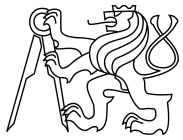
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Detekce a fyzická lokalizace útočníků v sítích WiFi
Student:	Bc. Jakub Samek
Vedoucí:	Mgr. Rudolf Bohumil Blažek, Ph.D.
Studijní program:	Informatika
Studijní obor:	Počítačové systémy a sítě
Katedra:	Katedra počítačových systémů
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

Nastudujte metody a technologie pro lokalizaci zařízení ve WiFi sítích. Zaměřte se na identifikaci nepovolených přístupových bodů a metody jejich blokování. Navrhněte a vytvořte vlastní nástroj pro lokalizaci a blokování vybraných bezdrátových klientských zařízení a přípojných bodů, které porušují síťové politiky. Nástroj koncipujte tak, aby ho bylo možné použít na vhodném běžně dostupném WiFi zařízení na vybrané open source platformě, například OpenWrt.
Součástí řešení by mělo být uživatelské rozhraní na prohlížení získaných lokalizačních údajů. Vytvořené řešení otestujte ve vlastní WiFi síti.

Seznam odborné literatury

Dodá vedoucí práce.

prof. Ing. Róbert Lórencz, CSc.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 15. prosince 2016

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra počítačových systémů

Diplomová práce

Detekce a fyzická lokalizace útočníků v sítích WiFi

Bc. Jakub Samek

Vedoucí práce: Mgr. Rudolf Bohumil Blažek, Ph.D.

30. června 2017

Poděkování

Děkuji vedoucímu diplomové práce Mgr. Rudolfu Bohumilu Blažkovi, Ph.D. za podnětné připomínky a užitečné rady při vypracování práce. Děkuji přítelkyni, rodině a přátelům za podporu během studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 30. června 2017 .....

Odkaz na tuto práci

Samek, Jakub. *Detekce a fyzická lokalizace útočníků v sítích WiFi.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Tato práce se zabývá vývojem a implementací distribuovaného systému pro ochranu WiFi sítí před útoky a porušováním uživatelských politik. Práce obsahuje analýzu existujících metod a dostupných nástrojů pro detekci, fyzickou lokalizaci a blokování vybraných bezdrátových zařízení v rámci ochrany WiFi sítí. Součástí práce je návrh a implementace distribuovaného systému, který pomocí běžných komerčně dostupných WiFi zařízení umožňuje detekovat nepovolená zařízení ve WiFi síti, fyzicky je lokalizovat a blokovat, včetně nepovolených přístupových bodů. Systém byl úspěšně otestován ve vlastní síti.

Klíčová slova WiFi, bezdrátová síť, prevence útoků, lokalizace WiFi zařízení, blokování WiFi zařízení, neautorizovaný přístupový bod, OpenWrt, MQTT

Abstract

This thesis deals with developing an implementation of a distributed system for defending WiFi networks against attacks and usage policy violations. The thesis analyzes existing methods and available tools for detection, physical localization, and blocking of selected wireless devices for the purpose of protecting WiFi networks. The thesis includes design and implementation of

a distributed computer system that employs affordable commercially available WiFi hardware to detect, localize, and block unauthorized WiFi devices, including unauthorized access points. The system was successfully tested in a testbed network.

Keywords WiFi, wireless network, attack prevention, WiFi device localization, WiFi device blocking, unauthorized access point, OpenWrt, MQTT

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Použití bezdrátových WiFi sítí je dnes běžný způsob počítačové komunikace. Na rozdíl od drátových počítačových sítí probíhá přenos dat bezdrátově vzduchem pomocí elektromagnetických vln, čímž odpadá nutnost fyzického připojení zařízení k drátovému rozvodu sítě. Tento způsob připojení k síti přináší uživatelům žádanou volnost a komfort, ale zároveň vznikají bezpečnostní rizika, která se drátových sítí netýkají.

Kromě toho, že samotná komunikace musí být dostatečně zabezpečená kvůli riziku odposlechnutí, které je v rádiových sítích evidentní, je nutné věnovat pozornost vzniku a provozování bezdrátových sítí.

V rozsáhlých prostorech s velkým pohybem lidí, jako jsou školy, kanceláře, hotely nebo nemocnice, nemá poskytoval sítě možnost při porušení síťových politik snadno lokalizovat konkrétní zařízení. Stejně tak nelze osobám fyzicky bránit ve vytváření vlastních bezdrátových přístupových bodů.

Pro útočníky jsou takováto prostředí vhodná k napadání poskytované sítě nebo provozování podvodného přístupového bodu (s motivací odposlouchávat komunikaci) a to bez rizika snadné fyzické lokalizace a odpojení připojeného zařízení, jako by tomu bylo v kontextu drátových sítí.

Provozovatelé sítí by v zájmu ochrany svých klientů měli mít nasazený systém pro detekci, lokalizaci a případné blokování nepovolených zařízení. Tato diplomová práce se zabývá návrhem a implementací takového nástroje na dostupném WiFi zařízení.

První kapitola analyzuje možnosti detekce a lokalizace zařízení v bezdrátových sítích a způsob blokování nepovolených přístupových bodů. V kapitole 2 je pak představen návrh, jak v implementaci řešit dílčí úlohy problému a jak je skloubit do funkčního celku. Kapitola 3 popisuje implementovaný nástroj pro detekci, lokalizaci a blokování zařízení ve WiFi sítích. Kapitola 4 obsahuje výsledky testování implementovaného řešení ve vlastní WiFi síti.

Kapitola 1

Analýza

1.1 WiFi komunikace a její slabiny

WiFi označujeme technologii pro počítačovou síťovou komunikaci založenou na standardu IEEE 802.11 [1]. Tento standard definuje spojovou a fyzickou vrstvu v ISO/OSI modelu[2]. WiFi sítě lze provozovat v několika frekvenčních pásmech. Ta se dále dělí na kanály. Aby spolu mohla dvě zařízení komunikovat, musí používat stejný kanál.

Rozlišujeme dva typy WiFi sítí:

- Ad-hoc sítě
- Infrastrukturní sítě

V této práci nebudeme uvažovat sítě typu Ad-Hoc umožňující přímé propojení zařízení bez přístupového bodu.

Infrastrukturní WiFi sítě obsahují jeden nebo více přístupových bodů (přípojný bod, access point, AP), které tvoří bezdrátovou síť. Klienti (client, station) se mohou bezdrátově připojit do sítě. Komunikace připojených klientů s ostatními zařízeními v síti probíhá skrz přípojný bod. Přípojné body bývají součástí LAN sítě, kterou svým klientům zpřístupňují.

Existují dva způsoby, jak klient může zjistit informace o okolních existujících WiFi sítích:

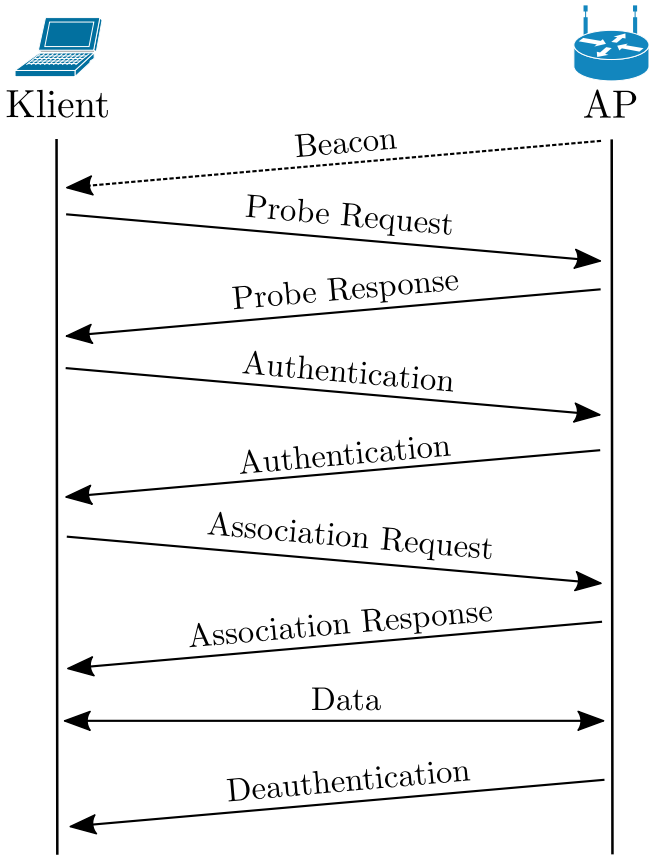
Aktivně: Klient posílá žádost Probe Request na daném kanálu a čeká, zda nepřijde odpověď Probe Response z nějakého přístupového bodu. Uvnitř rámce s odpovědí jsou parametry sítě (jako jsou například SSID název sítě nebo podporované přenosové rychlosti). Na základně zjištěných informací může klient zahájit připojení k síti. Klient posílá žádosti periodicky postupně na všech kanálech, pro zjištění sítí v celém pásmu.

Pasivně: Přístupové body vysílají na svém kanálu Beacon rámec s informacemi o síti, kterou vysílají. Obsah takového rámce je podobný s Probe

Response. Klient pasivně čeká na takové rámce, přičemž opět střídá kanály, aby zachytil informace o sítích v celém pásmu.

Probe Response, Probe Request i Beacon patří do skupiny Management rámců. Jsou obvykle posílány na co nejnižší přenosové rychlosti a nešifrované.

Podle zjištěných informací může klient zahájit proces autentizace a asociace (opět pomocí management rámců). Po úspěšné asociaci může klient začít posílat data a komunikovat s dalšími zařízeními v síti. Celý proces znázorňuje obr. 1.1.



Obrázek 1.1: Diagram WiFi komunikace

Přístupový bod může kdykoliv vyslat Deauthentication rámec klientovi a tím s ním ukončit komunikaci.

1.1.1 802.11 MAC rámec

Komunikace na spojové vrstvě probíhá pomocí rámců s následujícím formátem (podle standardu IEEE 802.11).

Rozlišujeme tři typy rámců:

2B	2B	6B	6B	6B	6B	2B	0-2312B	4B
Frame Control	Dur. /ID	Add. 1	Add. 2	Add. 3	Sequence Control	Add. 4	Frame Body	FCS

Tabulka 1.1: 802.11 MAC rámec

- Kontrolní (Control)
- Manažovací (Management)
- Datové (Data)

Typ a podtyp rámce je definován v poli Frame Control 1.2. Podle typu rámce obsahují adresní pole (Add. 1 - Add. 4) kombinaci následujících adres:

BSS Identifikátor (BSSID) identifikátor množiny zařízení, která spolu mohou komunikovat v bezdrátové síti; v infrastrukturních sítích je to adresa přípojného bodu

Cílová adresa (Destination Address - DA) cílová adresa zařízení, pro kterou je rámec určený

Zdrojová adresa (Source Address - SA) adresa původního zdroje, který rámec vytvořil a vyslal do sítě

Adresa přijímače (Receiver Address - RA) adresa příštího zařízení v bezdrátové síti, které přijme rámec

Adresa vysílače (Transmitter Address - TA) adresa zařízení, které vyslalo rámec do bezdrátové sítě

2b	2b	4b	1b	1b	1b	1b	1b	1b	1b	1b
Prot. ver.	Type	Subt.	To DS	Fr. DS	More Frag.	Ret.	Pwr Mgt.	MD	WEP	Order

Tabulka 1.2: Frame Control

Každý rámec má své sekvenční číslo a číslo části (fragmentu) 1.3, k tomu slouží pole Sequence Control. Maximální hodnota sekvenčního čísla je 4095, další v pořadí je 0.

Informace obsažené v rámcích lze použít k identifikaci komunikujících zařízení a k určení druhu zařízení.

12b	4b
Sequence Number	Fragment Number

Tabulka 1.3: Sequence Control

1.1.2 Slabiny WiFi komunikace

Při diskuzích o bezpečnosti a slabinách bezdrátových sítí se často rozebírají způsoby autentizace a bezpečnost šifrování přenosu dat. Tato práce se však zaměřuje na rizika spojená se samotným provozováním infrastrukturních bezdrátových sítí.

1.1.2.1 Podvodné přístupové body

Vytvořit přístupový bod pro WiFi síť je snadné. Stačí buď nakonfigurovat zakoupený přístupový bod, nebo ho vytvořit z počítače s WiFi kartou a vhodným softwarem. Přístupový bod pak může vysílat Beacon rámce se zvoleným názvem sítě a odpovídat na Probe Request. K nastavenému přístupovému bodu se mohou připojovat klienti.

Operační systémy implementují funkce, které mají uživatelům připojování k bezdrátovým sítím zjednodušit. Jednou z takových funkcí je možnost automatického připojování ke známým sítím (již dříve navštíveným sítím). Dále pak systémy často i po připojení k síti monitorují okolí a to z toho důvodu, aby mohly automaticky přepnout na další přípojný bod, který má ke klientovi blíž (má větší sílu signálu). Takové monitorování probíhá aktivním posíláním Probe Request s vyplněným SSID na aktuální síť, ke které je klient připojen (na takové rámce odpovídají pouze přípojné body, které tuto síť vysílají).

Technicky není problém mít v jednom frekvenčím pásmu a kanále více přístupových bodů nabízejících stejně nazvanou síť, používající stejné šifrování a stejné přístupové údaje. To jasně implikuje riziko: jak uživatel pozná, že se připojil k zamýšlenému přípojnému bodu a ne k podvrženému? Z pohledu útočníka lze dokonce docílit toho, že jeho přístupový bod odpovídá na Probe Request rámce s vyplněným jiným SSID, než jakou vysílá.

Klient pak s ním začne autentizaci a asociaci, ačkoliv se chtěl připojit k síti s jiným názvem [3]. Útočník dále může posláním deautentizačního rámce odpojit klienta (nebo všechny klienty, když použije všesměrovou adresu cíle v odeslaném podvrženém WiFi rámci) od sítě a donutit ho k tomu, aby se musel připojit k síti znovu. V tu chvíli má útočník šanci, že se klient připojí na připravený podvržený přístupový bod.

Žádná z těchto možností nepředstavuje technicky nic složitého a existují hotová řešení, pomocí kterých lze celý proces automatizovat [4, 5].

Motivací útočníků pro vytváření podvodných přístupových bodů je stát se prostředníkem (Man in the middle) v komunikaci oběti (klienta) a mít

možnost odposlouchávat jejich provoz. Pokud to prostředí dovoluje, můžou být podvodné přístupové body připojené do stejné LAN sítě jako regulérní přístupový bod, aby obětem dál fungovaly služby sítě jako při připojení přes správný AP.

U sítí používajících autentizační metody odvozené od EAP je vytvoření kopie (dvojčete, evil twin) přístupového bodu složitější [6]. Slabým článkem pak můžou být klientské přípojné programy, které neověřují správnost certifkátů a samotný proces vydávání certifikátů a manipulace s nimi [7].

Uživatel nemusí používat automatiku operačního systému, ale i když se bude připojovat manuálně pouze na místech, která zná, k sítím, které zná (kromě kontroly názvu sítě klidně i s kontrolou fyzické BSS MAC adresy přístupového bodu – ta ale jde také podvrhnout), nemůže si být jistý, že se nepřipojil na podvržený přístupový bod. Existují výzkumy a experimentální nástroje [8, 9], které klientům nabízejí možnost ověření, zda nejsou připojeni k podvrženému AP, ale nejsem přesvědčený o jejich spolehlivosti.

Z výše uvedených důvodů by bylo dobré, aby provozovatel bezdrátové sítě kontroloval, že se v oblasti, kterou svou sítí pokrývá, nevyskytuje nepovolený přístupový bod, a v případě detekce takového bodu se pokusil o jeho zablokování, a tím chránil své uživatele. Uživatelé by měli být poučeni tak, že bezdrátová síť není bezpečný kanál (i když k přístupu může být zapotřebí znát přístupové údaje a provoz může být šifrovaný) a pro bezpečnou komunikaci používali technologii na některé z dalších síťových vrstev (IPSec, VPN, SSL/TLS, SSH, atd.).

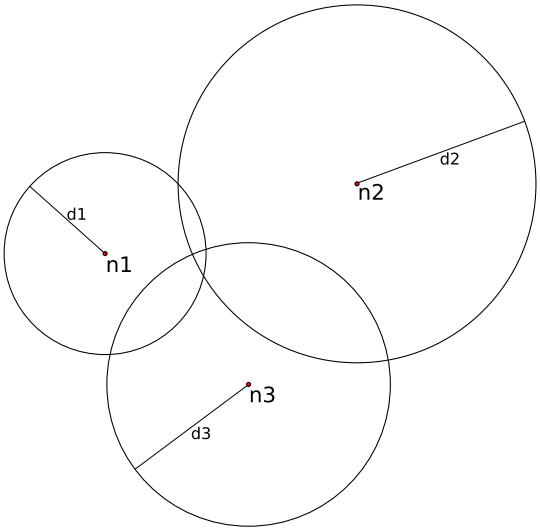
1.1.2.2 Útočník v síti

Předpokládejme, že je v síti nainstalovaný systém prevence průniku (IPS) či detekce průniku (IDS), který identifikuje zařízení (například pomocí IP a MAC adres) porušující síťové politiky. Taková zařízení lze blokovat a chránit tak síť a ostatní zařízení. Pro jejich úplné odstranění je ale nutná jejich fyzická lokalizace a odpojení. To v drátových sítích není problém, zařízení bude připojené na přístupovém portu některého z propojovacích prvků v síti. U bezdrátových sítí je situace složitější, obvykle lze snadno zjistit k jakému přístupovému bodu je útočník asociovaný (a bude tak možné ho odpojit od sítě), ale z této informace lze pro fyzickou lokalizaci vyvodit pouze prostor, který vymezuje pokrytí daného přístupového bodu. Pro přesnější lokalizaci je nutné použít informace z dalších bodů, které zachytily WiFi provoz útočníka.

1.2 Možnosti lokalizace

Většina známých metod je založené na monitorování WiFi provozu z několika bodů, jejichž polohu známe. Z takových měření lze získat vzdálenosti hledaného zařízení od měřících bodů a z nich pak určit polohu hledaného zařízení [10][s.96]. Pro představu uvádím příklad v obr. 1.2.

Mějme tři měřící stanice $n1$, $n2$, $n3$. Z naměřených údajů víme, že vzdálenost k neznámému zařízení je $d1$ z bodu $n1$, $d2$ z $n2$ a $d3$ z $n3$.



Obrázek 1.2: Lokalizace ze tří bodů

Hledané zařízení se nachází v průsečíku naměřených vzdáleností. Pro třírozměrnou lokalizaci je situace podobná, pouze naměřené vzdálenosti neurčují kružnici při jejímž obvodu by zařízení mělo být, ale kouli v prostoru.

Je tedy třeba řešit dva problémy: jak lze z monitorování WiFi provozu získat jednotlivé vzdálenosti k hledanému zařízení a jak z těchto naměřených vzdáleností určit polohu zařízení.

1.2.1 Odhad vzdáleností

Triviální možností jak určit vzdálenost hledaného zařízení od měřícího bodu je pouhá detekce přítomnosti zařízení v okolí, které monitoruje daný bod. To neurčuje přímo kružnici, ale celý kruh, kde může zařízení být (případně body uvnitř koule pro třírozměrný prostor). Při detekci zařízení z více bodů lze opět dělat průnik. Tento jednoduchý postup může být pro některá využití lokalizačního systému dostačující, ale existují i pokročilejší metody [11], jak získat přibližnou vzdálenost hledaného zařízení od sledovacího bodu a tím i zvýšit přesnost následně vypočtené pozice zařízení. Známé metody určení vzdálenosti z naměřených dat nyní rozeberu.

1.2.1.1 Čas zachycení

Někdy také Time of Arrival - ToA nebo Time Of Flight - ToF. Měříme dobu, kterou signálu trvá urazit vzdálenost mezi vysílačem a přijímačem [11][s. 1068].

Protože víme, že rychlost šíření radiových vln je přibližně rychlost světla, je snadné dopočítat hledanou vzdálenost.

$$d = c \cdot t, \tag{1.1}$$

kde

$$\begin{aligned} d &= \text{vzdálenost v metrech} \\ c &= \text{rychlost světla, přibližně } 300m/\mu s \\ t &= \text{čas v mikrovteřinách } (\mu s = 10^{-6}s) \end{aligned}$$

Tato metoda vyžaduje přesnou časovou synchronizaci jak měřících bodů, tak vysílacího zařízení [11] [10][s. 102]. Dále je nutné, aby komunikační zařízení uměla operovat s dostatečnou časovou granularitou. Například uvažujme, že lze měřit čas s přesností na mikrovteřiny. Podle 1.1 by pak takové zařízení umělo rozlišovat pouze vzdálenosti po 300 metrových úsecích.

1.2.1.2 Rozdíly v časech zachycení

Jinou časovou technikou je sledování rozdílů časů zachycených paketů na jednotlivých měřících bodech (jinak také Time Difference of Arrival - TDoA) [11][s. 1069]. Zde není potřeba znát čas začátku vysílání a odpadá nutnost časové synchronizace vysílacího (hledaného) zařízení s měřícími body (ty ale mezi sebou musí být synchronizované). Metoda pro výpočet používá relativní rozdíly v naměřených časech zachycení signálu na přijímačích. Z těchto rozdílů lze zkonstruovat hyperboly určující vzdálenosti hledaného zařízení od měřících bodů. Jejich průsečík opět označuje přibližnou polohu zařízení.

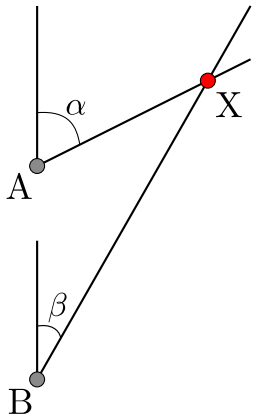
1.2.1.3 Úhel zachycení

Další metoda [11][s. 1069] (Angle of Arrival - AoA, Direction of Arrival - DoA) je založená na měření úhlu zachycení signálu vhledem k určenému referenčnímu směru. Jak zobrazuje diagram 1.3, průsečík přímek vycházejících z měřících bodů, svírajících s referenčním úhlem naměřené úhly, označuje polohu hledaného zařízení, kterou lze snadno dopočítat.

Nevýhodami této metody je nutnost použití speciálních měřících zařízení. Pro měření úhlu zachycení lze použít anténní pole z různě orientovaných antén nebo mít anténu umožňující mechanické natočení na nejsilnější signál a tak získat úhel zachycení.

1.2.1.4 Síla signálu

Jiným přístupem je pozorování síly signálu. Pokud známe podrobnosti o radiové komunikaci zařízení a komunikace probíhá ve volném prostoru, lze vzdálenost mezi nimi přesně spočítat podle Friisovy rovnice přenosu [12].



Obrázek 1.3: Lokalizace pomocí úhlů zachycených signálů

$$\begin{aligned} P_r &= P_t + G_t + G_r + FSPL \\ FSPL &= 20log_{10}\left(\frac{\lambda}{4\pi d}\right) \end{aligned}$$

kde

$$\begin{aligned} P_r, P_t &= \text{síla signálu na přijímači (respektive vysílači) v } dBm \\ G_r, G_t &= \text{zisk antény na přijímači (respektive vysílači) v } dB \\ FSPL &= \text{Free Space Path Loss - rovnice útlumu volného prostoru} \\ \lambda &= \text{velikost vlny v metrech} \\ d &= \text{vzdálenost mezi přijímačem a vysílačem v metrech} \end{aligned}$$

Pokud neznáme podrobnosti o zařízeních (použitý vysílací výkon a zisk antén), lze použít jiný model, který pracuje s naměřeným referenčním útlumem ve zvolené vzdálenosti [13][s. 102],[14] [10][s. 59]. Tento model je použitelný pro vnitřní prostředí.

$$P_r = P_{r_0} + 10\gamma log_{10}\frac{d}{d_0} \tag{1.2}$$

kde

$$\begin{aligned} P_r &= \text{síla signálu na přijímači v} dBm \\ P_{r_0} &= \text{síla signálu v referenční vzdálenosti v } dBm \\ \gamma &= \text{exponent útlumu, podle prostředí} \\ d &= \text{hledaná vzdálenost mezi přijímačem a vysílačem v metrech} \end{aligned}$$

d_0 = vzdálenost, ve které byl měřený referenční útlum v metrech

Model lze rozšířit [13][s. 104], [10][s. 59] na tvar:

$$P_r = P_{r_0} + 10\gamma log_{10}\frac{d}{d_0} + X_\sigma \tag{1.3}$$

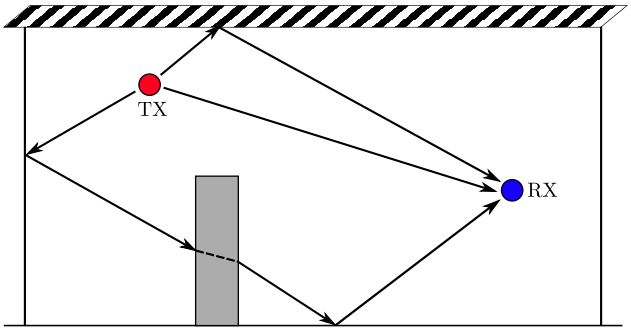
kde X_σ je náhodná proměnná s normálním rozdělením $N(0, \sigma^2)$ a popisuje náhodný efekt stínění.

V literatuře [13][s. 104] lze nalézt následující exponenty útlumu pro různá prostředí:

Prostředí	γ
volný prostor	2
mezi domy ve městě	2.7 - 3.5
mezi domy ve městě s vyšším zarušením	3 - 5
budova s přímou viditelností	1.6 - 1.8
budova s překážkami	4 - 6
industriální budova s překážkami	2 - 3

Tabulka 1.4: Přehled exponentů útlumu pro různá prostředí

Všechny popsané metody jsou citlivé na vícecestné šíření odeslaných signálů [10][s. 108]. Klientská zařízení používají všesměrové antény a svoje signály vysílají do všech směrů. Navíc se signály můžou odrážet od překážek nebo jimi procházet, ale se změněnými charakteristikami signálu. To má za důsledek, že v prostředí, kde existují překážky (stěny, nábytek, pohybující se lidé, apod) se signál může šířit několika cestami a přitom mění své vlastnosti. Jak je vidět na obr. 1.4, odražené signály zcela určitě urazí delší vzdálenost, což bude trvat delší čas, a také dorazí pod jiným úhlem, než ve kterém se vysílací zařízení skutečně nachází.



Obrázek 1.4: Více cest signálu mezi vysílačem TX a přijímačem RX

1.2.2 Měřící zařízení

Pro aplikaci všech uvedených metod je třeba naměřit vstupní data monitorováním probíhajícího provozu ve WiFi síti.

1.2.2.1 Monitorovací režim na WiFi kartě

Některé běžně dostupné WiFi karty lze s vhodných ovladačem provozovat v takzvaném monitorovacím režimu [15]. Ten umožňuje [15] sledování veškerého WiFi provozu na zvoleném kanálu bez nutnosti asociace do sítě. U vhodných WiFi adaptérů lze kromě informací, které nabízí 802.11 rámce, zjistit další podrobnosti o radiovém provozu. Například sílu přijatého signálu, přenosovou rychlost nebo identifikaci antény, která daný signál zachytila. Tyto a další informace jsou součástí takzvaných Radiotap hlaviček.

Monitorovací režim nijak nemění další vlastnosti WiFi karty. Stále může v jednu chvíli pracovat pouze na jednom kanálu a nelze tedy pomocí jedné WiFi karty sledovat veškerý probíhající okolní provoz v celém frekvenčním pásmu.

Příkladem vhodných karet podporujících monitorovací režim i Radiotap hlavičky jsou TP-Link TP-WN722N nebo Alfa AWUS036NHA. Probíhající okolní provoz včetně Radiotap hlaviček lze sledovat například v programech Wireshark, tcpdump nebo airodump-ng.

1.2.2.2 Spektrální analyzátor

Využití spektrálního analyzátoru ve WiFi sítích je užitečné spíše při jejich návrhu, případně při řešení problémů spojených s fyzickou komunikací mezi zařízeními v síti. S jeho pomocí lze přehledně vidět obsazenost jednotlivých frekvencí v daném pásmu, detekovat špatně pokrytá místa či nalézt zdroje rušení v daném pásmu (například mikrovlnné trouby).

Pro lokalizaci zařízení by bylo možné využít spektrálním analyzátorem naměřené síly signálů. Jelikož ve WiFi sítích více zařízení sdílí jednu frekvenci (pro vícenásobný přístup se používá CSMA-CA), je nutné naměřené hodnoty přiřadit ke konkrétním zařízením, identifikovatelných například pomocí adresy na spojové vrstvě ISO/OSI modelu. Nepodařilo se mi najít analyzátor, který by to uměl. Navíc se spektrální analyzátory často dodávají s proprietárním softwarem, což dále ztěžuje práci se zpracováním naměřených dat.

1.2.2.3 Softwarově definované rádio

Nevýhodou hotových radiových komunikačních zařízení je, že je třeba si vystačit s informacemi o komunikaci a možnostmi ovládání, které definoval výrobce a zpřístupnil je pomocí nějakého rozhraní. Řešením takového omezení je koncept softwarově definovaných rádií (SDR) [16].

Některé rádiové komponenty (demodulátory, filtry, zesilovače, apod.) nejsou v SDR realizovány hardwarově, ale pomocí nastavitelných číslicových obvodů

[16]. Na přijímači jsou signály co nejblíže anténě digitalizovány pomocí AD převodníku a pak už je lze zpracovávat softwarově. Na vysílači je postup analogický – signál se softwarově připraví, převede pomocí DA převodníku a přivede na anténu. Toto nahrazení umožňuje změnou softwaru používat různá frekvenční pásma, modulace, výkonové parametry a komunikační protokoly. To má za důsledek, že pomocí SDR lze simulovat [17, 18] nejrůznější komunikační zařízení a zkoumat příslušné protokoly (LTE, WiFi, GSM, GPS, apod.) nebo vyvíjet nové.

Použití SDR lze uvážit u případné konstrukce vlastních WiFi zařízení (například s anténním polem pro metodu se zachycováním úhlu) či pro studium WiFi radiové komunikace na nižší úrovni, než dovolují ovladače adaptérů.

1.2.2.4 Aplikace a GPS

Pro lokalizaci připojených zařízení ve WiFi síti by šlo použít [19] lokalizační modul přímo v zařízení. Většina mobilních zařízení má vestavěný GPS čip a umím si představit, že poskytovatel WiFi sítě umožní její plné používání, pouze tehdy uživatel povolí posílání polohy.

Tento případ uvádím pouze pro úplnost. Osobně si ale myslím, že se v budoucnu může něco takového ve veřejných sítích používat. Pro účely naší práce - detekce a lokalizace WiFi zařízení bez přístupu na tato zařízení - je tento způsob nevhodný.

1.3 Výpočet polohy

Dostáváme se k druhé části problému. Tím je najít způsob, jak z naměřených vzdáleností hledaného zařízení od jednotlivých měřících bodů určit či odhadnout jeho polohu v prostoru. Obecně se takové metody označují jako trilaterace či multilaterace (pro více naměřených hodnot) [20].

Mějme n měřících zařízeních se známými polohami o souřadnicích $[x_n, y_n, z_n]$. Dále pak uvažujme, že m měřících zařízení ($m \leq n$) monitorovalo zařízení X, a známe tedy vzdálenosti $d_0, d_1, \ldots, d_{m-1}$ mezi zařízením X a příslušným měřícím bodem. Chceme zjistit souřadnice $[x,\ y,\ z]$ hledaného zařízení X. Pak platí soustava rovnic:

$$\begin{aligned} d_0^2 &= (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 \\ d_1^2 &= (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 \\ &\ldots \\ d_{m-1}^2 &= (x - x_{m-1})^2 + (y - y_{m-1})^2 + (z - z_{m-1})^2 \end{aligned}$$

Takovou soustavu lze vyřešit a pro jednoznačnou polohu v prostoru stačí naměřené vzdálenosti ze čtyř bodů. Tento postup výpočtu polohy však vyža-

duje přesná měření, jinak se může stát, že soustava vůbec nebude mít řešení. Geometricky si to lze představit tak, že pomyslné koule o poloměrech naměřených vzdáleností se neprotnou v jednom místě (nebo se například neprotnou vůbec).

Jak již bylo zmíněno v předchozích sekcích, nelze předpokládat přesná měření vzdáleností. Pro účely této práce není nutné určit polohu absolutně přesně, ale odhadnout ji co nejlépe na základě naměřených dat. Jednou z možností, jak tento problém řešit, jsou iterativní optimalizační algoritmy [21], jejichž funkcí je minimalizace chyby (vzdálenosti) od naměřených dat. Mějme opět m naměřených vzdáleností z bodů M_i k hledanému zařízení X. Chybu e určené polohy lze spočítat jako:

$$e = \frac{\sum_{i=0}^{m-1} \left(d_i - dist\left(X, M_i\right)\right)^2}{m} \tag{1.4}$$

kde $dist(A, B)$ označuje vzdálenost mezi body A a B. Metody minimalizující tuto chybu lze označit [20] jako metody řešení nelineárních nejmenších čtverců. V matematických softwarech bývá pro řešení toho problému implementovaný Levenberg–Marquardtův algoritmus [21].

1.3.0.1 Lokalizace pomocí otisků prostředí

Jiným způsobem jak určit polohu zařízení v síti na základně naměřených informací ze síťového provozu je používat databázi otisků pro dané prostředí a tu pak využívat pro odhadnutí polohy. Tato metoda má dvě fáze [11][s. 1070]:

Kalibrační fáze V kalibrační fázi je třeba naměřit hodnoty zvolené metriky (například síly signálu) na místech, kde bude lokalizační systém fungovat, a k tomuto naměřenému vektoru hodnot si zaznamenat polohu měření. Tato dvojice – vektor hodnot a poloha v prostoru – označují otisk. V praxi to znamená obejít prostor s bezdrátovým kalibračním zařízením, jehož provoz (vektor naměřených hodnot) systém zaznamenává společně s polohou, kde se právě nachází. Takto vytvořená databáze otisků slouží pro určení polohy neznámých zařízení v operační fázi.

Operační fáze V operační fázi systém při detekci zařízení vyhledá naměřený vektor hodnot z měřících bodů v databázi a vrátí přiřazenou pozici v prostoru. Tato metoda tedy nepoužívá výpočet založený na postupu, kdy se nejprve zjistí vzdálenosti od jednotlivých měřících bodů a z nich se spočte výsledná pozice, ale určuje polohu přímo z vektoru naměřených dat. Triviální vyhledání v databázi ale není v praxi použitelné. Mapa hodnot z kalibrační fáze nebude pokrývat všechny kombinace naměřených dat a jak už víme, měření nejsou přesná a i menší změny v prostředí (otevřené dveře, přesun nábytku) ovlivňují jednotlivá měření a tím celý vektor hodnot. Proto se pro určení pozice na základě předem

naměřených hodnot používají složitější metody, od jednoduché podobnosti vzorků až po strojové učení.

Hlavní nevýhodou této metody je nutnost naměření otisků po celém prostoru a citlivost na změny v prostředí.

1.4 Blokování přístupových bodů

Jak již bylo zmíněno, přístupový bod může kdykoliv ukončit spojení s asociovaným klientem zasláním deautentizačního rámce na adresu klienta. Protože se deautentizační rámce posílají nešifrované a zatím se nepoužívá žádný mechanismus k ověření jejich pravosti (doplněk 802.11w popisuje možné řešení [22]), může kdokoliv klientům bezdrátové sítě poslat deautentetizační rámec s podvrženou zdrojovou adresou na adresu skutečného AP a tím je odpojit od sítě. Rámce není nutné posílat jednotlivým klientům, lze odpojit všechny najednou využitím cílové všesměrové adresy.

Na tento postup lze nahlížet jako na útok deautentizační útok odepření přístupu (Denial of Service - DoS) k přístupovému bodu, nebo naopak jako na blokování nepovoleného přístupového bodu s úmyslem ochránit klienty sítě tím, že od něj budou neustále odpojováni. Příkladem hotového nástroje pro posílaní podvržených deautentizačních rámců klientům zvolené WiFi síti je aireplay-ng [23].

Pří blokování přístupových bodů na reálných sítích je nutné seznámit se legislativou v příslušné zemi. Existuje případ hotelové sítě, která ve svých hotelech blokovala přístupové sítě svých hostů. Toto jednání bylo shledáno jako protiprávní, provozovatel dostal pokutu [24].

1.5 Existující řešení

Instalací a správou rozsáhlých bezdrátových WiFi sítí se zabývá několik firem, jejichž řešení zvládnou konfigurovat více než stovky přípojných bodů a obsloužit více než tisíce připojených klientů najednou.

Požadavky na takové systémy jsou hlavně na síťovou funkcionalitu (vysílání několika sítí najednou, různé VLAN, konfigurace kanálů a vysílacích výkonů, migrace klientů mezi AP při pohybu, možnosti monitoringu) a na správu přístupu uživatelů k síti (AAA - authentication, authorization and accounting; RADIUS, TACACS). Systémy dále bývají rozšiřovány například o bezpečnostní (WIDS/WIPS) a lokalizační funkce.

Architektura těchto systémů je podobná. V síti nebo v cloudu existuje centrální prvek (kontroler) a k němu jsou připojené přístupné body. Konfigurace se provádí na centrálním prvku a ten pak nastavuje přístupové body. Kontroler má nad přístupovými body plnou kontrolu a může je tak použít i pro monitorování provozu v celé síti.

1.5.1 Cisco

Podle mého názoru je Cisco Unified Wireless Network [25] nejpropracovanější systém pro rozsáhlé WiFi sítě. Centrálním bodem je WLC (Wireless LAN Controller) a k němu jsou připojené LAP (lightweight access point). Z našeho pohledu je nejzajímavější volitelnou komponentou tohoto systému Cisco Mobility Services Engine (MSE), který dále může být propojen s WIPS a CMX (Connected Mobile Experiences). Mezi hlavní funkce tohoto celku patří:

- detekce a lokalizace neznámých přístupových bodů, jejich klasifikace (Malicious, Friendly);
- blokování vybraných přístupových bodů a zařízení v síti;
- detekce a řešení útoků na WiFi síť podle nastavené politiky;
- detekce a lokalizace interferencí a DoS útoků na síť z frekvencí mimo 802.11 (Bluetooth, mikrovlnné trouby, apod.);
- zaznamenávání počtu zařízení v dosahu, zaznamenávání doby jejich návštěvy a jejich historie [1].

Cisco pro výpočet polohy používá metodu lokalizace pomocí otisků [27]. Přesnost je v řádu jednotek metrů, velmi záleží na konkrétním prostředí, počtu měřících AP a jejich rozprostření po lokalitě. Přesnost systému lze zvýšit přidáním dodatečných zařízení s anténním polem. To umožňuje rozšířit způsob výpočtu polohy o informace o úhlu přijatého signálu (AoA) [28].

Blokování vybraných detekovaných přístupových bodů a klientů probíhá zasíláním deautentizačních rámců [29].

Veškerá data a upozornění lze získávat i pomocí API, není tedy problém s využitím dat v dalších systémech. Nevýhodou řešení od Cisco je cena, síťová řešení této společnosti jsou velmi drahá. Nabídky a ceny se řeší se zákazníky individuálně podle jejich potřeb a licenční politika je složitá. Velmi zhruba si lze představit [30], že systém s MSE a wIPS pro desítky přístupových bodů by stál stovky tisíc korun.

1.5.2 Aruba, Ruckus, Meraki a další firmy

Existuje několik dalších firem, které se zabývají řešením pro provozování rozsáhlých WiFi sítí. Některé z nich umožňují získat polohu zařízení nebo detekovat nepovolené přístupové body. Funkční nabídka je velmi podobná tomu, co nabízí Cisco.

[1] To už lze považovat za shromažďování osobních údajů a provozovatel by si k tomu měl zajistit souhlas. Výrobci některých operačních systémů z obavy o soukromí svých uživatelů implementovali náhodné změny MAC adresy WiFi karty pro hledání okolních sítích. Ukazuje se, že ani to nemusí zabránit jednoznačné identifikaci zařízení [26].

1.5.3 OpenWISP

Za jediný obstojný zdarma dostupný systém na správu rozsáhlejších WiFi sítí považuji OpenWISP [31]. Jeho prostřednictvím lze centrálně spravovat AP a směrovače s operačními systémy OpenWrt nebo LEDE. Projekt je zaměřený na centrální konfiguraci sítě a zatím zde není žádná implementace detekce podvodných přístupových bodů nebo lokalizační služby. Projekt má otevřený zdrojový kód s licencí GPL 3.0 a je připravený na rozšíření o další funkce.

1.5.4 Výzkum

Při hledání podkladů pro tuto práci jsem narazil na desítky výzkumných projektů a studií řešící lokalizaci zařízení v bezdrátových sítích. I když většina z nich nevedla k použitelným nástrojům a jejich motivace nebo použité metody nejsou aplikovatelné na náš případ, pro zájemce jsem vybral ty nejzajímavější: [32, 33, 34, 35, 36].

Kapitola 2

Návrh

Cílem práce má být návrh a implementace nástroje pro lokalizaci zařízení ve WiFi sítích a jejich případné blokování. V této kapitole navrhnu strukturu takového nástroje a proberu možnosti, jak by jeho jednotlivé funkční části mohly být implementovány.

Z analýzy vyplývá, že pro lokalizaci zařízení ve WiFi sítích bude nutné nějakým způsobem monitorovat provoz v síti, a to z několika bodů. Podle naměřených dat lze identifikovat hledaná zařízení a spočítat jejich polohu, kterou by pak systém měl dále zpřístupnit pomocí nějakého rozhraní. Z takového postupu lze systém rozdělit na hlavní funkční celky:

- monitorování provozu
- zpracování dat a výpočet polohy
- prezentace výsledků a ovládání systému

Tyto celky nyní podrobněji rozeberu a dále rozdělím s cílem definovat funkční požadavky pro následnou implementaci navrženého systému.

2.1 Monitorování provozu

2.1.1 Hardware

Ze zadání je určené, že systém má být navržen tak, aby jej bylo možné provozovat na běžně dostupném WiFi zařízení. Jak už vyplývá i z analýzy, nejvhodnější bude použít WiFi kartu s podporou monitorovacího režimu. Monitorovací režim podle mého hledání podporují téměř všechny aktuálně používané čipsety v dostupných WiFi kartách, ale je nutné ověřit podporu pro provoz tohoto režimu v ovladači pro kartu v operačním systému, kde chceme kartu použít. Jednotlivé dostupné adaptéry se liší v použitém čipsetu, podporovaných frekvencích, přípojném rozhraní, možnosti připojení externí antény a také podporovanými operačními systémy.

Z dalších zařízení jsem uvažoval pouze softwarově definované rádio, ale jeho použití pro tuto práci má několik problémů. Komerční zařízení jsou drahá (například HackRF či Ettus stojí tisíce Kč), výroba vlastních SDR je náročná. Kromě přípravy takového zařízení je potřeba použít nebo implementovat vhodný software pro jeho ovládání a možnost WiFi komunikace. Vhodný hotový software pro SDR podporovaný sadou nástrojů GNU Radio [37] by mohl být Wime [38], ale prozkoumání takového řešení je nad rámec této práce.

Nenašel jsem žádné dostupné zařízení, které by bylo použitelné pro metodu zachycení úhlu.

2.1.2 Software

Jak již bylo zmíněno, WiFi adaptér je třeba někam připojit a pak ho pomocí ovladače ovládat a používat ho pro monitorování provozu. Všechny v současné době používané operační systémy mají podporu pro bezdrátové sítě a jsou v nich dostupné nástroje či rozhraní pro ovládání bezdrátových adaptérů. Odlišnosti práce a možnost použití WiFi adaptérů pro monitorování provozu v jednotlivých operačních systémech stručně popíšu.

Pro měření síly signálu u zachycených rámců je nutné, aby adaptér, ovladač i použitý software pro zachytávání WiFi provozu uměl zprostředkovat informace v Radiotap hlavičkách.

2.1.2.1 Linux

Ovladače pro většinu čipsetů adaptérů jsou dostupné přímo v jádře, kompatibility konkrétních adaptérů a podporu monitorovacího režimu je třeba zkontrolovat [39, 18]. Pro ovládání síťového adaptéru existuje mnoho programů, mezi základní programy po příkazové rozhraní patří iw, iwconfig a iwlist.

Pro zachycení provozu lze použít například programy wireshark, tshark, tcpdump nebo airodump. Všechny tyto programy používají knihovnu libpcap. Vzhledem ke koncepci jádra operačního systému a dostupnosti knihoven pro práci se síťovými zařízeními je další možností vyvinout pro monitorování provozu vlastní program.

Na trhu existuje velké množství WiFi karet s potřebnou podporou a jejich cena se pohybuje v řádu stokorun. Za vhodné adaptéry pro použití na monitorování sítě považuji ty založené na čipsetech Atheros AR9271, Ralink RT3070 a Ralink RT3572, jako jsou například USB adaptéry TP-LINK TL-WN722N, Alfa AWUS036NH nebo Alfa AWUS036NEH.

2.1.2.2 OpenWrt, LEDE, DD-WRT

Existují speciální distribuce Linuxu určené pro provoz na síťových zařízeních, jako jsou WiFi přístupové body a směrovače. Taková zařízení jsou většinou založená na omezenější výpočetní platformě (převážně MIPS) a mají limitovanou operační i perzistentní paměť (jednotky MB). Zmíněné operační systémy jsou

upravené na provoz s takto omezenými prostředky a lze je do podporovaných zařízení nahrát místo operačního systému (firmwaru) výrobce, čímž rozšíříme funkce těchto zařízení. Do takto pozměněných zařízení lze instalovat další software prostřednictvím balíčkovacího systému a používat většinu standardních síťových nástrojů dostupných na plnohodnotných distribucích Linuxu.

Z pohledu této práce by bylo vhodné, aby jako měřící body mohla být použita právě taková síťová zařízení. Například pokud už někdo má realizovanou WiFi síť pomocí přístupových bodů s OpenWrt nebo podobným systémem, mohl by nástroj na detekci a lokalizaci nepovolených zařízení nasadit bez větších změn ve stávající infrastruktuře. Ani případná instalace nových zařízení speciálně určených pro provoz navrhovaného nástroje nemusí představovat vysokou finanční investici. Existují levné (stovky Kč) domácí WiFi směrovače s vyhovujícím adaptérem a podporou OpenWrt nebo podobného systému.

2.1.2.3 FreeBSD, NetBSD, OpenBSD

Situace je podobná jako u Linuxu. Pokud je adaptér v systému podporovaný a ovladač podporuje jeho provoz v monitorovacím režimu, lze opět využít nástroje pro zachytávání provozu. Počet podporovaných čipsetů a adaptérů je v těchto systémech nižší než u Linuxu, ale na trhu lze najít vhodná zařízení.

2.1.2.4 macOS, OS X

V operačních systémech Apple macOS i starších OS X lze s výrobcem dodávanými WiFi adaptéry používat monitorovací režim. Přepnutí do monitorovacího režimu lze provést vestavěným programem airport[2]. Jinou možností je využít program pro zachycení síťového provozu, který si adaptér nastaví sám, například Wireshark.

U jiných adaptérů je nutné, aby monitorovací režim implementoval jeho výrobce v ovladači. Žádný takový běžně dostupný jsem nenašel.

2.1.2.5 Windows

Novější operační systémy Microsoft Windows (Windows Vista a novější) mají v rámci rozhraní Network Driver Interface Specification (NDIS) API podporu pro monitorovací režim, ale je nutné, aby tento režim podporoval i síťový adaptér a jeho ovladač. Našel jsem jediný takový adaptér - Riverbed AirPcap, v ceně několika tisíc korun. Alternativně lze použít placený WiFI ovladač firmy Acrylic, který u kompatibilních adaptérů zpřístupní monitorovací režim.

[2]dostupném na cestě /System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airport

2.2 Výpočet polohy

Nyní předpokládejme, že umíme monitorovat WiFi provoz na několika měřících bodech včetně Radiotap hlaviček. Ze zachycených rámců jsou pro detekci zařízení a výpočet jejich polohy vhodná k použití tato data:

- síla signálu zachyceného rámce
- použitý kanál
- sekvenční číslo rámce a číslo fragmentu
- identifikátor antény, která rámec zachytila
- typ 802.11 rámce
- podtyp 802.11 rámce
- dodatečné informace k danému typ a podtypu rámce (například SSID u Beacon)
- BSS adresa
- zdrojová adresa
- cílová adresa

V kapitole analýza jsem zmínil několik metod, jak z naměřených údajů lze zjistit vzdálenost hledaného zařízení od měřícího bodu. Z těchto vzdáleností pak lze spočítat polohu zařízení. Nyní zhodnotím použitelnost jednotlivých metod pro navrhovaný systém.

2.2.1 Vzdálenosti od měřících bodů

2.2.1.1 Čas zachycení

Předpokladem u této metody je časová synchronizace měřících bodů i hledaného zařízení. Vzdálenost se určuje pomocí doby pohybu dat ve vzduchu mezi vysílačem a přijímačem, je tedy nutné znát začátek vysílání na vysílači a čas zachycení na přijímači. Systém má detekovat a lokalizovat zařízení, nad kterými nemá provozovatel kontrolu a bez toho nelze zjistit začátek vysílání ani zajistit časovou synchronizaci s měřícími body.

2.2.1.2 Rozdíly v časech zachycení

Předpoklady pro použití této metody:

- časová synchronizace měřících bodů
- přesné časy zachycení jednotlivých rámců

Pro časovou synchronizaci mezi počítači existují nástroje a protokoly, mezi nejpoužívanější patří NTP a PTP. Pro tuto chvíli předpokládejme, že pomocí těchto protokolů lze po ethernetové síti zajistit časovou synchronizaci mezi měřícími body v řádu nanosekund.

Pro čas zachycení nelze použít časové razítko TSFT z Radiotap hlaviček, které slouží pro synchronizaci klientů v bezdrátové síti. Jako čas zachycení lze použít informaci o zachycení rámce ze softwaru monitorujícího síťový provoz. Testované programy využívající libpcap udávají časy zachycení s rozlišením v řádu mikrosekund. To má za důsledek, že nejmenší nenulový rozdíl mezi dvěma časy zachycení je 1 μs což odpovídá vzdálenosti 300 metrů. Takový nejmenší krok je pro lokalizaci ve WiFi sítích moc velký a tuhle metodu nelze použít.

2.2.1.3 Úhel zachycení

Metodu nelze pro navrhovaný systém použít z důvodu nedostupnosti vhodného adaptéru pro měření.

2.2.1.4 Síla signálu

U převádění síly zachyceného signálu na vzdálenost stačí identifikovat jednotlivá zařízení podle zdrojové MAC adresy. Pokud zařízení z jednoho místa vysílá například Beacon, Probe Response a data, tak na přijímači zachytíme tyto signály se stejnou silou, a ta stačí k přepočtu na vzdálenost mezi zařízením a měřícím bodem.

Stále je zde problém s vícecestnou propagací signálu, navíc se síla signálu při propagaci skrz nějaký objekt snižuje. V implementaci by bylo dobré zvážit, podobně jako jsem zmiňoval u časové metody, zda by nebylo vhodné brát v úvahu pouze nejsilnější přijatý signál v dané chvíli. Opět by síla takového signálu měla být nejblíže tomu, kdyby se měřilo v otevřeném prostoru s přímou viditelností.

Z analýzy víme, že do přepočtu síly signálu na vzdálenost vstupuje proměnná γ určující exponent útlumu pro konkrétní prostředí. V implementaci je třeba nechat tuto proměnnou volitelnou a lze předpokládat, že ji bude třeba experimentálně upravovat pro dobré výsledky lokalizace pro konkrétní místo použití.

Z existujících řešení jsem vypozoroval, že je vhodné implementovat hranici, od které vůbec bude naměřených signál možné aplikovat pro další použití.

Cisco například doporučuje takovou hranici na -75 dBm (signály s naměřenou horší sílou, než je -75 dBm, se tedy nebudou pro lokalizaci používat) [40]. V navrhovaném systému bude hranice jako nastavitelný parametr.

2.2.2 Výpočet polohy ze vzdáleností

Z analýzy a z přehledu výzkumů v sekci existujících řešení víme, že způsobů, jak z naměřených vzdáleností získat polohu, je několik. Program pro výpočet polohy by měl být připraven, aby bylo možné způsob výpočtu polohy zaměnit za jiný bez modifikace zbytku systému. Jako výchozí metodu pro implementaci zvolím iterativní řešení založené na minimalizaci chyby a další metody přidám, pokud se tato metoda ukáže pro navrhovaný systém nevhodná.

V tuto chvíli neuvažuji, že by nástroj uměl pro výpočet využít předem naměřenou mapu otisků signálů pro místa v lokalitě nasazení.

2.3 Detekce neznámých přístupových bodů a jejich blokování

Základní funkcí navrhovaného nástroje má být detekce neznámých (a možná podvodných) přístupových bodů a jejich lokalizace. Z pozorování WiFi provozu lze poznat přístupové body dvěma způsoby: BSS adresa a adresa vysílače je stejná, nebo jde o rámce, které posílají jen AP, jako je Beacon, Probe Response, Deauthentication a podobně. Otázkou zůstává, jak poznat podvodné přístupové body.

Podezřelé přístupové body jsou všechny, které nejsou známé. Za známé přístupové body považuji ty, které vysílají v daném místě pravou WiFi síť, a ty, které byly označené jako přístupové body neporušující síťové politiky v daném místě. Příkladem takového bodu můžou být přístupové body pro síť sousedního domu, které jsou ale tak blízko, že je náš systém detekuje. Podezřelé body bude možné blokovat.

Pro tuto práci navrhuji, aby systém uměl pracovat se seznamem MAC adres známých přístupových bodů zařízení [3] a všechny ostatní systém prezentoval jako podezřelé a umožnil jejich blokování.

Blokování vybraného bezdrátového bodu bude probíhat posíláním deautentizačních rámců s podvrženou adresou zdroje. Klienti připojení k podvodnému bezdrátovému bodu budou odpojeni a další pokusy o připojení budou neúspěšné. Tím dojde k ochránění klientů legitimní sítě a blokování přístupového bodu.

[3] Z předchozích poznatků je zřejmé, že podvodný přístupový bod může podvrhnout i MAC adresu tak, aby odpovídala adrese legitimního AP, a tedy tento způsob detekce neznámých zařízení by takový bod neodhalil. Řešením může být automatické občasné vypínání jednotlivých přístupových bodů a jejich vyřazení ze seznamu známých zařízení. To vyžaduje spolupráci s kontrolerem ovládajícím AP a tento proces nebude součástí práce.

2.4 Zpracování dat

Způsob výpočtu vzdáleností z monitorovaného WiFi provozu a následné vypočítání polohy je určený, nyní je třeba navrhnout, jak tyto procesy spojit do funkčního celku. Pro výpočet polohy je nutné mít k dispozici vzdálenosti detekovaných zařízení na jednom místě, ze kterého se pro tuto chvíli stává centrální bod systému. Měřící zařízení mohou naměřený provoz ukládat na své lokální úložiště a centrálnímu bodu tato data zpřístupňovat nebo lze naměřená data rovnou předávat centrálnímu bodu ke zpracování. Vzhledem k omezení prostředků navrhovaných měřících bodů je vhodnější data o síťovém provozu předávat rovnou centrálnímu bodu, u kterého není třeba uvažovat omezení prostředků. Návrh předpokládá, že měřící body jsou připojené k počítačové síti a lze jí využít pro předávání naměřených dat centrálnímu bodu.

Lze si představit situace, kdy navrhovaný systém nebude permanentní součástí sítě v místě, kde má systém fungovat, ale půjde pouze o jeho dočasné nebo krátkodobé nasazení za účelem vyřešení nějakého problému - například při občasné detekci nepovolených bodů nebo nutnosti fyzické lokalizace útočníka detekovaného jiným systémem. V takovém případě by šlo připravit speciální sadu měřících zařízení s dvěma WiFi adaptéry (jeden pro měření a jedno pro výměnu dat) a měřícího bodu, které by si v dané lokalitě vytvořili vlastní bezdrátovou síť pro vzájemnou komunikaci.

Centrální bod zpracovaná data a vypočtené pozice zpřístupní pro prezentaci a další zpracování. Pro navržený postup je třeba zvolit způsob výměny dat pomocí síťové komunikace.

2.5 Komunikace

Z měřících bodů je třeba na centrální bod přenášet informace o provozu v síti, pro výpočet stačí informace z hlaviček rámců a Radiotap hlaviček. Jako řešení pro síťovou komunikaci mezi měřícími a centrálním bodem v navrhovaném systému uvažuji následující možnosti:

síťový socket a vlastní protokol Triviální možností je pomocí vlastního programu a nebo hotového nástroje (netcat, ncat, ssh, stunnel, atd.) otevřít síťové spojení na počítač a port, který na daném portu naslouchá a spojení očekává. Je třeba implementovat obě strany komunikace včetně zpracování nestandardních situací (nedostupnost hostitele spojení, připojení více klientů, výpadky v průběhu komunikace, ...). Obsah komunikace nemusí být nijak strukturovaný, vše je plně v rukou programátora.

REST API REST - Representational state transfer umožňuje komunikaci se serverem pomocí standardních HTTP volání (GET, PUT, POST, DELETE, PATCH) s odpovědí. Položky, které lze pomocí těchto volání

vytvářet, měnit apod a mají jasně definovanou URI na kterou příslušný požadavek musí být zaslán. Serverová část implementuje metody zpracovávající požadavky nad objekty podle přijatých volání.

SOAP Simple Object Access Protocol - protokol pro výměnu dat ve formátu dat XML. Podobně jako u REST lze data posílat pomocí HTTP požadavků a na určenou URL. Obsah požadavků a odpovědí má danou strukturu podle připraveného schématu.

CORBA Common Object Request Broker Architecture. Pomocí CORBA lze vyměňovat instance objektů implementované v různýcg jazycích a běhových prostředích, včetně normalizace volání jejich metod. To umožňuje používat techniky objektového programování napříč heterogenním systémem spojeným sítí. Nevýhodou je složitost implementace.

systém založený na posílaní zpráv a zpracování jejich front V těchto systémech existují producenti zpráv, konzumenti zpráv a centrální bod. Producent vytváří zprávy a zasílá je do pojmenované fronty na centrálním bodu. Konzument se přihlásí k odběru zpráv z fronty a zprávy konzumuje. Odeslání a zpracování zpráv funguje asynchronně. Producent odešle zprávu, obsah zpráv nemá danou strukturu. Správci front a centrální bod, ke kterému se připojují producenti i konzumenti, se nazývá broker. Příkladem takových systému jsou AMQP, ActiveMQ, STOMP, ZeroMQ nebo MQTT.

Pro navrhovaný systém lze použít jakoukoliv z výše uvedených metod a zajistit tak spolehlivou komunikaci mezi funkčními celky. Pro implementaci nástroje považuji za nejvhodnější posílání zpráv. Tento způsob nejvíce odpovídá postupu zpracování dat a orchestraci systému jako celku.

Měřící bod nepotřebuje od centrálního bodu potvrzení, že data, která předal centrálnímu bodu, už byla použitá do výpočtu a naopak se hodí chování typické pro systémy s frontami zpráv - producent se po odeslání zprávy o ní dál nemusí zajímat a jeho odeslání pro něj není blokující. Podobně zpracování zpráv se odehrává tak, jak to centrální bod systému stíhá zpracovávat. V případě, že systém pro výpočet poloh nestíhá zpracovávat všechna zachycená data, se nic neděje, zprávy čekají ve frontě, dokud je nezkonzumuje. Takové případy lze ve velkých WiFi sítích předpokládat - například ve firmě přestane fungovat připojení k internetu, zaměstnanci si pro přístup na internet vyrobí přístupové body ze svých mobilních telefonů s datovým tarifem. V tu chvíli se mohou v síti objevit desítky neznámých přístupových bodů, které systém detekuje.

Konzumentů zpráv může být více. Tím lze snadno řešit nasazení záložního výpočetního bodu nebo testovat jeho novou verzi, stačí, když se přihlásí k odběru zpráv u fronty se zachycenými daty.

I pro opačný směr dává tento způsob komunikace smysl. Například předpokládejme, že IDS systém detekoval útočníka a zná jeho MAC adresu. Nyní stačí, aby IDS odeslal MAC adresu útočníka do fronty zpráv, ke které jsou připojené měřící body. Ty tak můžou zahrnout útočníkovu adresu do filtrování provozu ihned po tom, co jí IDS poslal do fronty, aniž by se všechny musely cyklicky dotazovat na IDS.

Jak jsem zmínil na začátku, všechny diskutované možnosti jsou použitelné, ale pro tuto práci navrhuji použít některý za systémů na posílaní zpráv. Tento způsob komunikace je vhodný na spojování rozdílných služeb, umožňuje rychlé změny a zároveň je dostatečně výkonný na zpracování velkého počtu zpráv, který může při monitorování provozu nastat.

2.6 Prezentace výsledků a ovládání systému

Implementovaný systém by měl umožnit grafickou prezentaci zjištěných poloh zařízení v prostoru a to nejlépe v reálném čase. Konkrétním uživatelským rozhraním může být webová stránka, program s grafickým prostředím nebo jen soubor dat definujících graf, který lze v nějakém softwaru zobrazit. Webový prohlížeč je dnes standardním vybavením operačních systémů a lze v něm zobrazovat graficky náročnější prvky, jako může být například 3D graf s polohami detekovaných zařízení. Považuji tuto možnost za nejlepší.

Získaná lokalizační data o zařízeních v síti jsou citlivá, a proto je třeba chránit přístup k systému a data přenášet bezpečně.

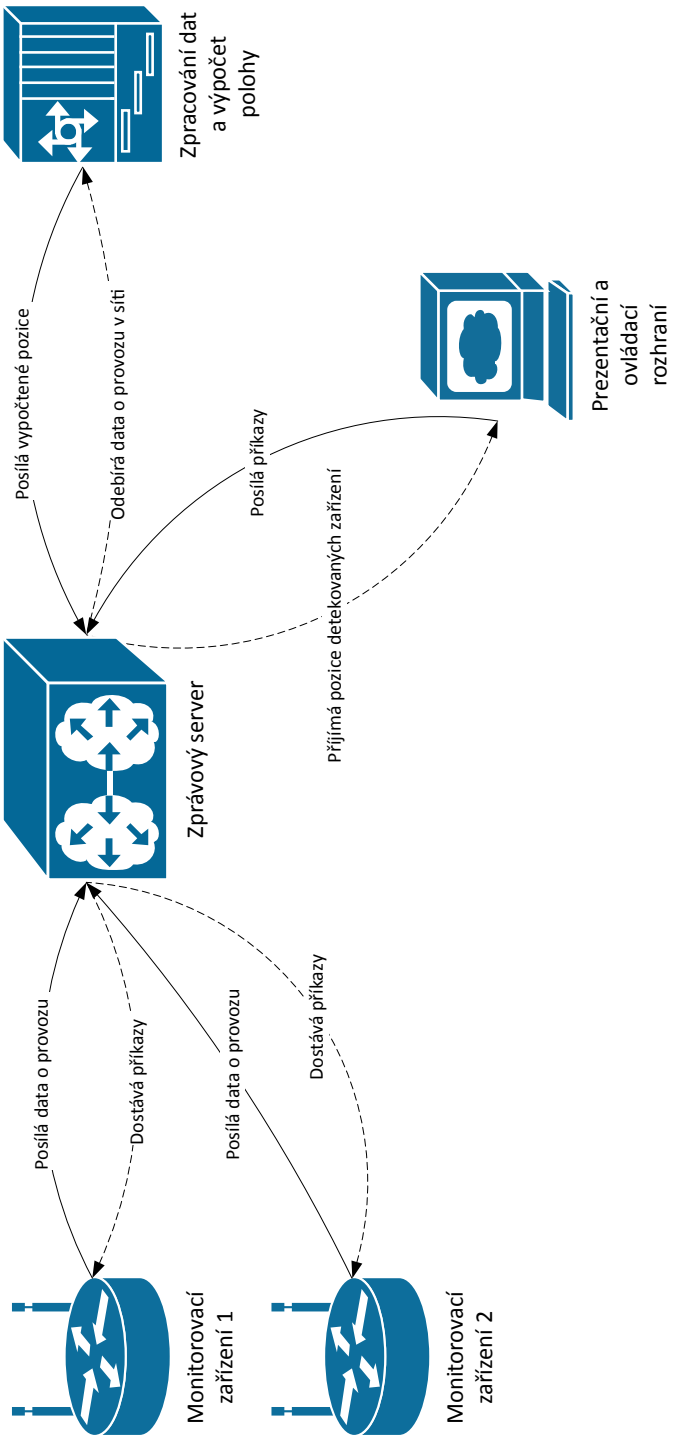
Kromě grafického znázornění by systém měl zpřístupnit zjištěné polohy detekovaných zařízení i pomocí rozhraní umožňující další zpracování a napojení na další software. Možnosti jsou obdobné rozebíraným v sekci komunikace 2.5 a i zde považuji za nejlepší použít některý ze systému na posílání zpráv. Zjištěné polohy budou ihned publikovány do určeného kanálu zpráv. K němu se pak může další program přihlásit a informace o polohách odebírat v reálném čase bez nutnosti periodického dotazování. Navrhované webové prezentační rozhraní může data čerpat právě tímto způsobem.

Se zobrazováním činnosti navrhovaného nástroje souvisí i možnosti jeho ovládání. Musí existovat rozhraní, pomocí kterého lze předávat vstupní data (seznam známých přístupových bodů, adresy útočníků detekovaných IPS/IDS) a kterým půjde ovládat jeho funkce (například zahájit blokování vybraného přístupového bodu). Zobrazení pro prezentaci výsledků tedy navrhuji rozšířit o možnosti ovládání systému, a to jak u grafické verze, tak i programového rozhraní.

2.7 Struktura systému a požadavky na implementaci

Z navrhovaných částí celého nástroje vyplývají dílčí požadavky na implementaci a zároveň je možné navrhnout základní architekturu celého nástroje 2.1. Konkrétní technologie, další rozvrstvení a možnosti konfigurace budou předmětem implementace.

- monitorování provozu
  - hardware: WiFi adaptér a monitorovací režim
  - software: program na sledování a zachycení síťového provozu, spustitelný na síťových zařízeních se systémem OpenWrt nebo podobném
- zpracování dat a výpočet polohy
  - zasílání dat z provozu na centrální bod pomocí zpráv
  - výpočet vzdáleností od měřících bodů pomocí síly signálu
  - určení polohy iterativní minimalizací chyby
- prezentace výsledků a ovládání systému
  - webové rozhraní
  - aplikační rozhraní



Obrázek 2.1: Návrh infrastruktury nástroje k implementaci

Kapitola 3

Implementace

Jak vyplývá z návrhu, systém se skládá z několika funkčních částí, které mezi sebou komunikují prostřednictvím zasílání zpráv. Každou z funkčních částí tvoří samostatný program. V této kapitole rozeberu použitý způsob komunikace a podrobnosti o implementaci jednotlivých programů. Pro navrhovaný systém jako celek používám označení **loc** a z tohoto názvu se odvíjí pojmenování jednotlivých programů.

Navrhovaný nástroj se skládá z těchto programů:

loc.sh Program pro měřící body na monitorování provozu a blokování nepovolených přístupových bodů.

loc.py Výpočetní program pro detekci a lokalizaci zařízení síti pomocí naměřených dat.

loc-viz Webová stránka k prohlížení pozic nalezených zařízení a k ovládání nástroje.

loc-db Program pro ukládání informací o detekovaných zařízeních do databáze.

loc-dashboard Webový přehledový panel, napojený na databázi a s integrací loc-viz.

3.1 Komunikace

Podle návrhu pro komunikaci a výměnu použiji technologii založenou na posílání zpráv do front. Pro implementaci jsem vybral protokol MQTT (Message Queue Telemetry Transport) a jako centrální bod použiji zprávový server (broker) Mosquitto [41]. Klíčové vlastnosti pro výběr tohoto způsobu pro navrhovaný nástroj byly:

- snadná instalace a konfigurace

- výkon
- existují úrovně front a možnost používat masky při odebírání zpráv
- nenáročné programy pro posílání i příjem zpráv
  (mosquitto_pub, mosquitto_sub)
- podpora omezení přístupu a zabezpečení přenosu dat

Při přihlášení odběru zpráv lze uvést konkrétní název fronty nebo použít masky. Název fronty zpráv je tvořen řetězcem znaků a jednotlivé úrovně jsou oddělené pomocí /. Producent zpráv posílá (publikuje zprávy) do fronty a konzument přijímá zprávy z fronty, ke které se přihlásil. Konzument (klient) se může připojit k odběru více témat.

3.1.1 Fronty zpráv

V implementovaném systému je hlavní úroveň určená řetězcem loc a používám následující fronty zpráv:

loc/sensor/<místo> Do této fronty odesílají měřící body zprávy s informacemi o zachycených WiFi rámcích.

loc/command/<místo> Měřící body jsou přihlášené k odběru tohoto tématu, obsah zpráv jsou příkazy pro program na měřících bodech.

loc/location/<místo> Spočtené pozice detekovaných zařízení se posílají na toto téma.

Pro větší instalace může být vhodné fronty dělit na další úrovně. Například pro kancelářský komplex s několika budovami lze zvolit strukturu: loc/sensor/<místo>/<budova>/<patro> a měřící body pak budou zasílat informace do tématu podle toho, kde se nacházejí.

Pomocí speciálních znaků + a # lze odebírat automaticky zprávy z více úrovní témat. Znak v názvu tématu + označuje jakoukoliv jednu úroveň a # označuje aktuální a všechny nižší úrovně. Klient přihlášený k frontě loc/sensor/<místo>/+/<patro> bude dostávat zprávy s naměřenými daty ze všech budov v daném místě a patře a klient přihlášený k loc/sensor/# bude odebírat data ze všech zařízení v místě. Programy v implementaci jsou na dělení do úrovní připravené, názvy front jsou konfigurační proměnné.

3.1.2 Kvalita služeb QoS

V MQTT můžeme určit úroveň kvality služby (QoS - Quality of Service) a zaručit tak doručení zprávy na broker. Lze použít:

QoS 0 zpráva bude na broker doručená maximálně jednou

QoS 1 zpráva bude na broker doručená alespoň jednou

QoS 2 zpráva bude na broker doručená právě jednou

V systému používám QoS 0 pro posílání informací o síťovém provozu a QoS 1 pro zasílání příkazů a vypočtených pozic zařízení.

Poslanou zprávu do fronty lze v MQTT označit tak, že bude ve frontě zanechaná (retained) i po zpracování právě připojenými konzumenty. Toho v systému využívám u fronty zpráv pro zadávání příkazů. Program na měřících bodech po připojení k frontě na přijímání příkazů má ihned k dispozici poslední zprávu s příkazem (ty jsou právě zasílané do fronty jako retained).

3.1.2.1 Zabezpečení přístupu a šifrování přenosu

Protokol MQTT probíhá pomocí TCP/IP a nabízí autentizaci pomocí uživatelského hesla a jména. Šifrování provozu, řízení přístupu k jednotlivým frontám (ACL) a způsoby řešení autentizace jsou záležitostí konkrétní implementace serveru (brokeru).

Mosquitto v instalaci bez rozšíření umožňuje šifrovaný přenos pomocí TLS a přihlášení pomocí uživatelského jména a hesla, které ověřuje oproti souboru se zahešovanými hesly nebo pomocí klientských X509 certifikátů podepsaných autoritou, kterou server podle konfigurace používá. Mosquitto dále umožňuje MQTT komunikaci přes protokol WebSocket, vhodný pro využití ve webový aplikacích.

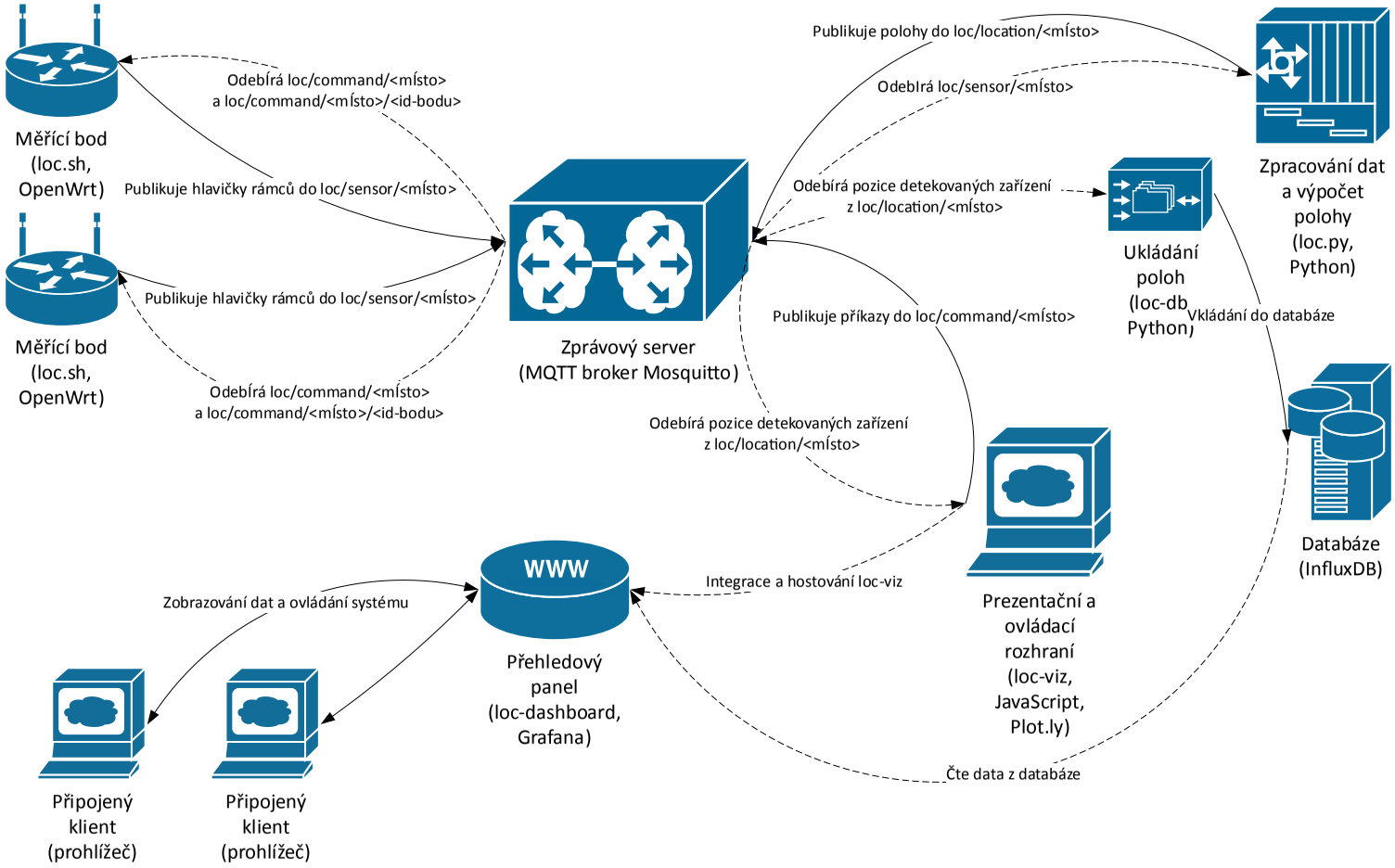
V implementaci je použita autentizace pomocí uživatelského jména a hesla, přenos dat je zabezpečený pomocí TLS a webová prezentační aplikace používá přístup k MQTT frontám přes WebSocket.

Se znalostí způsobu komunikace lze doplnit diagram fungování celého systému 3.1.

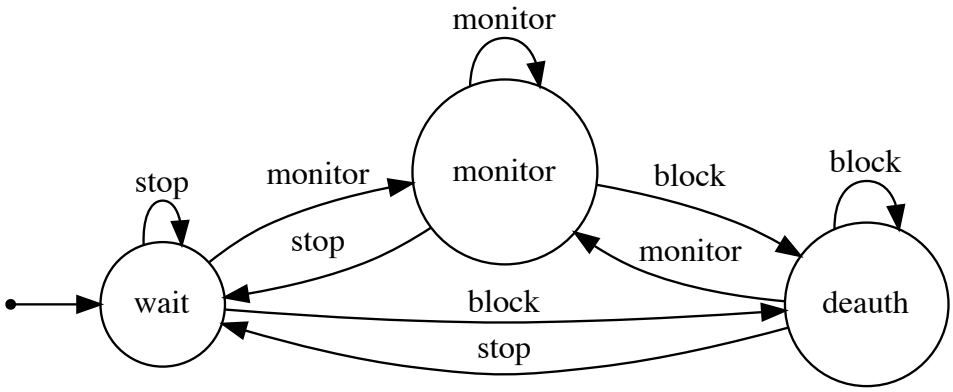
3.2 loc.sh

Program loc.sh je program pro měřící zařízení, který podle příkazů monitoruje a posílá zprávy o WiFi provozu nebo blokuje vybraný přístupový bod. Program je skript pro interpret příkazů a je navržen pro spuštění v operačním systému OpenWrt, kde je výchozím interpretem ash (Almquist Shell) v prostředí Busybox. Program je připraven pro fungování i na dalších linuxových operačních systémech s interpretem Bash.

Program si lze představit jako jednoduchý stavový automat 3.2. Podporované příkazy tvoří přechody mezi stavy a stavy reprezentují činnost, kterou v něm program vykonává.



Obrázek 3.1: Infrastruktura navrhovaného nástroje



Obrázek 3.2: Program loc.sh

3.2.1 Implementované příkazy

Program se po spuštění připojí na frontu na zprávovém serveru podle konfigurace a čeká na příkaz. Kromě hlavní příkazové fronty se program připojuje ještě k o jednu úroveň nižší frontě (loc/command/<místo>/<identifikátor-měřícího-bodu>). Toho systém využívá při blokování nedovolených přístupových bodů. Příkaz k blokování tak lze vyslat pouze vybraným nejbližším bodům a ostatní mohou dál monitorovat provoz.

Program reaguje přechodem do jiného stavu na tyto příkazy:

monitor Spustí sledování okolního WiFi provozu. Součástí zprávy je seznam známých zařízení, adresy útočníků k lokalizaci a seznam kanálů ke sledování.

block Zahájí blokování přístupového bodu na zvoleném kanále. Adresa a kanál jsou součástí zprávy.

stop Zastaví probíhající činnosti a program přejde do vyčkávacího stavu jako při spuštění.

3.2.2 Funkce

3.2.2.1 Monitorování provozu a jeho odesílání

Formát zprávy příkazu pro zahájení monitorování (stav monitor):

```
"monitor" "<seznam␣známých␣zařízení>" \
"<seznam␣útočníků>" "<seznam␣kanálů>"
```

kde seznam známých zařízení jsou mezerou oddělené síťové MAC adresy známých zařízení, obdobně seznam útočníků a seznam kanálů jsou mezerou oddělená čísla kanálů, na kterých bude monitorování probíhat.

Před samotným monitorováním provozu je nutné mít bezdrátové zařízení v monitorovacího režimu. Podle konfigurace buď takové zařízení vytvoří:

```
$ iw dev <zařízení> set monitor otherbss
```

nebo použije už připravené. Monitorování probíhá na množině kanálů, která je součástí zadaného příkazu. Automatická cyklická změna kanálu probíhá jako samostatný podproces s voláním příkazu iw:

```
$ iw dev <zařízení> set channel <číslo kanálu>
```

Monitorování provozu probíhá pomocí programu tcpdump. Pro zvolenou metodu vypočítávání vzdáleností ze síly signálu jsou dostačující vybrané textové informace, které umí tcpdump vypisovat na standardní výstup. Hlavičky zachycených rámců jsou vypisovány po řádcích a přes standardní přesměrování výstupu jsou tyto řádky poslány na vstup MQTT klienta mosquitto_pub. Ten má otevřené spojení pro posílání zpráv a řádky přijaté na standardním vstupu odesílá jako zprávy.

Součástí příkazové zprávy monitor je i seznam známých přípojných bodů a seznam adres útočníků k lokalizaci. Podle těchto polí je zkonstruován filtr pro tcpdump. Výsledný příkaz pro volání programu tcpdump hlavním skriptem má tři části:

parametry spuštění

```
$ tcpdump -e -y IEEE802_11_RADIO -l -n -tt \
-i <zařízení>
```

Přepínače popořadě určují: výpis informací o rámcích na spojové vrstvě, určení typu rámců, vypisování rámců po řádcích, vypnutí překladu adres na jména, vypisování časů zachycení jako vteřin od 1. ledna 1970 00:00:00 UTC a volbu zařízení k monitorování provozu.

filtr pro detekci přístupových bodů

```
((subtype beacon or subtype probe-resp
or subtype assoc-resp or subtype reassoc-resp
or subtype disassoc)
and wlan src not <známé zařízení 1>
or wlan src not <známé zařízení 2>
... or wlan src not <známé zařízení n>)
```

Filtrem projdou rámce vysílané přístupovými body, které nepocházejí ze známých zařízení.

filtr pro detekci útočníků

```
... or wlan src <adresa útočníka 1>
or wlan src <adresa útočníka 2>
... or wlan src <adresa útočníka n>
```

Pro hledání útočníků jsou užitečné jakékoliv zachycené rámce s jeho zdrojovou adresou.

3.2.2.2 Blokování přístupového bodu

Formát zprávy příkazu pro zahájení blokování (stav deauth):

```
"block" "<adresa zařízení>" "<kanál>"
```

Po přijetí příkazu na blokování přístupového bodu je spuštěn program aireplay-ng s následujícími parametry:

```
$ aireplay-ng -0 0 -a <adresa AP> <rozhraní>
```

Tím dochází k nepřetržitému odpojování klientů zvoleného přístupového bodu (a tím k jeho blokování) posíláním deautentizačních řámců na všesměrovou adresu s podvrženou zdrojovou adresou. Před samotným spuštěním blokování je síťový adaptér nastaven na kanál, kde přípojný bod vysílá. Číslo kanálu je součástí přijaté zprávy.

3.2.2.3 Čekání na příkaz

V tomto stavu (wait) se program nachází po spuštění a přejde do něj při přijetí zprávy:

```
"stop"
```

Přechodem do tohoto stavu program ukončí předchozí probíhající činnost.

3.2.3 Konfigurace a spuštění

Konfigurace je tvořená textovým souborem ve formátu klíč hodnota. Popis dostupných konfiguračních voleb:

Ovládání běhu programu:

spuštění

```
$ ./loc.sh -k start -c <konfigurační soubor>
```

zastavení

```
$ ./loc.sh -k stop -c <konfigurační soubor>
```

node_id	identifikátor měřícího bodu
device	zařízení pro monitorování
channel_hop_interval	interval pro změnu kanálů (ve vteřinách)
prepare_device	má program připravit zařízení pro monitorování?
pidfile	soubor pro zapsání čísla spuštěného procesu
mqtt_server	hostitel MQTT serveru
mqtt_port	port MQTT serveru
ca_certificate	soubor s certifikátem autority TLS spojení
username	uživatelské jméno pro MQTT připojení
password	heslo pro MQTT připojení
publish_topic	fronta zpráv pro odesílání naměřených dat
command_topic	fronta zpráv pro příjem příkazů

Tabulka 3.1: Volby konfiguračního souboru pro loc.sh

3.2.4 Požadavky

Pro běh programu musí být v systému dostupný následující software:

- interpret příkazů (zsh, bash)
- iw
- mosquitto_pub, mosquitto_sub
- tcpdump
- aireplay-ng
- pkill

Všechny požadované nástroje jsou dostupné v balíčkovacím systému operačního systému OpenWrt 15.05 Chaos Calmer i ostatních běžných distribucích Linuxu. Běh programu jsem testoval na bezdrátovém směrovači TP-LINK TL-WR842N, kde operační systém, potřebný další software a skript dohromady zabíraly 3.2MB diskového prostoru. Na síťových zařízeních s architekturou MIPS tedy dostačuje velikost úložiště 4MB, kterou disponuje většina zařízení podporovaných systémem OpenWrt 15.05.

3.3 loc.py

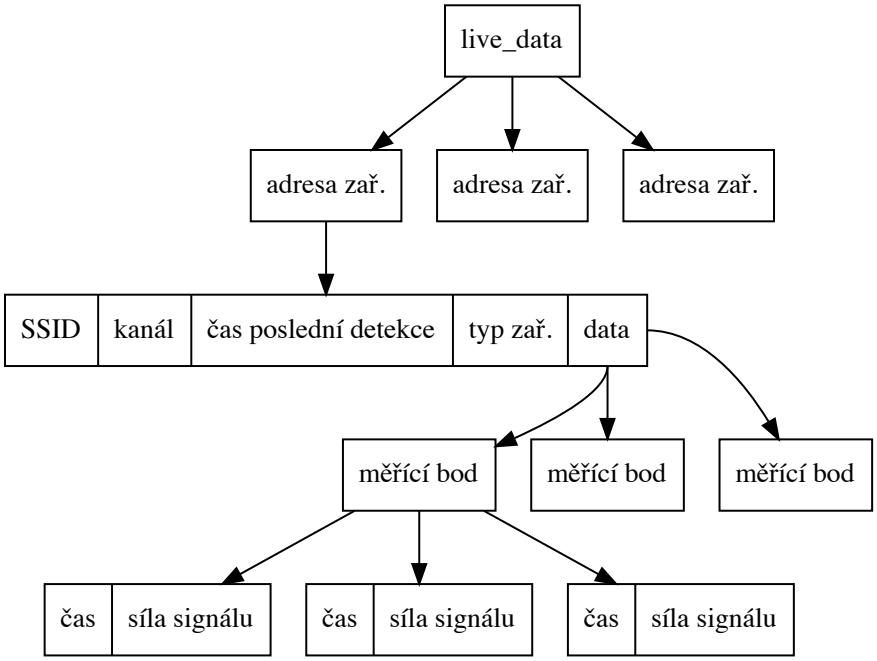
Program loc.py je přihlášen k odběru zpráv se zachycenými WiFi rámci v prostředí a na základě těchto informací detekuje hledaná zařízení a počítá jejich polohy. Ty pak odesílá opět jako zprávy k dalšímu zpracování a vizualizaci.

3.3.1 Fungování programu

3.3.1.1 Příjem zpráv a zpracování dat

Program se po startu ve vyhrazeném vlákně připojí k frontě podle konfigurace a očekává zprávy se zachycenými hlavičkami rámců.

Z přijatých zpráv jsou extrahovány jednotlivé položky a ukládají se do struktury live_data 3.3. Ta udržuje ke každému identifikovanému zařízení seznam zachycených signálů rozdělených podle jednotlivých měřících bodů, které příslušný rámec zachytily. Tato struktura představuje vstupní data pro zjištění vzdáleností hledaných zařízení od měřících bodů a následný výpočet jejich polohy.



Obrázek 3.3: Struktura live_data

Pro každé zachycené zařízení a každý měřící bod, který jej zachytil, se uchovávají seznamy zachycených signálů, respektive vypočtených vzdáleností.

3.3.1.2 Výpočet polohy zařízení

Hlavní vlákno programu funguje v nekonečném cyklu, jehož funkcí je výpočet poloh detekovaných zařízení, mazání starých dat ze struktury live_data a odesílání zjištěných informací o detekovaných zařízeních.

Výpočet polohy je pro detekované zařízení spuštěn pouze v případě, že existují alespoň čtyři měřící body, které zaměřily.

Do struktury live_data se měřením dostane ke každému detekovanému zařízení a ke každému bodu, který jej detekoval, více hodnot intenzit zachycených signálů. Pro výpočet polohy je z těchto hodnot vybraná ta nejvyšší, tedy nejsilnější signál.

K přepočtu signálu na vzdálenost se používá model pracující s naměřeným referenčním útlumem 1.2, bez rozšíření o efekt náhodného stínění. Hodnoty referenčních útlumů pro měřící body a parametr prostředí jsou součástí konfiguračních souborů při spuštění programu.

Ze získaných vzdáleností od měřících bodů je pak podle návrhu iterativně nalezena poloha s nejmenší chybou. K výpočtu je použitá funkce optimize.minimize z modulu Scipy.

Ze struktury získaných informací o detekovaných zařízeních se pravidelně (ve výchozím nastavení jednou za 5 vteřin) spouští cyklus pro výpočet poloh zařízení a mazání starších dat. Cyklus postupně prochází strukturu live_data. Pokud bylo neznámé zařízení detekováno alespoň čtyřmi měřícími body, je pro dané zřízení spuštěn výpočet polohy.

Naměřené hodnoty i detekovaná zařízení jsou na začátku každé iterace cyklu mazána ze struktury live_data podle nakonfigurovaných časových mezí.

3.3.2 Zpřístupnění

Po dokončení cyklu s výpočtem poloh detekovaných zařízení jsou tyto informace publikovány do kanálu zpráv podle konfigurace. Všechny informace jsou obsažené v jedné zprávě v notaci JSON s následujícím formátem:

```
{
  "type" : "intruders",
  "intruders" : seznam detekovaných zařízení
}
```

Kde položka v poli detekovaných zařízení má formát:

```
{
    "mac" : adresa zařízení,
    "device_type" : typ zařízení (ap neb sta),
    "ssid" : jméno vysílané sítě,
    "channel" : kanál,
    "time" : čas naposledy detekováného signálu
    "position" : {
```

```
      "x" : souřadnice x,
      "y" : souřadnice y,
      "z" : souřadnice z
    }
 }
```

Příklad zprávy se dvěma zařízeními:

```
{
  "type" : "intruders",
  "intruders" : [ {
    "ssid" : "internet",
    "mac" : "a0:a1:a2:a3:a4:a5",
    "device_type" : "ap",
    "time" : "2017-06-25T07:21:17.972748",
    "position" : {
      "y" : 2.77,
      "x" : 4.41,
      "z" : 5.47
    },
    "channel" : "3"
  }, {
    "mac" : "f0:f1:f2:f3:f4:f5",
    "device_type" : "sta",
    "channel" : "9",
    "position" : {
      "y" : 1.27,
      "x" : 5.44,
      "z" : 6.62
    },
    "time" : "2017-06-25T07:20:43.166781"
  }]
}
```

3.3.3 Konfigurace a spuštění

Program pro spuštění vyžaduje dva konfigurační soubory: konfiguraci programu a soubor s informacemi o měřících bodech. Parametry spuštění:

```
$ python loc.py -c <konfigurační soubor> \
-n <konfigurace měřících bodů>
```

3.3.3.1 Konfigurace programu

Konfigurační soubor programu má tvar vhodný pro zpracování modulem ConfigParser. Konfigurační soubor má čtyři sekce a následující položky:

sekce	parametr	popis
MQTT	Server	hostitel MQTT serveru
	Port	port MQTT serveru
	Username	už. jméno pro MQTT připojení
	Password	heslo pro MQTT připojení
	CAcert	certifikát autority pro TLS spojení
PropagationModel	N	parametr γ v modelu šíření signálu
Signal	CutoffTreshold	hranice síly signálu (v dBm) pro použití do výpočtu
Timeout	Device	doba platnosti detekce zařízení
	Data	doba platnosti naměřených signálů
	MainLoop	doba nečinnosti hlavního cyklu

Tabulka 3.2: Konfigurace programu loc.py

3.3.3.2 Definice měřících bodů

Pro výpočet poloh zařízení je třeba znát umístění měřících bodů a jejich referenční síly signálu. Konfigurační soubor používá stejný formát jako konfigurace programu. Název sekce představuje identifikátor měřícího bodu a parametry v sekci jeho vlastnosti.

Implementovaný nástroj nepoužívá zeměpisné souřadnice. Polohy měřících bodů je třeba zadat ve vlastním třírozměrném souřadném systému pro dané místo nasazení a v tomto systému budou i vypočtené pozice detekovaných zařízení.

parametr	popis
X	souřadnice pozice ve směru x (v metrech)
Y	souřadnice pozice ve směru y (v metrech)
Z	souřadnice pozice ve směru z (v metrech)
ReferenceSignal	referenční síla signálu ve vzdálenosti 1 m (v dBm)

Tabulka 3.3: Konfigurace parametrů měřících bodů

Příklad definice přístupového bodu v konfiguračním souboru:

```
[node-1]
X: 10
Y: 20
Z: 0
ReferenceSignal: -40
```

Pro větší instalace by bylo vhodné tento způsob konfigurace měřících bodů nahradit databází propojenou s konfigurací přístupu do MQTT serveru.

3.4 loc-viz

Program loc-viz je webová stránka k zobrazování pozic detekovaných zařízení v prostorovém grafu v reálném čase a k ovládání systému.

Program používá knihovny Paho JavaScript Client pro MQTT komunikaci a plotly.js pro zobrazování grafu. Pro MQTT komunikaci používá technologii WebSocket, kterou mosquitto broker podporuje. Komunikace probíhá šifrovaně přes TLS.

Po načtení stránky se program připojí k frontě zpráv s informacemi o detekovaných zařízeních.

Zařízení a jejich pozice jsou pak živě vykreslovány na graf typu 3D scatter. K běhu stránky není potřeba spouštět další kód na webovém serveru, veškerý kód běží v prohlížeči klienta. Pro lepší orientaci v grafu jsou v něm vykreslené i pozice měřících bodů. Informace o nich program dostane speciální zprávou typu "nodes"ve frontě s pozicemi. Pro vizuální kontrolu při testování je dále implementováno zobrazení známých zařízení na grafu, které lze opět poslat do fronty se speciální zprávou typu "known-devices".

Prohlížeč musí podporovat JavaScript a technologii WebGL pro zobrazování 3D grafu.

Pomocí rozhraní lze posílat příkazy měřícím bodům a spustit tak monitorování provozu, blokovat vybraný detekovaný přístupový bod nebo zastavit činnost příkazem stop. Příkazy jsou MQTT klientem poslané do příkazové fronty a lze zvolit, zda příkaz bude pro všechna monitorovací zařízení, nebo jen pro vybraná.

Snímek obrazovky 3.4 načteného rozhraní ukazuje jednotlivé ovládací prvky a interaktivní graf s pozicemi měřících bodů, známých zařízení a aktuálně detekovaných zařízení[4].

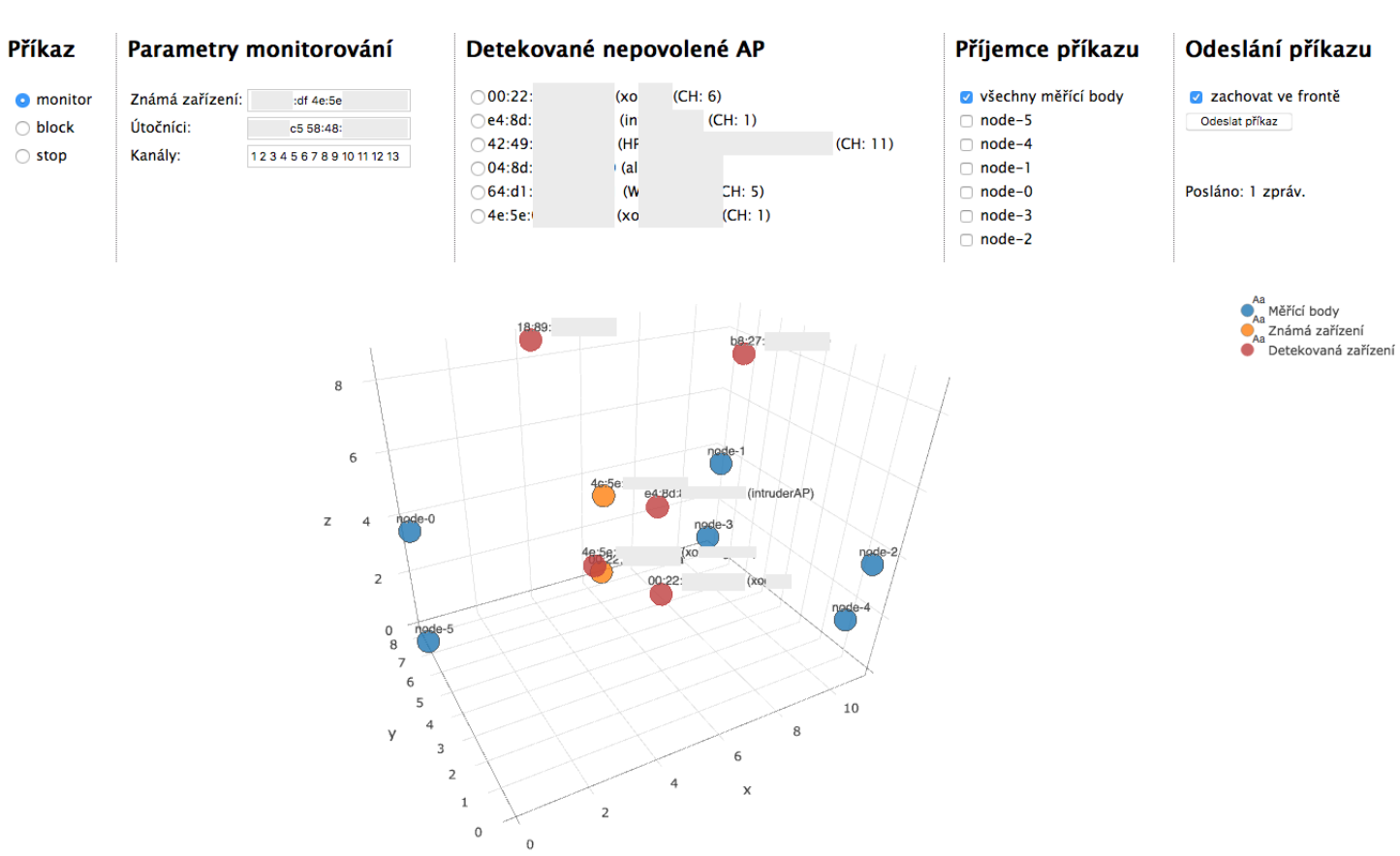
3.5 loc-db

Program loc-db.py je opět program v jazyce Python a jeho úkolem je ukládat polohy detekovaných zařízení do databáze jako perzistentního úložiště. Běh programu je triviální: po startu se připojí na frontu zpráv, kam loc.py posílá informace o zařízeních, a zprávu uloží do tabulky. Systém používá databázový software InfluxDB.

Databáze používá jednu tabulku. Klíčem řádku je čas ze zprávy a sloupce jsou hodnoty: MAC adresa zařízení, kanál, SSID a souřadnice polohy (x, y, z). Do databáze se ukládají i záznamy bez informace o pozici nebo SSID.

Konfigurační soubor používá stejnou syntaxi jako loc.py a obsahuje pouze dvě sekce: MQTT a InfluxDB. V sekci InfluxDB jsou následující parametry pro připojení k databázi.

[4]Na snímku jsou částečně skryté síťové adresy a názvy sítí z důvodu ochrany osobních údajů. Podobně budou upravené i všechny další snímky v práci.



Obrázek 3.4: Webové rozhraní loc-viz

parametr	popis
Server	hostitel InfluxDB serveru
Port	port InfluxDB serveru
Username	uživatelské jméno pro připojení k databázi
Password	heslo pro připojení k databázi
Database	databáze pro ukládání dat

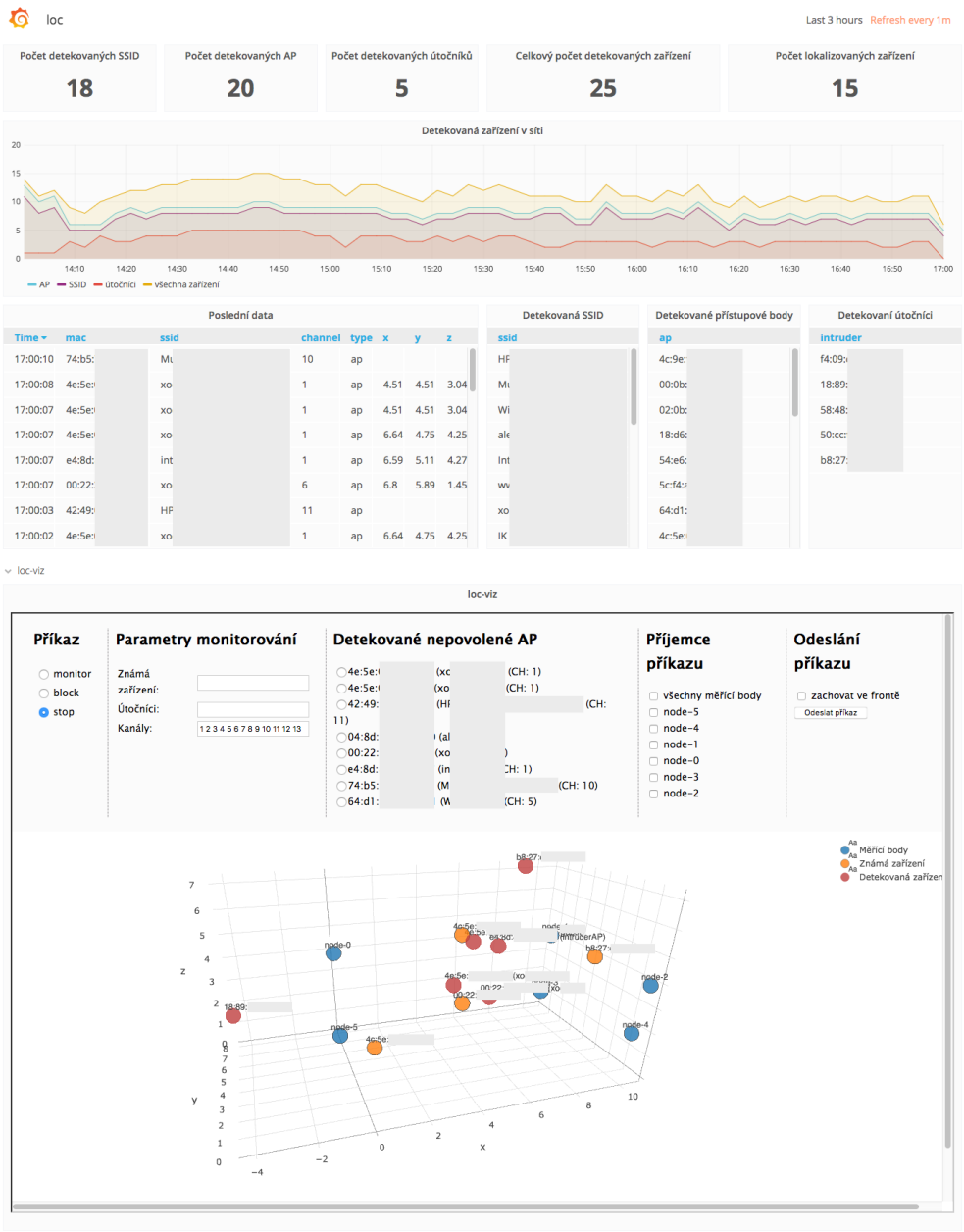
Tabulka 3.4: Konfigurace připojení k databázi v loc-db

3.6 loc-dashboard

Součástí implementace je definice přehledového panelu (dashboard) pro software Grafana. Tento panel pracuje s daty z databáze a zobrazuje následující informace pro zvolené časové období:

- počet detekovaných přístupových bodů
- počet rozdílných vysílaných názvů sítí (SSID)
- počet útočníků
- celkový počet detekovaných zařízení
- počet lokalizovaných zařízení
- graf průběhu výše uvedených metrik v čase
- seznam zachycených názvů sítí
- seznam adres detekovaných přístupových bodů
- seznam adres detekovaných útočníků
- posledních několik desítek naměřených hodnot

V panelu lze zvolit dynamický časových interval (například poslední hodinu) a automatické obnovování zobrazených dat a lze tak stránku používat i jako přehled o aktuálních informacích v síti. Do stránky je integrováno rozhraní loc-viz, které zobrazuje detekovaná zařízení v grafu a umožňuje posílat příkazy.



Obrázek 3.5: Přehledový panel loc-dashboard v softwaru Grafana

Kapitola 4

Testování

Implementovaný nástroj na detekci, lokalizaci a blokování zařízení v bezdrátových sítích jsem otestoval ve vlastním prostředí. Následující kapitola popisuje přípravu a testování implementovaného systému.

4.1 Instalace a příprava

4.1.1 Měřící body

Program loc.sh pro měřící body jsem otestoval na několika různých zařízeních a operačních systémech. Pro využití měřícího bodu v systému je nutné zajistit, aby pro měřící bod byl síťově dostupný server na zasílání zpráv na hostiteli a portu podle konfigurace.

4.1.1.1 Síťové zařízení

Pro ověření možnosti použití síťového zařízení jako měřícího bodu jsem použil bezdrátový směrovač TL-WR842N v ceně okolo 700 Kč. Tento směrovač má vestavěný WiFi adaptér, disponuje USB portem pro připojení dalších zařízení a originální software lze nahradit systémem OpenWrt 15.05. Takové zařízení je tedy vhodným kandidátem pro využití jednoho zařízení zároveň jako přístupového a měřícího bodu. Pro měření jsem použil vestavěný bezdrátový síťový adaptér.

4.1.1.2 Virtuální stroj

Běh programu jsem také otestoval ve virtualizovaném prostředí. Použil jsem připravený obraz operační systému OpenWrt 15.05 pro architekturu x86, který jsem rozšířil o potřebné nástroje a skripty. Výsledný virtuální stroj jsem otestoval ve virtualizačních nástrojích VirtualBox a Parallels.

Vzhledem k tomu, že většina dostupných procesorů a operačních systému podporuje virtualizaci a nároky na prostředky virtuálního stroje jsou malé

(jednotky MB diskového prostoru i operační paměti), lze provozovat měřící bod na pracovních stanicích souběžně s hlavním operačním systémem a vyhnout se tak použití dedikovaného zařízení pro měřící bod. Jediným dalším požadavkem je pak pořízení vhodného WiFi adaptéru v ceně stovek korun.

4.1.1.3 Počítač

Program lze provozovat i přímo na počítači s nainstalovaným Linuxovým operačním systémem. Kromě běžných počítačů jsem měl pro testování k dispozici několik tenkých klientů řady HP T5700, pro které jsem vzhledem k omezeným prostředků připravil opět systém OpenWrt.

4.1.1.4 Měřící body pro testovací prostředí

Pro testování jsem připravil 7 měřících bodů 4.1. Jde o kombinaci různých zařízení, operačních systémů a adaptérů. Tím se zároveň ověřilo, že systém lze provozovat na běžně dostupném a zároveň dosti rozdílném hardwaru s odpovídajícím softwarem. Referenční útlumy jsem měřil vysíláním sítě testovacím směrovačem Mikrotik RB951Ui-2nD ve vzdálenosti 1 metr do antény měřících bodů.

4.1.2 Komunikace

Server pro komunikaci pomocí zasílání zpráv je jediná část systému, která musí být pro všechny další části dostupná. MQTT broker Mosquitto jsem pro testování provozoval v pronajatém virtuálním privátním serveru (VPS) dostupném přes internet. VPS má 2 GB RAM, SSD uložiště a vyhrazené jedno vlákno procesoru Intel Xeon E5-2650L. Na VPS provozuji operační systém Debian 8.7, ve kterém je Mosquitto i další potřebný software dostupný k instalaci pomocí balíčků. Pro TLS komunikaci jsem použil doménově ověřený certifikát. Pro měřící body a další části celého nástroje jsem vytvořil uživatele a hesla pro komunikaci pomocí příkazu mosquitto_passwd. Veškerá komunikace s VPS probíhá přes internet.

4.1.3 loc-viz

Webové rozhraní pro zobrazování poloh detekovaných zařízení a ovládání systému lze spustit v prohlížeči přímo z disku bez webového serveru. Pro integraci do přehledového panelu je stránka loc-viz dále hostovaná webovým serverem na VPS.

4.1.4 Ostatní

Programy loc.py, loc-db (a InfluxDB) a nástroj Grafana jsem při vývoji testoval a spouštěl samostatně na různých strojích, ale pro potřeby měření jsou

identifikátor	zařízení	WiFi adaptér	operační systém	referenční útlum (dB)
node-0	HP T5740	TP-LINK TL-WN722N	OpenWrt 15.05	-32
node-1	TP-LINK TL-WR842N	vestavěný Atheros QCA9531	OpenWrt 15.05	-32
node-2	HP T5740	TP-LINK TL-WN722N	OpenWrt 15.05	-32
node-3	HP T5740e	vestavěný Atheros AR928X	OpenWrt 15.05	-20
node-4	Parallels na Apple Macbook5,2	TP-LINK TL-WN722N	OpenWrt 15.05	-32
node-5	Dell Latitude X1	Alfa AWUS036H	Ubuntu 14.04	-14
node-6	Dell Optiplex 755	TP-LINK TL-WN722N	Debian 8.7	-32

Tabulka 4.1: Měřící body pro testování

všechny spouštěny na VPS, ačkoliv to architektura nevyžaduje.

4.2 Detekce a lokalizace

Testování probíhalo ve zděném třípatrovém řadovém rodinném domě používaném jako kanceláře firmy. Na každém patře se nachází několik místností včetně vybavení (nábytek, počítače, tiskárny, kuchyň). Testovaní probíhalo za plného provozu firmy.

V prostoru domu jsem rozmístil připravené měřící body do nepravidelného tvaru, tak, jak to umožňovaly dispozice prostoru.

Pro měření přesnosti jsem do prostoru dále umístil tři přístupové body se známou polohou:

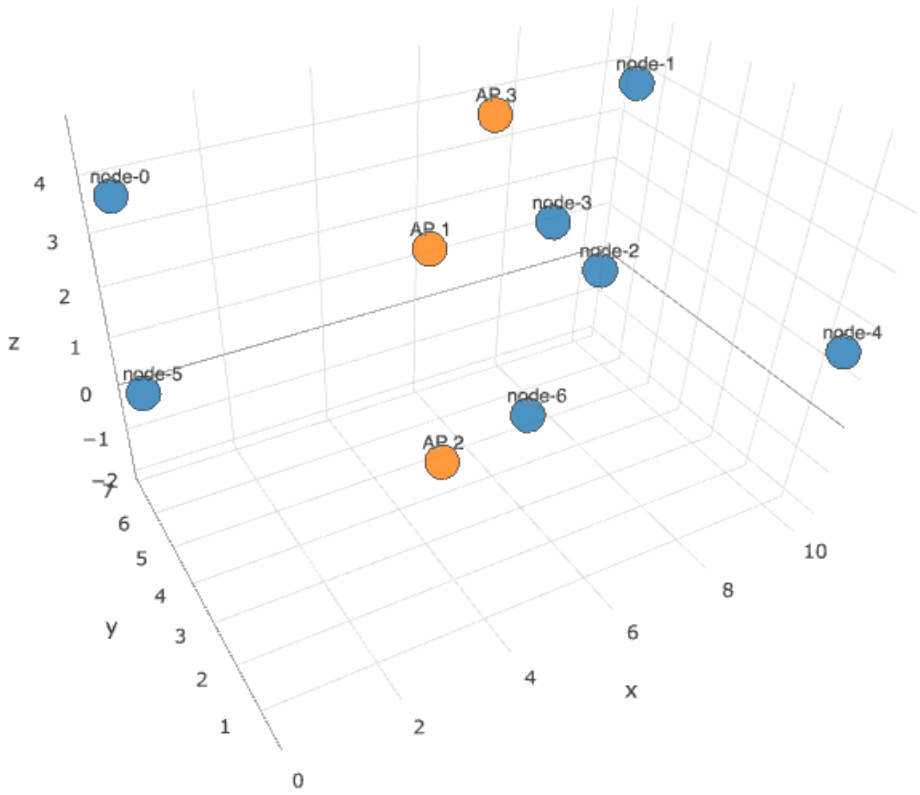
- AP 1
  - přístupový bod Netgear WPN802v2
  - v otevřeném prostoru s přímou viditelností na node-3
- AP 2
  - bezdrátový směrovač Mikrotik 2011UiAS-2HnD
  - umístění v rackovém rozvaděči obklopeném zdmi
  - žádný z měřících bodů nemá přímou viditelnost
- AP 3
  - bezdrátový směrovač Mikrotik 941-2nD
  - přímá viditelnost na node-2

Rozmístění je znázorněné na diagramu 4.1.

4.2.1 Detekce

Nástroj detekuje neznámé přístupové body a zařízení útočníků se zadanou adresou. Zpoždění detekce zařízení měřícím bodem je dané střídáním kanálů pro monitorování celého frekvenčního rozsahu. Pro 13 kanálů v 2.4 GHz může tedy trvat více než deset vteřin, než měřící bod zachytí vysílání neznámého zařízení. Další zpoždění je dáno tím, že procházení naměřených dat programem loc.py probíhá v cyklu s danou dobou nečinnosti.

V testovacím prostředí jsem zkoumal, jaké bude zpoždění detekce přístupového bodu. Ukázalo se, že z důvodu nezávislosti přístupových bodů a nesynchronním střídání kanálů, je detekce obvykle významně rychlejší, než je teoretické maximum, pokud je hledaný přístupový bod v dosahu více měřících bodů. Tabulka 4.2 ukazuje příklad detekce přístupového bodu v dosahu všech



Obrázek 4.1: Rozmístění testovacích zařízení

měřících bodů při monitorování 13 kanálů a dobou nečinnosti hlavního cyklu v loc.py 5 vteřin.

Během měření se nestalo, že by zařízení komunikující na některém z monitorovaných kanálů v dosahu měřících bodů nebylo detekováno.

4.2.2 Přesnost lokalizace

V prostředí jsem testoval různé hodnoty parametru γ a zkoumal jsem, jaký vliv bude mít na přesnost určení známých zařízení systémem. Chyba měření je určená vzdáleností mezi skutečnou pozicí a pozicí, kterou systém vypočítal na základě naměřených dat.

Pro další měření v testovacím prostředí jsem podle zjištěných hodnot 4.3 zvolil hodnotu $\gamma = 2.9$, protože u ní docházelo k nejpřesnějšímu určení polohy. Ani pro další dvě testované hodnoty γ se přesnost určené polohy zařízení AP 1 a AP 3 podstatně nelišila. Chyby vypočtených pozic pro $\gamma = 2.9$ jsou znázorněné na grafu 4.2.

Pro lepší porozumění, proč dochází k nepřesnému určení poloh zařízení, jsem na grafech znázornil ze signálů vypočtené vzdálenosti jednotlivých měřících bodů k AP 1 (obr. 4.3) a AP 2 (obr. 4.4). Záporná chyba znamená,

událost	čas (s)
čas povolení bezdrátové sítě na AP	0.00
čas detekce 1. měřícím bodem	0.58
čas detekce 2. měřícím bodem	0.70
čas detekce 3. měřícím bodem	3.00
čas detekce 4. měřícím bodem	4.89
čas 1. odeslané zprávy s pozicí	5.59
čas detekce 5. měřícím bodem	6.51
čas detekce 6. měřícím bodem	7.78
čas detekce 7. měřícím bodem	8.19

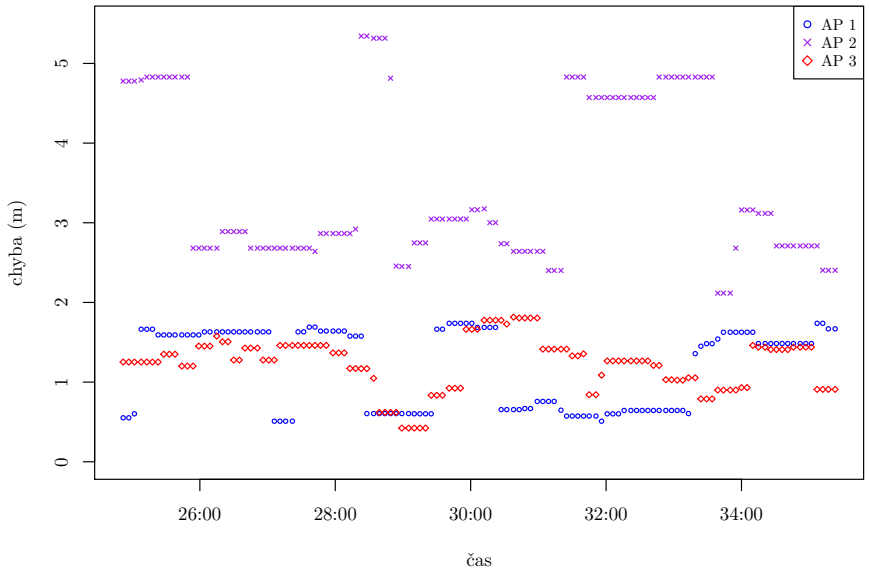
Tabulka 4.2: Zpoždění detekce přístupového bodu

	Zařízení	Min.	1Q	Medián	3Q	Max.	IQR
$\gamma = 2.5$	AP 1	0.32	0.90	1.10	1.56	9.90	0.66
	AP 2	2.24	3.70	6.15	6.38	7.71	2.68
	AP 3	0.53	1.02	1.40	1.60	1.91	0.58
$\gamma = 2.9$	AP 1	0.51	0.64	1.48	1.63	1.73	0.99
	AP 2	2.12	2.68	3.00	4.78	5.30	2.10
	AP 3	0.43	1.02	1.27	1.44	1.81	0.42
$\gamma = 3.3$	AP 1	0.69	1.14	1.81	1.86	2.03	0.72
	AP 2	2.99	3.62	3.80	4.16	4.48	0.54
	AP 3	0.77	1.36	1.58	1.66	1.81	0.30

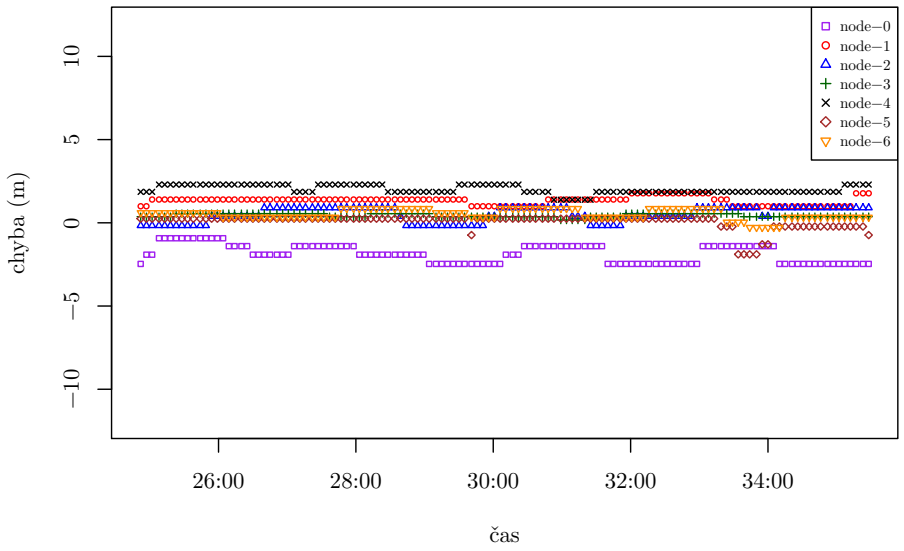
Tabulka 4.3: Vliv nastavení hodnoty γ na přesnost měření

že skutečná vzdálenost je mezi známým a měřícím bodem o danou hodnotu kratší. Naopak kladná znamená, že skutečná vzdálenost je o daný úsek delší.

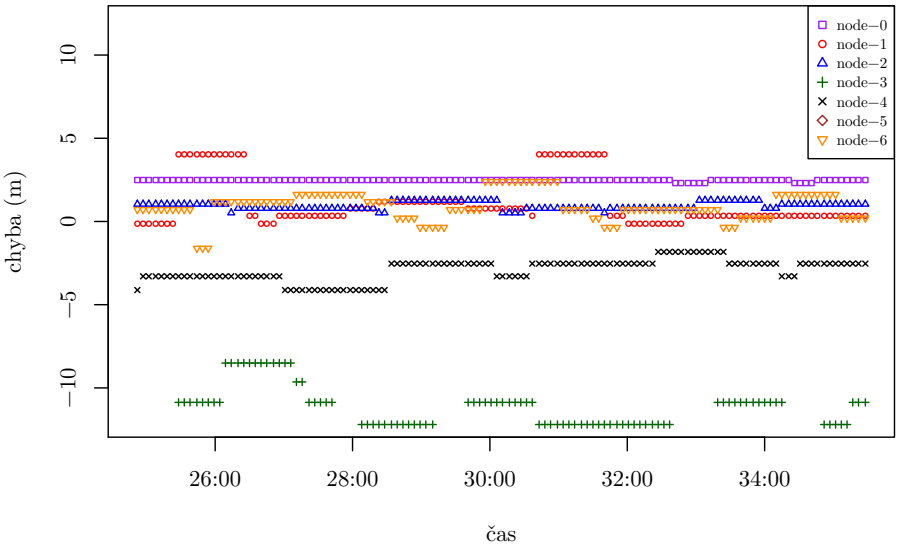
Vzdálenosti jsou vstupní data pro výpočet polohy. U vzdáleností k AP 2 dochází k větším chybám a je zřejmé, že čím nepřesnější jsou jednotlivé vzdálenosti, tím nepřesnější bude i vypočítaná výsledná pozice hledaného zařízení.



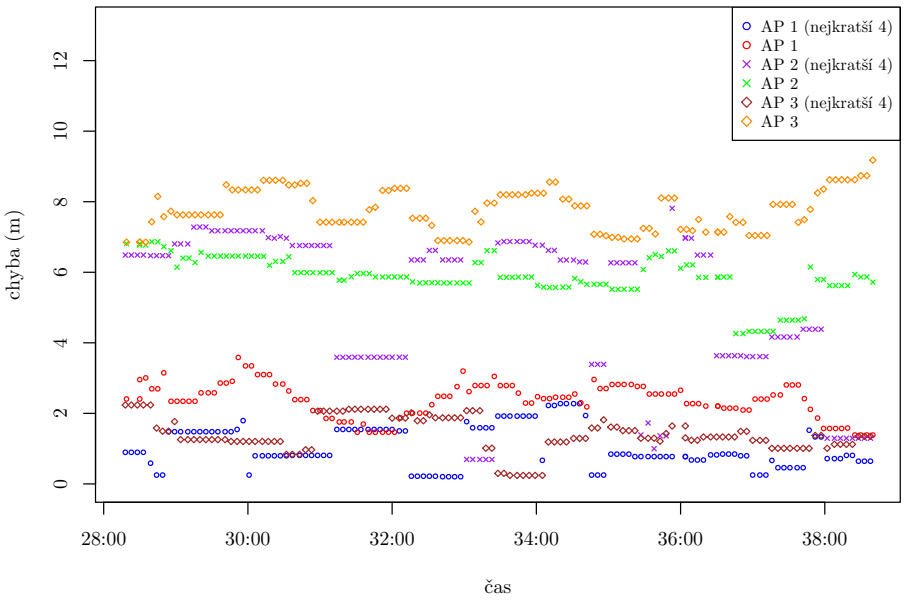
Obrázek 4.2: Chyba vypočítaných pozic testovacích zařízení



Obrázek 4.3: Chyba vzdáleností od měřících bodů k AP 1



Obrázek 4.4: Chyba vzdáleností od měřících bodů k AP 2



Obrázek 4.5: Vliv výběru pouze nejkratších 4 vzdáleností

4.2.2.1 Výpočet polohy z nejlepší čtveřice

Při implementaci a testování mě napadla teoreticky nepodložená idea, jak zlepšit přesnost výpočtu polohy. Do iterativního výpočtu polohy ze vzdáleností získaných měřícími body by se neuvažovaly vzdálenosti ze všech měřících bodů, ale pouze z těch, které jsou hledanému zařízení nejblíže (vzdálenost je nejkratší). Jejich měření považuji za méně ovlivněné prvky prostředí.

Tuto uvažovanou možnost jsem porovnal s původním řešením (obr. 4.5), kdy do výpočtu polohy vstupují vzdálenosti ze všech měřících bodů, které zařízení zachytila. Metoda použití pouze nejkratších čtyř vzdáleností se v testovaném prostředí ukázala jako přesnější a je implementována v systému.

4.2.3 Zhodnocení

V testovacím prostředí je implementovaný systém schopný detekovat neznámá zařízení. Pokud jsou detekována dostatečným počtem měřících bodů, je vypočítána jejich pozice.

Přesnost zjištěné pozice silně závisí na umístění konkrétního zařízení v prostředí, přestože je daný prostor hustě pokrytý měřícími body. Přesto hodnotím schopnost lokalizace zařízení v testovacím prostoru jako dobrou a použitelnou pro lokalizaci nepovolených zařízení a jejich odstranění.

Pro plné ověření přesnosti lokalizace by bylo nutné systém otestovat v rozsáhlejším prostředí. Je možné, že se systém bude potýkat s vlivy prostředí, které se v testovací lokalitě neprojevily a chybovost bude větší.

4.3 Blokování

Nástroj obsahuje funkci na blokování vybraného detekovaného bezdrátového přístupového bodu. Implementace neobsahuje funkci pro automatický výběr nejvhodnějšího měřícího bodu pro blokování, operátor systému musí bod zvolit sám.

Úkolem této části testování je ověřit, zda je blokovaný přístupový bod opravdu nedostupný. Stávající připojení klienti by měli být odpojeni a nové pokusy o připojení by měly být neúspěšné.

Jako podvodný přístupový bod jsem použil bezdrátový směrovač MikroTik RouterBoard RB951Ui-2nD. Směrovač vysílá síť "IntruderAP"se zabezpečením WPA2 a MAC adresa jeho bezdrátového adaptéru je e4:8d:8c:be:d6:a7. K síti mají přístup tato zařízení

- počítač Intel NUC 5I5RYH s operačním systémem Windows 10
- počítač Apple MacBook Air 6,2 s operačním systémem macOS 10.12
- mobilní telefon Samsung Galaxy S2 s operačním systémem Android 4.1.2

- mobilní telefon Apple iPhone 5S s operačním systémem iOS 10.3.2

a je u nich zvoleno automatické připojování k této síti.

Po zadání příkazu k blokování jsem měřil (tabulka 4.4), za jak dlouho přestane klientům fungovat síťové spojení, a zkoumal jsem, zda se jim ho podaří po dobu blokování obnovit.

	NUC	Macbook	Galaxy	iPhone
doba od poslání příkazu ke ztrátě spojení	< 1 s	< 1 s	< 1 s	< 1 s

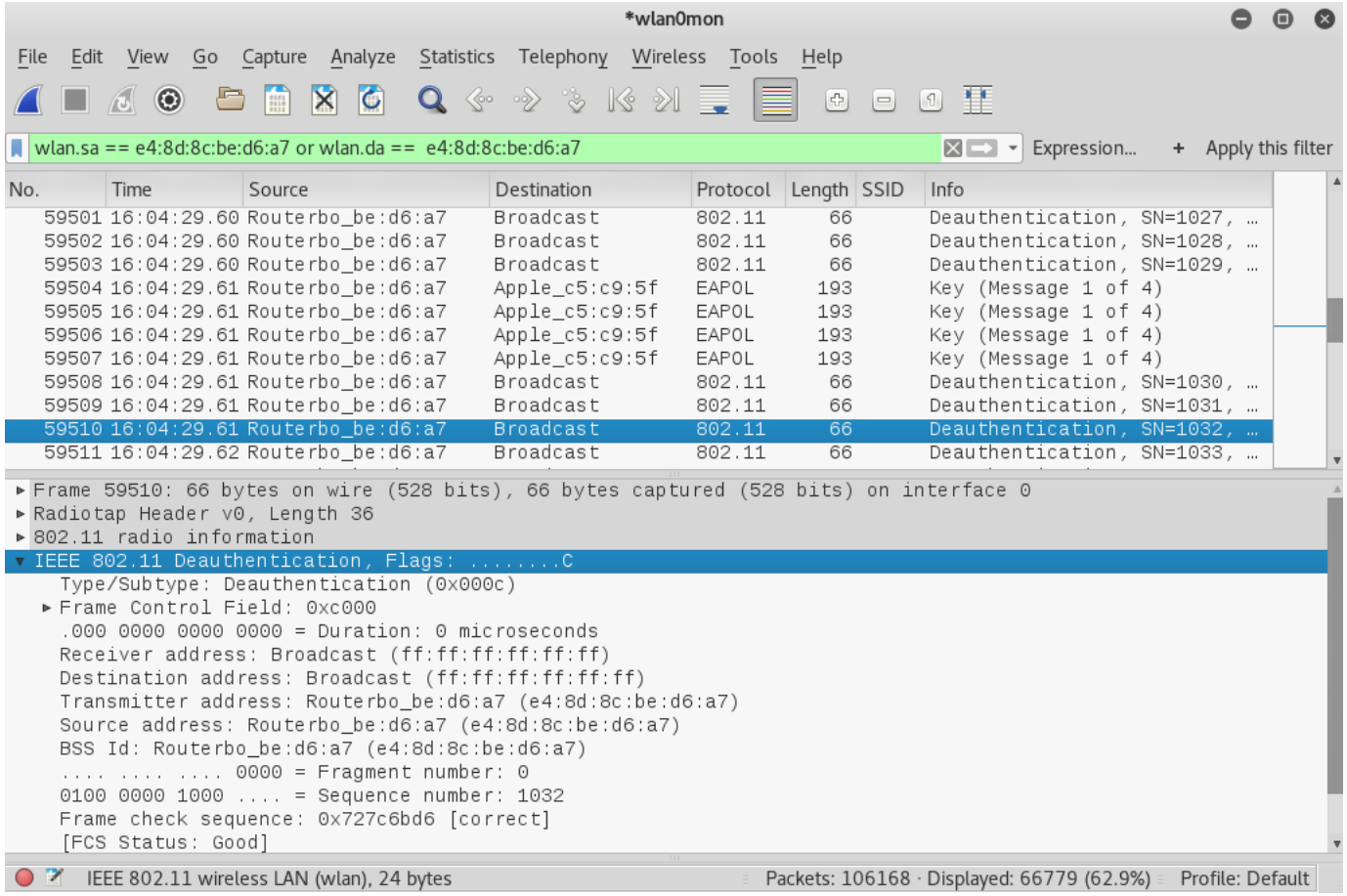
Tabulka 4.4: Blokování

Všechna testovaná zařízení se po přijetí deautentizačního rámce od sítě velmi rychle odpojila. Zařízení se pak pokoušela o opětovné připojení. Tento proces byl vždy v nějaké fázi přerušen dalším deautentizačním rámcem. Pokusy o opětovné připojení bylo možné sledovat i v záznamu informací na přípojném bodu:

```
15:23:23 wireless DC:04:46:00:00:00:11@wlan1: reassociating
15:23:23 wireless DC:04:46:00:00:00:11@wlan1: disconnected,
unicast key exchange timeout
15:23:23 wireless DC:04:46:00:00:00:11@wlan1: connected
15:23:23 wireless DC:04:46:00:00:00:1@wlan1: reassociating
15:23:23 wireless DC:04:46:00:00:00:11@wlan1: disconnected, ok
15:23:23 wireless DC:04:46:00:00:00:11@wlan1: connected
15:23:23 wireless DC:04:46:00:00:00:11@wlan1: reassociating
15:23:23 wireless DC:04:46:00:00:00:11@wlan1: disconnected,
extensive data loss
15:23:23 wireless DC:04:46:00:00:00:11@wlan1: connected
15:23:24 wireless DC:04:46:00:00:00:11@wlan1: reassociating
15:23:24 wireless DC:04:46:00:00:00:11@wlan1: disconnected, ok
15:23:24 wireless DC:04:46:00:00:00:11@wlan1: connected
```

Programem Wireshark jsem monitorováním provozu na dalším zařízení zkontroloval, že v síti opravdu dochází k posílání deautentizačních rámců a příslušné adresy jsou nastavené na adresu podvodného AP (obr. 4.6).

Z časových důvodů jsem neměl možnost otestovat větší vzorek klientských zařízení a operačních systémů. Přesto lze vyhodnotit, že implementovaný nástroj efektivně blokuje nepovolený přístupový bod zasíláním deautentizačních rámců, čímž chrání klienty před používáním tohoto podvodného AP.



Obrázek 4.6: Deautentizační rámec v programu Wireshark

Závěr

Cílem této diplomové práce bylo analyzovat metody detekce, lokalizace a blokování zařízení ve WiFi sítích a na základě zjištěných informací navrhnout a implementovat nástroj k tomuto procesu.

Analýza problematiky lokalizace zařízení ve WiFi sítích, včetně existujících řešení, je uvedena v první kapitole. Z této analýzy vyplývá, že pro lokalizaci je nutné monitorováním provozu zachytit informace o komunikujících zařízeních v síti a ty následně využít k výpočtu jejich polohy.

Dostupné metody pro monitorování a lokalizaci WiFi zařízení a koncepce celého nástroje jsou diskutovány v druhé kapitole Návrh. Vyvinutý nástroj je rozdělen na části pro monitorování provozu pomocí běžných WiFi adaptérů, výpočet polohy iterativní minimalizací chyby, ukládání získaných informací a jejich prezentaci. Jednotlivé části spolu komunikují pomocí serveru na zasílání zpráv, čímž je umožněné jejich provozování v distribuovaném systému.

V kapitole Implementace je popsané konkrétní řešení jednotlivých částí výsledného nástroje. Komunikace probíhá posíláním zpráv pomocí protokolu MQTT. K monitorování provozu lze použít běžná WiFi zařízení. Zachycená provozní data jsou prostřednictvím MQTT zpráv předána hlavnímu výpočetnímu programu, který z nich odhaduje polohy detekovaných zařízení a informace o nich zpřístupňuje opět zasíláním MQTT zpráv. Výsledné údaje o zařízeních v síti jsou ukládány do databáze a zpřístupněny pomocí webové stránky s prostorovým grafem a ovládacím rozhraním. Systém umožňuje napojení na další nástroje pomocí MQTT komunikace.

V kapitole testování je implementovaný systém vyzkoušen ve vlastní síti. Systém v testovaném prostředí úspěšně detekoval nepovolená zařízení a pokud bylo zařízení detekováno několika měřícími body, byla kromě informace o detekci zpřístupněna i jeho poloha. Přesnost lokalizace je ovlivněna prostředím a umístěním hledaných zařízení. V testovaném prostředí se průměrná chyba vypočtené pozice pohybovala od 1.5 do 3 metrů. Dále byla testována možnost blokování vybraného přístupového bodu, kdy docházelo úspěšně k odpojování připojených klientů a zamezení jeho dalšího používání.

Výsledkem práce je skupina propojených programů umožňující detekovat zařízení v bezdrátové síti, v případě dostatečného množství informací určit jejich polohu. Pokud jde o přístupový bod, je možné spustit jeho blokování.

Jednotlivé části celého nástroje lze dále zlepšovat. Například pro lepší kontrolu nad zachycenými rámci a posíláním deautentizačních rámců by bylo vhodné napsat vlastní program nad knihovnou s přímým přístupem k zařízení (například libpcap). Rozsáhlejším měřením ve více testovacích lokacích by bylo možné identifikovat příčiny nepřesností a najít cestu, jak automaticky definovat nejen optimální parametr prostředí, ale i jak zvolit délku cyklu hlavního programu, platnost dat, způsob výběru hodnot pro výpočet polohy a další. Nabízí se možnost kombinovat metodu výpočtu polohy s metodou lokalizace pomocí otisků prostředí.

Problém, kterým se zabývá tato práce, je rozsáhlý a vyžaduje souhru všech jeho součástí. Výsledný návrh a implementaci považuji za dobrý základ k dalšímu vylepšování.

Literatura

[1] The Institute of Electrical and Electronics Engineers, I.: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Březen 2012. Dostupné z: http://standards.ieee.org/getieee802/download/802.11-2012.pdf

[2] Recommendation, I.: 200 (1994)| ISO/IEC 7498-1: 1994. *Information technology–Open systems interconnection–Basic reference model: The basic model*, 1994.

[3] Wood, R.: Karma - DigiNinja. Dostupné z: https://digi.ninja/karma/

[4] xtr4nge: FruityWifi - Wireless Network Auditing Platform. Dostupné z: http://www.fruitywifi.com/index_eng.html

[5] Hak5: The WiFi Pineapple: WIRELESS AUDITING PLATFORM. Dostupné z: https://www.wifipineapple.com

[6] Cassola, A.; Robertson, W. K.; Kirda, E.; aj.: A Practical, Targeted, and Stealthy Attack Against WPA Enterprise Authentication. In *NDSS*, 2013.

[7] Brenza, S.; Pawlowski, A.; Pöpper, C.: A Practical Investigation of Identity Theft Vulnerabilities in Eduroam. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '15, New York, NY, USA: ACM, 2015, ISBN 978-1-4503-3623-9, s. 14:1–14:11, doi:10.1145/2766498.2766512. Dostupné z: http://doi.acm.org/10.1145/2766498.2766512

[8] Song, Y.; Yang, C.; Gu, G.: *Who is peeping at your passwords at Starbucks? To catch an evil twin access point*. ISBN 978-1-4244-7500-1 978-1-4244-7501-8 978-1-4244-7499-8. Dostupné z: https://www.infona.pl//resource/bwmeta1.element.ieee-art-000005544302

[9] Mónica, D.; Ribeiro, C.: WiFiHop - Mitigating the Evil Twin Attack Through Multi-hop Detection. In *Proceedings of the 16th European Conference on Research in Computer Security*, ESORICS'11, Berlin, Heidelberg: Springer-Verlag, 2011, ISBN 978-3-642-23821-5, s. 21–39. Dostupné z: `http://dl.acm.org/citation.cfm?id=2041225.2041228`

[10] Mao, G.: *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking: Monitoring and Surveillance Techniques for Target Tracking*. IGI Global, 2009.

[11] Liu, H.; Darabi, H.; Banerjee, P.; aj.: Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, ročník 37, č. 6, 2007: s. 1067–1080.

[12] Friis, H. T.: A note on a simple transmission formula. *Proceedings of the IRE*, ročník 34, č. 5, 1946: s. 254–256.

[13] Rappaport, T.: *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall PTR, druhé vydání, 2001, ISBN 978-0-13-042232-3.

[14] Kang, J.; Kim, D.; Kim, Y.: RSS Self-calibration Protocol for WSN Localization. In *2007 2nd International Symposium on Wireless Pervasive Computing*, Únor 2007, doi:10.1109/ISWPC.2007.342597.

[15] WLAN (IEEE 802.11) capture setup. Červen 2017. Dostupné z: `https://wiki.wireshark.org/CaptureSetup/WLAN`

[16] Dillinger, M.; Madani, K.; Alonistioti, N.: *Software defined radio: Architectures, systems and functions*. John Wiley & Sons, 2005.

[17] Receiving and acquiring GPS positions with an RTL-SDR dongle and GPS antenna. Prosinec 2015. Dostupné z: `http://www.rtl-sdr.com/receiving-gps-with-an-rtl-sdr-dongle-and-gps-antenna/`

[18] Comparison of open-source wireless drivers. Červen 2017, page Version ID: 784443858. Dostupné z: `https://en.wikipedia.org/w/index.php?title=Comparison_of_open-source_wireless_drivers&oldid=784443858`

[19] Zirari, S.; Canalda, P.; Spies, F.: WiFi GPS based combined positioning algorithm. In *Wireless communications, networking and information security (WCNIS), 2010 IEEE international conference on*, IEEE, 2010, s. 684–688.

[20] Yang, J.; Chen, Y.: Indoor localization using improved RSS-based lateration methods. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, IEEE, 2009, s. 1–6.

[21] Gill, P. E.; Murray, W.; Wright, M. H.: *Practical optimization*. Academic Press, 1981, ISBN 978-0-12-283950-4.

[22] WLAN 802.11w Technology. Červen 2014. Dostupné z: http://h20566.www2.hpe.com/hpsc/doc/public/display?sp4ts.oid=5377832&docLocale=en_US&docId=emr_na-c04498304

[23] Aireplay-ng. Červen 2017. Dostupné z: https://www.aircrack-ng.org/doku.php?id=aireplay-ng

[24] CNN, B. K. H.: FCC fines Marriott $600,000 for blocking guests' Wi-Fi - CNN.com. Dostupné z: http://www.cnn.com/2014/10/03/travel/marriott-fcc-wi-fi-fine/index.html

[25] Cisco Unified Wireless Network. Červen 2013. Dostupné z: http://www.cisco.com/c/en_vn/solutions/unified-wireless-network.html

[26] Vanhoef, M.; Matte, C.; Cunche, M.; aj.: Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. ACM Press, 2016, ISBN 978-1-4503-4233-9, s. 413–424, doi:10.1145/2897845.2897883. Dostupné z: http://dl.acm.org/citation.cfm?doid=2897845.2897883

[27] Cisco 2700 Series Wireless Location Appliance Deployment Guide. Červen 2007. Dostupné z: http://www.cisco.com/c/en/us/td/docs/wireless/technology/location/deployment/guide/depgd.html

[28] Cisco Hyperlocation Module with Advanced Security. Leden 2017. Dostupné z: http://www.cisco.com/c/en/us/products/collateral/interfaces-modules/aironet-hyperlocation-module-advanced-security/datasheet-c78-734901.pdf

[29] Rogue Management in a Unified Wireless Network using v7.4. Dostupné z: http://www.cisco.com/c/en/us/td/docs/wireless/technology/roguedetection_deploy/Rogue_Detection.html

[30] CISCO GPL 2017. Červen 2016. Dostupné z: http://itprice.com/cisco-gpl/WIPS

[31] OpenWISP Project - Homepage. Červen 2017. Dostupné z: http://openwisp.org

[32] Kotaru, M.; Joshi, K.; Bharadia, D.; aj.: Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM Computer Communication Review*, ročník 45, ACM, 2015, s. 269–282.

[33] Xiong, J.: *Pushing the Limits of Indoor Localization in Today's Wi-Fi Networks*. Doctoral, UCL (University College London), Září 2015. Dostupné z: http://discovery.ucl.ac.uk/1470731/

[34] Xiong, J.; Jamieson, K.: ArrayTrack: A Fine-Grained Indoor Location System. In *NSDI*, 2013, s. 71–84.

[35] Sen, S.; Choudhury, R. R.; Nelakuditi, S.: SpinLoc: Spin once to know your location. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, ACM, 2012, str. 12.

[36] Vasisht, D.; Kumar, S.; Katabi, D.: Decimeter-level localization with a single wifi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, USENIX Association, 2016, s. 165–178.

[37] GNU Radio - Free & Open-Source Toolkit for Software Radio. Dostupné z: https://www.gnuradio.org/

[38] Wime Project. Dostupné z: https://www.wime-project.net/

[39] Existing Linux Wireless drivers. Dostupné z: https://wireless.wiki.kernel.org/en/users/drivers

[40] Wi-Fi Location-Based Services 4.1 Design Guide - Installation and Configuration [Design Zone for Mobility]. Dostupné z: http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/WiFiLBS-DG/wifich4.html

[41] A Light, R.: Mosquitto: server and client implementation of the MQTT protocol. *The Journal of Open Source Software*, ročník 2, č. 13, Květen 2017, ISSN 2475-9066, doi:10.21105/joss.00265. Dostupné z: http://joss.theoj.org/papers/10.21105/joss.00265

Příloha A

Seznam použitých zkratek

AAA Authentication, authorization and accounting

AoA Angle of arrival

ACL Access control list

AP Access point

API Application programming interface

BSSID Basic service set identifier

CSMA-CA Carrier-sense multiple access with collision avoidance

DoS Denial of service

EAP Extensible authentication protocol

IDS Intrusion detection system

IPS Intrusion prevention System

LAN Local area network

MAC Media access control

MQTT Message queue telemetry transport

QoS Quality of service

RADIUS Remote authentication dial in user service

TACACS Terminal access controller access-control system

SDR Software-defined radio

SSID Service set identifier

ToA Time of arrival

ToF Time of flight

TDoA Time difference of arrival

TLS Transport layer security

WIDS Wireless intrusion detection system

WIPS Wireless intrusion prevention system

Příloha **B**

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `impl` ..... zdrojové kódy implementace
    - `loc-dashboard` ..... definice panelu do softwaru Grafana
    - `loc-db` ..... zdrojové kódy programu loc-db
    - `loc-viz` ..... zdrojové kódy programu loc-db
    - `loc.py` ..... zdrojové kódy programu loc.py
    - `loc.sh` ..... zdrojové kódy programu loc.sh
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF