Access port[2] je port sloužící k připojení koncových zařízení. Port přijímá rámce pouze pro jednu VLAN, a odesílá rámce bez tagu. (documentation.meraki.com, 2024) Tag může být přidán na směrovači.

VLAN muže být portu přidělena staticky, podle MAC adresy zařízení, podle provozu síťové vrstvy nebo z autorizačního serveru. (Bouška, 2007c)

1.2 Základní síťové prvky

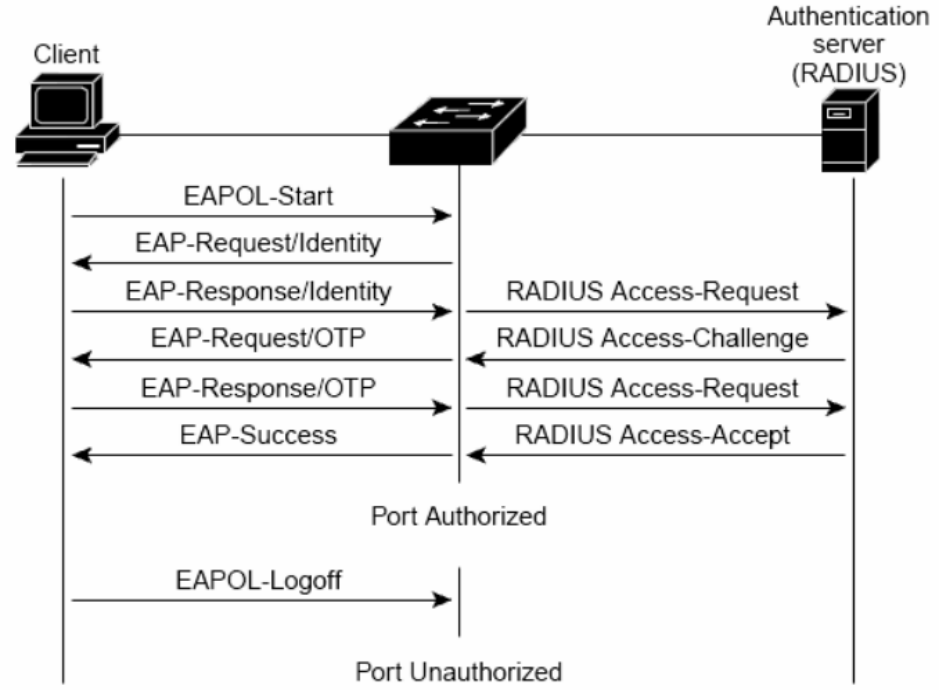
Počítačové sítě jsou tvořeny technickými prostředky (síťovými prvky). **Pasivní síťové prvky** nemanipulují s přenášenými daty a nezasahují do přenášených dat. Nevyžadují externí napájení. Aktivní síťové prvky vyžadují externí napájení a manipulují s přenášenými daty a mohou do nich i zasahovat. **Aktivní síťové prvky** nabízejí různou funkcionalitu, podle které se dále rozlišují.

Aktivní síťové prvky:

- síťový adaptér,
- přepínač (switch),
- přístupový bod (access point),
- směrovač (router).

Mezi dnes již málo používané aktivní síťové prvky lze zařadit:

[2] přístupový port



Obrázek 1.2: Průběh 802.1X / RADIUS autentizace. (Stankuš, 2007)

- opakovač (repeater),
- most (bridge).

Pasivní síťové prvky:

- kabeláž,
- spojky,
- pasivní rozdělovače,
- konektory,
- filtry.

1.3 802.1X

802.1X je standard vydaný organizací IEEE[3] pro zabezpečení přístupu do sítě. Tento standard zvyšuje bezpečnost tím, že zajišťuje autorizaci uživatelů sítě. Přístup je kontrolován na linkové vrstvě ISO/OSI modelu.

V architektuře autentizace pomocí IEEE 802.1X jsou tři hlavní prvky. Prvním je samotný klient (tzv. suplikant), druhý je switch/přístupový bod (autentizátor) a třetím je autentizační server.

Suplikant komunikuje s autentizačním serverem pomocí EAP (Extensible Authentication Protocol) zpráv, které jsou specifikovány RFC 2284. Pro přenos EAP v prostředí Ethernetu slouží EAPoE

[3]Institute of Electrical and Electronics Engineers

(EAP over Ethernet), kde EAP rámce používají EtherType `0x888E`. Komunikace mezi Suplikantem a autentizátorem probíhá pouze na L2 vrstvě ISO/OSI modelu. Autentizátor EAP zprávy od klienta (suplikanta) neinterpretuje, zabalí je do dalšího protokolu a přepošle je autentizačnímu serveru přes IP síť.

Autentizátor komunikuje s autentizačním serverem pomocí jednoho z AAA[4] protokolů. Tyto protokoly se *používají pro zajištění zabezpečení síťové infrastruktury* (Lešek, 2019). Mezi nejvíce používané patří RADIUS, TACACS, TACACS+, KERBEROS a DIAMETER. (Stankuš, 2007). Autentizační server ověřuje EAP zprávy a odpovídá autentizátoru, který na základě těchto odpovědí upravuje přístup suplikanta k síti a přeposílá příslušné EAP odpovědi suplikantovi. Komunikace mezi Autentizátorem a autentizačním serverem probíhá na úrovní aplikační vrstvy.

Proces autorizace klienta pomocí RADIUS serveru je znázorněn na obrázku 1.2.Po připojení klienta (suplikanta) do sítě je veškerý síťový provoz na portu blokován, dokud není dokončena autentizace Autentizátor akceptuje pouze EAPoL zprávy. Klient (suplikant) zahájí komunikaci s autentizátorem **EAPoL-Start**, kterou zahájí autentizační proces. Autentizátor vyzve klienta (suplikant) zprávou **EAP-Request/Identity** ke sdělení identity. Klient odpoví zprávou **EAP-Response/Identity**, která obsahuje jeho identifikaci (například uživatelské jméno). Tuto zprávu autentizátor předá autentizačnímu serveru jako **RADIUS Access-Request**. V tomto protokolu je zapouzdřena EAP zpráva. (Čuhel, 2020)

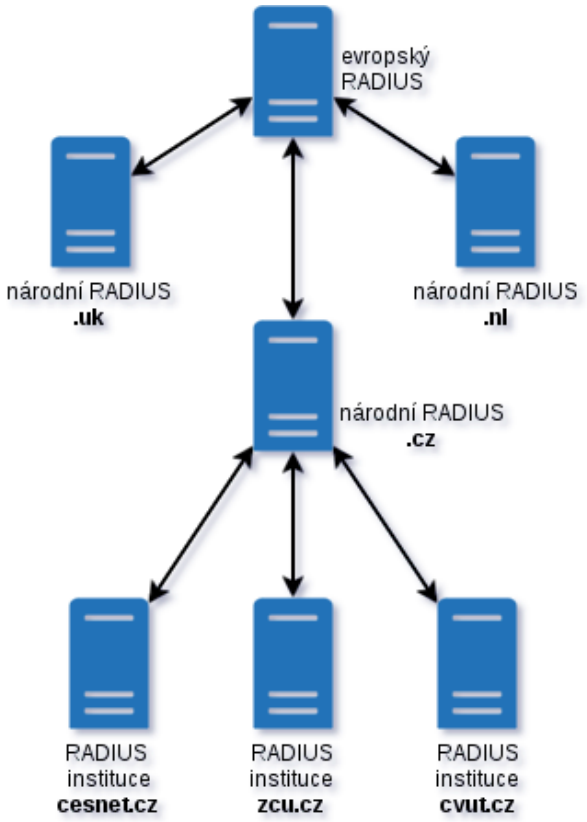
Autentizační server může odpovědět výzvou k zadání dalšího ověření ve formě hesla pomocí **RADIUS Access-Challenge**, tento proces pokračuje dokud server neověří klientovu identitu. Po úspěšném ověření identity server odpoví zprávou **RADIUS Access-Challenge**, kterou autentizátor předá jako **EAP-Success** a povolí klientovi přístup k síti. (Lešek, 2019)

Identita uživatele je uložena v databázi autentizačního serveru, v externí databázi, v systému pro správu identit, nebo na jiném autentizačním serveru.

EAP nezajišťuje šifrování ani ochranu přenášených dat. Toto mají na starosti konkrétní autentizační metody. Mezi nejrozšířenější metody patří:

- **EAP-TLS**: používá pro vzájemnou autentizaci certifikáty klienta a serveru.
- **PEAP**: Vytváří šifrovaný tunel TLS, uvnitř tunelu probíhá sekundární autentizace např. pomocí MS-CHAPv2. Vyžaduje certifikát pouze na straně serveru.
- **EAP-TTLS**: Funguje podobně jako PEAP, nabízí více možností sekundární autentizace (PAP, CHAP, nebo MS-CHAPv2). (networkencyclopedia.com, 2024)

[4]Authentication, Authorization, Accounting - autentizace, autorizace, účtování



Obrázek 1.3: Hierarchické uspořádání autentizačních serverů. (eduroam.cz, 2019)

1.3.1 eduroam

eduroam je mezinárodní projekt s cílem umožnit studentům a pracovníkům vzdělávacích a výzkumných institucí jednoduchý, bezpečný a spolehlivý přístup k veřejné síti Internet v participující institucích kdekoli na světě. Participující instituce využívají jednotný identifikátor Wi-Fi sítě `eduroam`. Přístup k internetu je zařízením umožněn pomocí Wi-Fi nebo ethernetového rozhraní. eduroam využívá specifikace 802.1X pro udělení či zamítnutí přístupu.

K tomu využívá hierarchické uspořádání autentizačních serverů RADIUS (obrázek 1.3), kde každá instituce spravující své identity provozuje svůj autentizační server.

Tímto se snaží napodobit roaming (přepnutí ze sítě jednoho operátora do sítě jiného operátora) v celulárních sítích.

Kapitola 2

Sada standardů Wi-Fi

S rychlým rozvojem technologií a rostoucí dostupností nositelné elektroniky pro širokou veřejnost vzrostla poptávka po spolehlivém a efektivním bezdrátovém přenosu dat.

V roce 1997 byl zveřejněn standard IEEE 802.11 organizací IEEE jako reakce na rostoucí potřebu uživatelů přenášet data prostřednictvím bezdrátových technologií. Tento standard specifikoval základy pro bezdrátové přenosy dat, přičemž umožnil vytvoření bezdrátových lokálních sítí (WLAN). Na základě tohoto standardu byla následně založena organizace Wi-Fi Alliance, která se zaměřila na testování a certifikaci zařízení implementujících tento soubor standardů. Název „Wi-Fi“, odvozený od činnosti této aliance, se postupně stal celosvětově rozšířeným označením pro tuto technologii. Příslušné standardy specifikují mechanismy přenosu dat, přístupu ke sdílenému médiu, autentizace a další podpůrné funkce, jako je například roaming mezi přístupovými body, zajištění kvality služeb nebo řízení vysílacího výkonu.

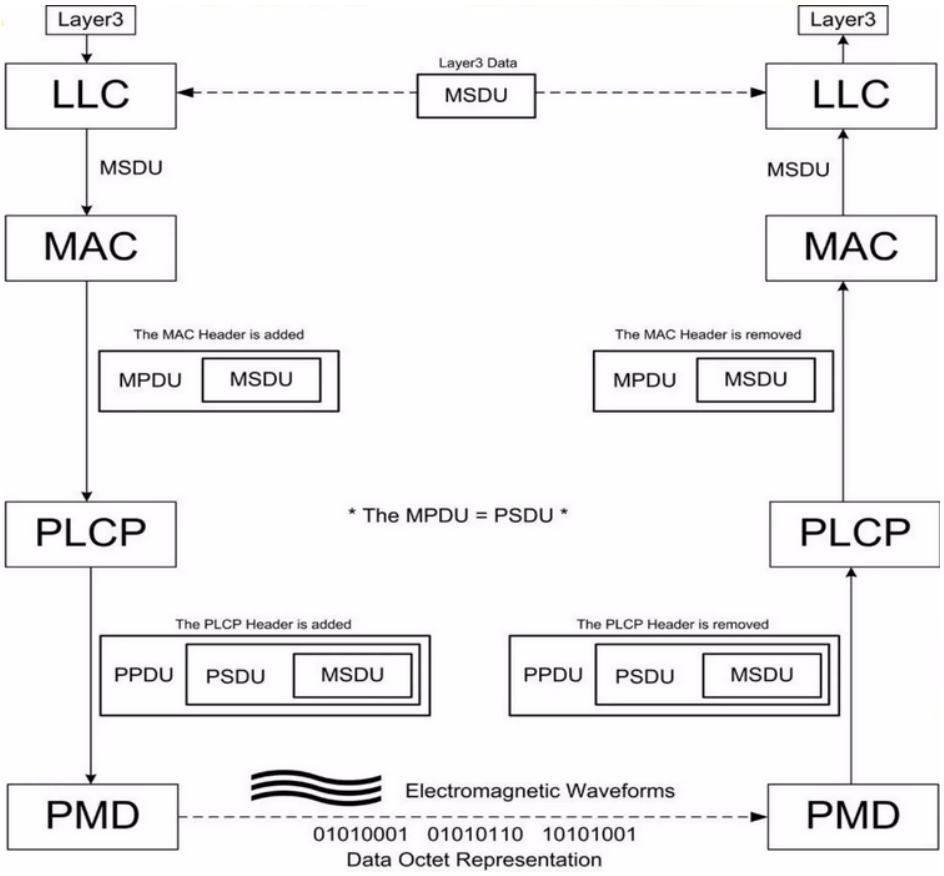
2.1 Vrstvy 802.11

Standard IEEE 802.11 definuje fungování fyzické vrstvy a linkové vrstvy referenčního modelu ISO/OSI. Na fyzické vrstvě stanovuje technické parametry přenosu, jako jsou modulační techniky či frekvenční pásma, zatímco na linkové vrstvě se zabývá otázkami řízení přístupu k médiu (MAC) a řízení spojení (LLC). (IEEE, 2016)

2.1.1 Linková vrstva

Linková vrstva je rozdělena na dvě podvrstvy **Logical Link Control (LLC)** a **Media Access Control (MAC)**.

LLC je vrstva zajišťující řízení logického spoje, definovaná ve standardu IEEE 802.2. Přidáním své hlavičky k síťovému paketu umožňuje multiplexování různých síťových protokolů na jedné fyzické vrstvě. Tato funkce je srovnatelná s polem EtherType, které je využíváno v ethernetových rámcích. Poskytuje informaci cílovému systému o tom jakým způsobem má být paket zpracován.



Obrázek 2.1: Proces enkapsulace a dekapsulace u Wi-Fi. (Namasivayam, 2018)

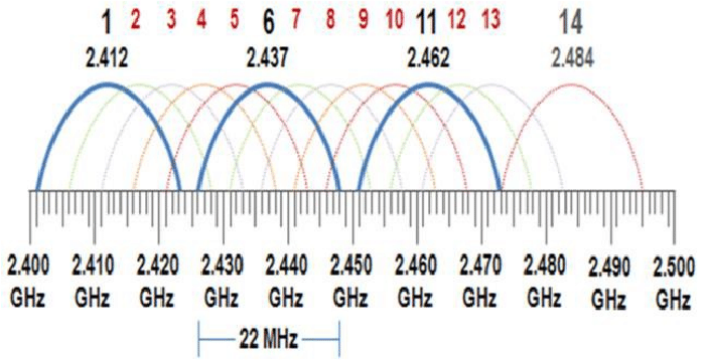
MAC vrstva je zodpovědná za přístup k sdílenému médiu, kontrolu integrity přenášených dat a zabezpečení komunikace. Dále definuje formát rámce, který je použit pro přenos dat v síti. Pro řízení přístupu k médiu MAC vrstva využívá dvě hlavní koordinační funkce: Distributed Coordination Function (DCF) a volitelné rozšíření Point Coordination Function (PCF), přičemž PCF je v praxi zřídka nasazováno (Byeong Gi Lee, 2008). DCF implementuje metodu CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), která minimalizuje kolize mezi zařízeními tím, že jim předchází.

2.1.2 Fyzická vrstva

Aby nebyla MAC vrstva závislá na konkrétní technologii je rozdělena dvě podvrstvy.(Worthman, 2015) Podvrstva **Physical Layer Convergence Protocol (PLCP)**, která je zodpovědná za přípravu dat pro podvrstvu **Physical Medium Dependent (PMD)** zodpovědnou za samotný přenos. (Peterka, 2014)

Hlavní funkcí **PLCP vrstvy** je informovat MAC vrstvu o stavu přenosového kanálu. PLCP vrstva přidává k 802.11 rámci (MPDU[1]) preambuli pro detekci rámců a synchronizaci. A dále hlavičku, která slouží pro informování příjmových zařízení především o délce vysílání a použitých rychlostech. Tyto

[1]MAC Protocol Data Unit



Obrázek 2.2: Kanály pásma 2,4GHz při použití DSSS (802.11b) (Kurnaz et al., 2017)

informace získá z tzv. TXVECTORu od MAC vrstvy, který obsahuje informace o vysílací rychlosti (rate), délce MPDU, typu preambule, modulace a o požadovaném vysílacím výkonu. Tyto parametry specifikují nastavení rádia při přenosu. (Byeong Gi Lee, 2008)

Vrstva PMD zajišťuje samotný přenos dat prostřednictvím elektromagnetického spektra. Stará se o modulaci/demodulaci signálů, rozprostření do spektra (např. OFDM, DSSS), spojení/rozdělení (IFFT/FFT)[2] a jejich vysílání/příjem. To zahrnuje řízení vysílacího výkonu a nastavení kanálu. Je řízena PLCP vrstvou.(Worthman, 2015)

K přenosu jsou využity různé techniky k přenosu dat v elektromagnetickém spektru. Především rozprostřené signály ve spektru a modulace.

Kanály Wi-Fi jsou definovány v rámci normy IEEE 802.11. Mezinárodní i národní regulační orgány určují, které kanály a pásma lze legálně používat v konkrétní zemi. Tyto orgány rozlišují mezi licencovanými a nelicencovanými frekvenčními pásmy, přičemž obě kategorie podléhají regulačním opatřením. Nelicencovaná pásma, jsou zpřístupněna široké veřejnosti za předpokladu dodržení stanovených technických a provozních podmínek. Využité pásma jednotlivými specifikacemi jsou uvedeny v tabulce 2.1. Na obrázku 2.2 jsou uvedeny kanály Wi-Fi v pásmu 2,4 Ghz, které se překrývají. U kanálů v pásmu 5GHz k překryvu nedochází. (Tremer, 2014)

2.2 Druhy rámců

IEEE 802.11 specifikuje tři druhy rámců. Označení druhu rámce je uvedeno v hlavičce MAC.

Datové rámce přenášejí samotná data mezi klientem a přístupovým bodem nebo mezi zařízeními v síti. Obsahují:

- Datovou část: Přenášená informace (např. obsah e-mailu, soubory).

[2](Inverted) Fast Fourier Transformation

- Záhlaví: Obsahuje informace o zdrojové a cílové adrese, QoS (Quality of Service) a další metadata.
- Kontrolní součet (FCS): Slouží k detekci chyb v přenosu.

Rámce pro management slouží k inicializaci a správě spojení mezi zařízeními. Obsahují informace potřebné pro zajištění provozu bezdrátové sítě. Mezi hlavní Management zprávy patří: (Spencer, 2002)

- **Beacon**: Maják - Vysílá přístupový bod (AP) pro oznámení své existence, poskytuje informace o síti, jako je SSID, frekvence a bezpečnostní nastavení.
- **Probe Request/Response**: Klient hledá dostupné sítě (request), AP odpovídá s detaily své sítě (response).
- **Authentication**: Používá se k ověření, že klient má právo se připojit k síti.
- **Deauthentication**: Ukončuje autentizaci mezi klientem a AP.
- **Association Request/Response**: Slouží k přidružení klienta k AP. AP alokuje potřebné síťové zdroje pro komunikaci s klientským zařízením.
- **Disassociation**: Ukončuje přidružení klienta k AP.

Řídící rámce se používají k řízení přenosu dat a zajištění efektivního využití bezdrátového média. Mezi základní řídící rámce patří (Parsi, 2012):

- **RTS (Request to Send)**: Klient žádá o přístup k přenosovému médiu pro přenos velkých rámců.
- **CTS (Clear to Send)**: AP potvrzuje, že klient může vysílat.
- **ACK (Acknowledgement)**: Potvrzení přijetí datového rámce.

2.3 Standardy Wi-Fi

Od prvního vydání standardu proběhl významný vývoj v technologii Wi-Fi, který přinesl vyšší rychlosti, lepší spolehlivost a větší efektivitu. Standardy se označují *IEEE 802.11* a nové revize jsou doplňovány o písmenné označení. Některé ze standardů se zároveň označují číselnou verzí. Tabulka 2.1 porovnává jednotlivé důležité standardy a jejich základní vlastnosti.

2.4 Autentizační metody a bezpečnost

Vzhledem k používání sdíleného média, kterým je elektromagnetické spektrum, je ve Wi-Fi sítích vhodné zajistit šifrování přenosu a autentizaci zařízení nebo uživatelů. Tyto aspekty byly zohledněny již při návrhu původních standardů IEEE 802.11. Šifrování probíhá na úrovni MAC vrstvy.

Standard	Verze Wi-Fi	Rok vydání	Rychlost	Pásmo
802.11b	1 (neoficiálně)	1999	11 Mbps	2.4 GHz
802.11a	2 (neoficiálně)	1999	54 Mbps	5 GHz
802.11g	3 (neoficiálně)	2003	54 Mbps	2.4 GHz
802.11n	4	2009	600 Mbps	2.4 GHz, 5 GHz
802.11ac	5	2013	3.46 Gbps	5 GHz
802.11ax	6	2019	10.53 Gbps	2.4 GHz, 5 GHz
802.11ax	6E	2019	10.53 Gbps	2.4 GHz, 5 GHz, 6 GHz
802.11be	7	Nevydáno	40 Gbps	2.4 GHz, 5 GHz, 6 GHz
802.11bn	8	Nevydáno		

Tabulka 2.1: Seznam standardů IEEE 802.11 a jejich základní vlastnosti. (Brawley, 2023)

2.4.1 Autentizace na bázi sdíleného klíče

Autentizace na bázi sdíleného klíče využívá předem sdílený klíč (heslo), který je nastaven na přístupovém bodě a musí být manuálně zadán na každém klientském zařízení. Tento přístup je jednoduchý, avšak omezený z hlediska bezpečnosti.

Wired Equivalent Privacy - WEP

První metoda zabezpečení ve 802.11 je **WEP (Wired Equivalent Privacy)**. Používá proudové šifrování RC4, které kombinuje sdílený klíč s inicializačním vektorem (IV). Použití IV zabraňuje opakování sekvence textu při šifrování dat. (Awati, 2022) Dochází pouze k autentizaci zařízení, která používají stejný sdílený klíč jako ostatní zařízení. Tento způsob zabezpečení komunikace a autentizace zařízení obsahuje zranitelnosti, a proto není bezpečný. (Singh, 2022) **Open System Authentication**, varianta autentizace v WEP, pouze šifruje provoz a neřeší autentizaci.

Wi-Fi Protected Access - WPA

Byl zaveden jako náhrada za WEP, aby překonal jeho nedostatky. WPA využívá Temporal Key Integrity Protocol (TKIP), který dynamicky mění šifrovací klíč při každém přenosu dat. Tento přístup výrazně zvyšuje odolnost vůči útokům založeným na opakování šifrovacích sekvencí. Kromě toho WPA zavádí mechanismus integrity zpráv (Message Integrity Check), který chrání data před neoprávněnými změnami během přenosu. (Wang et al., 2015)

Ačkoli WPA představuje významné zlepšení oproti WEP, stále zůstává citlivý na některé typy útoků, zejména pokud je používán s předem sdíleným klíčem (PSK). Tento nedostatek vedl k vývoji jeho nástupce, WPA2.

Wi-Fi Protected Access 2 - WPA2

Přinesl zásadní zlepšení bezpečnosti oproti WPA díky zavedení šifrovacího algoritmu AES (Advanced Encryption Standard) a protokolu CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol). Tyto technologie poskytují silné šifrování a ochranu integrity dat, což zajišťuje vyšší úroveň bezpečnosti. (Wang et al., 2015)

WPA2 je dnes nejrozšířenějším standardem zabezpečení Wi-Fi sítí, zejména v kombinaci s Enterprise režimem, který využívá autentizaci na bázi 802.1X.

Wi-Fi Protected Access 3 - WPA3

Představuje nejnovější generaci zabezpečení Wi-Fi sítí. WPA3 nahrazuje sdílený klíč PSK metodou SAE (Simultaneous Authentication of Equals), která je odolnější vůči offline útokům hrubou silou. Dále přidává šifrovací algoritmus GCMP (Galois/Counter Mode Protocol), jenž poskytuje ještě silnější ochranu dat než CCMP. (Vanhoef et al., 2020)

2.4.2 Autentizace na bázi 802.1X/EAP

Autentizace na bázi protokolu 802.1X/EAP je považována za nejbezpečnější metodu zabezpečení Wi-Fi sítí, neboť neautentizuje pouze dané zařízení ale konkrétního uživatele. Je podporována všemi standardy WPA. Tento přístup využívá RADIUS server pro centralizovanou autentizaci uživatelů. Označuje se také jako WPA-enterprise.

Každý uživatel se autentizuje vůči RADIUS serveru prostřednictvím jedné z podporovaných metod EAP, které jsou specifikovány v sekci 1.3.

Kapitola 3

Deklarace požadavků na modelovou Wi-Fi síť

Implementace rozsáhlé Wi-Fi sítě vychází z požadavků školského zařízení, které zahrnuje rozsáhlý hlavní objekt a další geograficky oddělené pracoviště. Cílem je zajistit plné pokrytí obou lokalit sítěmi Wi-Fi a umožnit přístup uživatelům s různými rolemi, jako jsou učitelé, studenti a návštěvníci. Hlavním požadavkem je integrace této sítě do projektu `eduroam`.

Pro zajištění kvalitního pokrytí obou lokalit bude využito více přístupových bodů (AP), které budou podporovat připojení v pásmech 5 GHz a 2,4 GHz, podle dostupných zařízení. Každé z pásem bude nabízet dvě samostatné Wi-Fi sítě, které budou plnit rozdílné funkce:

- **Síť eduroam** – slouží studentům a vyučujícím. Připojení k této síti bude vyžadovat autentizaci a autorizaci uživatelů prostřednictvím centrálního autorizačního serveru. Na základě role uživatele budou přiřazeny specifické VLAN, které zajistí odpovídající přístup k lokálním službám a internetu. Tato síť bude splňovat požadavky projektu eduroam na bezpečnost a interoperabilitu.
- **Veřejná síť** – poskytne snadný přístup k internetu pro návštěvníky, bez nutnosti autentizace na úrovni Wi-Fi. Tato síť bude izolována od interních systémů školy, aby byla zajištěna bezpečnost.

Důležitou součástí návrhu je centralizovaná správa přístupových bodů. Ta umožní efektivní konfiguraci, monitorování a údržbu celé Wi-Fi infrastruktury, čímž se sníží administrativní náročnost a zvýší spolehlivost poskytovaných služeb.

V neposlední řadě je nutné zajistit monitoring přístupových bodů a provozních údajů jednak z důvodu zvýšení spolehlivosti sítě tak i z důvodu legislativních požadavků vyplývajících ze zákona o elektronických komunikacích, který říká, že právnická osoba zajišťující veřejnou komunikační síť je povinna uchovávat po dobu 6 měsíců provozní a lokalizační údaje, které jsou vytvářeny nebo zpracovávány při zajišťování jejích veřejných komunikačních sítí. (Česko, 2005) Obdobný požadavek klade i národní koordinátor sítě eduroam ve svých technických požadavcích a doporučeních.

Kapitola 4

Analýza dostupných nástrojů

4.1 Systémy pro přístupové body

Na trhu je široká nabídka síťového hardwaru od různých výrobců, přičemž tato zařízení často obsahují rozšířený a dobře dokumentovaný hardware, který je podporován v Linuxovém jádře. Tento fakt umožňuje využití existujících projektů zaměřených na vývoj operačních systémů pro tato zařízení.

Pro umožnění použití co největšího množství přístupových bodů různých výrobců je nutné nahradit jejich proprietární systém otevřenou alternativou. Tento přístup rovněž odpovídá cílům práce, která si klade za úkol implementovat přístupovou Wi-Fi síť výhradně s využitím open-source technologií.

Open-source alternativy přinášejí několik výhod. Zejména to, že je podporován komunitou i po ukončení oficiální podpory výrobce zařízení, což zvyšuje bezpečnost a odolnost vůči novým zranitelnostem (Haidarzhy, 2024). Také mnohdy rozšiřují funkce zařízení nad rámec toho, co je běžně poskytováno výrobcem.

Pro tento účel existuje mnoho softwarových řešení, která se zaměřují na vylepšení a přizpůsobení fungování domácího nebo firemního routeru. Tato firmware řešení často poskytují více funkcí a flexibilitu než standardní verze, které poskytují výrobci. Informace o těchto projektech byly čerpány z článků DD-WRT vs. Tomato vs. OpenWRT (Asim, 2022), DD-WRT vs OpenWrt (Twain, 2024) a Which Router Firmware Is the Best? (Crowder, 2023). Z těchto článku vyplývá, že mezi nejznámější a nejpoužívanější patří následující softwarová řešení.

DD-WRT

DD-WRT je firmware, který se vyznačuje kompatibilitou s různými zařízeními díky podpoře mnoha ovladačů. To znamená, že jej lze nainstalovat na mnoho různých modelů routerů, což z něj tvoří velmi univerzální systém. Tento firmware má jednu z největších a nejaktivnějších komunit uživatelů a vývojářů, což znamená, že pro něj existuje spousta dokumentace.

Nicméně, aktualizace pro DD-WRT nevycházejí tak často, což může znamenat menší rychlost imple-

mentace nových funkcí a oprav. Firmware je stabilní, dobře optimalizovaný a spolehlivý, ale rozhraní pro uživatele je méně intuitivní a pro některé uživatele může být nastavení a konfigurace náročnější.

OpenWrt

OpenWrt je známý pro svou vysokou flexibilitu, která je dána možností přidávat různé balíčky a moduly, což umožňuje přizpůsobit firmware přesně podle potřeb uživatele. Tento firmware podporuje širokou škálu zařízení.

Disponuje systémem `uci` pro konfiguraci přes příkazovou řádku, což umožňuje pokročilým uživatelům detailní nastavení systému. Nabízí pouze open-source ovladače, což znamená, že všechny ovladače jsou dostupné zdarma a jejich kód je otevřený pro komunitní vylepšení.

OpenWrt má velkou a aktivní komunitu, pravidelně vycházejí aktualizace a opravy, což zajišťuje bezpečnost a nové funkce. Na druhou stranu, webové rozhraní není tak přívětivé jako u některých jiných softwarových řešení.

Tomato

Tomato je firmware zaměřený především na stabilitu a jednoduchost použití. Jeho hlavní výhodou je přehledné, moderní uživatelské rozhraní, které usnadňuje konfiguraci a správu zařízení.

Tento firmware je obzvláště vhodný pro zařízení, která používají Broadcom Wi-Fi čipy, na které je dobře optimalizován. Tomato nabízí velmi jednoduché a rychlé nastavení. Nicméně, oproti OpenWrt nebo DD-WRT nabízí omezené možnosti přizpůsobení a menší podporu pro různé modely routerů. Komunita je výrazně menší a aktualizace jsou méně časté.

4.2 Nástroje pro automatizaci konfigurace

Wi-Fi kontroléry jsou nástrojem pro centralizovanou a automatizovanou správu rozsáhlých sítí přístupových bodů. Umožňují jednotnou konfiguraci, distribuci aktualizací firmware, monitorování provozu a řešení problémů, čímž eliminují nutnost individuální konfigurace jednotlivých zařízení. Tento přístup šetří čas, snižuje riziko chyb a zajišťuje konzistenci napříč celou sítí.

Proprietární řešení, jako jsou Cisco Wireless Controller, Ubiquiti UniFi Controller nebo Aruba Mobility Controller, poskytují širokou škálu funkcí, které umožňují efektivní správu bezdrátových sítí. Avšak tato řešení obvykle vyžadují použití specifického hardwaru a softwaru daného výrobce. Pro organizace hledající otevřenou alternativu k těmto proprietárním systémům se nabízí **OpenWISP**, open-source platforma určená pro správu Wi-Fi sítí. OpenWISP podporuje škálovatelné nasazení a umožňuje nejen konfiguraci přístupových bodů, ale rovněž správu autentizace uživatelů. Nicméně, tento nástroj má jedno zásadní omezení: je primárně navržen pro zařízení běžící na operačním systému OpenWRT, což

může omezit jeho použitelnost v prostředích s jinými typy zařízení nebo operačními systémy. (Shi, 2018)

Alternativním přístupem k hromadné konfiguraci přístupových bodů je využití nástrojů pro **obecnou automatizaci konfigurace**. Tyto nástroje, původně vyvinuté pro správu serverů a IT infrastruktury, umožňují automatizovat konfiguraci libovolných zařízení. (Heap, 2016)

Těmito nástroji lze definovat požadovaný stav systému. Tento přístup, označovaný jako Infrastructure as Code (IaC), zapisuje požadovanou konfiguraci infrastruktury ve strojově čitelném formátu. Softwarové nástroje pak na základě tohoto zápisu automaticky aplikují konfiguraci na jednotlivé prvky infrastruktury.

Použití IaC přináší několik výhod: (Maiseyeu, 2019)

- **Snížení chybovosti:** Automatizace eliminuje lidské chyby při konfiguraci více zařízení.
- **Konzistentní nastavení:** Zajišťuje homogenní konfiguraci napříč celou infrastrukturou.
- **Verzování:** S verzovacími systémy je možné sledovat změny konfigurace, zajišťovat editovatelnost a zpětnou dohledatelnost.
- **Zvýšená efektivita:** Automatizace šetří čas a usnadňuje škálování infrastruktury.

Existuje několik známých open-source nástrojů pro správu infrastruktury jako kódu, každý se svými specifiky a způsob správy. Mezi nejpoužívanější opensource nástroje patří **Ansible**, **Chef**, **Puppet** a **SaltStack** (Nemeth Evi, 2018, s. 841). Jejich jednotlivé vlastnosti jsou uvedeny v tabulce 4.1 na stránce 28.

4.2.1 Přístupy k zápisu konfigurace

Existují dvě základní paradigmata zápisu konfigurace. (Sen, 2024)

Imperativní (procedurální) definuje jednotlivé kroky, které mají být provedeny v přesném pořadí, pro dosažení požadovaného stavu infrastruktury. Tento přístup je vhodný tam, kde je důležité mít detailní kontrolu nad průběhem změn a kde je třeba přesně specifikovat jednotlivé akce.

Deklarativní zápis popisuje požadovaný stav infrastruktury, tedy jaký má být cílený stav. Jedná se o deklarativní abstrakci, kdy dosažení cílové konfigurace je implementováno imperativně. Nástroj pro automatizaci konfigurace vyhodnotí aktuální stav a provede pouze nezbytné kroky k dosažení cílového stavu. Tento přístup je často preferovaný díky své jednoduchosti a čitelnosti, protože umožňuje soustředit se na výsledek místo na proces.

4.3 Nástroje pro logování

Systémové logy hrají zásadní roli při monitorování stavu zařízení. Poskytují důležité informace o bezpečnostních incidentech, hardwarových a softwarových problémech nebo jiných událostech, které vyžadují pozornost. Aby byla tato data užitečná doporučuje se logy uchovávat alespoň jeden měsíc. Důvodem je skutečnost, že odhalení bezpečnostního incidentu může také zabrat určitý čas. (Nemeth Evi, 2008)

Standardy RFC 3164 a RFC 5424 definují formát zpráv syslog. Tyto standardy určují strukturu logovacích zpráv, včetně časové značky, úrovně závažnosti, identifikace zařízení a dalších informací. Většina moderních nástrojů pro logování tento standard dodržuje, což umožňuje snadné sdílení logů mezi servery a nástroji.

Jednou z důležitých funkcí těchto nástrojů je podpora vzdáleného logování. To znamená, že logovací démon může odesílat zprávy na externí server, což je užitečné pro centralizaci logů z více zařízení.

Systémové logování na Linuxu obvykle zajišťuje některý z těchto nástrojů: (Nemeth Evi, 2018, s. 299-304)

- `rsyslog`: Rychlý a široce používaný logovací démon s podporou rozšířených filtrů a vzdáleného logování.
- `syslog-ng`: Flexibilní a výkonný démon pro logování s podporou strukturovaných formátů a šifrovaného přenosu logů.
- `systemd-journald`: Součást systemd; Ukládá logy v binárním formátu.

4.4 Nástroje pro monitoring

Kromě systémového logu mohou historické záznamy o stavu systému a hardwarových prostředcích sloužit ke zpětné analýze příčin vzniklých problémů, což usnadňuje jejich diagnostiku a řešení. Monitoring také umožňuje včas identifikovat potenciální problémy, jako je vyčerpání operační paměti nebo nedostatek výpočetní kapacity CPU, a tím zabránit výpadkům.

Dle článku Monitoring and Visualization Options for OpenWRT (Ram, 2024) a repozitářů softwaru systému DD-WRT (dd-wrt.com, 2024) a Tomato (FreshTomato.org, 2011) jsou k dispozici pro monitoring tyto nástroje:

- **zabbix-agentd:** je klientský nástroj, který se instaluje na monitorované systémy. Shromažďuje data o systému (jako je využití CPU, paměti, síťový provoz) a odesílá je do Zabbix Serveru nebo Proxy pro další zpracování
- **collectd:** je démon pro sběr a export metrik o výkonu systému a aplikací. Je lehký, modulární a snadno rozšiřitelný.

- **prometheus-node-exporter:** je standardní komponenta databáze časových řad Promethes, která poskytuje základní metriky systému.
- **snmpd:** poskytuje informace o zařízení a systému přes protokol SNMP (Simple Network Management Protocol). SNMP je standardizovaný protokol pro monitorování a správu síťových zařízení. Nabízejí ho všechny systémy ze sekce 4.1.
- **telegraf:** je plugin-based agent pro sběr metrik a logů, který je součástí TICK stacku od InfluxData. Data může odesílat do InfluxDB nebo dalších úložišť.

4.5 Nástroje pro záznam síťových toků

Při použití překladu IP adres (NAT) je nezbytné zaznamenávat provozní údaje veřejných komunikačních sítí, aby byly splněny zákonné požadavky a pravidla stanovená národním operátorem `eduroam`.

Provozní údaje jsou legislativně definovány jako údaje umožňující dohledání a identifikaci zdroje a adresáta, dále údaje vedoucí ke zjištění data, času, způsobu a doby trvání komunikace (Česko, 2005). V kontextu TCP/IP sítí zahrnují tyto údaje IP adresy a porty zdroje i cíle, MAC adresu zdrojového zařízení, typ provozu (např. UDP, TCP, ICMP), čas zahájení komunikace a dobu jejího trvání.

Pro sledování síťových toků v TCP/IP sítích je široce využíván protokol NetFlow. Tento protokol je navržen pro přenos informací o síťových tocích, které vznikají agregací paketů na základě metadat obsažených v jejich hlavičkách. (Petryschuk, 2024) Směrovače propojující sítě jsou vybaveny softwarem pro agregaci těchto paketů do tzv. flow dat. Tento software se nazývá exportér. Záznamy o tocích se pomocí protokolu NetFlow posílá na tzv. collector. Ten tyto data přijímá a ukládá, případně dále zpracovává.

Agregace paketů má několik výhod. Záznam síťových toků významně snižuje nároky na výpočetní výkon a šířku přenosového pásma, protože místo jednotlivých paketů jsou přenášeny pouze sumarizované informace o komunikaci. Agregace toků zároveň zvyšuje úroveň soukromí uživatelů, neboť poskytuje přehled o síťové aktivitě na úrovni toků, nikoliv jednotlivých paketů (Hofstede et al., 2014).

Pro získání informací o MAC adresách zařízení je nezbytné použít NetFlow ve verzi V9 nebo IPFIX, které tyto údaje podporují. (Claise et al., 2008) (Trammell et al., 2013) Tyto pokročilé verze protokolu umožňují detailnější sledování a analýzu síťových toků, čímž přispívají k efektivnějšímu zajištění legislativních a provozních požadavků.

Mezi používané exportéry patří:

- **softflowd** je jednoduchý a efektivní NetFlow exportér, který podporuje verze NetFlow, především v5 a v9. Je zaměřen na snadnou konfiguraci a nasazení v malých a středních sítích. Tento nástroj běží v příkazové řádce a je vhodný pro uživatele, kteří potřebují rychlý způsob exportování síťových dat do NetFlow kolektorů. Je ideální pro situace, kdy není potřeba složité

nastavení.

- **fprobe** je dalším jednoduchým NetFlow exportérem, který je primárně zaměřen na export NetFlow v5. Tento nástroj je velmi nenáročný na systémové prostředky, což z něj činí dobrou volbu pro malé sítě a testovací prostředí. Jeho výhodou je jednoduchost a minimální požadavky na konfiguraci. Na druhou stranu, fprobe má omezené možnosti v oblasti novějších verzí NetFlow
- **pmacct** je vysoce flexibilní a výkonný nástroj pro sběr. Podporuje NetFlow, sFlow a IPFIX. Výhodou pmacct je jeho schopnost pracovat s velkými objemy dat. Je vhodný pro střední a velké organizace, které potřebují pokročilé sledování a reporting síťového provozu. Nevýhodou může být složitější konfigurace a nároky na hardware při větších nasazeních. (Lucente, 2014)
- **nProbe** je pokročilý NetFlow a IPFIX exportér, který se vyznačuje vysokým výkonem. Pochází z nástroje ntop.
- **Exportér implementovaný v systému**: Systémů routerů nabízí možnost nastavení zaznamenávání síťových toků a jejich zasílání na kolektor. Např. OpnSense Netflow Export (Deciso B.V., 2024) nebo proprietární Mikrotik Traffic Flow (R., 2024).

Mezi používané kolektory patří:

- **pmacct** je vysoce flexibilní a výkonný NetFlow, sFlow a IPFIX collector. Nabízí možnost ukládání dat do různých databází, jako jsou PostgreSQL, MySQL nebo InfluxDB, což usnadňuje jejich následnou analýzu. Je vhodný jak pro malé sítě, tak pro rozsáhlé organizace, které potřebují pokročilé sledování a reportování síťového provozu. Díky široké podpoře protokolů a možnostem přizpůsobení je oblíbený mezi administrátory, kteří potřebují škálovatelný nástroj. (Lucente, 2014)
- **ntop** je pokročilý nástroj pro monitorování a analýzu síťového provozu. Podporuje různé formáty dat, včetně NetFlow, IPFIX a sFlow, což z něj činí univerzální řešení. Pro prohlížení nabízí webové uživatelské prostředí.
- **nfcapd** je NetFlow collector specializovaný na sběr a ukládání NetFlow dat do binárních souborů. Je navržen s důrazem na vysoký výkon a efektivitu, což ho činí ideálním pro prostředí s velkým množstvím dat. Pro přístup k těmto datům a jejich analýzu se používá nástroj `nfdump`, který umožňuje snadné filtrování a vytváření reportů. Tento systém je často volen pro jeho jednoduchost a spolehlivost při dlouhodobém ukládání NetFlow dat.
- **GoFlow** je moderní a lehký NetFlow collector napsaný v programovacím jazyce Go. Podporuje NetFlow v5/v9 a IPFIX, přičemž je optimalizovaný pro vysoký výkon a nízké systémové nároky. Net flow data předává do nástroje Kafka nebo vypisuje do příkazové řádky. Neřeší jejich uložení, tím klade důraz na modularitu. GoFlow je oblíbený zejména v moderních sítích, které kladou důraz na jednoduchost a flexibilitu. (Cloudflare, Inc, 2024)

	Ansible	**Chef**	**Puppet**	**SaltStack**
Programovací jazyk	Python	Ruby	C++, Clojure	Python
Konfigurační jazyk	YAML, JSON	Ruby	Vlastní jazyk	YAML
Zápis konfigurace	Procedurální Deklarativní	Procedurální	Deklarativní	Deklarativní
Architektura	Agentless	Server/client	Server/client, Agentless	Server/client, Agentless
Model získání konfigurace	Push	Pull	Pull / Push	Push
Způsob přenosu konfigurace	SSH	HTTP(S)	MCollective, HTTPS	ZeroMQ

Tabulka 4.1: Srovnání nástrojů pro automatizaci správy infrastruktury

Kapitola 5

Výběr vhodných nástrojů

Tato sekce se věnuje výběru specifických technologií, které jsou využity v modelové implementaci, viz kapitola 6. Jejich výběr navazuje na kritéria stanovená v kapitole 3.

5.1 Systém pro síťová zařízení

Po zvážení možností uvedených v sekci 4.1 byl pro implementaci vybrán systém **OpenWrt**. Tento systém byl zvolen díky několika klíčovým vlastnostem, které jsou pro danou aplikaci zásadní. Mezi hlavní důvody patří:

- Široká komunita vývojářů, která zajišťuje pravidelné aktualizace, vylepšení a širokou podporu pro různé modely zařízení.
- Obsáhlé repozitáře doplňkového softwaru, které umožňují snadnou integraci dalších funkcí a nástrojů, což je důležité pro přizpůsobení systému specifickým potřebám.
- Flexibilita konfigurace, která poskytuje možnosti pro detailní přizpůsobení nastavení síťových zařízení.
- Možnost konfigurace pomocí nástroje `uci`, což usnadňuje automatizaci a skriptování konfigurací zařízení, což je klíčové pro efektivní správu velkých síťových prostředí.

Díky těmto vlastnostem je **OpenWrt** ideální volbou pro zajištění efektivního řešení pro správu síťových zařízení v rámci modelové implementace.

5.2 Nástroj pro automatizaci konfigurace

Výhodou nástroje OpenWISP je jeho specializace na správu Wi-Fi sítí, včetně autentizace uživatelů. Tento zaměřený přístup však omezuje jeho schopnosti v oblasti konfigurace dalších síťových zařízení. Naopak, použití nástroje pro obecnou automatizaci konfigurace umožňuje širší možnosti správy, zahrnující nejen přístupové body, ale i další prvky infrastruktury, jako jsou směrovače, přepínaše, servery

a další zařízení. Implementací takového nástroje je možné dosáhnout stavu, kdy je celá IT infrastruktura definována „jako kód“ (IaC).

Tabulka 4.1 poskytuje přehled a porovnává vybrané nástroje pro automatizaci konfigurace.

Kritéria pro výběr nástroje:

- **Instalace agenta:** Preferovány jsou nástroje, které nevyžadují instalaci agenta (menší využití hardwarových prostředků).
- **Jazyk pro zápis konfigurace:** YAML je preferován díky své čitelnosti a jednoduchosti.

Těmto požadavkům vyhovuje nástroj **Ansible**. Byl vybrán pro jeho širokou komunitní podporu a existenci projektu `ansible-OpenWrt`[1], jenž přepisuje příkazy některých modulů Ansiblu vyžadující python interpretr za jejich alternativy interpretované v unixovém příkazovém procesoru a obsahuje modul uci pro konfiguraci OpenWrt. Díky interpretaci příkazů modulů v příkazovém procesoru není nutné na zařízení s omezenými hardwarovými prostředky, jako jsou přístupové body, instalovat Python, který může zabrat více než 9 MiB paměti.

Ansible umožňuje zápis požadované konfigurace kombinací způsobů imperativního i deklarativního. Je však doporučeno upřednostňovat deklarativní způsob zápisu konfigurace a k imperativnímu přistoupit až v momentě kdy jsme omezeni schopnostmi tohoto nástroje. (Appnel, 2023)

Ansible se skládá z následujících komponent, které jsou důležité při implementaci modelového řešení v kapitole 6:

- **Inventory**: Seznam spravovaných zařízení, který umožňuje seskupovat jednotlivé zařízení do skupin podle jejich vlastností nebo umístění.
- **Playbook**: Soubor, který obsahuje definice úkolů, které mají být vykonány na cílových zařízeních, nebo skupinách cílových zařízení.
- **Moduly**: Předpřipravené funkce, které Ansible používá k provádění úkolů. Příkladem může být modul `apt` pro instalaci balíčků, nebo `filesystem` pro správu filesystémů.
- **Role**: Strukturované jednotky obsahující playbooky, proměnné, šablony a soubory tak, aby byly znovu použitelné na různých systémech či v jiných kontextech.
- **Proměnné**: Slouží k ukládání hodnot využitelných v playboocích, šablonách nebo modulech. Lze definovat specifické hodnoty proměnných pro konkrétní zařízení, pro skupiny zařízení nebo pro daný playbook.
- **Templates (šablony)**: Slouží pro generování konfiguračních souborů na základě proměnných. Zapisují se v jazyce Jinja2.
- **Handlers**: Speciální úkoly, které jsou spuštěny pouze tehdy, dojde-li ke změně na konfigurovaném zařízení. Používají se například pro restart služby při změně konfigurace.

[1] `https://github.com/gekmihesg/ansible-OpenWrt`

- **Facts (fakta)**: Shromážděné informace o konfigurovaném zařízení, které lze dále použít v playboocích k individuální konfiguraci.

5.3 Nástroj pro logování

Pro účely logování nebyl zvolen žádný externí nástroj vzešlý z analýzy dostupných softwarových nástrojů, neboť systém OpenWrt obsahuje vestavěný systémový log s názvem *logd*. Tento systémový log podporuje přesměrování logových zpráv na externí server, což umožňuje splnit požadavek formulovaný v kapitole 3 na centrální logování.

5.4 Nástroj pro monitoring

Pro monitoring byl vybrán nástroj `prometheus-node-exporter-lua`, který se ukázal jako ideální volba díky své jednoduchosti a efektivitě. Zároveň umožňuje rozšíření prostřednictvím rozšiřujícího modulu `prometheus-node-exporter-lua-hostapd_stations`. Tento modul poskytuje podrobné informace o připojených klientech k jednotlivým Wi-Fi sítím pomocí démona `hostapd`. Díky této integraci odpadá nutnost vytvářet vlastní složitá řešení, což výrazně zjednodušuje proces monitorování a zajišťuje rychlý a spolehlivý přístup k relevantním datům o monitorovaných zařízeních. Pro vizualizaci dat lze využít nástroj Grafana, který umožňuje vytvářet přehledné grady a dashboardy z dat uložených v Prometheu.

5.5 Nástroje pro záznam síťových toků

Monitoring síťových toků spadá do kompetence zařízení, která operují na síťové vrstvě a zajišťují přeposílání IP paketů mezi jednotlivými sítěmi. V rámci této práce není konfigurace síťového směrovače specifikována, neboť školní infrastruktura již disponuje existujícím síťovým směrovačem. Z tohoto důvodu není výběr konkrétního exportéru součástí navrhovaného řešení. Místo toho bude použit nástroj odpovídající typu směrovače, který je již v současnosti součástí školní sítě, čímž se zohlední kompatibilita současného školního hardwaru.

Je však vybrán kolektor. Pro sběr netflow dat byl z nabízených kolektorů vybrán **nfcapd**. Byl vybrán z důvodu jeho vysokého výkonu, spolehlivosti a jednoduchosti. Nenabízí žádné pokročilé analytické možnosti, slouží pouze k ukládání dat.

Kapitola 6

Implementace modelové Wi-Fi sítě

Tato sekce se zaměřuje na konfiguraci síťových zařízení a dalších softwarových komponent nezbytných pro správnou funkčnost modelové implementace v souladu s požadavky definovanými v kapitole 3.

Hlavní důraz je kladen na:

- **Konfiguraci přístupových bodů**: Detailně popisuje nastavení potřebné k zajištění funkcionality a možností sběru dat pro účely monitoringu sítě.
- **Získávání dat pro monitoring:** Uvádí přístupy k extrakci a zpracování relevantních dat o provozu a stavu sítě.
- **Centrální autentizaci a autorizaci:** Popisuje implementaci řešení umožňujícího jednotné řízení přístupu, včetně jeho integrace do sítě `eduroam`, která zajišťuje bezpečný a standardizovaný přístup pro uživatele.

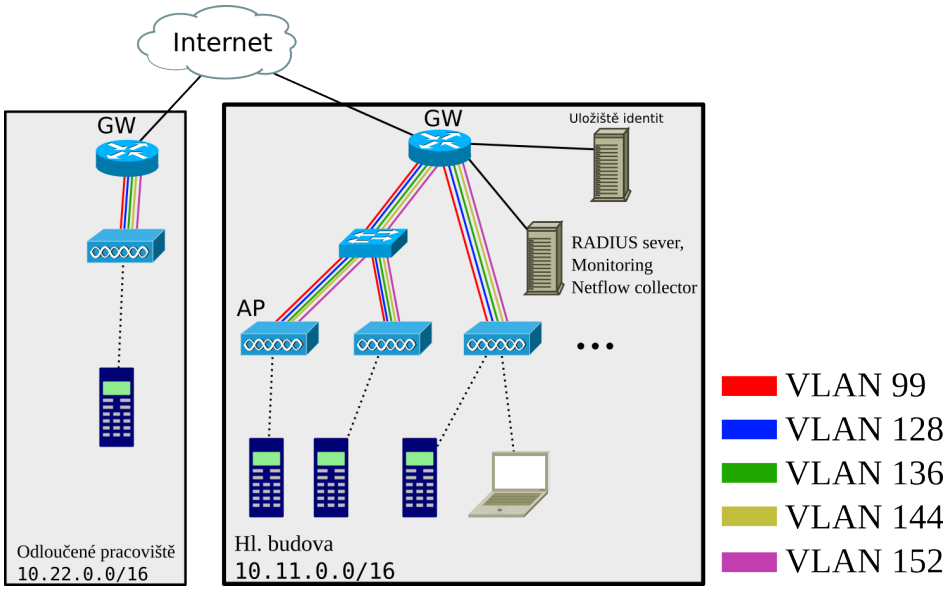
6.1 Síťové prostředí

Síťové prostředí je tvořeno dvěma geograficky oddělenými lokalitami: hlavní budovou a detašovaným pracovištěm. Obě lokality disponují bezdrátovými sítěmi `eduroam` a `free wifi`, které jsou poskytovány prostřednictvím přístupových bodů Wi-Fi. Síť je rozdělena do několika logických podsítí pomocí technologie EEEI 802.1Q VLAN, které slouží pro síťový provoz uživatelů a správu sítě. Použité VLAN tagy jsou v obou lokalitách totožné.

Každá lokalita je vybavena hlavní bránou (gateway), která propojuje lokální infrastrukturu s Internetem a umožňuje komunikaci mezi lokalitami. Trunkové linky zajišťují přenos dat mezi směrovačem, přepínači a přístupovými body, přičemž každá VLAN má svou specifickou funkci:

- **VLAN 99** slouží pro správu síťových prvků a další specializovaný provoz.
- **VLAN 128** je vyhrazena pro hosty.
- **VLAN 136** je určena pro studenty.
- **VLAN 144** je určena pro vyučující.
- **VLAN 152** je určena pro uživatele `free wifi` sítě.

Hlavní budova využívá adresní prostor `10.11.0.0/16`, zatímco detašovanému pracovišti je přidělen adresní prostor `10.22.0.0/16`. Obě lokality jsou propojeny prostřednictvím VPN, která slouží nejen k zajištění autentizace klientů, ale také k managementu přístupových bodů a sdílení dalších síťových zdrojů mezi oběma lokalitami.



Obrázek 6.1: Schéma sítě modelové implementace

V modelové implementaci je pro management vyčleněna síť VLAN 99 s rozsahy `10.11.99.0/24` a `10.xx.99.0/24`. Tato síť obsahuje DHCP server, který přiděluje IP adresy v rozsahu `10.xx.99.100`–`10.xx.99.254`. Tento rozsah slouží pro nově přidané přístupové body, které ještě nebyly finálně nakonfigurovány. Nakonfigurované přístupové body pak používají statické IP adresy v rozmezí `10.xx.99.2`–`10.xx.99.99`.

Další síťové rozsahy není nutné specifikovat, avšak z důvodu přehlednosti a konzistence je doporučeno přizpůsobit třetí oktet síťového rozsahu číslu VLAN tagu, nebo naopak, jak doporučuje zdroj (Lainhart, 2022). Je důležité zajistit, aby zvolené rozsahy byly dostatečně velké, a tím umožnily adresaci všech zařízení v síti.

V síti hlavní budovy se nachází server, který zajišťuje více funkcí:

- **RADIUS server:** Slouží k autentizaci uživatelů sítě `eduroam`, čímž podporuje bezpečný a standardizovaný přístup.
- **Monitoring infrastruktury:** Uchovává data získaná při monitoringu, mezi něž patří logy, záznamy o komunikaci a stavy jednotlivých přístupových bodů.

Dále je zde přítomen druhý server, který poskytuje identity potřebné pro činnost RADIUS serveru. Tento návrh centralizuje všechny funkce, které jsou důležité pro správu sítě a zajišťuje jejich integraci do jednotného systému.

6.2 Příprava zařízení

Přístupový bod je z výroby vybaven proprietárním operačním systémem dodávaným výrobcem zařízení. Tento systém je nutné nahradit zvoleným operačním systémem, což se provádí přepsáním původního obsahu paměti zařízení.

6.2.1 Kompilace a sestavení OpenWrt

Základní možností jak získat systém OpenWrt je stáhnout již předkompilovaný a sestavený obraz systému OpenWrt pro dané zařízení. Tento předkompilovaný a sestavený systém obsahuje software pro provoz síťového zařízení v režimu směrovače/Wi-Fi směrovače (obsahuje firewall a DHCP servery). Tato možnost je pro uživatele jednoduchou cestou jak získat OpenWrt pro své zařízení, neumožňuje však úpravy samotného systému a změny nastavení, které by se aplikovali před prvním spuštěním.

Jedním z požadavků modelové implementace je centrální management síťových prvků. K naplnění tohoto cíle je nutné mít nové zařízení po prvním spuštění již integrované do existující sítě, tak aby bylo možné k němu přistoupit a provést dodatečnou konfiguraci pomocí nástroje pro automatizaci konfigurace.

Z tohoto důvodu je vhodné zvolit druhou z možností jak získat systém OpenWrt a to zkompilovat a sestavit svůj vlastní obraz systému. Tímto způsobem lze upravit obraz systému, přidat požadovanou inicializační konfiguraci a následně jej nahrát do paměti zařízení. Další výhodou je to, že můžeme systém OpenWrt sestavit bez zbytného softwaru pro účely přístupového bodu a ušetřit tím místo v paměti zařízení.

Zkompilovat a sestavit vlastní obraz systému je možné na lokálním počítači po naklonování repozitáře se zdrojovým kódem, nebo si ho nechat zkompilovat a sestavit přímo na serverech samotného projektu OpenWrt. Pro účely této práce využijeme možnosti sestavení systému na serverech OpenWrt. Tato možnost umožňuje provést potřebné úpravy před sestavením obrazu systému. Vyhneme se tak potřebě vytvoření prostředí pro křížovou kompilaci[1] a sestavení obrazu OpenWrt na vlastním počítači.

Webová aplikace `https://firmware-selector.OpenWrt.org/` umožňuje získat již předkompilované obrazy systému OpenWrt i možnost vytvořit si obraz vlastní. Obrazy systému jsou pro různé zařízení unikátní, protože různá zařízení využívají různé procesorové architektury, mají různé rozložení paměti a obsahují různé periferie pro které je nutné mít adekvátní ovladače. V první části je nutné vybrat model zařízení, pro který chceme získat obraz systému. Následně vybereme verzi systému, kterou chceme získat.

Aplikace standardně ve spodní části stránky nabízí ke stažení tři druhy předkompilovaných obrazů.

[1] Též „cross compilation“ - proces vytvoření spustitelného kódu pro jinou instrukční sadu procesoru než na jaké se kód kompiluje

První je samotné jádro operačního systému - kernel. Další možnosti jsou *factory* obraz a *sysupgrade* obraz. Oba obrazy obsahují stejný systém s tím rozdílem, že factory obraz je doplněný o hlavičky a skripty pro interakci s továrním softwarem zařízení, tak aby tento obraz akceptovaly proprietární nástroje výrobce zařízení pro upgrade firmwaru (zapsání obrazu do paměti zařízení). Factory obraz je tedy určený pro prvotní instalaci OpenWrt, sysupgrade obraz slouží k aktualizaci systému z již funkčního systému OpenWrt.(OpenWrt, 2021)

Po rozkliknutí sekce `Customize installed packages and/or first boot script` můžeme definovat jaké balíčky budou zakomponovány do vytvořeného obrazu systému a jaká se provede inicializační konfigurace. Tlačítkem `Request build` zahájíme sestavení vlastního obrazu OpenWrt, který si následně stáhneme.

Ve vstupu `Installed Packages` jsou předvyplněny softwarové balíčky pro konkrétní zařízení, se kterými se sestaví systém OpenWrt. V tomto seznamu jsou zahrnuty i balíčky nepotřebné pro funkci zařízení v režimu přístupového bodu. Systém sestavíme bez těchto balíčků.

Seznam balíků, které nejsou potřeba pro provoz zařízení v režimu přístupového bodu:

- `dnsmasq` – DNS a DHCPv4 server,
- `firewall4` – překladač uci pravidel firewallu na nftables pravidla,
- `kmod-nft-offload` – kernel modul pro podporu harwarové akcelerace směrování a NATu,
- `luci` – webové rozhraní pro konfiguraci zařízení,
- `nftables` – framework pro filtrování paketů a další práci se síťovým provozem,
- `odhcpd-ipv6only` – démon pro správu IPv6 v síti,
- `ppp` – podpora Point-to-Point protokolu,
- `ppp-mod-pppoe` – rozšíření o podporu PPPoE funkcionality.

Protože budeme dle definovaných požadavků v kapitole 3 používat k autentizaci a autorizaci WPA Enterprise nahradíme balík `wpad-basic-mbedtls` za jeho plnohodnotnou alternativu `wpad-mbedtls`, který mimo jiné implementuje WPA-Enterprise metody pro autorizaci a autentizací klientů.

Balíky lze odebrat připsáním názvů balíků s prefixem `-`:

```
-dnsmasq -firewall4 -kmod-nft-offload -luci -nftables -odhcpd-ipv6only -ppp -ppp-mod-pppoe
↪  -wpad-basic-mbedtls wpad-mbedtls
```

Do vstupu `Script to run on first boot (uci-defaults)` zadáme ash unix shell script, který provede konfiguraci systému při prvním spuštění. Náhled skriptu je uveden v ukázce kódu 7.

Tento skript odstraní výchozí síťové nastavení systému OpenWrt. Ze souboru `/etc/board.json` (popis výchozí konfigurace pro dané zařízení) zjistí původní WAN port zařízení a nad ním vytvoří nové síťové rozhraní s VLAN tagem `99` pro příchozí i odchozí provoz. DHCP client na tomto novém rozhraní

požádá o přidělení IP adresy. Toto síťové rozhraní bude jediné síťové rozhraní, kterému bude přidělena IP adresa. Bude sloužit pro management přístupového bodu.

Skript detekuje, zda dané zařízení obsahuje switch a rozpozná jestli je spravován pomocí DSA nebo nástrojem `swconfig`. DSA nebo-li Distributed Switch Architecture je subsystém Linuxového jádra pro unifikovanou správu specifických embedded přepínačů (switch). Vytváří virtuální síťová rozhraní pro každý port přepínače, což umožňuje jejich správu standardními Linuxovými nástroji. (kernel.org, 2024)

Při použití DSA přepínače OpenWrt automaticky nastaví VLAN tagování pro síťové rozhraní managementu. U přepínače spravovaného nástrojem `swconfig` je potřeba dodatečně nakonfigurovat, aby WAN port přenášel rámce s VLAN tagem 99 na port připojený k ethernetovému rozhraní CPU přístupového bodu.

Skript dále provede změnu root hesla, které je definováno v proměnné `root_password` na začátku skriptu a bezpečnostní nastavení SSH démona s přidáním SSH klíče z proměnné `ssh_key_rsa`. SSH démon není pro některé architektury kompilován s podporou novějších typů klíčů, proto použijeme klíč typu RSA.

6.2.2 Nahrání systému do paměti zařízení

Proces nahrávání obrazu systému OpenWrt se liší podle typu zařízení. Na většinu zařízení lze factory obraz systému OpenWrt snadno nahrát pomocí nástroje pro aktualizaci firmwaru v webovém rozhraní původního systému. Některé zařízení mohou vyžadovat složitější postup. Je doporučeno seznámit se s postupem instalace OpenWrt v dokumentaci k danému zařízení v dokumentaci k OpenWrt.

Tímto jsme získali přístupový bod s open-source systémem OpenWrt a se síťovým přístupem připravený pro následnou konfiguraci dalšími nástroji.

6.3 Konfigurace a správa přístupových bodů

Ke správě konfigurace přístupových bodů byl vytvořen adresář, který odpovídá standardní adresářové struktuře používané v Ansible, jak je znázorněno v příkladu **??**. Tento adresář je uložen v Git repozitáři, který je dostupný na adrese `https://github.com/Pixxcz/thesis-network`.

Na uvedený repozitář je v následujících sekcích pravidelně odkazováno, protože zdrojové soubory jsou často příliš rozsáhlé na to, aby mohly být rozumně vloženy přímo do textu práce. Odkazy na jednotlivé soubory obsahují jejich přesné umístění a názvy. U každé této anotace je navíc poznámka pod čarou s přímým odkazem na zmíněný soubor, což jej činní snadno dohledatelným v repozitáři. Výhodou tohoto přístupu je zároveň to, že repozitář vždy obsahuje aktuální verzi konfigurace, což umožňuje snadnou implementaci případných změn a vylepšení, aniž by bylo nutné upravovat text práce.

```
$ tree -L 2 --dirsfirst thesis-network
.
+-- group_vars
|   +-- accessPoints.yaml
|   +-- dsa.yaml
|   +-- idp_vault.yaml
|   +-- openwrt.yaml
|   \-- swconfig.yaml
+-- host_vars
|   +-- ap_0c806307e88a.yaml
|   +-- ap_107c61992bd8.yaml
|   +-- ...
|   \-- <realm>.yaml
+-- include
|   \-- createHostVars.yml
+-- roles
|   +-- freeradius/
|   +-- gekmihesg.openwrt/
|   +-- network/
|   +-- system/
|   \-- wireless/
+-- templates
|   \-- host_vars.j2
+-- addNewAPs.yml
+-- ansible.cfg
+-- debug.yml
+-- inventory.yml
+-- reboot.yml
\-- setupAPs.yml
```

Kód 1: Adresářová struktura Ansible

V adresáři `roles/gekmihesg.OpenWrt` je naklonován repozitář `ansible-OpenWrt`[2], který přidává roli pro konfiguraci zařízení OpenWrt. Pro použití modulů této role je potřeba umístit spravovaná zařízení do skupiny `OpenWrt` v inventory Ansiblu a použít roli `gekmihesg.OpenWrt` v direktivě `roles:` v playbooku.

6.3.1 Nasazení nových přístupových bodů

Pro nasazení nových přístupových bodů slouží playbook `addNewAPs.yaml`[3], který je umístěný v kořenovém adresáři.

V úvodu bloku úkolů playbooku *Detekce zařízení v síti* jsou v proměnné `subnets` definovány podsítě, které budou skenovány programem `ping`. Skenován je adresní rozsah DHCP serveru, kde jsou hledány dynamicky přidělené IP adresy nově připojených přístupových bodů a taky adresní rozsah mimo DHCP server, aby byly zjištěny volné IP adresy, ze kterých bude novým přístupovým bodům přidělena statická IP adresa. Nalezená nová zařízení jsou přidána do dynamického inventáře se kterým se pracuje

[2] `https://github.com/gekmihesg/ansible-OpenWrt`

[3] `https://github.com/Pixxcz/thesis-network/blob/main/addNewAPs.yaml`

v dalším úkolu. Volné IP adresy jsou uloženy do proměnné.

Další blok úkolů playbooku *Získání hostů a generování hostname* získá fakta o nalezeném zařízení a vyextrahuje z něj MAC adresu, která bude sloužit jako unikátní identifikátor zařízení. Je vytvořen seznam nalezených zařízení indexovaný jejich hostnamem, který je vytvořený z MAC adresy a prefixu `ap_`.

V bloku úkolů playbooku *Přidání nového access pointu do správy* je vytvořen prostřednictvím playbooku `include/createHostVars.yaml`[4] soubor s definicí specifických proměnných pro dané zařízení. Soubor je umístěn v adresáři `host_vars` a je vytvořen podle šablony `templates/host_vars.j2`[5]. Název souboru a hostname zařízení používaný v rámci současného playbooku je získán v předchozím kroku. Při vytváření souboru je pro dané zařízení přiřazena do proměnné `device_ip_address` IP adresa z rozsahu volných IP adres. Nalezené zařízení je s přiřazenou IP adresou přidáno do `invetory.yaml`[6].

V bloku úkolů playbooku *Nastavení statické ip na mgmnt interface* je prostřednictvím role `network` na nalezených nových zařízeních nastavena statická adresa zapsána v předchozím kroku do proměnné `device_ip_address` a provedeno další síťové nastavení.

Playbook se spouští příkazem `ansible-playbook addNewAPs.yaml`.

6.3.2 Konfigurace přístupových bodů

Výchozí situace před finálním nakonfigurováním je taková, že všechny přístupové body jsou přidány do souboru `inventory.yaml` a mají nastavenou statickou IP adresu. Zařízení jsou v `inventory.yaml` rozdělena do skupin podle toho zda používají ke konfiguraci switche DSA[7] nebo nástroj `swconfig`, protože switch je nutno pro oba případy konfigurovat odlišně. Princip DSA je popsán v závěru sekce 6.2.1. Skupiny `dsa` a `swcofig` jsou v meta-skupině `accessPoints`, která je v další meta-skupině `OpenWrt` pro správnou funkci role `gekmihesg.OpenWrt`.

Zápis globální systémové konfigurace

Konfigurace je zapsána deklarativním způsobem v souboru `group_vars/openwrt.yaml`[8] a také v soouboru `group_vars/accessPoints.yaml`[9]. Tato konfigurace se aplikuje na všechny přístupové body, je zde tedy zapsána konfigurace, která bude na všech přístupových bodech stejná. Jedná se o nastavení

[4] `https://github.com/Pixxcz/thesis-network/blob/main/include/createHostVars.yaml`

[5] `https://github.com/Pixxcz/thesis-network/blob/main/templates/host_vars.j2`

[6] `https://github.com/Pixxcz/thesis-network/blob/main/inventory.yaml`

[7] Distributed switch Architecture

[8] `https://github.com/Pixxcz/thesis-network/blob/main/group_vars/openwrt.yaml`

[9] `https://github.com/Pixxcz/thesis-network/blob/main/group_vars/accessPoints.yaml`

```yaml
openwrt:
  children:
    accessPoints:

accessPoints:
  vars:
    ansible_scp_extra_args: "-O"
  children:
    dsa:
      hosts:
        ap_107c61992bd8:
          ansible_host: "10.11.99.2"
        ap_c47154393f26:
          ansible_host: "10.22.99.2"
    swconfig:
      hosts:
        ap_b04e26bbc7e3:
          ansible_host: "10.11.99.3"
        ap_0c806307e88a:
          ansible_host: "10.11.99.4"
```

Kód 2: YAML konfigurace inventáře pro Ansible

systému, softwarových mostů (bridge), síťových rozhraní a Wi-Fi rozhraní.

```yaml
system:
  hostname: "{{ device_hostname | default(inventory_hostname_short) }}"
  timezone: "CET-1CEST,M3.5.0,M10.5.0/3"
  zonename: "Europe/Prague"
  log_ip: "10.11.99.51"
```

Kód 3: Skupinové proměnné v souboru `group_vars/openwrt.yaml`

Sekce System YAML souboru složí pro konfiguraci systému. V sekci se konfiguruje časové pásmo a nastavuje se logování na externí server. Toto nastavení ovlivňuje soubor `/etc/config/system` v systému OpenWrt. Konfigurace systému je provedena Ansible rolí `system`.

Role k provedení systémové konfigurace

Role `system` v `roles/system/tasks/main.yaml`[10] využívající modul `uci` upraví konfiguraci sekce systém v prostředí operačního systému OpenWrt. Po provedení úprav odchází ke commitu[11] změn. V případě, že došlo ke změnám je restartována systémové služba pomocí handleru, který je definován v souboru `roles/system/handlers/main.yaml`[12].

[10] `https://github.com/Pixxcz/thesis-network/blob/main/roles/system/tasks/main.yaml`

[11] uložení

[12] `https://github.com/Pixxcz/thesis-network/blob/main/roles/system/handlers/main.yaml`

Zápis globální síťové konfigurace

Konfigurací v souboru `group_vars/accessPoints.yaml` jsou ovlivněny tyto konfigurační soubory systému OpenWrt:

- `/etc/config/network`[13]
- `/etc/config/wireless`[14]

```
device_2g_radio: "{{ (ansible_facts.openwrt_wireless | dict2items | selectattr('value.config.band',
↪  'equalto', '2g') | map(attribute='key') | first) | default('none') }}"
device_5g_radio: "{{ (ansible_facts.openwrt_wireless | dict2items | selectattr('value.config.band',
↪  'equalto', '5g') | map(attribute='key') | first) | default('none') }}"
```

V úvodu se z Ansible faktů zjistí logické identifikátory 5Ghz a 2,4GHz Wi-Fi rádií a uloží se do proměnných `device_2g_radio` a `device_5g_radio`. Tyto proměnné slouží k individuální konfiguraci přístupového bodu, kdy pomocí těchto proměnných lze omezit vybrané Wi-Fi sítě pouze na dané rádia.

```
network_devices:
  - name: "br-vlan128"
    state: "present"
    type: "bridge"
    ports: ["{{ device_bridge_port }}.128"]
  - ...
```

Sekce `network_devices` definuje jaká síťová zařízení budou vytvořena. V tomto případě budou podle kapitoly 6.1 vytvořena čtyři síťová zařízení typu bridge. Bridge nebo také síťový most je logické síťové zařízení, které spojuje více fyzických nebo virtuálních rozhraní do jednoho síťového segmentu. V tomto případě budou vytvořeny síťové mosty (bridge) pro propojení ethernetového portu s VLAN a adekvátní Wi-Fi sítě.

```
network_interfaces:
  - id: "mgmnt"
    proto: "static"
    device: "{{ device_bridge_port }}.99"
    ipaddr: "{{ device_ip_address }}/24"
    gateway: "{{ device_ip_address | regex_replace('\\.[0-9]+$', '.1') }}"
    dns: ["{{ device_ip_address | regex_replace('\\.[0-9]+$', '.1') }}"]
  - id: "hoste_wifi"
    proto: "none"
    device: "br-vlan128"
  - ...
```

[13] `https://OpenWrt.org/docs/guide-user/network/network_configuration`

[14] `https://OpenWrt.org/docs/guide-user/network/wifi/basic`

Sekce `network_interfaces` definuje čtyři nová síťová rozhraní, která budou vytvořena nad vytvořenými síťovými mosty (bridge).

```
wireless_devices_default:
  disabled: 0
  country: CZ
```

Sekcí `wireless_device_default` jsou nastaveny výchozí hodnoty pro všechna Wi-Fi rádia zařízení. V tomto případě je nastaveno, že rádio bude ve výchozím stavu zapnuto a bude nastaven „country code“ na hodnotu CZ. Toto nastavení zajišťuje dodržování právních předpisů týkajících se používání elektromagnetického spektra definovaných ve všeobecného oprávnění ČTÚ[15]. Zejména se jedná o omezení počtu kanálů a maximálního vysílacího výkonu.

```
wireless_interfaces:
  - ssid: "eduroam"
    mode: "ap"
    ifname: "eduroam"
    network: ["hoste_wifi"]
    encryption: "wpa2+ccmp"
    auth_server: "10.11.99.99"
    auth_secret: "Jednokolka123"
    dynamic_vlan: "2"
    vlan_tagged_interface: "{{ ansible_facts.openwrt_interfaces.mgmnt.device |
    ↪  regex_search('eth[0-9]+|wan') }}"
    vlan_bridge: "br-vlan"
    vlan_naming: "1"
    isolate: "1"
    ieee80211r: "1"
    ft_over_ds: "0"
    device: "{{ ansible_facts.openwrt_wireless | list }}"
  - ...
```

V poslední sekci `wireless_interfaces` jsou definovány dvě Wi-Fi sítě podle požadavků v kapitole 3. První síť má SSID `eduroam` a k zabezpečení využívá WPA2 Enterprise. Adresa autentizačního a autorizačního serveru je definovaná v proměnné `auth_server`, heslo je uloženo v proměnné `auth_secret`. Je zde umožněn roaming dle 802.1r. Jedním z požadavků na modelovou implementaci je dynamické přidělování uživatelů do příslušných VLAN, tohoto je docíleno pomocí parametrů v proměnných `dynamic_vlan`, `vlan_tagged_interface`, `vlan_bridge` a `vlan_naming`, které ovlivňují chování démona `hostapd`.

`Hostapd` je nástroj určený ke správě a vytváření Wi-Fi sítí. Kromě toho zajišťuje autorizaci zařízení a funguje jako 802.1X autentizátor, tedy zprostředkovává autentizaci mezi klientským zařízením a autentizačním serverem. V OpenWrt je součástí balíčku `wpad-mbedtls`, který jsme nainstalovali

[15]Český telekomunikační úřad

ve své plnohodnotné verzi při kompilaci v podsekci 6.2.1. Parametr `dynamic_vlan hostapd` definuje zda je klientské zařízení dynamicky přidělováno do vvlan. Jsou definovány tři stavy. 0 - neprobíhá dynamické přiřazování vlan, 1 – VLAN není požadováno, použije se výchozí síťové rozhraní, 2 – VLAN je od RADIUS serveru požadována jinak klientské zařízení nepřipojí. Pokud je dynamické přidělování zapnuto `hostapd` Wi-Fi klienty přiřazuje do adekvátních bridgů. K tomu potřebuje znát prefix síťového mostu (bridge), který je definovaný v `hostapd` parametru `vlan_bridge`. `vlan_naming` s hodnotou 1 říká, že do síťového mostu (bridge) bude přiraženo ethernetové rozhraní ve formátu `<vlan_tagged_interface>.<VLANtag>`, v případě hodnoty 0 je přiřazeno rozhraní s názvem `vlan<vlantag>`. V parametru `vlan_tagged_interface` je tedy uvedeno ethernetové rozhraní propojující přístupový bod a směrovač sítě. (OpenWrt, 2024)

Druhou definovanou Wi-Fi síťí je `free wifi` která nevyžaduje žádnou autentizaci a je přiřazena k síťovému rozhraní `free_wifi`.

Obě dvě Wi-Fi sítě jsou ve výchozím nastavení spuštěny na všech Wi-Fi rádiích zařízení.

Zápis unikátní konfigurace pro jednotlivé zařízení

Specifická konfigurace zařízení je zapsána v `host_vars/ap_<mac_adresa>.yaml`. Tato konfigurace slouží k nastavování IP adresy zařízení, hostname zařízení a nastavení přepínače. Zde je nutné nastavit jednotlivé VLANy a jejich porty. Na jednotlivých přístupových bodech je pomocí této konfigurace také možné nastavovat kanál Wi-Fi rádií, vysílací výkon nebo specifikovat kanály jednotlivých rádií.

```yaml
device_ip_address: "10.11.99.2"
device_hostname: "ap-ucebna-101"
patro: "1"
ucebna: "02"

network_devices_append:
  - name: "br0"
    state: "present"
    type: "bridge"
    ports: ["wan", "lan1", "lan2", "lan3"]
network_bridge_vlan_filtering:
  - vlan: "99"
    device: "{{ device_bridge_port }}"
    ports: ["wan:t"]
  - vlan: "128"
    device: "{{ device_bridge_port }}"
    ports: ["wan:t", "lan1", "lan2:t"]
  - ...
```

U přístupových bodů implementující DSA se nejdříve definuje v sekci `network_devices_append` nový síťový most (bridge) k vytvoření, který obsahuje fyzické ethernetové porty zařízení. V sekci `network_bridge_vlan_filtering` se definují čísla VLAN a ethernetové porty, na kterých budou tyto VLAN aktivní. Dále se určí, které porty budou označeny tagem a které nikoli.

```
network_swconfig:
  - vlan: "128"
    ports: "0t 1t"
    description: "hoste_wifi"
  - ...
```

U zařízení, kde se switch spravuje nástrojem `swconfig`, konfiguraci provádí sekce `network_swconfig`. Je zde uvedeno číslo VLAN a ethernetový port. V případě `swconfig` switche je potřeba nahlédnout do dokumentace k zařízení, protože jednotlivé porty jsou na každém zařízení číslovány jinak.

```
wireless_devices:
  - id: "{{ device_2g_radio }}"
    channel:  "11"
    txpower: "3"
  - id: "{{ device_5g_radio }}"
    channel:  "52"
```

Sekce `wireless_devices` slouží k nastavení kanálů a vysílacích výkonů jednotlivých rádií. Výkon se zapisuje v dBm a je nutné se podívat do dokumentace k danému zařízení jaké výkony umožňuje nastavit. Stejně tomu je i u kanálu. Některá zařízení mají omezené možnosti nastavení kanálu, zejména v pásmu 5GHz.

```
wireless_interfaces_override:
  # - ssid: "eduroam"
  #   device: "{{ [device_2g_radio] }}"
   - ssid: "free wifi"
     device: "{{ [device_2g_radio] }}"
```

Sekcí `wireless_interfaces_override` lze omezit danou Wi-Fi síť pouze na určité rádio.

Role k provedení síťové konfigurace

Role `network` s hlavním souborem v `roles/network/tasks/main.yaml`[16] nejdříve iteruje sekcí konfigurace `network_swconfig` a konfiguruje prostřednictvím playbooku `swconfig.yaml` switch u zařízení kde je tato sekce definována. Dále jsou playbookem `device.yaml` nastaveny síťové mosty (bridge) opět iterací přes sekci `network_devices` doplněnou o VLAN filtrující bridge z `network_devices_append` u zařízení implementujících DSA.

Ve třetím kroku jsou nakonfigurovány VLANy pomocí playbooku role `vlan_filtering.vlan`. V posledním kroku jsou vytvořeny síťové rozhraní playbookem role `interface.yaml`, provedené změny

[16] `https://github.com/Pixxcz/thesis-network/blob/main/roles/network/tasks/main.yaml`

jsou aplikovány a je restartováno síťování na přístupovém bodu. Pro případ změny IP adresy zařízení jsou na konci úkoly měnící IP zařízení v inventory běžícího playbooku.

Role `wireless` s hlavním souborem v `roles/wireless/tasks/main.yaml`[17] provádí konfigurací Wi-Fi rádií a jejich sítí. Role iteruje všemi dostupnými rádii zařízení a nastavuje jejich výchozí hodnoty z proměnné `wireless_devices_default` a kanály s vysílacími výkony, tak jak jsou definovány u konkrétních zařízení. Toto je provedeno v playbooku role `device.yaml`[18]

Následně vytváří Wi-Fi sítě tak jak jsou definovány v `group_vars/accessPoints.yaml` pomocí playbooku role `interface.yaml`[19]. Při vytváření se zohledňují individuální nastavení pro jednotlivé přístupové body v sekci YAML konfigurace `wireless_interfaces_override`.

Ve všech případech je použit pro konfigurace přístupových bodů modul `uci` z role `gekmihesg.OpenWrt`.

Playbook k provedení finální konfigurace

Playbook využívá role `system`, `network` a `wireless`, které provedou konfiguraci podle nastavení v proměnných v adresářích `group_vars` a `host_vars`. Po nastavení zařízení jsou spuštěny další úkoly v sekci `post_tasts`. Pomocí modulu `opkg` jsou na všechny přístupové body nainstalován balíčky `prometheus-node-exporter-lua` a `prometheus-node-exporter-lua-hostapd_station`. Tyto balíčky jsou modulem `uci` nakonfigurovány a všechna zařízení jsou restartována.

Playbook se spouští příkazem `$ ansible-playbook -i inventory.yaml setupAPs.yaml`. Tímto je dokončena finální konfigurace všech přístupových bodů infrastruktury.

6.4 Konfigurace autentizačního serveru

Národní operátor sítě eduroam v české republice CESNET z. s. p. o. nabízí Ansible roli pro konfiguraci autentizačního serveru freeRADIUS. Tato role je dostupná v repozitáři `https://github.com/CESNET/ansible-freeradius`. Naklonováním repozitáře přidáme roli do adresáře `roles/` adresářové struktury Ansible. Tato role bude využita při konfiguraci freeRADIUS serveru v této modelové implementaci.

Role provádějí kompletní instalaci a konfiguraci freeRADIUS serveru. Role předpokládá požití jednoho z linuxových systémů založených na distribuci Debian (s balíčkovacím systémem `apt`) nebo CentOS/RHEL (s balíkovacím systémem `dnf`) Na autentizačním serveru musí být zprovozněn SSH server a je vhodné přidat autentizační klíče pro přístup Ansible.

Po naklonování role repozitáře je nutné definovat adresu freeRADIUS serveru v Ansible inventory a konfiguraci v adresářích `group_vars`, `host_vars`. Dále pak přidat certifikáty podle podsekce 6.4.2.

[17] `https://github.com/Pixxcz/thesis-network/blob/main/roles/wireless/tasks/main.yaml`

[18] `https://github.com/Pixxcz/thesis-network/blob/main/roles/wireless/tasks/device.yaml`

[19] `https://github.com/Pixxcz/thesis-network/blob/main/roles/wireless/tasks/interface.yaml`

Role je distribuována s příklady konfigurace. Požadavkům modelové implementace odpovídá příklad konfigurace `semik-dev.cesnet.cz-IdPSP.yaml` z adresáře role `examles/`. Tento příklad nakopírujeme do `host_vars/<realm>.yaml`. Proměnná `eduroam.realm` slouží k definování doménového prostoru (realm), který je specifický pro danou instituci.

6.4.1 Zdroj identit

```
ldap:
  URL: ldaps://ldap1.skola.cz:636
  CAChain: certs/chain_TERENA_SSL_CA_3.pem
  eduroam:
    bindDN: uid=rad1,ou=Special Users,dc=skola,dc=cz
    bindPass: '{{ skola_cz.ldap_passwd }}'
  peopleDN: dc=skola,dc=cz
  attrs:
    uid: uid
    eduroamPassword: radiusPassword
```

Použitá konfigurace ověřuje identity vůči LDAP serveru. V sekci `ldap` konfigurace se definuje LDAP server (`URL`), Certifikát CA (`CAChain`) a přihlašovací údaje k LDAP. Dále se nastavují názvy atributů, tedy `attrs.uid`, který definuje název atributu s uživatelským jménem, `attrs.eduroamPassword` který definuje název atributu s uživatelským heslem. `peopleDN` vybírá atributy, podle kterých budou vyhledáváni uživatelé. Dalším důležitým atributem v LDAP je `memberOf`, podle kterého je v podsekci 6.4.4 přiřazena konkrétní VLAN.

6.4.2 Certifikáty

freeRADIUS pro zajištění různých typů komunikace potřebuje certifikáty. Ty vytváření bezpečnou komunikaci při komunikaci se suplikanty, jinými RADIUS servery nebo LDAP serverem.

Na obrázku 6.2 jsou znázorněna spojení, která jsou vytvořena při autentizaci klientského zařízení napříč infrastrukturou eduroam. Jedná se o spojení RADIUS serveru s národním RADIUS serverem prostřednictvím zabezpečeného protokolu RADSEC a spojení EAP mezi klientským zařízením (suplikantem) a autorizačním serverem. Tyto spojení jsou zabezpečena certifikáty, které je potřeba nastavit freeRADIUS serveru. Je zde nastavena lokace RADSEC certifikátu s veřejným a privátním klíčem:

```
  radsec:
    certificate: certs/radius.skola.cz.crt
    private_key: certs/radius.skola.cz.key
    private_key_password: "{{ skola_cz.radsec_key_password }}"
  EAP:
    certificate: certs/radius.skola.cz.crt
    private_key: certs/radius.skola.cz.key
    private_key_password: "{{ skola_cz.eap_key_password }}"
```

Proměnná `eduroam.radsec.private_key_password` s heslem k privátnímu klíči odkazuje na proměnnou `<realm>.radsec_key_password` v souboru `group_vars/idp_vault.yaml`. Stejně tak je lokace certifikátu pro metody EAP PEAP a TTLS definována v proměnné `eduroam.EAP.certificate` a privátní klíč v `eduroam.EAP.private_key`.

Autor role oddělil použitá hesla privátních klíčů a heslo k LDAP do samostatného souboru, aby tyto hesla bylo možné šifrovat pomocí funkce Ansible Vault[20].

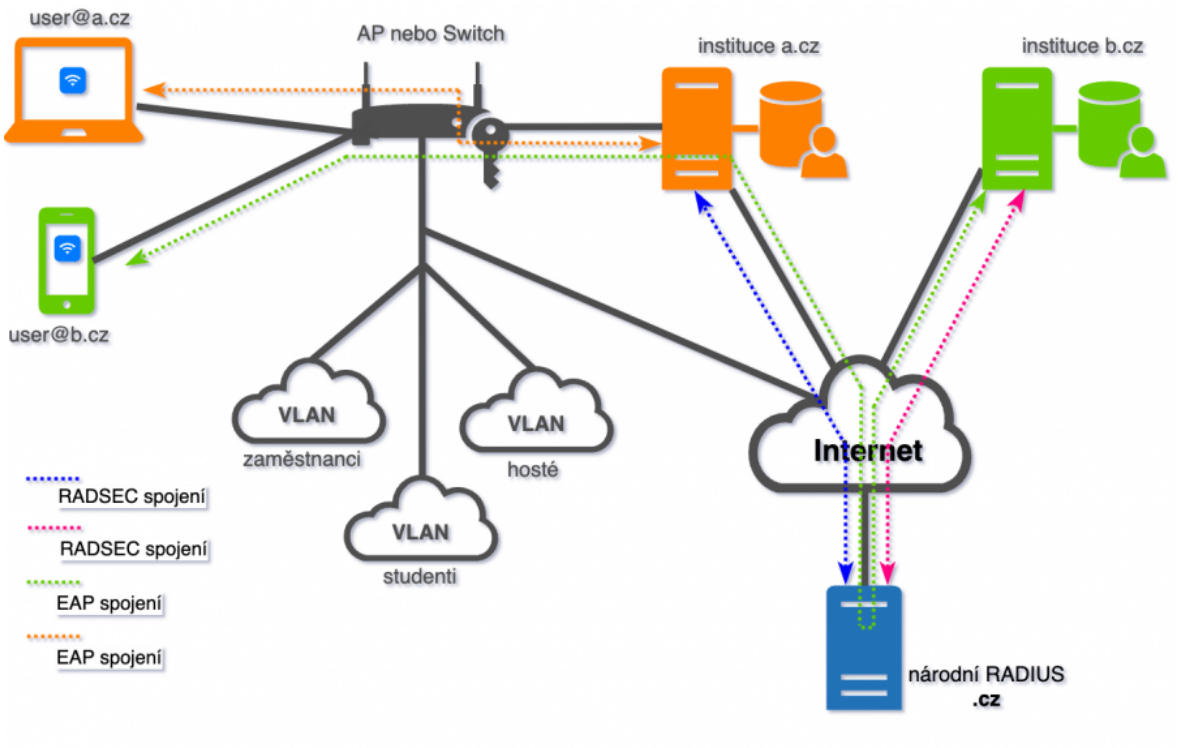
```
skola_cz:
  ermon_secret: <heslo_k_ermon.cesnet.cz>
  ldap_passwd: <heslo_k_LDAP>
  radsec_key_password: <heslo_k_privatnimu_klici_pro_radsec>
  eap_key_password: <heslo_k_privatnimu_klici_pro_eap>
```

Kód 4: Obsah souboru `idp_vault.yaml`

Národní RADIUS server důvěřuje pouze certifikátům, které vydala certifikační autorita CESNET, tedy sám národní operátor nebo certifikační autorita společnosti GÉANT. Musíme tedy požádat o vydání certifikátu. (eduroam.cz, 2024)

Pro EAP spojení lze použít certifikát podepsaný vlastní CA nebo certifikát od CESNETu (používaný v RADSEC). Pak je nutné konfigurovat klienty pomocí nástroje eduroam CAT nebo jim manuálně importovat kořenové certifikáty vlastních CA. Třetí možností je použít certifikát podepsaný důvěryhodnou kořenovou CA.

[20] `https://docs.ansible.com/ansible/2.9/user_guide/vault.html`



Obrázek 6.2: Spojení při autentizaci 802.1X v eduroam. (eduroam.cz, 2024)

6.4.3 Autentizátory

Přístup přístupových bodů (autentizátorů) k freeRADIUS serveru se spravuje v sekci `radius.NAS`. Zde je definováno heslo použité v proměnné `auth_secret` z podsekce 6.3.2. V modelové implementaci jsou přístupové body na dvou geograficky oddělených lokacích, přístupové body jsou v každé lokaci v jiné síti, proto jsou zde uvedeny záznamy pro dvě sítě.

```
radius:
  NAS:
  - ipaddr: 10.11.99.1/24
    secret: Jednokolka123
    shortname: SiteA
  - ipaddr: 10.22.99.1/24
    secret: Jednokolka123
    shortname: SiteB
```

6.4.4 Přiřazení VLAN

Odpověď **RADIUS Access-Accept** freeRADIUS serveru obsahuje informaci o použitém VLAN tagu. Ta je obsažena ve specifický atributech:

- Atribut **Tunnel-Type** určuje typ tunelu, respektive logického oddělení síťového prostoru. Hodnota `VLAN` označuje, že tunel bude používat technologii VLAN.
- **Tunnel-Medium-Type** definuje typ média (technologii), které se používá pro tunelování.
- **Tunnel-Private-Group-ID** určuje identifikátor skupiny, což je také číslo VLAN tagu.

Pro správné zasílání těchto atributů serverem freeRADIUS je nutné upravit jeho konfiguraci. Podpora dynamického přiřazení VLAN tagu je konfigurována v sekci `post_tasks` v Ansible playbooku pro nastavení freeRADIUS serveru `freeradius.yaml`[21].

Úkol *Pridani VLAN pro lokální identity podle skupiny* v sekci `post_tasks` upravuje konfiguraci tak, aby server přiřazoval atributy pro zařazení do VLAN v rámci odpovědi typu RADIUS Access-Accept. Tento proces je aplikován pouze pro identity, které patří naší instituci. VLAN tagy jsou přiřazeny na základě uživatelských skupin definovaných v LDAP podle atributu `memberOf`.

Další úkol s názvem *Přidaní VLAN pro cizí identity* upravuje konfiguraci tak, aby přidávala návštěvníkům naší instituce VLAN tag, který je určen pro přístup do sítě pro hosty.

Úkol *Filtrování odchozích atributu* zahrnuje konfiguraci filtrace atributů při zasílání odpovědi na externí autorizační server v rámci sítě eduroam, což zajišťuje, že atributy definující použitou VLAN v naší lokální síti nejsou zasílány při požadavcích na autentizaci z jiných externích RADIUS serverů.

Úkol *Nastavit use_tunneled_reply* v části peap zajišťuje, že freeRADIUS předá atributy pro konfiguraci VLAN ze sekundárního (inner) EAP tunelu, kde jsou přiřazeny na základě identity do primárního (outer) EAP tunelu, aby mohly být zaslány v rámci **RADIUS Access-Accept** zprávy.

Spuštěním playbooku `freeradius.yaml` získáme nakonfigurovaný freeRADIUS server, který je možné registrovat u národního operátora sítě eduroam.

6.5 Monitoring a logování

Pro získání komplexního přehledu o přístupových bodech a síti je potřeba shromáždit informace ze samotných přístupových bodů, z RADIUS serveru a z gatewaye (směrovače).

6.5.1 Systémový log

Systém OpenWrt používá nástroj `logd` z projektu ubox[22] k řízení systémového logování. Tento nástroj sbírá logovací zprávy od všech programů běžících na systému OpenWrt a ukládá je do centrálního systémového logu. Ve výchozí konfiguraci je tento log uložen do cyklické vyrovnávací paměti umístěné v operační paměti (RAM). Tento přístup minimalizuje zápisy do trvalé paměti zařízení, což je důležité

[21] `https://github.com/Pixxcz/thesis-network/blob/main/freeradius.yaml`

[22] `https://git.OpenWrt.org/?p=project/ubox.git`

zejména u zařízení s NAND pamětí, která by mohla nadměrnými zápisy degradovat.

Kvůli omezené kapacitě cyklické vyrovnávací paměti v RAM přesměrujeme logy z přístupových bodů na centrální logovací server přes síť. Nástroj `logd` tuto funkci podporuje a modelová implementace výše tuto konfiguraci zahrnuje. Toto je docíleno pomocí proměnné `log_ip` v `group_vars/openwrt.yaml`.

Na centrálním serveru mohou být logy dále zpracovávány a analyzovány pomocí dalších nástrojů.

6.5.2 Log RADIUS serveru

Pro získání detailnějších informací o lokálních autentizovaných klientech, zejména prostřednictvím sítě eduroam je potřeba provést úpravy serveru. Logování na FreeRADIUS serveru zajišťuje modul linelog, který umožňuje zapisovat záznamy do systémového logu.

Konfigurace[23] dostupná na webu freeradius.com ukazuje příklad několika instancí modulu `linelog`, které slouží k logování příchozích RADIUS požadavků nebo odpovědi. Tyto instance jsou následně volány v hlavním konfiguračním souboru FreeRADIUS `sites-available/default` na příslušných místech v rámci zpracování RADIUS požadavků. Zalogovány jsou lokální požadavky na autorizaci, odpovědi na autentizační požadavky, odpovědi na požadavky přijaté od jiných institucí a odeslané proxy požadavky směrem k jiným institucím.

Ansible úkolem *Kopírování linelog* v sekci `post_tasks` playbooku `ansible.yaml` je soubor linelog nakopírován do konfiguračního adresáře freeRADIUS serveru.

Blokem úkolů **Edit default** je na adekvátní pasáže konfiguračního souboru `sites-available/default` vloženo volání funkcí k logování.

Výsledkem této konfigurace je detailnější log, včetně například atributů NAS-IP-Address, Operator-Name, Calling-Station-Id, tedy IP adresy autentizátoru, název instituce, ze které přišel nebo kam se odesílá požadavek/odpověď a MAC adresa zařízení. Tyto záznamy je nutné dle Technických požadavků a doporučení pro členy federace eduroam.cz[24] archivovat alespoň 6 měsíců.

Ze systémového logu RADIUS serveru lze tyto záznamy opět přeposlat na centrální logovací server.

6.5.3 Monitoring systému a hardwarových prostředků

Pro monitoring systému byl v kapitole 5 vybrán nástroj `prometheus-node-exporter-lua`, který byl nainstalován a nakonfigurován v rámci playbooku `setupAPs.yaml` v sekci `post_tasks`.

`prometheus-node-exporter` shromážděné data o systému publikuje v textovém formátu na vlastním http serveru. Prometheus server tato data pravidelně dotazuje, ukládá je do své databáze časových

[23] `https://wiki.freeradius.org/guide/eduroam`

[24] `https://www.eduroam.cz/_media/cs/technicke_pozadavky_eduroam.pdf`

```
linelog linelog_recv_request {
  filename = syslog
  syslog_facility = local0
  syslog_severity = debug
  format = "action = Recv-Request, %{pairs:request:}"
}

linelog linelog_send_accept {
  filename = syslog
  syslog_facility = local0
  syslog_severity = debug
  format = "action = Send-Accept, %{pairs:request:}"
}

linelog linelog_send_reject {
  filename = syslog
  syslog_facility = local0
  syslog_severity = debug
  format = "action = Send-Reject, %{pairs:request:}"
}

linelog linelog_send_proxy_request {
  filename = syslog
  syslog_facility = local0
  syslog_severity = debug
  format = "action = Send-Proxy-Request, %{pairs:proxy-request:}"
}

linelog linelog_recv_proxy_response {
  filename = syslog
  syslog_facility = local0
  syslog_severity = debug
  reference = "messages.%{proxy-reply:Response-Packet-Type}"
  messages {
    Access-Accept = "action = Recv-Proxy-Accept, User-Name = %{User-Name}, Calling-Station-Id =
    ↪  %{Calling-Station-Id}, %{pairs:proxy-reply:}"
    Access-Reject = "action = Recv-Proxy-Reject, User-Name = %{User-Name}, Calling-Station-Id =
    ↪  %{Calling-Station-Id}, %{pairs:proxy-reply:}"
    Access-Challenge = "action = Recv-Proxy-Challenge, User-Name = %{User-Name}, Calling-Station-ID
    ↪  = %{Calling-Station-Id}, %{pairs:proxy-reply:}"
  }
}
```

Kód 5: Soubor `files/linelog`

řad a umožňuje jejich další zpracování.

6.5.4 Ukládání netflow dat

Pro ukládání NetFlow dat byl v kapitole 5 vybrán nástroj `nfcapd`. Tento nástroj je určen pro instalaci na monitorovací server, kde běží jako systémový démon, tedy proces běžící na pozadí. Při jeho spuštění je nutné parametry specifikovat port, na kterém bude naslouchat příchozím NetFlow datům, a také

```
...
post-auth {
    ...
    # Zalogování úspěšné autentizace
    linelog_send_accept

    Post-Auth-Type REJECT {
        ...
        # Zalogování neúspěšné autentizace
        linelog_send_reject
    }
    ...
}
...
pre-proxy {
    ...
    # Zalogování požadavku do jiné instituce
    linelog_send_proxy_request
}

post-proxy {
    ...
    # Zalogování požadavku přijatého z~jiné instituce
    linelog_recv_proxy_response
}
```

Kód 6: Nastavení logování v `sites-available/default`

adresář, do něhož budou data ukládána.

```
$ nfexpire -u flows/ -t 182d
```

Pomocí nástroje **nfexpire** se v adresáři s daty nastaví jak dlouho budou netflow data uchovávána. Systémový démon poté bude starší data mazat. Podobně jako u logů RADIUS serveru musí být tyto údaje uchovávány alespoň 6 měsíců.

Data v binárním formátu lze prohlížet nástrojem **nfdump**.

```
$ nfdump -R flows/ \
    -t "2024/11/11.20:00:00-infinity" \
    -o "fmt:%ts %td %pr %sap -> %dap %pkt %byt %ismc -> %odmc %idmc -> %osmc" \
    "dst ip <cílová IP adresa> and src port <zdrojový port>"
```

V případě požadavku na sdělení provozních a lokalizačních údajů nejčastěji ze strany orgánů veřejné moci lze pomocí příkazu výše listovat a filtrovat netflow data na základě cílové IP adresy a zdrojového portu.

Kapitola 7

Hodnocení modelové implementace

V této kapitole je hodnocena modelová implementace sítě z kapitoly 6. Cílem této části je zhodnotit, jak navržená implementace odpovídá požadavkům stanovených v kapitole 3 a jaký je její přínos pro plánované provozní nasazení.

Modelová implementace byla nasazena a otestována v prostředí střední školy, která k otestování poskytla 14 existujících přístupových bodů rozmístěných ve dvou patrech školy a dílnách na geograficky oddělené lokalitě. Po ověření jejich kompatibility se systémem OpenWrt bylo zjištěno, že 12 z nich splňuje požadavky na nasazení.

U deseti přístupových bodů bylo možné nahrát firmware OpenWrt přímo prostřednictvím webového rozhraní výrobce, zatímco zbývající dvě zařízení vyžadovala alternativní přístup. Konkrétně bylo nutné aktivovat SSH server a provést přepsání paměti zařízení pomocí příkazového řádku. Následná konfigurace prostřednictvím nástroje Ansible proběhla na všech zařízeních bez komplikací.

Škola poskytla virtuální server, který byl využit pro autentizaci uživatelů prostřednictvím RADIUS serveru propojeného k uložišti identit LDAP a virtuální server pro ukládání dat vzniklých z monitoringu. K datu dokončení této práce nebyl RADIUS server školy z administrativních důvodů plně integrován do sítě eduroam, nicméně ověřování lokálních identit je plně funkční.

Logování a monitoring je řešen pouze ve smyslu získání a uložení potřebných dat z infrastruktury. Nejsou využity žádné pokročilé analytické nebo vizualizační nástroje, ale všechny podstatné informace jsou uloženy, čímž je zajištěna jejich dostupnost pro případné budoucí vizualizace a analýzy.

Správa infrastruktury přístupových bodů a RADIUS serveru pomocí nástroje Asible se při modelové implementaci na testovacích přístupových bodech ukázala jako plnohodnotně funkční. Určitou výhodou může být i to, že obě komponenty eduroam sítě (přístupové body a RADIUS server) je možné konfigurovat deklarativním způsobem pomocí jednoho nástroje. Síťoví administrátoři ocení úsporu času při konfiguraci (zejména během přidávání nových přístupových bodů) a konzistentní konfiguraci napříč infrastrukturou.

Ansible je flexibilní nástroj avšak složitý nástroj pro konfiguraci, což může být nevýhodnou pro administrátory, kteří tento nástroj neznají, protože zápis konfigurace není tak uživatelsky přívětivý jako

konfigurace pomocí grafického rozhraní (GUI).

Při reálné implementaci bylo zjištěno, že se manuální konfigurace vysílacích kanálů a vysílacího výkonu jeví jako problematická pro dynamické prostředí bezdrátových sítí. V průběhu konfigurace je nezbytné, aby administrátor navrhl plán optimálního využití elektromagnetického spektra jednotlivými přístupovými body, avšak tento plán může být snadno narušen vnějšími faktory, například jinými zařízeními.

7.1 Návrhy na vylepšení systému

Jedním z důležitých navrhovaných vylepšení je automatizace procesu aktualizace firmwaru přístupových bodů. Tento proces se musí v modelové implementaci provádět manuálně, což může představovat bezpečnostní riziko a také zbytečnou časovou náročnost pro administrátora. Toto lze vyřešit přidáním role do Ansible, která provede lokální kompilaci systému OpenWrt pro jednotlivé modely přístupových bodů v síti. Sestavený systém lze následně pomocí Ansible nahrát do paměti zařízení.

Vzhledem k plánovanému přechodu systému OpenWrt na `apk` v nadcházející verzi (Krčmář, 2024) bude pro zajištění budoucího provozu infrastruktury nezbytné doplnit modul umožňující instalaci balíčků prostřednictvím `apk`. Je nutné, aby buď správce projektu `ansible-openwrt` na tuto změnu reagoval, nebo aby byla příslušná Ansible role upravena.

Za účelem zvýšení efektivity správy sítě a minimalizace rušení by bylo vhodné navrhnout a implementovat Ansible playbook, který by byl spouštěn periodicky. Tento playbook by umožnil automatizované skenování aktuálního využití spektra na jednotlivých přístupových bodech, na základě čehož by dynamicky přizpůsoboval nastavení vysílacích kanálů a výkonů tak, aby odpovídaly aktuálním podmínkám a minimalizovali vzájemné rušení přístupových bodů.

Posledním volitelným navrhovaným vylepšením je zavedení pokročilých monitorovacích a vizualizačních nástrojů. Toto umožní efektivní sledování provozu sítě a poskytne detailní přehled o stavu přístupových bodů, stavu autentizace a další užitečné informace důležité pro zajištění stability systému.

Závěr

V rámci této práce byla úspěšně implementována rozsáhlá přístupová Wi-Fi síť pomocí open-source nástrojů. Analýza dostupných softwarových řešení pro nasazení, správu a monitoring Wi-Fi sítí poskytla ucelený přehled o aktuálních možnostech a posloužila jako základ pro návrh modelové implementace.

První část práce je zaměřena na teoretické základy a důležité aspekty potřebné pro nasazení rozsáhlých bezdrátových sítí, včetně principů fungování sítí a možností jejich zabezpečení.

Samotná implementace zahrnovala tvorbu Ansible skriptů pro automatizovanou konfiguraci přístupových bodů a nasazení autentizačního serveru. Modelová síť byla testována v reálném prostředí školní sítě, což prokázalo, že open-source technologie mohou být efektivní a spolehlivé řešení pro správu a monitoring rozsáhlých Wi-Fi sítí.

Testování však zároveň odhalilo některé výzvy, zejména vyšší nároky na technické znalosti při správě a údržbě těchto systémů. Na základě zjištění byly navrženy změny a doporučení pro další iterace, zahrnující zejména rozšíření funkcionality monitorovacích nástrojů a zvýšení automatizace.

Tato práce poskytuje praktický návod pro realizaci podobných technologických řešení, přičemž spojuje teoretické poznatky s konkrétními příklady implementace. Výsledky mohou být užitečné pro školy i jiné organizace při budování moderních a efektivních Wi-Fi sítí.

Bibliografie

APPNEL, Timothy, 2023. *The Zen of Ansible* [online]. [cit. 2024-10-15]. Dostupné z: `https://www.ansible.com/blog/the-zen-of-ansible/`.

ASIM, Uneeb, 2022. *DD-WRT vs. Tomato vs. OpenWRT* [online]. [cit. 2024-11-13]. Dostupné z: `https://www.thetechlounge.com/dd-wrt-vs-tomato-vs-openwrt/`.

AWATI, Rahul, 2022. *initialization vector* [online]. [cit. 2024-10-09]. Dostupné z: `https://tbhaxor.com/decrypt-wep-traffic-with-insufficient-ivs/`.

BOUŠKA, Petr, 2007a. *Ethernet - CSMA/CD, kolizní doména, duplex* [online]. [cit. 2024-09-17]. Dostupné z: `https://www.samuraj-cz.com/clanek/ethernet-csma-cd-kolizni-domena-duplex/`.

BOUŠKA, Petr, 2007b. *TCP/IP a ethernet - cesta v síti, aktivní síťové prvky* [online]. [cit. 2024-09-17]. Dostupné z: `https://www.samuraj-cz.com/clanek/tcp-ip-a-ethernet-cesta-v-siti-aktivni-sitove-prvky/`.

BOUŠKA, Petr, 2007c. *VLAN - Virtual Local Area Network* [online]. [cit. 2024-09-18]. Dostupné z: `https://www.samuraj-cz.com/clanek/vlan-virtual-local-area-network/`.

BRAWLEY, William, 2023. *From 802.11b to Wi-Fi 7: What Do Wi-Fi Numbers Mean?* [online]. [cit. 2024-10-09]. Dostupné z: `https://www.pcmag.com/explainers/from-80211b-to-wi-fi-7-what-do-wi-fi-numbers-mean`.

BYEONG GI LEE, Sunghyun Choi, 2008. *Broadband wireless access and local networks : mobile WiMax and WiFi*. Norwood: Artech House. ISBN 978-1-59693-293-7.

CLAISE, Benoît; TRAMMELL, Brian; AITKEN, Paul, 2008. *Information Model for IP Flow Information Export* [online]. 2008-01. [cit. 2024-10-01]. RFC, 5102. Internet Engineering Task Force. Dostupné z: `https://www.rfc-editor.org/rfc/rfc5102`.

CLOUDFLARE, INC, 2024. *GoFlow: High performance, scalable and reliable NetFlow/IPFIX/sFlow collector* [`https://github.com/cloudflare/goflow`]. Accessed: 2024-10-02.

CROWDER, Crystal, 2023. *DD-WRT vs. Tomato vs. OpenWRT: Which Router Firmware Is the Best?* [online]. [cit. 2024-11-01]. Dostupné z: `https://www.maketecheasier.com/dd-wrt-vs-tomato-vs-openwrt-router-firmware/`.

ČESKO, 2005. *Zákon č. 127/2005 Sb., o elektronických komunikacích a o změně některých souvisejících zákonů (zákon o elektronických komunikacích), § 97 odst. 3* [Sbírka zákonů České republiky]. [cit. 2024-01-01]. Dostupné z: https://www.zakonyprolidi.cz/cs/2005-127#p97-3. Aktualizováno k 1. lednu 2024.

ČUHEL, Radim, 2020. *Řízení přístupu k lokální síti pomocí protokolu IEEE 802.1x* [online]. [cit. 2024-09-26]. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=210204.

DECISO B.V., 2024. *Netflow Export & Analyses* [online]. [cit. 2024-11-23]. Dostupné z: https://docs.opnsense.org/manual/netflow.html.

DOCUMENTATION.MERAKI.COM, 2024. *Fundamentals of 802.1Q VLAN Tagging* [online]. [cit. 2024-09-25]. Dostupné z: https://documentation.meraki.com/General_Administration/Tools_and_Troubleshooting/Fundamentals_of_802.1Q_VLAN_Tagging.

EDUROAM.CZ, 2019. *Realm* [online]. [cit. 2024-09-27]. Dostupné z: https://www.eduroam.cz/cs/spravce/pripojovani/realm.

EDUROAM.CZ, 2024. *Certifikaty* [online]. [cit. 2024-09-22]. Dostupné z: https://www.eduroam.cz/cs/spravce/pripojovani/serverove_certifikaty.

FAIRHURST, Gorry, 2012. *Advanced VLANs* [online]. [cit. 2024-09-20]. Dostupné z: https://erg.abdn.ac.uk/users/gorry/course/lan-pages/vlan-advanced.html.

FREECCNASTUDYGUIDE.COM, 2024. *7-4 VLAN Trunking: ISL and 802.1Q* [online]. [cit. 2024-09-21]. Dostupné z: https://www.freeccnastudyguide.com/study-guides/ccna/ch7/7-4-vlan-trunking-isl-802-1q/.

FRESHTOMATO.ORG, 2011. *Quick list of Optware packages* [online]. [cit. 2024-10-23]. Dostupné z: https://wiki.dd-wrt.com/wiki/index.php/Quick_list_of_Optware_packages.

HAIDARZHY, Valerii, 2024. *Open Source: The Future of Router Defense?* [online]. [cit. 2024-11-18]. Dostupné z: https://sirinsoftware.com/blog/open-source-the-future-of-router-defense.

HEAP, Michael, 2016. *Ansible: From beginner to pro*. Apress. ISBN 978-1-4842-1659-0.

HOFSTEDE, Rick; ČELEDA, Pavel; TRAMMELL, Brian; DRAGO, Idilio; SADRE, Ramin; SPEROTTO, Anna; PRAS, Aiko, 2014. Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX. *IEEE Communications Surveys & Tutorials*. Roč. 16, č. 4, s. 2037–2064. Dostupné z DOI: 10.1109/COMST.2014.2321898.

IEEE, 2016. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, s. 1–3534. Dostupné z DOI: 10.1109/IEEESTD.2016.7786995.

IEEE, 2018. *802.3-2018 - IEEE Standard for Ethernet* [online]. [cit. 2024-09-29]. Dostupné z: https://ieeexplore.ieee.org/document/8457469.

IJS2.8U.CZ, 2024. *Ethernet* [online]. [cit. 2024-09-17]. Dostupné z: http://ijs2.8u.cz/index.php?option=com_content&view=article&id=20&Itemid=125.

KERNEL.ORG, 2024. *Distributed Switch Architecture (DSA)* [online]. [cit. 2024-10-09]. Dostupné z: https://docs.kernel.org/networking/dsa/dsa.html.

KRČMÁŘ, Petr, 2024. *OpenWrt mění balíčkovací systém, z vlastního opkg přechází na apk* [online]. [cit. 2024-11-18]. Dostupné z: https://www.root.cz/zpravicky/openwrt-meni-balickovaci-system-z-vlastniho-opkg-prechazi-na-apk/.

KURNAZ, Cetin; ENGIZ, Begum; KOSE, Ugur, 2017. *Investigating the effect of number of users on signal strength level and throughput for Wi-Fi system* [online]. [cit. 2024-11-13]. Dostupné z: https://www.researchgate.net/figure/Wi-Fi-channels-of-24-GHz-range_fig1_321233408.

LAINHART, Brittany, 2022. *10 VLAN Numbering Best Practices* [online]. [cit. 2024-11-18]. Dostupné z: https://climbtheladder.com/10-vlan-numbering-best-practices/.

LEŠEK, Vladimír, 2019. *Autentizace v lokálních sítích pomocí IEEE 802.1x* [online]. [cit. 2024-09-27]. Dostupné z: https://dspace.cvut.cz/bitstream/handle/10467/82727/F3-BP-2019-Lesek-Vladimir-Autentizace%20v%20lokalnich%20sitich%20pomoci%20IEEE%20802.1x.pdf?sequence=-1&isAllowed=y.

LUCENTE, Paolo, 2014. *Collec'ng NetFlow with pmacct* [online]. [cit. 2024-10-26]. Dostupné z: http://www.pmacct.net/Lucente_collecting_netflow_with_pmacct_v1.2.pdf.

MAISEYEU, Aliaksei, 2019. *Dawn of the Infrastructure as Code* [online]. [cit. 2024-10-15]. Dostupné z: https://lean-delivery.com/2019/12/infrastructure_as_code.html.

NAMASIVAYAM, Arun, 2018. *Wlan 802.11n* [online]. [cit. 2024-10-07]. Dostupné z: https://www.slideshare.net/slideshow/wlan-80211n-85952420/85952420#2.

NEMETH EVI Snyder Garth, Hein Trent R., 2008. *Linux : kompletní příručka administrátora*. Brno: Computer Press. ISBN 978-80-251-2410-9.

NEMETH EVI Snyder Garth, Hein Trent R., 2018. *Unix and Linux system administration handbook*. Boston: Addison-Wesley. ISBN 978-01-342-7755-4.

NETWORKENCYCLOPEDIA.COM, 2024. *Decoding EAP Protocol: A Guide to Extensible Authentication* [online]. [cit. 2024-09-26]. Dostupné z: https://networkencyclopedia.com/decoding-eap-protocol-a-guide-to-extensible-authentication/.

OPENWRT, Project, 2021. *FAQ before installing OpenWrt* [online]. [cit. 2024-07-03]. Dostupné z: https://openwrt.org/docs/guide-user/installation/before.installation#what_is_the_difference_between_the_different_image_formats.

OPENWRT, Project, 2024. *Introduction to 802.1X* [online]. [cit. 2024-11-10]. Dostupné z: https://openwrt.org/docs/guide-user/network/wifi/wireless.security.8021x.

PARSI, Nagababu, 2012. *802.11 Frame types* [online]. [cit. 2024-11-16]. Dostupné z: http://ilovewifi.blogspot.com/2012/07/80211-frame-types.html.

PETERKA, Jiří, 2014. *rámce 802.11* [online]. [cit. 2024-10-08]. Dostupné z: https://www.earchiv.cz/l226/slide.php3?l=16&me=24.

PETRYSCHUK, Steve, 2024. *NetFlow Basics: An Introduction to Monitoring Network Traffic* [online]. [cit. 2024-11-03]. Dostupné z: https://www.auvik.com/franklyit/blog/netflow-basics/.

R., Normunds, 2024. *Traffic Flow* [online]. [cit. 2024-11-23]. Dostupné z: https://help.mikrotik.com/docs/spaces/ROS/pages/21102653/Traffic+Flow.

RAM, 2024. *Monitoring and Visualization Options for OpenWRT* [online]. [cit. 2024-10-15]. Dostupné z: https://nramkumar.org/tech/blog/2024/06/21/monitoring-and-visualization-options-for-openwrt/.

RODRIGUES, Tiago, 2024. Wi-Fi Trends for 2024 and Beyond. *Pipeline Hub*.

SEN, Kaushik, 2024. *Declarative vs. Imperative Models for Configuration Management: Which Is Really Better?* [online]. [cit. 2024-11-23]. Dostupné z: https://www.upguard.com/blog/declarative-vs-imperative-models-for-configuration-management.

SHI, William, 2018. *OpenWISP Blog Post* [online]. [cit. 2024-10-25]. Dostupné z: https://medium.com/@williamchoudhury/openwisp-blog-post-ac6987d6b734.

SINGH, Gurkirat, 2022. *Decrypt WEP Traffic using Bruteforce with Insufficient IVs* [online]. [cit. 2024-09-10]. Dostupné z: https://tbhaxor.com/decrypt-wep-traffic-with-insufficient-ivs/.

SPENCER, Jamie, 2002. *Understanding 802.11 Frame Types* [online]. [cit. 2024-10-08]. Dostupné z: https://wi-fiplanet.com/understanding-802-11-frame-types/.

STANKUŠ, Martin, 2007. *Autentizace, autorizace a accounting v prostředí IEEE 802.1X* [online]. [cit. 2024-09-26]. Dostupné z: http://www.cs.vsb.cz/grygarek/SPS/projekty0607/RADIUS-Stankus.pdf.

TRAMMELL, Brian; CLAISE, Benoît, 2013. *Information Model for IP Flow Information Export (IPFIX)* [online]. 2013-09. [cit. 2024-10-01]. RFC, 7012. Internet Engineering Task Force. Dostupné z: https://www.rfc-editor.org/rfc/rfc7012.

TREMER, Michael, 2014. *About polluting the 5GHz WiFi band* [online]. [cit. 2024-10-08]. Dostupné z: https://www.ipfire.org/blog/about-polluting-the-5ghz-wifi-band.

TWAIN, Kurt, 2024. *DD-WRT vs OpenWrt: The Better Router Firmware in 2024?* [online]. [cit. 2024-11-01]. Dostupné z: https://www.homeowner.com/connectivity/routers/dd-wrt-vs-openwrt.

VANHOEF, Mathy; RONEN, Eyal, 2020. Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd. In: *IEEE Symposium on Security & Privacy (SP)*. IEEE.

WANG, Jie; KISSEL, Zachary A., 2015. Wireless Network Security. In: *Introduction to Network Security: Theory and Practice*. IEEE, s. 211–252. Dostupné z DOI: 10.1002/9781119113102.ch6.

WORTHMAN, Ernest, 2015. *A Primer For The 802.XX Physical Layer* [online]. [cit. 2024-10-07]. Dostupné z: https://semiengineering.com/a-primer-for-the-802-xx-physical-layer/.

DD-WRT.COM, 2024. *Feature matrix* [online]. [cit. 2024-10-23]. Dostupné z: https://wiki.freshtomato.org/doku.php/feature_matrix.

Prohlašuji, že při tvorbě této práce byly nástroje umělé inteligence využity výhradně jako podpora pro stylistické úpravy, opravy pravopisu a návrhy textových formulací. Tyto nástroje byly používány způsobem respektujícím všechny platné předpisy a pravidla, včetně Etického kodexu Univerzity Karlovy. Práce přitom zachovává originalitu a integritu mé vlastní tvorby.

Ukázky zdrojového kódu

Kód 7: Script spouštěný při prvním bootu *(uci-defaults.sh)*

```
#!/bin/sh

root_password="Jednokolka"
ssh_key_rsa="ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILQqqspfQuf2aNIbu6riGMTU3g5ZRcKgnRUqDSHe3/VU
↪  pixx@desktop"


. /lib/functions.sh
. /usr/share/libubox/jshn.sh

delete_wireless() {
    local name="$1"
    uci -q del wireless."$name"
}

delete_network() {
    local name="$1"
    if [[ $name != "loopback" ]]; then
        uci -q del network."$name"
    fi
}

# log potential errors
exec >/tmp/setup.log 2>&1

#odstraneni vychozi konfigurace
config_load wireless
config_foreach delete_wireless wifi-iface
config_load network
config_foreach delete_network interface
config_foreach delete_network device

#ziskani puvodniho wan rozhrani
json_load_file /etc/board.json
json_select "network"
if json_is_a "wan" object; then
        json_select "wan"
else
  json_select "lan"
fi
json_get_var wan_device "device"

#ziskani nastaveni switche
json_select ..
json_select ..
```

```
if json_is_a "switch" object; then
  echo "not DSA"
        json_get_keys keys switch
        json_select "switch"
        for key in $keys; do
                json_select $key
                json_select "roles"
                idx=1
                while json_is_a $idx object
                do
                        json_select $idx
                        json_get_var device "device"
                        if [[ $device == $wan_device ]]; then
                                json_get_var ports "ports"
                                ports=$(echo $ports | sed -r 's/\b([0-9]+)\b/\1t/g')
        switchid=$key
                        fi
                        idx=$(( idx + 1 ))
                        json_select ..
                done
        done
else
  echo "is DSA"
fi

wan_device=$(echo $wan_device | sed -r 's/\..*$|$/\.99/g')

#nastaveni switche
if [ ! -z ${ports+x} ]; then
  config_foreach delete_network switch_vlan

  uci add network switch_vlan
  uci set network.@switch_vlan[-1].device="${switchid}"
  uci set network.@switch_vlan[-1].vlan='99'
  uci set network.@switch_vlan[-1].ports="${ports}"
  uci set network.@switch_vlan[-1].vid='99'
  uci set network.@switch_vlan[-1].description='mgmnt'
fi

#vytvoreni sitoveho rozhrani pro management
uci set network.mgmnt=interface
uci set network.mgmnt.proto='dhcp'
uci set network.mgmnt.device=${wan_device}

#nastaveni root hesla
if [ -n "$root_password" ]; then
  (echo "$root_password"; sleep 1; echo "$root_password") | passwd > /dev/null
fi

#nastaveni ssh
uci -q set dropbear.@dropbear[0].PasswordAuth='off'
uci -q set dropbear.@dropbear[0].RootPasswordAuth='off'
if [ -n "$ssh_key_rsa" ]; then
  echo $ssh_key_rsa >> /etc/dropbear/authorized_keys
fi

uci commit network
uci commit wireless
uci commit dropbear
```

```
echo "All done!"
```