VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY
DEPARTMENT OF CONTROL AND INSTRUMENTATION

MONITORING A ANALÝZA PROVOZU SÍTÍ WI-FI A BLUETOOTH
WIFI AND BLUETOOTH MONITOR TOOL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Samuel Petráš**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Soběslav Valach**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Samuel Petráš ***ID:*** 203317
Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Monitoring a analýza provozu sítí Wi-Fi a Bluetooth

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je návrh analyzátoru monitoringu zařízení pracujících na frekvenci 2,4GHz a 5GHz. Především by se mělo jednat o zařízení WiFi a Bluethooth. Ze získaných dat bude třeba vytěžit informace o pohybu zařízení, počtu průchodu monitorovanými stanovišti a odhadem počtu osob, které prošly daným stanovištěm za jednotku času. Cílová platforma by měla využívat procesor s jádrem ARM (např. modul Raspberry Pi).

1) Prostudujte principy a techniky monitorování a sledování sítí
2) Vytvořte aplikaci pro monitorování Wi-Fi a BT provozu, ověřte její funkci
3) Navrhněte vhodné uspořádání monitorovacích stanovišť v 2D prostoru
4) Navrhněte vhodný komunikační protokol pro předávání dat nadřazenému systému
5) Zpracujete data získaná data a vizualizujte vhodným způsobem
6) Analyzujte spolehlivost a funkčnost navrženého řešení

DOPORUČENÁ LITERATURA:

https://www.raspberrypi.org/

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 8.6.2020

Vedoucí práce: Ing. Soběslav Valach

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

Abstrakt

Táto bakalárska práca sa zaoberá monitoringom okolitých zariadení používajúcich bezdrôtové komunikačné rozhrania Wi-Fi a Bluetooth. V teoretickej časti práce sú tieto štandardy predstavené spolu s možnosťami ich monitoringu a komplikáciami s tým spojenými. Vlastné riešenie predstavuje implementáciu a automatizáciu takéhoto monitoringu na platforme Raspberry Pi, čo zahŕňa zachytávanie dát, ukladanie dát v MySQL databáze, analýzu dát a prezentáciu výsledkov.

Kľúčové slová

Wi-Fi, Bluetooth, pásmo 2,4 GHz, pásmo 5 GHz, monitoring, Raspberry Pi

Abstract

This bachelor thesis is concerned with the monitoring of surrounding devices using wireless communication interfaces Wi-Fi and Bluetooth. In the theoretical section, these standards are presented together with the possibilities of monitoring of said wireless standards and complications associated with it. The proposed solution outlines implementation and automation of such monitoring on a Raspberry Pi platform, including data capture, data storage in a MySQL database, data analysis and representation of results.

Keywords

Wi-Fi, Bluetooth, 2.4 GHz band, 5 GHz band, monitoring, Raspberry Pi

Bibliografická citácia:

PETRÁŠ, Samuel. Monitoring a analýza provozu sítí Wi-Fi a Bluetooth. Brno, 2020. Dostupné také z: https://www.vutbr.cz/studenti/zav-prace/detail/126966. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Soběslav Valach.

Prehlásenie

Prehlasujem, že svoju bakalársku prácu na tému Monitoring a analýza provozu sítí Wi-Fi a Bluetooth som vypracoval samostatne pod vedením vedúceho bakalárskej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, predovšetkým som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Zb., vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníku č. 40/2009 Zb. Českej Republiky

V Brne dňa: **8. júna 2020** ……

podpis autora

Poďakovanie

Ďakujem vedúcemu bakalárskej práce Ing. Soběslavovi Valachovi za účinnú metodickú, pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovaní mojej bakalárskej práce.

V Brne dňa: **8. júna 2020** ……

podpis autora

Obsah

Zoznam skratiek

AP	…	Access Point
BD_ADDR	…	Bluetooth Device Address
CID	…	Company Identifier
dBm	…	Decibel miliwatt
FCC	…	Federal Communications Commission
GHz	…	Gigahertz
IEEE	…	Institute of Electrical and Electronics Engineers
IoT	…	Internet of Things
ISM	…	Industrial, Scientific, and Medical
MAC	…	Medium Access Control
MB	…	Megabajt
NIC	…	Network Interface Controller
OS	…	Operačný Systém
OUI	…	Organizationally Unique Identifier
PAN	…	Personal Area Network
PNL	…	Preferred Network List
SSH	…	Secure Shell
SSID	…	Service Set Identifier
U-NII	…	Unlicensed National Information Infrastructure
VM	…	Virtual Machine

Zoznam obrázkov

Zoznam tabuliek

1 ÚVOD

Bakalárska práca sa zaoberá monitoringom bezdrôtových komunikačných protokolov Wi-Fi a Bluetooth a následným spracovaním zozbieraných dát. Vďaka masívnemu rozšíreniu spomenutých protokolov predstavuje takýto monitoring atraktívny a v praxi používaný zdroj dát s mnohými využitiami.

V teoretickom úvode práce sú opísané bezdrôtové pásma a štandardy, ktoré sú v tejto práci využité v rámci monitoringu, a samotná problematika monitoringu. Ďalej je v teoretickom úvode uvedený príklad využitia z praxe, a tiež pravidlá GDPR, ktoré je pritom potrebné v rámci Európskej únie dodržiavať.

Tretia kapitola práce predstavuje vlastnú implementáciu monitoringu od zachytávania dát až po ich spracovanie a zobrazenie výsledkov v podobe grafov, tabuliek a doplňujúceho textu. Riešenie je komplexné, zahŕňa niekoľko programov, skriptov a algoritmov na spracovanie dát, a preto sa významná časť textu práce venuje popisu funkcií, ovládania a spolupráce jednotlivých častí. Riešenie vzniklo na platforme Raspberry Pi a dôvody tohto výberu sú opísané v úvode tretej kapitoly. Využitie riešenia práce v praxi je diskutované na konci tretej kapitoly.

Vyhodnotenie dát z monitoringu, ktorý sa uskutočnil v dvoch obdobiach a celkovo trval 45 dní, je uvedené v štvrtej kapitole. Okrem ukážky výsledkov v podobe, ako ich ponúka riešenie práce, sa v tejto kapitole nachádza aj podrobnejšia analýza dát z monitoringu. Keďže monitoring prebiehal počas pandémie koronavírusu, trendom v dátach v súvislosti s touto pandémiou je venovaná jedna podkapitola.

Bash skripty použité v tejto práci boli prevzaté z riešenia semestrálnej práce na rovnakú tému [20]. Bolo potrebné implementovať niekoľko vylepšení, najmä podporu monitorovania Wi-Fi zariadení pracujúcich v pásme 5 GHz.

2 TEORETICKÝ ÚVOD

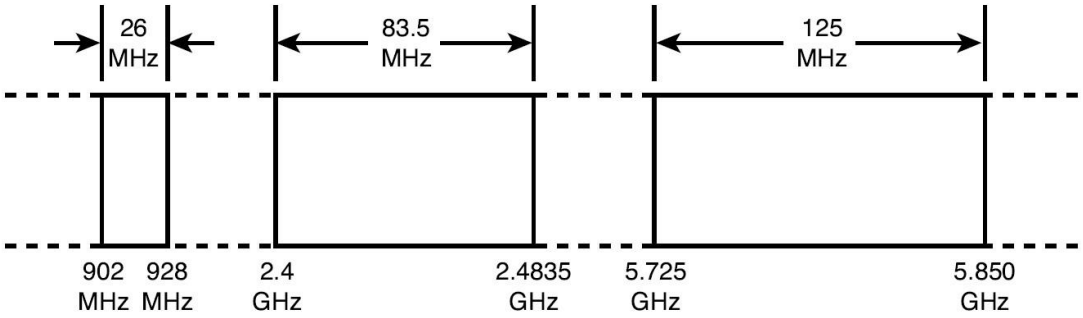
Teoretické predpoklady pre túto prácu sú rozobrané v jednotlivých podkapitolách. Najskôr sú predstavené jednotlivé bezdrôtové komunikačné a štandardy a pásma, ktorými sa práca zaoberá, a nakoniec je priblížená problematika monitoringu týchto štandardov.

Štandardov pracujúcich v pásme 2,4 GHz a 5 GHz je oveľa viac ako táto práca pokrýva. Avšak pre monitoring sú práve nižšie rozoberané štandardy najzaujímavejšie z dôvodu ich masívneho rozšírenia. Tak isto existujú aj iné komunikačné pásma, ktorým sa táto práca nevenuje.

Jednotlivé štandardy sa líšia napríklad tým, na akú vzdialenosť dokážu komunikovať, koľko energie na to potrebujú a počtom zariadení, s ktorými dokážu naraz pracovať. Štandard Bluetooth je určený na komunikáciu na krátke vzdialenosti medzi zariadeniami, typicky používanými jedným užívateľom (bezdrôtová myš, slúchadlá, produkty IoT a iné). Naopak Wi-Fi sa typicky využíva na komunikáciu viacerých zariadení s jedným prístupovým bodom, za účelom prístupu na internet. [24]

2.1 Pásmo 2,4 GHz

Jedno z najpoužívanejších pásiem spadajúcich pod ISM (Industrial, Scientific and Medical) je práve pásmo 2,4 GHz. Pôvodne sa používalo napríklad pre vojenské radary, ale aj iné prístroje, ktoré vysielali rádiové vlny, ako napríklad priemyselné ohrievače a mikrovlnné rúry, a nie pre komunikáciu. Využitie pre komunikáciu povolila komisia Federal Communications Commission (FCC) až v roku 1985. [23]



Obr. 1 Pásmo ISM [43]

2.2 Pásmo 5 GHz

Okrem ISM pásma 5,725 – 5,850 GHz sa v komunikačných štandardoch využíva pásmo Unlicensed National Information Infrastructure (U-NII). Pásmo U-NII sa delí na niekoľko menších pásiem a štandardy Wi-Fi z toho využívajú rozsah 5,150 – 5,850 GHz. Z toho vyplýva, že určitá časť pásiem ISM a U-NII sa prekrýva.

Výhodou pásma 5GHz je vyššia prenosová rýchlosť (*data rate*) a menšia obsadenosť, čo znamená aj menej rušenia. Naopak nevýhodou je menší dosah a väčšie tlmenie signálu pri prechádzaní cez prekážky, v porovnaní s nižšími frekvenciami. [43]

Nevýhoda vyššieho tlmenia prekážkami a kratšieho dosahu môže byť pri monitoringu naopak výhodou, pokiaľ nás zaujímajú signály len z určitej malej uzatvorenej oblasti. Tieto rozdiely oproti pásmu 2,4 GHz je pri monitoringu potrebné brať do úvahy.

2.3 Wi-Fi

Hlavnou motiváciou vývoja je mobilita zariadení, ktorú Wi-Fi ponúka. Umožňuje prístup na internet bez priameho káblového pripojenia. Najčastejšie je využitá ako posledný článok reťaze medzi existujúcou káblovou sieťou a užívateľmi. Rozšírenie Wi-Fi do podoby, ako ju vidíme dnes, je možné vďaka štandardizácii, ktorá zabezpečuje kompatibilitu medzi rôznymi výrobcami. [25]

Obr. 2 Logo Wi-Fi [26]

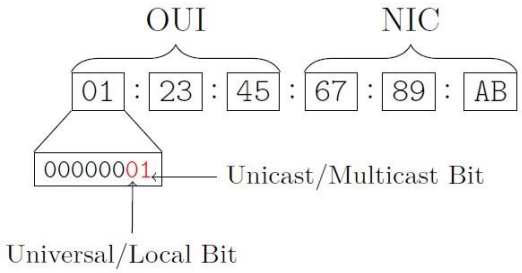
Wi-Fi je založená na štandarde IEEE 802.11, ktorý bol ratifikovaný v roku 1997 inštitútom Institute of Electrical and Electronics Engineers (IEEE). Tak ako všetky IEEE 802 štandardy, aj tento sa sústredí na dve najspodnejšie vrstvy modelu OSI. [25]

Frekvenčné pásmo 2,4 GHz, v ktorom Wi-Fi zariadenia založené na prvom štandarde IEEE 802.11 pracujú, sa delí na 14 kanálov, širokých vždy 22 MHz. Susedné kanále sa z časti prekrývajú, pričom sa dajú vybrať tri bez akéhokoľvek prekrytia. Štandardy pracujúce vo frekvenčnom pásme 5 GHz ponúkajú väčší počet neprekrývajúcich sa kanálov, keďže je toto pásmo širšie. Vysielanie prebieha vždy na jednom kanáli bez „skákania“ medzi nimi. Ako je možné vidieť v tabuľke *Tab. 1 Porovnanie štandardov IEEE 802.11 [28]*, šírka pásma sa pre jednotlivé vylepšenia štandardu líši. [25][43]

MAC adresa

Zariadenia spĺňajúce jeden zo štandardov IEEE 802, teda aj 802.11, sú identifikované globálne unikátnou MAC adresou. Táto adresa je trvalo priradená výrobcom. Každé zariadenie v lokálnej sieti je podľa nej možné adresovať. Skladá sa z 48 bitov a najčastejšie sa zapisuje v podobe 12 miestneho hexadecimálneho čísla. Prvých 6 čísel sa nazýva Company Identifier (CID) ale aj Organizationally Unique Identifier (OUI) a identifikujú výrobcu, pričom rozdelenie týchto adries zabezpečuje IEEE. Zvyšných 6 čísel priraďuje sám výrobca sieťovej karty (NIC). [12][25][27]

Moderné zariadenia často okrem globálne unikátnej MAC adresy používajú aj lokálne priradenú adresu. Rozlišujú sa bitom *universal/local* v OUI časti adresy (1 – adresa je lokálna; 0 – adresa je globálne unikátna). Nie je zaručené, že lokálne priradená adresa bude aj globálne unikátna, a často sa používa len dočasne. Čo je však pre túto prácu najpodstatnejšie, lokálne priradené adresy sa používajú aj ako bezpečnostný prvok – randomizácia MAC adresy. Výrobca zariadenia si môže kúpiť prvé tri byty adresy (OUI) s tým, že bude lokálny bit nastavený. Tieto adresy nebudú priradené sieťovým kartám pri výrobe, ale budú použité iba na randomizáciu MAC adries. Viac informácií sa nachádza v kapitole *2.5.2 Komplikácie*. [27]



Obr. 3 Štruktúra MAC adresy [27]

Vylepšenia štandardu IEEE 802.11

Od prvého vydania štandardu IEEE 802.11 v roku 1997 už vyšlo niekoľko jeho vylepšení. V nasledujúcej tabuľke sú uvedené technické špecifikácie štandardov IEEE 802.11*a,b,g,n,ac*.

Tab. 1 Porovnanie štandardov IEEE 802.11 [28]

Štandard 802.11	***Dátum vydania***	***Frekvenčné pásmo [GHz]***	***Šírka kanálu [MHz]***	***Rýchlosť prenosu (Rýchlosť MIMO prenosu) [min-max Mbps]***	***Počet súčasných komunikácií (MIMO)***
802.11	1997	2,4	22	1 - 2	1
a	1999	3,7 a 5	20	6 - 54	1
b	1999	2,4	22	1 - 11	1
g	2003	2,4	20	6 - 54	1
n	2009	2,4 a 5	20	7.2 - 72.2 (6.5 - 65)	4
			40	15 - 150 (13.5 - 135)	
ac	2013	5	20	7.2 - 96.3 (6.5 - 86.7)	8
			40	15 – 200 (13.5 - 180)	
			80	32.5 - 433.3 (29.2 - 390)	
			160	65 - 866.7 (58.5 - 780)	

Režim sieťovej karty

Sieťové karty rozlišujú štyri prevádzkové režimy [29]:

1. **Master mode**

Používa sa na vytváranie prístupového bodu. Sieťová karta vytvorí sieť s určitým menom (SSID) a kanálom. V tomto režime spravuje celú komunikáciu odohrávajúcu sa na danej sieti a overuje klientov, ktorý sa chcú pripojiť. Môže komunikovať výhradne len so zariadeniami v klientskom režime.

2. **Managed mode (klient)**

Dokáže sa pripájať na siete vytvorené nadriadeným zariadením (v režime *master mode*), pokiaľ prejde overením. Automaticky prispôsobuje svoj kanál. Na označenie klienta, ktorý je pripojený k prístupovému bodu, sa používa výraz *associated*.

3. **Ad-hoc mode**

Vytvára sieť v ktorej sa môže spájať každý s každým v dosahu a na rovnakom kanáli. V takejto sieti nemajú zariadenia rozdelené role na *master* a *slave*.

4. **Monitor mode**

Je využívaný monitorovacími programami na pasívne zachytávanie všetkého okolitého prenosu na danom kanále. V tomto režime sa nevysielajú žiadne dáta. Nevyužíva sa v bežnej komunikácii.

2.4 Bluetooth

Bluetooth vyvinula spoločnosť Ericsson Mobile Platforms v roku 1994 a neskôr bol v roku 1999 formalizovaný skupinou Bluetooth Special Interest Group, ktorá ho vyvíja dodnes. Vznikol za účelom prenosu malých objemov dát na krátku vzdialenosť pri nízkej spotrebe energie. Tieto parametre závisia od triedy zariadenia. Prvé verzie boli ratifikované ako štandard IEEE 802.15.1, pre súčasné to už neplatí. [24][30][31]

Obr. 4 Logo Bluetooth [33]

Pásmo, v ktorom komunikujú Bluetooth zariadenia sa začína frekvenciou 2,4 GHz a končí 2,4835 GHz. Delí sa na 79 kanálov (v niektorých krajinách iba 23). Počas komunikácie „skáče“ medzi týmito kanálmi pseudonáhodne, aby sa zvýšila bezpečnosť a spoľahlivosť pri rušení. [31]

Pri komunikácií tvoria zariadenia *ad-hoc* PAN nazývaný *piconet,* v ktorom majú zariadenia rozdelené role *master* a *slave*, pričom *master* môže komunikovať až so

siedmimi zariadeniami naraz, ale *slave* zariadenia navzájom komunikovať nemôžu. Tieto role sú čisto softwarové, každé zariadenie môže byť *master* aj *slave*. [31][32]

Tab. 2 Triedy Bluetooth vysielačov [24]

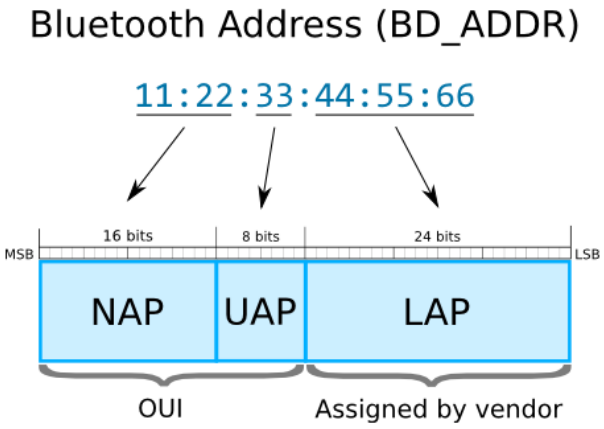
Trieda	*Maximálny povolený výkon [mW]*	*Dosah [m]*
1	100	~100
2	2.5	~10
3	1	~1

BD_ADDR adresa

Tak ako všetky zariadenia postavené na štandarde IEEE 802, každý Bluetooth vysielač má priradenú globálne unikátnu 48-bitovú adresu, v tomto prípade nazývanú Bluetooth Device Address (BD_ADDR), ktorá pozostáva z troch častí:

- LAP – spodná časť adresy, 24 bitov (lower address part)
- UAP – horná časť adresy, 8 bitov (upper address part)
- NAP – nevýznamná časť adresy, 16 bitov (nonsignificant address part)

Používa sa na vytvorenie spojenia medzi zariadeniami a na určenie vzorca, podľa ktorého sa má pri komunikácií „skákať“ medzi kanálmi. Rovnako ako v prípade Wi-Fi zariadení, zvyčajne sa zobrazuje v hexadecimálnom tvare a horných 24 bitov (NAP a UAP) tvorí identifikátor výrobcu OUI. Spodných 24 bitov určuje výrobca daného zariadenia. [12][25][31][34]



Obr. 5 Štruktúra BD_ADDR adresy [34]

2.5 Monitoring

Pojmom monitoring sa myslí zber určitých dát za cieľom získania informácií. Zdrojom dát v tejto práci sú prenosné zariadenia s bezdrôtovými komunikačnými rozhraniami, najmä smartfóny, využívajúce Wi-Fi a Bluetooth. Dáta z neprenosných zariadení, ako sú napríklad tlačiarne, sú nežiadúce, a v algoritmoch na spracovanie dát bolo implementované opatrenie na ich filtrovanie.

2.5.1 Programy

Pre operačné systémy s linuxovým jadrom existuje obrovské množstvo rôznych programov a nástrojov na monitorovanie, analýzu alebo aj prelomenie bezdrôtových sietí a ich bezpečnosti [41][42]. V tejto kapitole je uvedených niekoľko najznámejších.

Aircrack-ng je kompletná sada nástrojov na posudzovanie bezpečnosti Wi-Fi sietí. Všetky nástroje sú prístupné cez príkazový riadok, čo umožňuje úplnú automatizáciu pomocou skriptov. Je podporovaný širokým výberom operačných systémov. Zameriava sa na rôzne oblasti zabezpečenia **Wi-Fi**: monitorovanie, útok, testovanie a prelomenie šifry. Z oblasti monitorovania ponúka zachytávanie paketov a ich export do textových súborov pre ďalšie spracovanie. [47]

Kismet je detektor bezdrôtových zariadení a sietí, a ich bezpečnostných narušení. Pracuje s rozhraniami **Wi-Fi**, **Bluetooth** a iným špecializovaným hardvérom na zachytávanie paketov. Podporuje operačné systémy Linux, OSX a do určitej miery aj Windows 10. Na Linuxe funguje s väčšinou Wi-Fi a Bluetooth rozhraní. Ponúka kompletný monitorovací reťazec od zachytenia paketov až po zobrazenie výsledkov na webe. [48]

Wireshark je popredný a široko používaný analyzátor sieťových protokolov. Dokáže zachytávať prenos Ethernetu, **Wi-Fi**, **Bluetooth** a mnoho iných. Umožňuje nahliadnuť na to, čo sa deje v sieti na mikroskopickej úrovni. Zachytené dáta je možné podrobne skúmať neskôr. Ponúka prehliadač jednotlivých paketov. Pracuje v príkazovom riadku ale aj ako GUI rozhranie. [49]

Bluelog je linuxový **Bluetooth** skener, ktorý dokáže bežať na pozadí. Je určený na dlhodobý zber dát za účelom zistenia počtu Bluetooth zariadení vo viditeľnom režime (discoverable) v okolí. Funguje len v príkazovom riadku a po spustení nevyžaduje žiadnu interakciu s užívateľom. Ponúka konfigurovateľný tvar logových súborov, ale aj automaticky generované webové rozhranie. Jedná sa o relatívne novší nástroj s poslednou aktualizáciou v roku 2017. [5]

Redfang je linuxový nástroj používaný na vyhľadávanie skrytých (*hidden*) **Bluetooth** zariadení. Dokáže to tým, že prejde všetky adresy BD_ADDR v zadanom rozsahu (*brute-force attack*). Táto metóda je veľmi pomalá, za hodinu prejde iba okolo 150 adries, a preto nie je použiteľná pre monitoring prítomnosti osôb v reálnom čase. [50]

2.5.2 Komplikácie

Pri monitoringu osôb využitím technológií Wi-Fi a Bluetooth sa predpokladá, že drvivá väčšina ľudí so sebou nosí smartfón, ktorý ich má zabudované. Vzhľadom na nasledujúce okolnosti sa však vždy bude jednať iba o odhad, nie presné číslo [35]:

- Niektorí ľudia nemusia mať pri sebe zariadenie s Wi-Fi alebo Bluetooth rozhraním
- Niektoré zariadenia nemusia mať zapnuté Wi-Fi alebo Bluetooth rozhranie

- Niektorí ľudia môžu mať pri sebe viac ako jedno zariadenie
- Môžu byť zachytené zariadenia z okolia meranej oblasti
- Dĺžka pobytu v dosahu nemusí byť dosť dlhá na detekciu zariadenia
- Vzhľadom na existenciu viacerých kanálov, ale možnosť monitorovania len jedného v určitý moment, nemusí dôjsť zachyteniu potrebných údajov

Pokrytie kanálov

Problém uvedený v poslednom bode zoznamu je možné minimalizovať použitím viacerých Wi-Fi rozhraní (napríklad USB Wi-Fi adaptérov). Ideálne by bolo použiť jedno rozhranie pre každý kanál so 100% pokrytím. Avšak to nie je vždy možné, napríklad kvôli cene, obmedzenému počtu USB portov alebo obmedzeniam vo výkone zariadenia. Preto sa využíva menší počet adaptérov, pričom každému sa priradí podmnožina kanálov, ktoré má sledovať. [35]

Randomizácia MAC adries

S príchodom náhodne priradených lokálnych MAC adries, ktoré sa menia v čase, sa sledovanie bezdrôtových zariadení stáva obtiažne. Keď sa zariadenie nachádza v stave *disassociated* (nie je pripojené k prístupovému bodu), namiesto globálne unikátnej MAC adresy vysiela lokálnu (náhodnú z určitej množiny). Až keď sa zariadenie pokúša pripojiť, alebo je pripojené k prístupovému bodu (stav *associated*), odhalí svoju pravú MAC adresu. [27]

V čase keď je Wi-Fi rozhranie zariadenia zapnuté, nepretržite vysiela do svojho okolia žiadosti (*probe request*) pre okolité prístupové body, aby sa identifikovali. Nezáleží na tom, či je pripojené k prístupovému bodu alebo nie. Aj keď je pripojené, hľadá známu sieť, ku ktorej sa už aspoň raz pripojilo, so silnejším signálom. Povinnou súčasťou takejto žiadosti je MAC adresa. [27][35]

Pokiaľ by zariadenie vysielalo svoju globálne unikátnu MAC adresu, ako to bolo v minulosti, jeho sledovanie by bolo triviálne ľahké. V boji proti takémuto sledovaniu začali zariadenia s operačným systémom Android a Apple iOS implementovať randomizáciu MAC adries. Neuľahčuje ho ani fakt, že každý výrobca implementuje túto taktiku trochu inak. Pre ilustráciu, zariadenia iPhone s operačným systémom iOS 10.1.1 menia svoju MAC adresu vždy keď nastane jedno z nasledujúcich: (i) zariadenie sa zamkne/odomkne; (ii) Wi-Fi rozhranie je aktivované/deaktivované; (iii) spojenie s prístupovým bodom je nadviazané, alebo bol o to pokus. Tým pádom nie je možné stanoviť presný časový interval medzi zmenou MAC adresy, pretože záleží na interakcií zariadenia s užívateľom. [27][35]

Aj napriek tomu je monitoring stále úspešne realizovateľný. A to práve vďaka žiadostiam *probe request,* ktoré obsahujú SSID zapamätaných sietí, na základe ktorých je možné identifikovať jednotlivých užívateľov. [35]

2.5.3 Výber USB Wi-Fi adaptérov

Väčšina zariadení má zabudovanú sieťovú kartu, ktorá režim *monitor mode* nepodporuje, pretože slúžia iba na pripojenie zariadenia k prístupovým bodom. Jedným z nich je aj Raspberry Pi 3B+, použité v tejto práci. Riešením sú externé USB Wi-Fi adaptéry. Avšak aj pri nich platí, že väčšina tento režim nepodporuje. Je preto potrebný pozorný výber, pri ktorom treba zohľadniť *chipset* (čip ovládajúci adaptér), typ antény a štandardy IEEE 802.11, ktoré adaptér podporuje, teda aj frekvenčné pásma 2,4 GHz a 5 GHz. Zoznamy adaptérov podporujúcich *monitor mode* je možné nájsť na internete. [36]

Obr. 6 USB Wi-Fi adaptér Alfa AWUS036ACH [19]

Nástroj airodump-ng použitý v tejto práci podporuje štandardy 802.11*a,b,g* a je preto potrebné vybrať taký adaptér, ktorý ich tiež podporuje [4]. Nemusia to však byť všetky tri naraz. Ako je možné vidieť v tabuľke *Tab. 1 Porovnanie štandardov IEEE 802.11 [28]*, štandardy 802.11*b,g* pracujú s pásmom 2,4GHz, zatiaľ čo štandard 802.11*a* pracuje s pásmom 5GHz.

Pri vypracovaní tejto práce bol použitý USB Wi-Fi adaptér Alfa AWUS036ACH, pretože podporuje *monitor mode* a zároveň štandardy 802.11*a,b,g,n,ac*. [19][36]

2.5.4 GDPR

MAC adresa je považovaná za súkromný údaj, aj vo forme *hashu*, pretože je pomocou nej možné identifikovať konkrétnu osobu a spadá preto pod GDPR. Až po anonymizácii, ktorá zabezpečí, že nie je dáta nijako možné spätne priradiť ku konkrétnej osobe, je možné s nimi voľne pracovať. Prvým krokom k tomuto cieľu je zber dát o jednotlivých osobách a preto je najskôr potrebné nájsť legálny spôsob, ako tento zber dát uskutočniť.

Monitorovanie v súkromných (komerčných) priestoroch je legálne oveľa viac legitimizované ako monitorovanie verejných priestorov. Každopádne, monitorovaná osoba musí byť na to upozornená, či už pri prihlasovaní na sieť alebo nápisom na vstupe do priestoru. Zároveň musí byť uvedená aj informácia ako sa takémuto monitorovaniu

vyhnúť – napríklad vypnutím Wi-Fi rozhrania. Monitorované osoby by mali byť tiež informované o svojich právach – napríklad o práve na vymazanie zozbieraných dát. Zozbierané dáta musia byť v čo najmenších rozmeroch nutných na splnenie účelu. Taktiež by nemali byť skladované dlhšie ako je nutné.

Právne sa rozlišuje monitorovanie rôznych štatistík, kde identita osoby nie je podstatná, ako napríklad počet návštevníkov obchodu v určitom čase, a monitorovanie konkrétnych osôb, napríklad za účelom cielených ponúk a podobne. [37]

Obr. 7 Upozornenie na Wi-Fi monitoring [37]

Súhlas monitorovanej osoby

Súhlas musí byť dobrovoľný – nie podmienkou. To znamená, že užívateľ nesmie byť nijako znevýhodnený, ani mu nesmie na základe nesúhlasu s monitoringom byť odmietnutý prístup k akejkoľvek službe.

Typická implementácia zberu súhlasu môže prebiehať pri prihlásení na voľnú sieť Wi-Fi alebo ako dobrovoľná súčasť vernostného programu. V takom prípade by bolo možné riešenie predstavené v tejto práci upraviť, aby filtrovalo zachytené MAC adresy podľa databázy užívateľov, ktorý udelili súhlas. Ďalšia legálna možnosť ako získať súhlas je výmena za nejakú konkrétnu odmenu, a poslednou možnosťou je takzvaný legitímny záujem, ktorý prinesie zrejmé výhody, napríklad zvýšenú bezpečnosť alebo zlepšenie poskytovaných služieb. [37]

Príklad z praxe

Príkladom implementácie monitoringu Wi-Fi zariadení v praxi sú holandské železnice Nederlandse Spoorwegen. Monitoring používajú na zvýšenie bezpečnosti v priestoroch a blízkom okolí vlakových staníc ale aj pre zlepšenie ponúkaných komerčných služieb. Na webovej stránke majú detailné informácie o celom procese monitoringu a na vchodoch do staníc sú nalepené upozornenia o monitoringu Wi-Fi zariadení s odkazom na túto webovú stránku. Zozbierané dáta dôkladne anonymizujú a po určitej dobe trvalo odstraňujú. [37][38]

3 VLASTNÉ RIEŠENIE

Riešenie je postavené na platforme Raspberry Pi, ktorá je mimoriadne vhodná na daný účel vzhľadom na svoju veľkosť, mobilitu, možnosť rozširovania schopností použitím USB adaptérov, podporu operačného systému s linuxovým jadrom a v neposlednom rade aj cenu. Operačný systém s linuxovým jadrom umožňuje použitie služieb *systemd service*, časovačov *systemd timer* a *bash* skriptov na automatizáciu celého procesu od spustenia monitoringu až po uloženie dát do MySQL databázy. Spracovanie dát uložených v MySQL databáze prebieha pomocou webového rozhrania.

Bash skripty spájajú existujúce programy do väčšieho, modulárneho celku. Ovládanie celého procesu monitoringu prebieha výlučne pomocou vytvorených *bash* skriptov – užívateľ teda nie je odkázaný na znalosť využitých programov, stačí poznať niekoľko jednoduchých skriptov. Algoritmy webového rozhrania sa nastavujú pomocou HTML formulárov. Mojou snahou bolo implementovať čo najväčší počet nastaviteľných parametrov v oboch prípadoch.

Zároveň však riešenie nie je limitované len na platformu Raspberry Pi. Riešenie je prenosné aj na iné OS s linuxovým jadrom. Počas práce na zadaní som využil virtuálne stroje (VM) s OS Ubuntu a Lubuntu, na ktorých riešenie fungovalo bez problémov. Niektoré OS však môžu vyžadovať miernu úpravu skriptov, keďže nemusia podporovať rovnakú sadu nástrojov.

Riešenie umožňuje aby monitorovanie prebiehalo na jednom samostatnom zariadení, ale aj ako sieť, v ktorej majú zariadenia rozdelené úlohy. Viac o tomto rozdelení je napísané v nasledujúcej podkapitole a v kapitole *3.5 Využitie.*

Obr. 8 Raspberry Pi 3B+

3.1 Rozdelenie zariadení

Ako už bolo spomenuté, monitorovanie môže prebiehať na jednom, alebo viacerých zariadeniach. Riešenie pomocou *bash* skriptov podporuje použitie teoreticky neobmedzeného množstva zariadení s tým, že si rozdelia úlohy. Prvý typ zariadení je monitorovacia jednotka - *node* (ďalej bude monitorovacia jednotka nazývaná už len *node*) a jej úlohou je zachytávať dáta zo svojho okolia a zapisovať ich do logov. Tieto dáta môžu byť následne cez internet presunuté do zariadenia druhého typu, serveru, v ktorom sa uložia do MySQL databázy. Spracovanie dát uložených v MySQL databáze prebieha pomocou webového rozhrania. Tomu sa venuje kapitola *3.4 Webové rozhranie*.

V prípade použitia jedného zariadenia na oba účely prebieha celý proces lokálne a nie je preto potrebné, aby malo zariadenie prístup na internet. Výsledky môžu byť spracované kedykoľvek spätne. Avšak pri použití samostatného zariadenia nie je možné využiť jednu z metód spracovania dát – sledovanie presunu z bodu A do bodu B.

Druhou možnosťou je použitie viacerých spoločne prepojených zariadení. V tomto prípade je vhodné jedno zariadenie použiť ako server a ostatné ako *node*.

Poslednou možnosťou je použitie viacerých samostatných zariadení fungujúcich ako *node* a server zároveň, ktoré nemajú vzájomné prepojenie. Keďže zariadenia nie sú prepojené, pred spracovaním dát v jednotlivých databázach spoločne je nutné tieto databázy manuálne presunúť (klonovať) na jedno zariadenie. Tento proces môže prebiehať cez internet a slúži na to skript `server_clone`, ktorý je bližšie opísaný v kapitole *3.3.2 Ovládanie serveru*.

3.1.1 Monitorovacia jednotka – *node*

Na monitoring okolitých zariadení sú v práci použité programy aircrack-ng (konkrétne nástroj airodump-ng) a Bluelog. Tieto programy poskytujú všetky relevantné informácie o okolitých zariadeniach, ktoré sú na monitoring potrebné a zároveň sú efektívne použiteľné v *bash* skriptoch. Vzhľadom na to, že jediný výstup týchto programov je logový súbor, všetky ostatné súčasti monitoringu, ako ukladanie dát do MySQL databázy a algoritmy na spracovanie dát bolo potrebné implementovať dodatočne, čo mi poskytlo voľnosť v spôsobe implementácie. [4][5]

3.1.2 Server

Na ukladanie logov z monitoringu je použitá MySQL databáza MariaDB. Komunikácia MySQL databázy s webovým rozhraním a aj samotné výpočtové algoritmy prebiehajú pomocou jazyka PHP. Keďže sú algoritmy implementované v jazyku PHP, výpočty prebiehajú na serveri. Ako webový server je použitý software Apache. Pôvodne bol v semestrálnej práci použitý webový server NGINX, ako sa však ukázalo, jeho spolupráca s PHP si vyžadovala manuálne nastavenie konfiguračných súborov a automatizácia *bash*

skriptami by preto bola menej spoľahlivá. Software Apache manuálnu konfiguráciu nevyžaduje.

Výhodou tohto spôsobu spracovania dát je možnosť prístupu k dátam cez internet pomocou webového prehliadača bez nutnosti inštalácie dodatočného softwaru a je tým pádom nezávislé na platforme. [1][2][3]

3.2 Inicializácia zariadení

Každé zariadenie Raspberry Pi použité v monitorovacej sieti je najskôr potrebné inicializovať. Zariadenia potrebujú operačný systém a programy, ktoré zabezpečia požadovanú funkčnosť. Všetky aspekty inicializácie zariadení, či už *node* alebo serveru, sú opísané v nasledujúcich podkapitolách.

3.2.1 Operačný systém

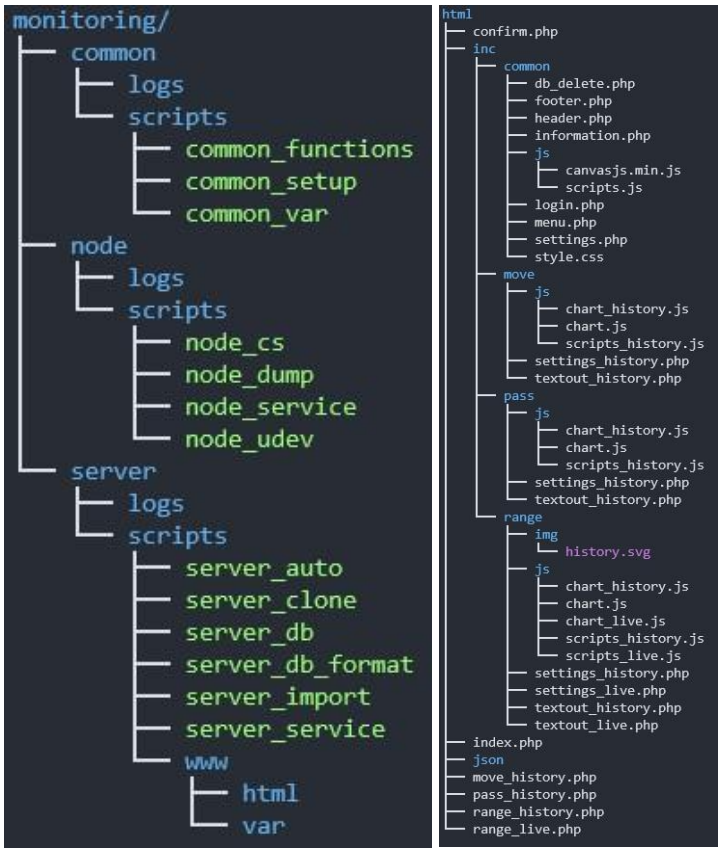
Postup pri inštalácii OS je rovnaký pre *node* aj server. Zvolil som Raspbian, pretože je to oficiálny OS pre Raspberry Pi, podporovaný výrobcom. Za predpokladu výlučného použitia vzdialeného prístupu k zariadeniu nie je potrebné inštalovať *desktop* verziu, a z ponúkaného výberu troch verzií je teda najvhodnejšia verzia *lite*. Táto verzia šetrí miesto na *micro SD* karte. V prípade záujmu o priamy prístup do zariadenia pomocou monitoru, myši a klávesnice je vhodné nainštalovať *desktop* verziu. Jedným z možných spôsobov, ako inštalovať operačný systém na *micro SD* kartu, je použiť oficiálny software Raspberry Pi Imager. [6]

Aby sa hneď po prvom zapnutí bolo možné na zariadenie pripojiť pomocou SSH, po inštalácii OS je nutné do *boot* sekcie *micro SD* karty vložiť prázdny súbor s názvom *ssh* bez prípony. Prvé prihlásenie prebieha pomocou prednastavených prihlasovacích údajov (meno: *pi*, heslo: *raspberry*). Z bezpečnostných dôvodov je odporúčané heslo ihneď zmeniť pomocou príkazu `passwd`. [7][8]

3.2.2 Adresárový strom

Počiatok stromu monitorovacích adresárov sa nachádza v domovskom adresári pre užívateľa *pi* - `/home/pi/monitoring`. Tento definovaný adresárový strom dodržiavajú všetky skripty a zabezpečuje ich spoluprácu lokálne, ale aj v monitorovacej sieti.

V adresári `monitoring/server/www/html` sa nachádza druhý definovaný adresárový strom, použitý pre webové rozhranie. Webový server Apache však k týmto súborom nepristupuje. Súčasťou inicializácie zariadenia fungujúceho ako server je kopírovanie súborov z tohto miesta do adresáru v ktorom ich Apache očakáva - `/var/www/html`. Tak ako všetky ostatné kroky inicializácie, aj tento je súčasťou inicializačného skriptu `common_setup`. Jeho použitiu sa venuje nasledujúca kapitola *3.2.3 Inicializačný skript*.



Obr. 9 Definované adresárové stromy

Do adresáru `/var/www/html/json` sú ukladané výsledky algoritmov webového rozhrania. Aby nedochádzalo ku kolízii názvov súborov, bolo implementované ich skladanie z troch častí:

1. Typ dát
2. Skratka algoritmu
3. Relácia (*session*) webového prehliadača

3.2.3 Inicializačný skript

Po inštalácii operačného systému je nutné zariadenie nastaviť a nainštalovať potrebné programy. Za týmto účelom vznikol skript `common_setup`. Informácie o tom ako sa skript používa je možné získať po jeho zavolaní s argumentom `--help`.

Oproti klonovaniu *micro SD* kariet má tento prístup tú výhodu, že je možné použiť ho kedykoľvek, vzdialene, a nie len bezprostredne po inštalácii OS, pričom sa tým nenaruší iné prípadné využitie zariadenia (nepremaže sa OS). Taktiež veľkosť kopírovaných, prípadne klonovaných súborov je dramaticky nižšia ako veľkosť obrazu celej *micro SD* karty. Nevýhodou je nutnosť prístupu na internet. Preto môže byť klonovanie *micro SD* kariet v určitých situáciách výhodnejšie.

Vzhľadom na to, že musí byť skript použiteľný aj pred tým, ako je na zariadení k dispozícii súbor s globálnymi premennými (`common_var`), má všetky potrebné premenné definované priamo v sebe. V prípade zmien na jednom mieste je preto potrebné vykonať rovnaké zmeny na druhom mieste.

Pri prvej inicializácii zariadenia bez definovaného adresárového stromu je skript možné použiť dvoma spôsobmi popísanými nižšie.

Použitie s inicializačným USB diskom

K zariadeniu sa pripojí USB disk obsahujúci definovaný adresárový strom a z neho sa skopíruje skript `common_setup` do zariadenia, napríklad domovského adresára užívateľa *pi*. Je potrebné, aby bolo možné skript spustiť. To sa dá docieliť nasledujúcimi príkazmi, za predpokladu, že je adresárový súbor na USB disku umiestnený podľa požiadaviek skriptu.

```
cp /media/pi/<názov USB>/monitoring/common/scripts/common_setup /home/pi
chmod +x /home/pi/common_setup
```

Takto by už mal byť skript pripravený na spustenie s názvom USB disku ako jedným z argumentov.

Použitie s klonovaním GitHub repozitára

V tomto prípade stačí pred spustením skriptu `common_setup` naklonovať verejný GitHub repozitár [13] k tejto práci do domovského adresára užívateľa *pi* a premenovať ho z *bachelors-thesis* na *monitoring*. Mali by na to stačiť dva príkazy.

```
git clone https://github.com/Samuell08/bachelors-thesis.git
mv /home/pi/bachelors-thesis /home/pi/monitoring
```

Následne sa skript spustí priamo z adresárového stromu, bez argumentu s názvom USB disku – čím sa preskočí krok s kopírovaním adresára a vykonajú sa všetky ostatné kroky. Priradenie práv na spustenie nie je potrebné, pretože skripty v GitHub repozitári už tieto práva majú.

3.2.3.1 Popis jednotlivých krokov

Všetky kroky čakajú na užívateľove rozhodnutie vo forme odpovede `y/n`. Tieto rozhodnutia je možné automaticky potvrdiť pridaním argumentu `-y`, preskočia sa tým však kroky vyžadujúce vstup od užívateľa – nastavenie hesla, *hostname* a časovej zóny. Pri inštalácii MySQL databázy sa spúšťa služba na zvýšenie bezpečnosti, vyžadujúca vstup od užívateľa, a nie je možné preskočiť ju argumentom `-y`. Rovnako sa nepreskočí ani vytváranie užívateľa MySQL databázy, pretože je to neoddeliteľná súčasť jej inicializácie.

Spoločná časť pre *node* a server

1. **Vytvorenie adresárového stromu (len ak je použitý argument s názvom USB disku)**

Spustí kopírovanie adresárového stromu so všetkými skriptami z USB disku do domovského adresáru užívateľa *pi*. Pre tento krok je nutné, aby bol k zariadeniu pripojený USB disk s definovaným adresárovým stromom. Po skopírovaní sa automaticky prenesie vlastníctvo na užívateľa *pi* a všetkým skriptom sa pridá právo na spustenie.

2. **Nastavenie hesla**

Prebieha pomocou služby `passwd`.

3. **Nastavenie *hostname***

Hostname, teda meno, pod ktorým bude zariadenie vystupovať v sieti sa nastavuje zápisom do systémového súboru `/etc/hostname`. Zmena sa prejaví až po nasledujúcom *boote* zariadenia.

4. **Zmena časovej zóny**

Prebieha pomocou služby `timedatectl`. Najskôr ponúkne výpis zoznamu všetkých časových zón, z ktorých je možné vybrať, a až potom prebehne výber konkrétnej časovej zóny.

5. **Aktualizácia zoznamu programových balíkov**

Stiahne zoznamy najnovších verzií programov. Používa sa na to služba `apt-get update`.

6. **Aktualizácia programových balíkov (len ak prebehol predchádzajúci krok)**

Stiahne a nainštaluje aktualizácie všetkých programov na zariadení pomocou príkazu apt-get upgrade. Tento krok môže v závislosti od rýchlosti internetového pripojenia a počtu aktualizácií trvať dlhšiu dobu.

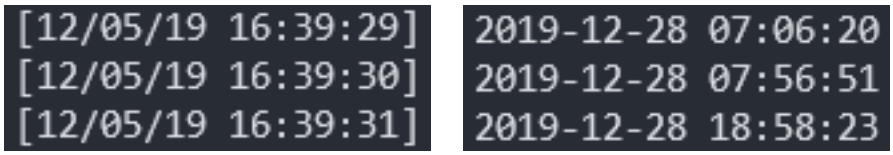
Časť špecifická pre *node*

1. **Inštalácia programu aircrack-ng**

Prebieha pomocou služby `apt-get`.

2. **Inštalácia programu Bluelog**

Najskôr nainštaluje knižnicu *libbluetooth-dev*, ktorá je pre funkčnosť programu Bluelog nevyhnutná [5]. Program Bluelog je dostupný z GitHub repozitára a do zariadenia sa skopíruje pomocou služby `git clone`. Pred kompiláciou je zdrojový kód upravený tak, aby bol výstup do logového súboru vo formáte, ktorý dokáže MySQL databáza spracovať [14].



a) **b)**

Obr. 10 Formát časovej známky programu Bluelog; a) pôvodný b) po úprave

3. **Vytvorenie OUI súboru pre Bluelog**

Zabezpečí databázu výrobcov na základe OUI častí BD_ADDR adresy, ktorú následne využíva program Bluelog. Využíva sa skript zabudovaný v programe Bluelog, ktorý túto databázu stiahne z internetu a upraví do vyžadovaného formátu.

4. **Inštalácia driveru *rtl8812au***

Driver *rtl8812au* je dostupný z GitHub repozitára a do zariadenia sa skopíruje pomocou služby `git clone`. Na inštaláciu používa vlastný skript. Jeho úlohou je zabezpečiť bezproblémové použitie monitorovacieho režimu na adaptéroch s *chipsetom* Realtek 8811, 8812, 8814 a 8821. [16]

Časť špecifická pre server

1. **Inštalácia webového serveru Apache**

Prebieha pomocou služby `apt-get`. Webový server je ihneď pripravený na použitie, nie je potrebný reštart.

2. **Inštalácia PHP (len ak je nainštalovaný webový server Apache)**

Pred inštaláciou sa kontroluje existencia konfiguračného adresára webového serveru Apache. Pokiaľ sa nájde, inštalácia prebieha pomocou služby `apt-get`.

3. **Inštalácia MySQL databázy MariaDB**

Prebieha pomocou služby `apt-get`. Po inštalácii MariaDB sa nainštaluje modul *php-mysql,* ktorý zabezpečí jej spoluprácu s webovým rozhraním prostredníctvom jazyka PHP. Nakoniec sa zavolá služba *mysql_secure_installation*, ktorá vyžaduje vstup od užívateľa a zvýši bezpečnosť nainštalovanej MySQL databázy [15].

4. **Vytvorenie MySQL užívateľa *mon***

Spustí sa iba po zadaní hesla pre *root* užívateľa MySQL databázy. Vyzve užívateľa, aby zadal heslo pre nového užívateľa MySQL databázy, ktorý bude použitý na prístup k dátam z webového rozhrania. Vytvorenie užívateľa potom prebieha priamo posielaním príkazov do MySQL databázy.

5. **Skopírovanie webového rozhrania na webový server**

Kopíruje adresárový strom webového rozhrania z monitorovacieho adresárového stromu do adresáru v ktorom tieto súbory webový server očakáva. Pred kopírovaním pre istotu premaže cieľový adresár. Po kopírovaní prenesie vlastníctvo adresáru na užívateľa *www-data,* čo zabezpečí bezproblémovú spoluprácu s PHP aj JavaScript skriptami. Viac o adresárových stromoch je uvedené v kapitole *3.2.2 Adresárový strom*.

3.2.4 USB Wi-Fi adaptéry

Pred spustením monitorovania je potrebné zabezpečiť predvídateľné správanie, a teda aj kompatibilitu so skriptami, všetkých použitých USB Wi-Fi adaptérov. Po pripojení USB Wi-Fi adaptéru sa mu priradí meno, podľa ktorého je možné naň odkazovať pri spúšťaní skriptov. Problémom je, že tieto mená systém priraďuje na základe poradia v akom boli pripojené. Pokiaľ sa pripojí viacero adaptérov a nedodrží sa vždy rovnaké poradie, tieto mená si zamenia. Taktiež pri *boote* sa môže stať, že sa adaptérom nepriradia mená v rovnakom poradí ako naposledy alebo si dokonca vymenia poradie so zabudovaným Wi-Fi čipom. Takéto nepredvídateľné správanie znemožňuje automatizáciu spúšťania programov pomocou skriptov. Problém sa dá vyriešiť priradením statického mena každému adaptéru na základe jeho MAC adresy pomocou *udev* pravidiel.

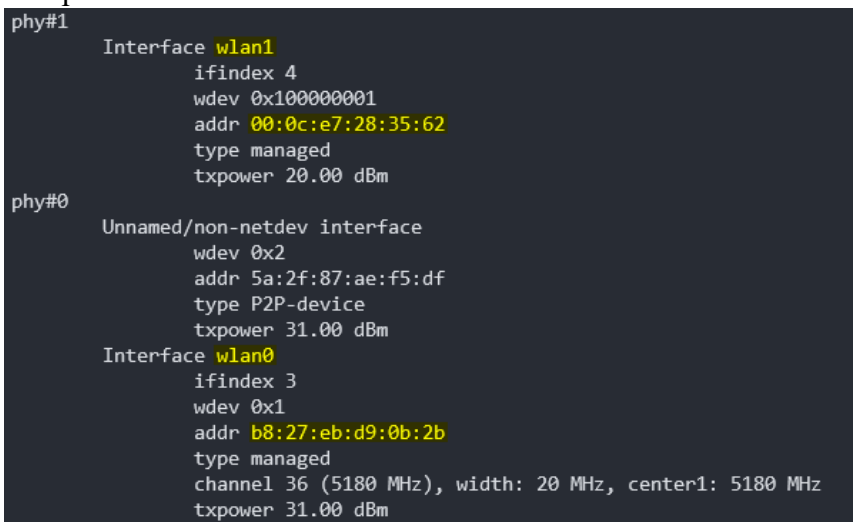
Udev pravidlá slúžia na obsluhu udalostí pri pripojení alebo odpojení zariadení, takže aj USB Wi-Fi adaptéru. Pre tieto udalosti je možné napísať pravidlo vyvolávajúce nejakú akciu. Akcia môže byť pridelenie práv, spustenie programu alebo pridelenie atribútu. Z týchto možností som využil pridelenie atribútu, konkrétne mena. Pravidlá vytvorené užívateľom sa ukladajú do adresáru `/etc/udev/rules.d` s príponou *.rules*. Súbory s pravidlami sa uplatňujú v abecednom poradí, preto sa pre prehľadnosť a zabezpečenie predvídaného správania pred názov píše číslo. V tomto prípade sa používa názov `70-persistent-net.rules`. [18]

V súčasnosti je táto metóda považovaná za zastaralú a je pravdepodobné, že v budúcich vydaniach OS Raspbian bude nahradená novým spôsobom generovania predvídateľných mien nazývaným Predictable Network Interface Names. Aj napriek tomu však ostane možnosť tento nový spôsob deaktivovať a používať starší spôsob, implementovaný v tejto práci. [17]

3.2.4.1 Vzorové vytvorenie *udev* pravidla

Pokiaľ nie je MAC adresa Wi-Fi adaptéru známa predom, veľmi jednoducho sa dá zistiť po pripojení k zariadeniu, napríklad príkazom `iw dev`. Výpis tohto príkazu po pripojení jedného USB Wi-Fi adaptéru je uvedený na obrázku *Obr. 11* nižšie. V obrázku sú žltou farbou zvýraznené mená a MAC adresy adaptéru a zabudovaného Wi-Fi čipu.

Rovnakým príkazom som najskôr zistil MAC adresu zabudovaného Wi-Fi čipu bez pripojeného adaptéru, aby som po jeho pripojení vedel jednoznačne rozoznať, ktorá MAC adresa patrí adaptéru.



Obr. 11 Výpis príkazu `iw dev` pred vytvorením *udev* pravidla

V tomto prípade je MAC adresa adaptéru `00:0c:e7:28:35:62`. Takto získanú MAC adresu som využil na vytvorenie *udev* pravidla skriptom `node_udev` s parametrom `-a`, ktorý pravidlo pridá k ostatným, ak už nejaké existujú.

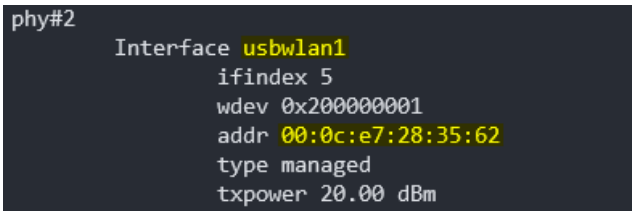
```
sudo ./node_udev -a persistent-net 00:0c:e7:28:35:62 usbwlan1
```

Skript vložil pravidlo do súboru `70-persistent-net.rules`. Jeho obsah je uvedený na obrázku nižšie.

```
SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="00:0c:e7:28:35:62", NAME="usbwlan1"
```

Obr. 12 *Udev* pravidlo vytvorené skriptom `node_udev`

Vytvorené pravidlo zaručí, že sa adaptéru s danou MAC adresou vždy pri zapojení alebo *boote* priradí meno *usbwlan1*. Výstup príkazu `iw dev` po vytvorení pravidla a opätovnom zapojení USB Wi-Fi adaptéru už ukazuje správne meno priradené k danej MAC adrese.



Obr. 13 Výpis príkazu `iw dev` po vytvorení *udev* pravidla

3.3 Ovládanie zariadení

Vytvorené *bash* skripty minimalizujú nutnosť zásahu užívateľa do procesu monitorovania. Skripty proces automatizujú alebo v prípade potreby zásahu interakciu s užívateľom minimalizujú na volanie týchto skriptov. Každý skript má v sebe zabudované

výpisy *help* (parameter `--help`) a *usage* objasňujúce jeho funkciu a parametre, s ktorými je možné daný skript volať, spolu s niekoľkými príkladmi použitia. Mojím cieľom pri tvorení *bash* skriptov nebola len funkčnosť, ale aj čo najväčšia sofistikovanosť a sú preto implementované po vzore profesionálnych skriptov. Aby poskytovali čo najviac bezproblémovú interakciu, boli implementované postupy ako napríklad opakované zadanie hesla pri jeho nastavovaní alebo dialógy, ktoré dávajú užívateľovi na výber, čo sa má stať ďalej a iné.

Pokiaľ bola funkcionalita niektorého skriptu oproti stavu v semestrálnej práci vylepšená, je o tom zmienka v odseku, v ktorom je skript opísaný v nasledujúcich podkapitolách.

3.3.1 Ovládanie monitorovacej jednotky - *node*

Nasledujúce skripty sú používateľovi dostupné v adresári `node/scripts` vnútri adresárového stromu.

Skript `node_udev`

Použitie tohto skriptu je opísané v kapitole *3.2.4 USB Wi-Fi adaptéry*. Jeho funkcionalita sa od vytvorenia v semestrálnej práci nezmenila.

Skript `node_dump`

Tento skript slúži na spustenie monitoringu Wi-Fi zariadení pracujúcich vo frekvenčných pásmach 2,4 GHz a 5 GHz a taktiež aj Bluetooth zariadení. Pri spúšťaní monitoringu Wi-Fi zariadení najskôr prepne použitý adaptér do režimu *monitor* a následne s týmto adaptérom spustí nástroj airodump-ng. Monitorovanie Wi-Fi zariadení je možné zapnúť s viacerými USB Wi-Fi adaptérmi naraz. V takom prípade budú „skákať“ medzi kanálmi tak, aby v žiadnom okamihu nepokrývali jeden kanál duplicitne. Spustenie monitoringu Bluetooth zariadení je možné iba s jedným Bluetooth čipom a prebieha volaním programu Bluelog s menom použitého čipu.

Pri zamýšľanom použití však nie je potrebné priame volanie tohto skriptu užívateľom. Na jeho obsluhu bol vytvorený skript `node_service` (spolu so skriptom `node_cs`), ktorý na to používa službu *systemd service*.

Volané programy nemajú žiadny výstup do konzole. Výstupy vo forme logu z monitoringu Wi-Fi aj Bluetooth zariadení sa ukladajú do adresáru `node/logs`. Program Bluelog ukladá všetok svoj výstup do jedného súboru aj pri vypnutí a opätovnom zapnutí monitoringu, pričom o týchto udalostiach zanechá zmienku. Naopak nástroj airodump-ng vytvára nový súbor pri každom zapnutí monitoringu a je s tým potrebné počítať pri ďalšom nastavovaní. Každý nový log nástroju airodump-ng má za menom najvyššie nepoužité číslo v danom adresári. Ďalším zásadným rozdielom logov týchto dvoch programov je, že Bluelog zapisuje každý záznam do nového riadku, a je teda možné v logoch vidieť celú históriu monitoringu, zatiaľ čo airodump-ng každým zápisom do logu prepisuje riadky na základe MAC adresy. Pokiaľ zaznamená zariadenie s určitou

MAC adresou v dosahu a v logu už existuje záznam o tomto zariadení, celý riadok sa prepíše. Takéto zapisovanie logov má za následok, že nie je možné získať históriu celého monitoringu, ale zakaždým iba stav v konkrétnom okamihu. Je s tým potrebné počítať pri importovaní do MySQL databázy. Záznam histórie monitoringu Wi-Fi zariadení v MySQL databáze existuje vďaka tomu, že sa riadky neprepisujú, ale pre každú unikátnu kombináciu MAC adresy a časovej známky posledného výskytu v dosahu sa vytvorí nový riadok.

Oproti stavu v semestrálnej práci sa funkcionalita tohto skriptu rozšírila o podporu monitorovania Wi-Fi zariadení pracujúcich v pásme 5 GHz a možnosť spustiť program Bluelog v takzvanom *amnesia* režime. Tento režim umožňuje „zabudnúť“ zaznamenané zariadenia po špecifikovanom čase a tým sa umožní jeho opätovné zapísanie do logu. Bez použitia tohto režimu by každé zariadenie bolo zapísané iba raz počas každého spustenia monitoringu.

Skript `node_cs`

Skratka „`cs`“ je odvodená od slov *create service*, keďže tento skript slúži na vytváranie *systemd service* služieb. Úlohou takto vytvorených *systemd service* služieb je ovládanie skriptu `node_dump`. Umožňuje to automatické spúšťanie prednastaveného monitoringu pri *boote* alebo manuálne spúšťanie kedykoľvek počas chodu zariadenia. Jeho volanie a teda aj argumenty sú rovnaké ako v prípade skriptu `node_dump`.

Z toho dôvodu boli implementované rovnaké rozšírenia funkcionality ako pri skripte `node_dump`.

Skript `node_service`

Účelom tohto skriptu je ovládanie *systemd service* vytvorenej skriptom `node_cs`. Na výber je 5 rôznych akcií. *Systemd service* je možné zapnúť a vypnúť do nasledujúceho *bootu* alebo povoliť a zakázať jej automatické spustenie pri každom nasledujúcom *boote*. Poslednou možnosťou je zobrazenie aktuálneho stavu *systemd service*.

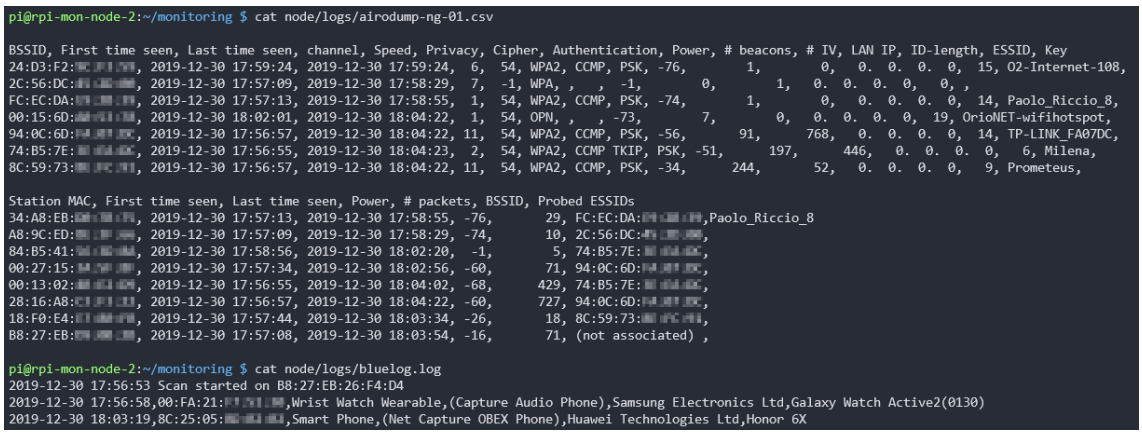
V tomto skripte bolo implementované rozšírenie funkcionality o prácu so službami *systemd service* pre monitoring Wi-Fi zariadení pracujúcich v pásme 5 GHz.

3.3.1.1 Vzorové spustenie monitoringu

Uvedený postup nastaví automatické spúšťanie monitoringu Wi-Fi (2,4 GHz) a Bluetooth zariadení pri každom *boote*. Bluetooth čip na Raspberry Pi má meno `hci0`.

```
sudo ./node_cs -w usbwlan1 -b hci0
sudo ./node_service -w enable -b enable
```

Po nasledujúcom *boote* sa spustil monitoring a v logoch sa začali objavovať zachytené zariadenia. Výpis týchto logov je pre ilustráciu na nasledujúcom obrázku.



Obr. 14 Výpis logov pre Wi-Fi a Bluetooth

Takto vygenerované logy je následne možné importovať do MySQL databázy. Na to slúžia skripty z adresáru `server/scripts` a ich funkcia je popísaná v nasledujúcej kapitole *3.3.2 Ovládanie serveru*.

3.3.2 Ovládanie serveru

Nasledujúce skripty sú používateľovi dostupné v adresári `server/scripts` vnútri adresárového stromu.

Skript `server_db`

Skôr ako sa do MySQL databázy môžu začať importovať logy z monitoringu, je potrebné ju vytvoriť. Na tento účel vznikol skript `server_db`, ktorý dokáže vytvoriť, ale aj odstrániť databázu na základe *hostname* zariadenia, pre ktoré je určená. Každé zariadenie zbierajúce dáta má teda vlastnú databázu a tá sa ďalej delí na tri tabuľky. Sú to tabuľky pre Wi-Fi bezdrôtové prístupové body (`AccessPoints`), Wi-Fi klientov (`Clients`) a pre Bluetooth zariadenia (`Bluetooth`). Formát tabuliek odpovedá formátu logových súborov vznikajúcich pri monitoringu, tým pádom sa do databázy importujú všetky zachytené informácie. Pre prehľadnosť kódu je formát tabuliek uložený v samostatnom súbore (`server_db_format`), z ktorého sa pri vytváraní databázy číta.

Skript `server_import`

Tento skript slúži na importovanie logových súborov do databázy. Pred samotným importom je potrebné požadovaný logový súbor skopírovať do adresáru MySQL databázy, kde môže byť spracovaný bez narušenia pôvodného súboru. Na importovanie zo vzdialených zariadení používa službu `scp`. V MySQL adresári je logový súbor následne spracovaný do *csv* formátu vhodného na vloženie do databázy. Spracovanie je podrobnejšie opísané v semestrálnej práci [20]. V zásade sa však jedná len o odstránenie

nežiadúcich častí logov a úpravu oddeľovačov stĺpcov. O samotný import sa postará služba `mysqlimport`, ktorá je súčasťou nainštalovanej MySQL databázy.

Podobne ako v prípade skriptu `node_dump`, na obsluhu tohto skriptu slúži skript `server_service` a `server_auto`, a pri bežnej prevádzke nie je potrebné jeho priame volanie užívateľom.

Primárny kľúč bol vo všetkých tabuľkách MySQL databázy zvolený ako kombinácia unikátneho identifikátoru a časovej známky posledného záznamu zariadenia. Tým sa zabezpečí, že sa z logov do MySQL databázy prenesú iba zmeny v prítomnosti zariadení, a nebude sa zbytočne zahlcovať opakovaným importovaním identických riadkov pre zariadenia, ktoré sú mimo dosahu. V prípade Wi-Fi zariadení (tabuľky `AccessPoints` a `Clients`) je za unikátny identifikátor považovaný stĺpec s MAC adresou a v tabuľke `Bluetooth` je to stĺpec `BD_ADDR` (Bluetooth Device Address).

Každé zavolanie skriptu, úspešné aj neúspešné, sa zapisuje do logu, ktorý sa nachádza v adresári `server/logs`. Zápis do logu rozoznáva úspešný import do databázy, chybu pri kopírovaní logového súboru a chybu pri zápise do databázy. Príklad výpisu z tohto logu je možné vidieť na obrázku *Obr. 15 Výpis logového súboru importu do databázy*.

Oproti stavu v semestrálnej práci sa funkcionalita tohto skriptu rozšírila o podporu importovania logov z Wi-Fi zariadení pracujúcich v pásme 5 GHz.

Skript `server_auto`

Hlavnou funkciou skriptu je automatické periodické importovanie dát do MySQL databázy. Využíva na to časovače *systemd timer*, ktoré s nastavenou periódou aktivujú služby *systemd service* a tie volajú skript `server_import`. Periódu je možné nastaviť ľubovoľnou kombináciou hodín, minút a sekúnd. Ďalej sa nastavuje *hostname* zariadenia, čiže názov MySQL databázy do ktorej sa majú logy importovať, typ logov a prípadne IP adresa vzdialeného zariadenia.

Na to aby bolo možné importovať logy zo vzdialených zariadení bez zadávania hesla je potrebné spárovanie SSH kľúčom. Aj na to slúži práve tento skript.

Detailnejšie informácie o použití skriptu sa nachádzajú v semestrálnej práci [20]. Avšak všetky tieto informácie je tak isto možné nájsť v *help* výpise samotného skriptu.

Oproti stavu v semestrálnej práci sa funkcionalita tohto skriptu rozšírila o podporu Wi-Fi zariadení pracujúcich v pásme 5 GHz a ukladanie informácie o perióde pre webové rozhranie.

Skript `server_service`

Úlohou tohto skriptu je ovládať každý vzniknutý časovač automatického importu do databázy. Rovnako ako v prípade skriptu `node_service` zabezpečuje 5 rôznych akcií, ktoré môže časovač *systemd timer* vykonať. Oproti týmto piatim akciám, ponúka navyše parameter `-w` spúšťajúci sledovanie všetkých časovačov *systemd timer* za pomoci služieb `watch` a `systemctl`.

Skript `server_clone`

Skript slúži na klonovanie MySQL databáz zo vzdialených zariadení cez internet. Používa na to služby `ssh`, `scp` a služby `mysqldump` a `mysqlimport` ktoré sú súčasťou MySQL databázy. Scenár, v ktorom sa tento skript môže hodiť je opísaný v poslednom odseku kapitoly *3.1 Rozdelenie zariadení*.[21][22]

3.3.2.1 Vzorové nastavenie periodického importu do MySQL databázy

Postup predpokladá, že sa na vzdialenom zariadení nachádzajú logové súbory, ktoré sa pokúša importovať a že zariadenia ešte neboli spárované SSH kľúčom. Pokiaľ by sa požadované logové súbory na vzdialenom zariadení nenachádzali, server by sa ich aj tak pokúšal periodicky importovať a do svojho logu by zapisoval neúspech pri kopírovaní, a pokiaľ by po čase vznikli, importoval by ich ako bolo pôvodne zamýšľané.

```
sudo ./server_db -c node_1
sudo -u pi ./server_auto -p 147.229.75.79
sudo ./server_auto -s 30 -d node_1 -a 147.229.75.79 -b -w 01
sudo ./server_service -d node_1 start
```

Najskôr je vytvorená nová MySQL databáza, potom sú zariadenia spárované a nakoniec je definovaný a aktivovaný časovač *systemd timer*, ktorý do novovzniknutej MySQL databázy každých 30 sekúnd importuje Bluetooth a Wi-Fi logy zo vzdialeného zariadenia s IP adresou `147.229.75.79`. Na nasledujúcom je zobrazená časť logu z importovania do databázy.

```
2020-06-03 20:27:24     successful      bt      rpi_mon_node_1  pi@147.229.75.79:/home/pi/monitoring/node/logs/bluelog.log
2020-06-03 20:27:24     successful      wlan    rpi_mon_node_1  pi@147.229.75.79:/home/pi/monitoring/node/logs/airodump-ng-01.csv
2020-06-03 20:27:53     successful      bt      rpi_mon_node_1  pi@147.229.75.79:/home/pi/monitoring/node/logs/bluelog.log
2020-06-03 20:27:54     successful      wlan    rpi_mon_node_1  pi@147.229.75.79:/home/pi/monitoring/node/logs/airodump-ng-01.csv
2020-06-03 20:28:37     successful      bt      rpi_mon_node_1  pi@147.229.75.79:/home/pi/monitoring/node/logs/bluelog.log
2020-06-03 20:28:37     successful      wlan    rpi_mon_node_1  pi@147.229.75.79:/home/pi/monitoring/node/logs/airodump-ng-01.csv
```

Obr. 15 Výpis logového súboru importu do databázy

3.4 Webové rozhranie

Po tom, ako sa dáta z monitoringu použitím *bash* skriptov uložia do MySQL databázy, ich spracovanie prebieha cez webové rozhranie. Užívateľ má na výber z troch hlavných podstránok, pričom každá implementuje odlišný algoritmus, a teda hľadá v dátach iné informácie. Na vytvorenie webového rozhrania sú využité štyri jazyky:

- HTML – základná štruktúra stránok, formuláre, textový výstup
- CSS – štýl webového rozhrania
- JavaScript – grafy (CanvasJS [39]) a aktualizovanie dynamického obsahu stránky
- PHP – spojenie s MySQL databázou a výpočtové algoritmy

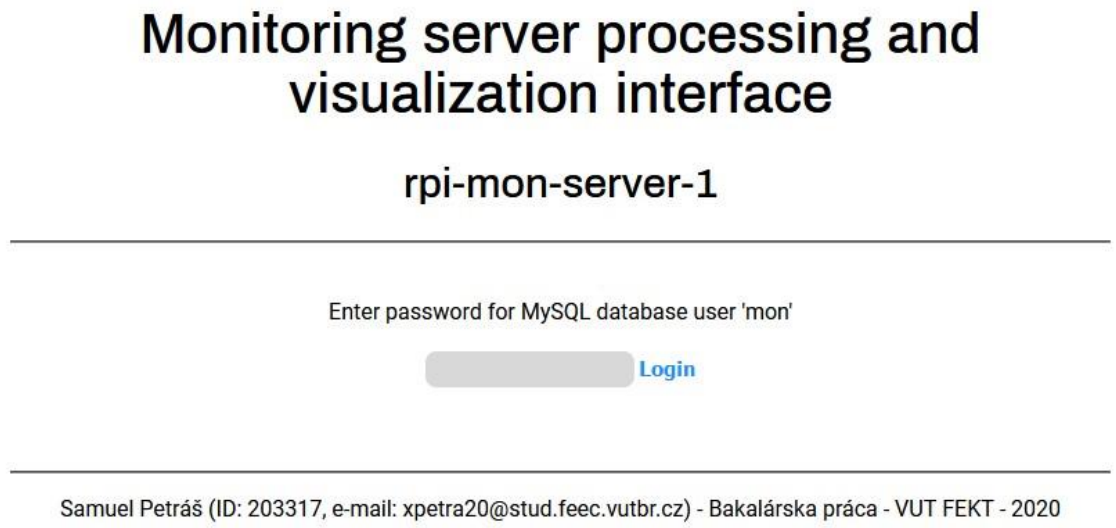
Rovnako ako v prípade *bash* skriptov, aj pri tvorbe webového rozhrania som sa okrem funkčnosti sústredil aj na bezproblémovú interakciu pre užívateľa a taktiež aj o estetiku stránky.

Dynamické prvky stránky sú implementované technikou AJAX. To umožňuje načítanie častí stránky bez toho, aby bolo potrebné ju celú obnoviť. Pre túto techniku som

sa rozhodol z toho dôvodu, aby mohol byť formulár s nastaveniami na rovnakej stránke ako výsledky algoritmu. Konkrétne je použitá na načítavanie kontajnerov (HTML *div*) *Information* a *Text Output*. Vďaka tomu užívateľ vidí, aké nastavenia použil pre aktuálny výpočet a taktiež to redukuje počet podstránok, čím sa orientácia na webovom rozhraní zjednodušuje. V praxi to vyzerá tak, že po odoslaní formuláru užívateľ čaká na rovnakej stránke a keď PHP algoritmus skončí, automaticky sa aktualizuje kontajner s výsledkom. V prípade kontajneru *Information* umožňuje použitie techniky AJAX aktualizáciu údajov informačnej tabuľky kedykoľvek, bez straty aktuálnych výsledkov alebo narušenia chodu PHP algoritmu.

Keďže technika AJAX narúša postupnosť vykonávania kódu, na predanie premenných do načítavaných PHP súborov je použitá PHP premenná _SESSION. To zaručuje dostupnosť potrebných údajov naprieč celým webovým rozhraním pre danú reláciu (*session*).

Na prístup do webové rozhrania je potrebné prihlásenie sa heslom užívateľa MySQL databázy s menom *mon*. Tento užívateľ má prístup len k monitorovacím databázam.



Obr. 16 Index webového rozhrania

3.4.1 Štruktúra

Všetky tri hlavné podstránky majú jednotnú štruktúru ktorá sa skladá z niekoľkých častí – kontajnerov. Ich obsah je opísaný v odsekoch nižšie.

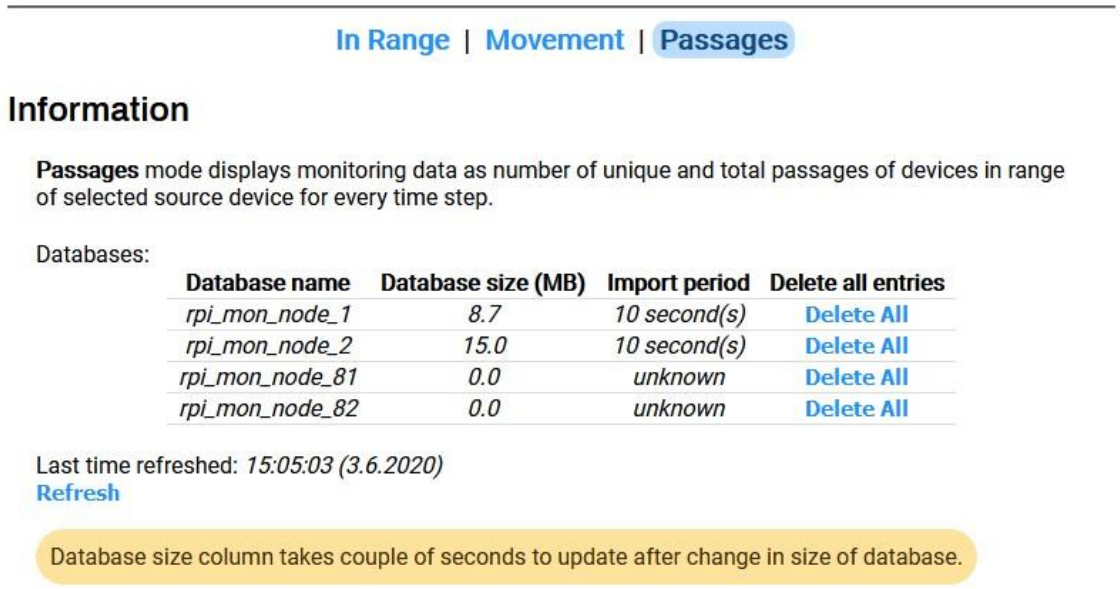
Žlté bubliny obsahujú informácie objasňujúce funkciu a obmedzenia niektorých prvkov stránky. Hlavička, pätička a menu sú importované zo spoločných zdrojových súborov a sú preto identické na každej podstránke. Menu sa nachádza medzi hlavičkou a kontajnerom *Information*.

Hlavička

Okrem názvu webového rozhrania obsahuje dynamický prvok – *hostname* zariadenia na ktorom sa nachádza server zobrazujúci webové rozhranie. Hlavičku je možné vidieť na obrázku *Obr. 16 Index webového rozhrania*.

Informácie (*Information*)

Obsahuje tabuľku s informáciami o dostupných MySQL databázach. Obsah je aktualizovaný pomocou JavaScript funkcie vždy pri načítaní stránky alebo manuálne tlačidlom *Refresh* pod tabuľkou. Okrem spoločnej časti obsahuje aj stručný opis funkcie aktuálnej podstránky. To, ako môže menu a informačný panel vyzerať, je uvedené na obrázku nižšie.



Obr. 17 Kontajner *Information* webového rozhrania

Informácia o perióde importovania logov do MySQL databázy je čítaná z priečinku `home/pi/monitoring/server/scripts/www/var`. Keďže informácie na tomto mieste vznikajú resp. zanikajú použitím skriptu `server_auto` na vytvorenie resp. odstránenie časovačov *systemd timer*, môže sa stať, že sa požadovaná informácia nenájde. V takom prípade sa v danom riadku zobrazí „*unknown*“ a znamená to, že časovač pre danú databázu neexistuje – žiadne logy sa neimportujú. Aj v prípade že sa informácia úspešne prečíta a zobrazí, nemusí to znamenať, že časovač importovania je spustený, môže byť zastavený. Pre podrobnejšie informácie o stave importovania je potrebné použiť *bash* skript `server_service` s argumentom `-w` alebo `-d` s možnosťou *status*.

Okrem toho obsahuje informačná tabuľka ešte veľkosť databázy v MB a tlačidlo *Delete All*, ktoré vymaže celý jej obsah. Vymazanie je potrebné opätovne potvrdiť, pretože sa jedná o trvalé odstránenie celého obsahu databázy, a tým pádom aj histórie monitoringu Wi-Fi zariadení, ktorá nie je obnoviteľná. Pre odstránenie samotnej databázy je potrebné použiť skript `server_db` s argumentom `-d`.

Obr. 18 Mazanie obsahu MySQL databázy cez webové rozhranie

Nastavenia (*Settings*)

Táto časť obsahuje HTML formulár s nastaveniami pre PHP algoritmus. V najjednoduchšom prípade má 5 častí (*In Range – Live data*) a v najkomplexnejšom 12 (*Movement – History*). Formuláre boli navrhnuté tak, aby užívateľovi poskytli čo najväčšiu kontrolu nad PHP algoritmom. Niektoré nastavenia majú predvolené hodnoty.

Hlavný HTML kontajner sa delí menšie kontajnery – jeden pre každú položku nastavenia. Vďaka tomu sa webové rozhranie jednoducho prispôsobuje šírke okna a tiež to umožňuje jednoduché pridávanie alebo odoberanie položiek formuláru v prípade potreby.

Graf (*Chart*)

Obsahuje prázdny graf (v prípade podstránky *Passages – History* obsahuje dva grafy), čakajúci na vygenerovanie dát PHP algoritmom v JSON formáte. Grafy sú implementované použitím študentskej licencie CanvasJS [39]. Načítanie dát do grafu je potrebné spustiť manuálne až po dokončení PHP algoritmu. Slúži na to tlačidlo *Update Chart* umiestnené pod grafom. Dokončenie PHP algoritmu je signalizované načítaním textového výstupu v časti *Text Output*.

Textový výstup (*Text Output*)

V tejto časti sa po dokončení PHP algoritmu zobrazí textový výstup. Obsah je automaticky načítaný pomocou JavaScript funkcie a po odoslaní HTML formuláru s nastaveniami nie je potrebná žiadna ďalšia akcia. V nasledujúcej tabuľke je uvedené, aký textový výstup ponúkajú jednotlivé podstránky.

Prípadné vylepšenia riešenia tejto práce by mali mať za cieľ, aby každý algoritmus ponúkal všetky typy textových výstupov a prípadne aj ďalšie, tu neuvedené.

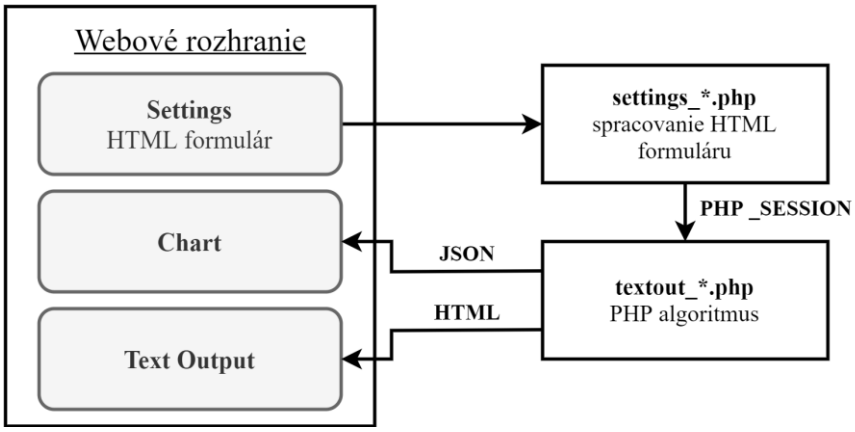
Tab. 3 Obsah textového výstupu jednotlivých podstránok

	Informácie o spracovanom časovom rozsahu	*Podrobnosti o jednotlivých zariadeniach*	*Tabuľka so štatistikami*	*Informácia o dĺžke trvania PHP algoritmu*	*Informácia o spotrebe pamäti PHP algoritmom*
In Range – Live data	Nie	Nie	**Áno**	Nie	Nie
In Range - History	**Áno**	Nie	Nie	**Áno**	**Áno**
Movement - History	**Áno**	**Áno**	**Áno**	**Áno**	**Áno**
Passages - History	**Áno**	**Áno**	**Áno**	**Áno**	**Áno**

Pätička
Obsahuje základné informácie ako meno, ID a emailovú adresu.

3.4.2 Algoritmy

Spolupráca podstránok s výpočtovými algoritmami taktiež dodržiava určitý formát. Ako napovedá obrázok *Obr. 9 Definované adresárové stromy,* každá podstránka dodržiava rovnaké rozdelenie kódu do súborov a adresárovú štruktúru. Súbory sú rozdelené aj podľa toho, či slúžia na spracovanie dát naživo – prípona `_live`, alebo spätne – prípona `_history`. Živé spracovanie dát je implementované len pre podstránku *In Range* a má slúžiť ako demonštrácia toho, že je to možné. Prepínač medzi *Live data* a *History* algoritmami sa nachádza nad HTML formulárom v časti *Settings*. Avšak omnoho podstatnejšia je implementácia spracovania histórie dát. Jedným z mnohých dôvodov je možnosť spracovania rovnakého obdobia s rozdielnymi nastaveniami algoritmu.



Obr. 19 Tok dát webového rozhrania

Všetky algoritmy začínajú prevzatím nastavení z PHP _SESSION premennej. Keďže sa jedná o vstup od užívateľa, skontroluje sa korektnosť týchto nastavení. Všetky algoritmy kontrolujú bežné chyby ako napríklad nevyplnené pole, alebo časový rozsah s hornou hranicou v budúcnosti.

Využitím informácií o MAC adresách z kapitoly *2.3 Wi-Fi* boli algoritmy nastavené tak, aby rozdeľovali MAC adresy podľa tabuľky *Tab. 4*. Keďže najnižší bit druhého hexadecimálneho znaku určuje, či je daná MAC adresa *unicast* (0) alebo *multicast* (1), a jednotlivé fyzické zariadenia z princípu nemôžu mať priradenú *multicast* adresu, v algoritme sa s týmito adresami nepočíta. Potvrdzuje to aj fakt, že sa počas žiadneho monitoringu nevyskytla v logoch ani jedna *multicast* MAC adresa.

Tab. 4 Rozdelenie MAC adries na globálne a lokálne

Globálne MAC adresy	***Lokálne MAC adresy***
_0:__:__:__:__:__	_2:__:__:__:__:__
_4:__:__:__:__:__	_6:__:__:__:__:__
_8:__:__:__:__:__	_A:__:__:__:__:__
_C:__:__:__:__:__	_E:__:__:__:__:__
„_“ označuje ľubovoľný hexadecimálny znak	

Zariadenia s lokálnou MAC adresou sa identifikujú pomocou zoznamu hľadaných ESSID. Tento zoznam sa tiež nazýva Preferred Network List (PNL) [40]. Keďže poradie v akom sa jednotlivé ESSID zaznamenajú nie je definované, je potrebné hľadať v zozname PNL anagramy – obsahujú rovnaké ESSID, len v inom poradí. Algoritmy takéto anagramy považujú za jedno zariadenie a zoznamy ich časových známok spoja do jedného.

Jednotlivé kroky vývojových diagramov na obrázkoch v prílohe *Príloha 2 - Vývojové diagramy algoritmov webového rozhrania* sú očíslované a opis v nasledujúcich podkapitolách je štruktúrovaný do bodov s rovnakým číslovaním.

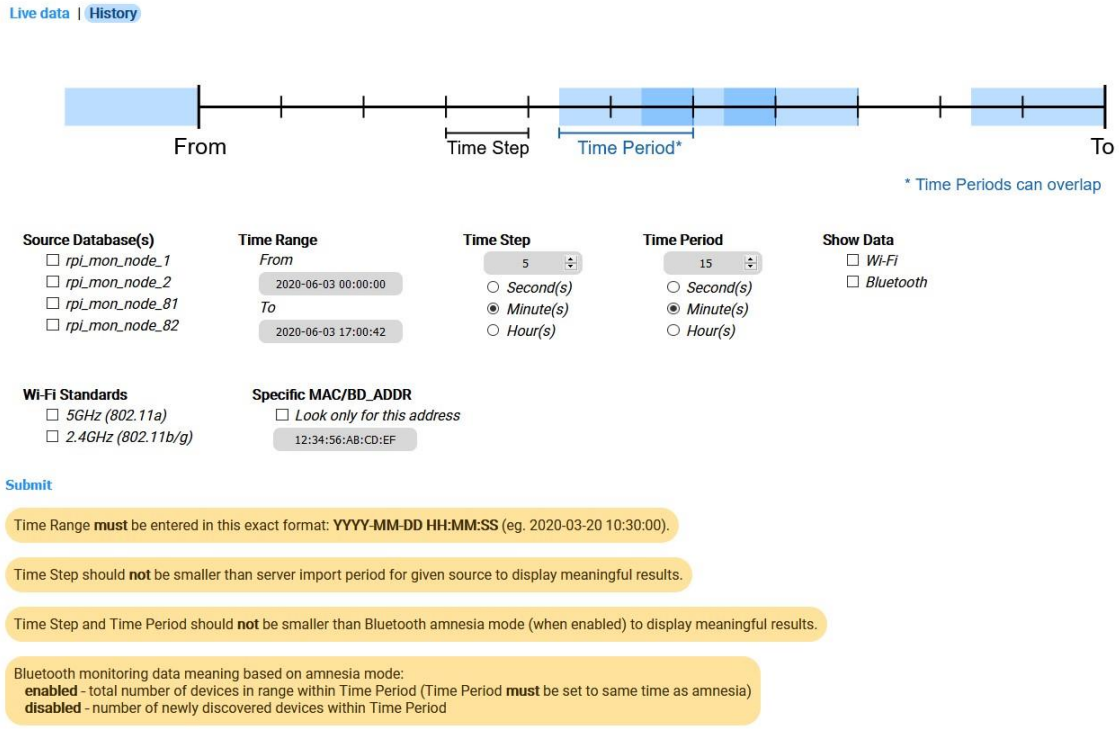
V nasledujúcich podkapitolách je pre MAC adresy, BD_ADDR adresy a PNL používané spoločné pomenovanie – kľúč.

3.4.2.1 Zariadenia v dosahu (*In Range – History*)

Monitorovacie dáta z daného časového rozsahu zobrazuje so špecifikovaným krokom ako graf počtu prítomných zariadení.

Časový krok sa nastavuje nezávisle od periódy, teda času, ako ďaleko do histórie sa algoritmus pozerá v každom časovom kroku. Jednotlivé periódy sa môžu prekrývať.

V nastaveniach na webovom rozhraní je vložený obrázok, ktorý vysvetľuje túto závislosť medzi krokom a periódou.



Obr. 20 Nastavenia algoritmu *In Range – History*

Vývojový diagram sa nachádza na obrázku *Príloha 2 - Obr. 38 Vývojový diagram algoritmu In Range – History*.

1. Nastavenia z PHP _SESSION premennej sa uložia do lokálnych premenných.
2. Skontroluje sa vstup od užívateľa.
    a. Po úspešnej kontrole vstupu sa algoritmus začína pripojením na MySQL databázu. Pokiaľ užívateľ vyberie viac MySQL databáz, algoritmus sa pripojí na všetky naraz a ďalej pracuje s referenciami na tieto pripojenia.
    b. Po neúspešnej kontrole sa v časti *Text Output* zobrazí chybová hláška vo forme nápovedy, ktoré nastavenie nie je akceptované.
3. Podľa toho, či užívateľ zadal hľadanie konkrétneho kľúča (v tomto prípade je možné zadať iba MAC alebo BD_ADDR adresu) alebo nie, sa spustí jeden z dvoch *for* cyklov.
    a. V tomto cykle sa do grafu pre jednotlivé kroky vkladá súčet všetkých nájdených kľúčov, ktoré spĺňajú nastavené podmienky. To sa opakuje pre každú zvolenú databázu, pričom hodnoty sa sčítajú do jedného grafu.

b. V prípade že užívateľ špecifikoval konkrétnu adresu, v grafe sa zobrazia hodnoty 1 – zariadenie je v danom kroku prítomné, alebo 0 – zariadenie nie je v danom kroku prítomné.

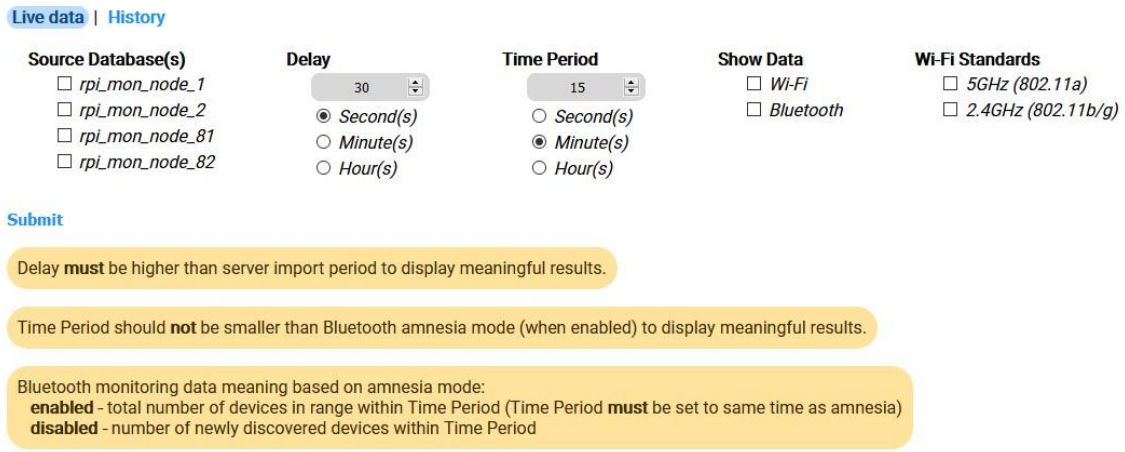
4. Algoritmus naformátuje do HTML informácie o časovom rozsahu a kroku a prípadne aj o tom aká konkrétna adresa bola hľadaná.
5. Nakoniec algoritmus uloží polia v JSON formáte do súborov v pamäti zariadenia. Po stlačení tlačidla *Update Chart* sa tieto súbory načítajú do grafov.

3.4.2.2 Zariadenia v dosahu (*In Range – Live data*)

Monitorovacie dáta zobrazuje v reálnom čase s nastaveným oneskorením a krokom 30 sekúnd ako:

- Graf počtu prítomných zariadení.
- Tabuľku štatistík počtu zariadení pre posledné meranie.

Periodické volanie algoritmu zabezpečuje samotná stránka použitím JavaScript metódy `setInterval`. Nastavenia z HTML formuláru sa medzi jednotlivými volaniami držia v premennej PHP _SESSION. Okrem nastavení sa v tejto premennej taktiež držia polia, ktoré sa na konci algoritmu prevádzajú do formátu JSON a ukladajú do pamäte, odkiaľ sú automaticky načítané do grafu. Pri každom zavolaní sa do týchto polí pridávajú aktuálne hodnoty. Pri opätovnom odoslaní formuláru sa meranie resetuje. Resetovaním sa vymažú polia uchovávajúce stav grafu z PHP _SESSION a začnú sa plniť odznova.



Obr. 21 Nastavenia algoritmu *In Range – Live data*

Oneskorenie väčšie ako perióda importovania logov do MySQL databázy zaručuje to, aby boli v momente spracovávania potrebnej oblasti všetky logy importované a tým pádom dáta v spracovávanej časovej oblasti kompletné. Pri nastavovaní tohto parametru je potrebné počítať s časom samotného vkladania dát do MySQL databázy, ktorý nie je zanedbateľný.

Vývojový diagram sa nachádza na obrázku *Príloha 2 - Obr. 39 Vývojový diagram algoritmu In Range – Live.*

1. Nastavenia a polia z PHP _SESSION premennej sa uložia do lokálnych premenných.
2. Skontroluje sa vstup od užívateľa.
    a. Po úspešnej kontrole vstupu sa algoritmus začína pripojením na MySQL databázu. Pokiaľ užívateľ vyberie viac MySQL databáz, algoritmus sa pripojí na všetky naraz a ďalej pracuje s referenciami na tieto pripojenia.
    b. Po neúspešnej kontrole sa v časti *Text Output* zobrazí chybová hláška vo forme nápovedy, ktoré nastavenie nie je akceptované.
3. Počty nájdených kľúčov, ktoré spĺňajú nastavené podmienky, sa vložia na koniec lokálnej kópie polí.
4. V tomto momente už sú všetky výpočty hotové a výsledné počty zariadení sa naformátujú do HTML textového výstupu.
5. Nakoniec algoritmus uloží polia v JSON formáte do súborov v pamäti zariadenia.

3.4.2.3 Presun zariadení (*Movement – History*)

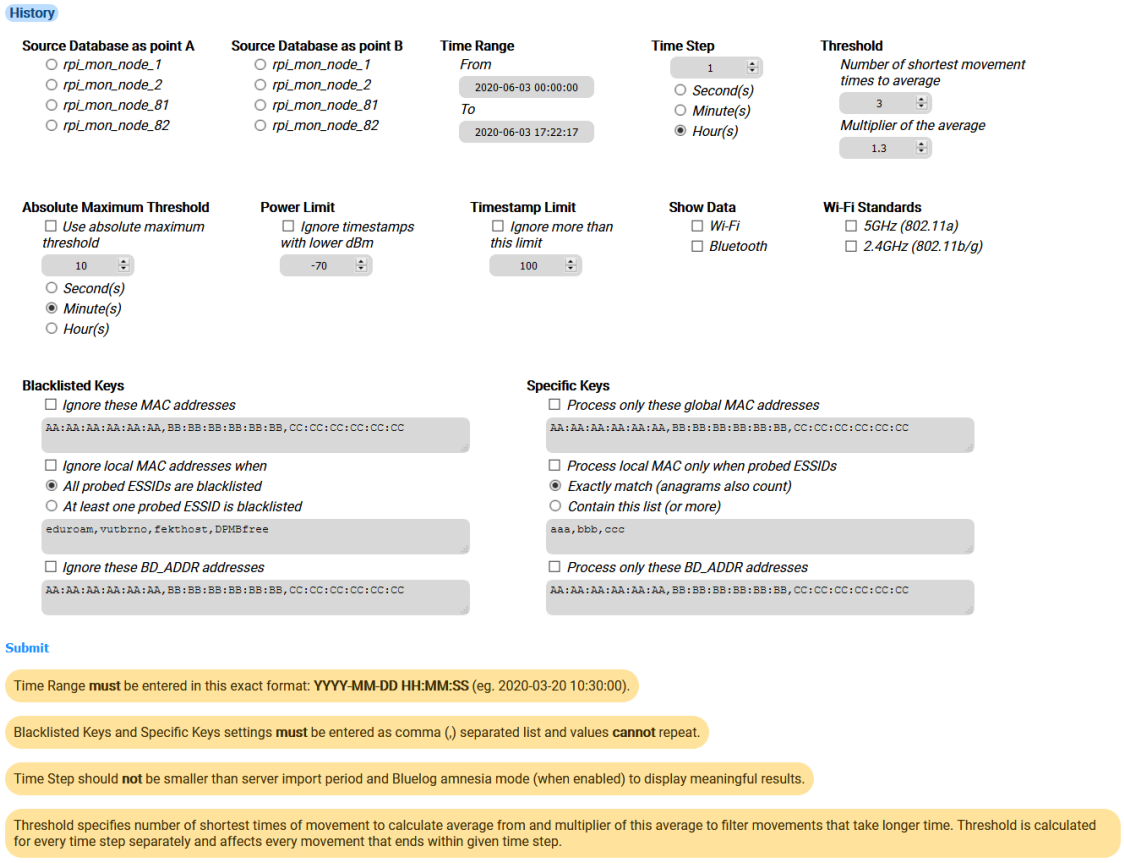
Monitorovacie dáta z daného časového rozsahu zobrazuje ako:

- Graf priemerného času presunu z bodu A do bodu B a naopak pre každý časový krok.
- Tabuľku štatistík počtu zariadení a presunov z celého časového obdobia.
- Zoznam zariadení a ich jednotlivých presunov.

V algoritme je na zoskupenie informácií implementovaná trieda *Movement* a obsahuje verejné premenné:

- Kľúč.
- Príznak *blacklisted.*
- Príznak *over_timestamp_limit.*
- A dve polia. Jedno pole pre presuny z bodu A do bodu B a druhé pre opačné presuny. Polia presunov sú dvojrozmerné, pričom v prvom stĺpci ukladajú časovú známku odchodu z prvého bodu, v druhom stĺpci časovú známku príchodu do druhého bodu a v treťom časový rozdiel týchto dvoch hodnôt.

V prípade nastavenia filtrovania časových známok podľa sily signálu (*Power Limit*) nie je možné použiť ako zdroj dát Bluetooth, pretože program Bluelog nezaznamenáva silu zachyteného signálu.



Obr. 22 Nastavenia algoritmu *Movement – History*

Algoritmus súčasne hľadá presuny z bodu A do B a naopak. Pre zjednodušenie je algoritmus opísaný len pre jeden smer. Pre druhý smer sú body A a B vymenené a spracovanie prebieha analogicky. Vývojový diagram sa nachádza na obrázku *Príloha 2 - Obr. 41 Vývojový diagram algoritmu Movement – History*.

1. Nastavenia z PHP _SESSION premennej sa uložia do lokálnych premenných.
2. Skontroluje sa vstup od užívateľa.
    a. Po úspešnej kontrole vstupu sa algoritmus začína pripojením na MySQL databázy, ktoré boli nastavené ako bod A a B.
    b. Po neúspešnej kontrole sa v časti *Text Output* zobrazí chybová hláška vo forme nápovedy, ktoré nastavenie nie je akceptované.
3. Zoznam kľúčov, ktorý sa predá na spracovanie, záleží od toho či užívateľ špecifikoval ich zoznam alebo nie.

a. Z databázy A a B sa získa zoznam všetkých globálnych MAC adries, všetkých reťazcov zo stĺpca `probed_ESSIDs` patriacich lokálnym MAC adresám a všetkých BD_ADDR adries v danom časovom rozsahu.

b. Z lokálnej kópie nastavení z HTML formuláru sa získajú zoznamy špecifikovaných kľúčov. V prípade špecifikovania ESSID sa požadované kľúče hľadajú v MySQL databáze. Je možné vybrať medzi hľadaním kľúčov, ktoré sú zostavené výlučne zo špecifikovaného zoznamu, alebo je možné povoliť aby obsahovali aj iné ESSID.

4. Odfiltrujú sa kľúče, ktoré sa nenachádzajú v oboch databázach naraz, pretože v takom prípade sa určite žiadny presun nenájde. Následne sa hotový zoznam kľúčov predá do *for* cyklu kde sa po jednom spracujú (kroky 5-10).
5. Pokiaľ užívateľ špecifikoval zoznam zariadení ktoré sa nemajú spracovať – *blacklist*, tento zoznam sa pri každom cykle prejde a skontroluje sa, či sa v ňom nenachádza práve spracovávaný kľúč. Pokiaľ sa v ňom kľúč nájde, v triede *Movement* sa nastaví príznak *blacklisted* a spracovanie daného kľúču sa preskočí.
6. Z MySQL databázy bodu A a B sa získajú všetky časové známky pre aktuálny kľúč. V prípade použitia filtrovania na základe štandardu alebo sily signálu sa vyberú len tie časové známky, ktoré spĺňajú nastavené podmienky.
7. Skontroluje sa počet súčtu časových známok z MySQL databázy A a B. Pokiaľ sa našiel väčší počet časových známok ako je nastavený limit, v triede *Movement* sa nastaví príznak *over_timestamp_limit* a spracovanie daného kľúča sa preskočí.
8. Nájde sa minimálny časový rozdiel medzi časovou známkou z bodu A a časovou známkou z bodu B. Tento údaj sa považuje za minimálny čas presunu aktuálneho kľúča medzi bodmi A a B a využije sa v nasledujúcich krokoch.
9. Je veľmi pravdepodobné, že časové známky budú v databáze uložené v zhlukoch. Tieto zhluky je potrebné odfiltrovať tak, aby ostali iba časy odchodu a príchodu. Nasledujúca tabuľka *Tab. 5* zobrazuje jednoduchý príklad. Minimálny čas presunu z bodu A do bodu B je medzi 16:15 a 16:45, čiže 30 minút. V časových známkach pre bod A sa odstránia všetky časové známky medzi ktorými je menší rozdiel ako 30 minút, pričom sa ponechá vždy tá posledná, pretože ide o čas odchodu. V časových známkach pre bod B sa naopak ponechá vždy tá prvá, keďže ide o čas príchodu.

Tab. 5 Ukážka kroku 9 z algoritmu pre presun zariadení

	časové známky						
bod A	~~16:00~~	~~16:05~~	~~16:10~~	**16:15**	~~18:00~~	~~18:05~~	**18:10**
bod B	**16:45**	~~16:50~~	**19:00**	~~19:05~~	~~19:10~~	~~19:15~~	

10. V tomto kroku sa postupne porovnávajú všetky časové známky z bodu A s časovými známkami z bodu B. Pokiaľ sa nájde v časových známkach pre bod B taký čas, ktorý je väčší ako aktuálne porovnávaný čas z bodu A a zároveň je menší ako dvojnásobok minimálneho času presunu, uloží sa. Toto obmedzenie je potrebné na zabránenie chybného spojenia časových známok. V nasledujúcej tabuľke *Tab. 6* je príklad toho, kedy sa toto obmedzenie uplatní. Keďže sa časové známky pre bod A od seba nachádzajú ďalej ako je minimálny čas presunu (30 minút), neboli správne odfiltrované v minulom kroku. Aby sa zabránilo zaznamenaniu dvoch presunov, je uplatnený limit, a keďže prechod od 9:00 do 10:30 by trval viac ako 60 minút, neuloží sa.

Tab. 6 Ukážka kroku 10 z algoritmu pre presun zariadení

	časové známky	
bod A	**9:00**	**10:00**
bod B	**10:30**	

11. Nájdené presuny sa ukladajú do triedy *Movement*. Výstupom *for* cyklu je pole týchto tried.
12. Na základe nastavenia *Threshold* a *Absolute Maximum Threshold* sa odfiltrujú príliš dlhé presuny. Pre každý časový krok výpočtu sa spriemeruje požadovaný počet najkratších časov presunu (spomedzi všetkých kľúčov) ktoré v danom časovom kroku končia a táto hodnota sa vynásobí požadovaným koeficientom. Výsledné číslo je horný limit času presunu pre daný časový krok. Druhá hranica je pevne nastavená užívateľom a platí pre všetky časové kroky. Presuny ktoré prekročia jednu z týchto hraníc sa stále zobrazia v textovom výstupe, bude však pri nich informácia o tom že prekročili horný limit a ich hodnoty sa neprejavia vo výslednom grafe.
13. V tomto momente už sú všetky výpočty hotové a výsledné pole tried *Movement* sa naformátuje do HTML textového výstupu.
14. Nakoniec algoritmus prevedie polia tried *Movement* do polí vo formáte čitateľnom pre grafy a v JSON formáte ich uloží do súborov v pamäti zariadenia. Po stlačení tlačidla *Update Chart* sa tieto súbory načítajú do grafov.

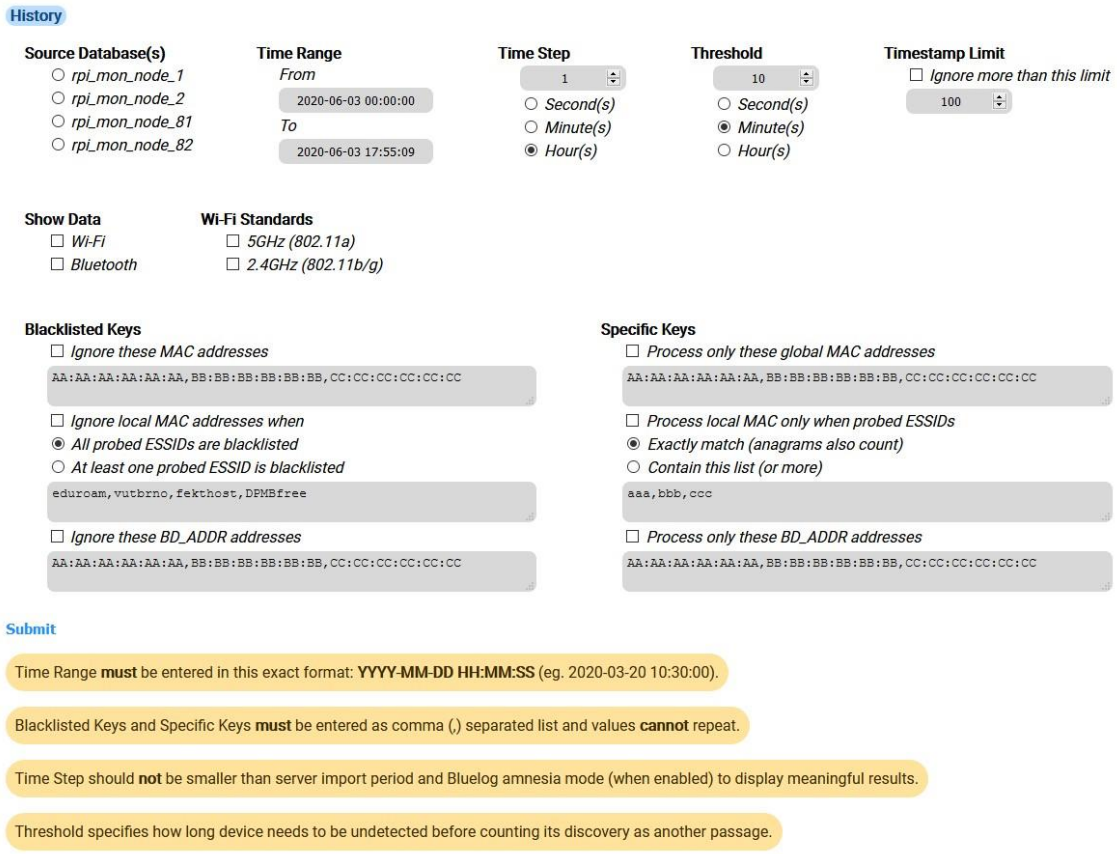
3.4.2.4 Počet priechodov (*Passages – History*)

Monitorovacie dáta z daného časového rozsahu zobrazuje ako:

- Graf počtu unikátnych zariadení ktoré prešli v dosahu spolu s celkovým počtom prechodov pre každý časový krok.
- Tabuľku štatistík počtu zariadení z celého časového obdobia.
- Zoznam zariadení a ich časových známok.

V algoritme je na zoskupenie informácií implementovaná trieda *Passenger* a obsahuje verejné premenné:

- Kľúč.
- Príznak *blacklisted.*
- Príznak *over_timestamp_limit.*
- A jedno pole časových známok. Toto pole je dvojrozmerné, pričom v prvom stĺpci ukladá časovú známku a v druhom stĺpci značku, či sa jedná o priechod alebo dlhší pobyt zariadenia v dosahu.



Obr. 23 Nastavenia algoritmu *Passages – History*

Vývojový diagram sa nachádza na obrázku *Príloha 2 - Obr. 40 Vývojový diagram algoritmu Passages – History*.

1. Nastavenia z PHP _SESSION premennej sa uložia do lokálnych premenných.
2. Skontroluje sa vstup od užívateľa.
    a. Po úspešnej kontrole vstupu sa algoritmus začína pripojením na MySQL databázu.
    b. Po neúspešnej kontrole sa v časti *Text Output* zobrazí chybová hláška vo forme nápovedy, ktoré nastavenie nie je akceptované.

3. Zoznam kľúčov, ktorý sa predá na spracovanie, záleží od toho či užívateľ špecifikoval ich zoznam alebo nie.
    a. Z databázy sa získa zoznam všetkých kľúčov v danom časovom rozsahu.
    b. Z lokálnej kópie nastavení z HTML formuláru sa získajú zoznamy špecifikovaných kľúčov. V prípade špecifikovania ESSID sa požadované kľúče hľadajú v MySQL databáze. Je možné vybrať medzi hľadaním kľúčov, ktoré sú zostavené výlučne zo špecifikovaného zoznamu, alebo je možné povoliť aby obsahovali aj iné ESSID.
4. Pokiaľ užívateľ špecifikoval zoznam zariadení ktoré sa nemajú spracovať – *blacklist*, tento zoznam sa pri každom cykle prejde a skontroluje sa, či sa v ňom nenachádza práve spracovávaný kľúč. Pokiaľ sa v ňom kľúč nájde, v triede *Passenger* sa nastaví príznak *blacklisted* a spracovanie daného kľúču sa preskočí.
5. Z MySQL databázy sa získajú všetky časové známky pre aktuálny kľúč. V prípade použitia filtrovania na základe štandardu sa vyberú len tie časové známky, ktoré spĺňajú nastavené podmienky.
6. Skontroluje sa počet časových známok. Pokiaľ sa v MySQL databáze našiel väčší počet časových známok ako je nastavený limit, v triede *Passenger* sa nastaví príznak *over_timestamp_limit* a spracovanie daného kľúča sa preskočí.
7. V cykle sa prejdú všetky časové známky a pokiaľ je rozdiel od aktuálnej po minulú väčší ako nastavená hodnota *Threshold*, označí sa ako prechod. Prvá časová známka sa označí ako prechod vždy.
8. Všetky časové známky spolu so značkou sa uložia do triedy *Passenger*. Výstupom *for* cyklu je pole týchto tried.
9. V tomto momente už sú všetky výpočty hotové a výsledné pole tried *Passenger* sa naformátuje do HTML textového výstupu.
10. Nakoniec algoritmus prevedie polia tried *Passenger* do polí vo formáte čitateľnom pre grafy a v JSON formáte ich uloží do súborov v pamäti zariadenia. Po stlačení tlačidla *Update Chart* sa tieto súbory načítajú do grafov.

3.5 Využitie

Vďaka predávaniu dát cez internet je vzdialenosť zariadení Raspberry Pi teoreticky neobmedzená, rovnako ako počet spolupracujúcich zariadení. Avšak v prípade použitia viacerých zariadení *node* sa ako najvhodnejšie ukázalo použiť jedno zariadenie výlučne ako server. Je to dôsledok relatívne nízkeho výkonu Raspberry Pi. Ešte vhodnejšie je použiť ako server výkonnejšie zariadenie, napríklad stolný počítač. To je možné vďaka univerzálnosti skriptov. Spracovanie dát cez webové rozhranie a vkladanie dát do

MySQL databázy tak trvá rádovo kratšie ako v prípade použitia Raspberry Pi, čo umožňuje voľbu kratšej periódy importovania logov do MySQL databázy.

Ďalším parametrom rozhodujúcim o možnostiach umiestnenia zariadení je ich malá veľkosť a spotreba energie. Krátkodobé merania tak môžu prebiehať napájaním z *powerbanky*.

Ovplyvniť oblasť, z ktorej budú dáta zbierané, je možné citlivosťou a typom antény. Citlivosť je prípadne možné ovplyvniť aj pri spracovaní dát pomocou nastavenia limitu sily zachyteného signálu.

Opísané vlastnosti naznačujú, že využití predstaveného riešenia práce je mnoho a rozmery monitorovacej siete sa môžu výrazne líšiť. Pre ilustráciu rôznorodosti týchto využití je ich niekoľko vymenovaných. Ako je z tohto zoznamu zrejmé umiestnenie zariadení si vyžaduje individuálny prístup na základe konkrétneho využitia. Môže to byť meranie:

- Počtu prítomných zákazníkov v priestoroch obchodu
- Pomeru nový a vracajúcich sa zákazníkov
- Rušnosti priestorov podľa dennej doby
- Vývoja rýchlosti presunu z bodu A do bodu B (obojsmerne)
    - Turistické trasy
    - Iné pešie trasy
    - Cesty
- Počtu účastníkov rôznych udalostí
- Času prítomnosti konkrétnych zariadení
- Počtu ľudí, ktorý prešli určitým priestorom
    - Celkovo
    - Pomer nových a vracajúcich sa

Využitie je limitované hlavne implementovanými algoritmami na spracovanie dát. Vzhľadom na aktuálnu pandémiu koronavírusu SARS-CoV-2 by mohli byť dáta v MySQL databáze využité na identifikovanie osôb, ktoré sa vyskytli v blízkosti pozitívne testovanej osoby. Za predpokladu známej polohy monitorovacích zariadení by k tomu nebola potrebná žiadna úprava monitoringu. Dáta v MySQL databáze v takom prípade tieto informácie obsahujú. Stačilo by implementovať vhodný algoritmus na ich získanie. Vo výsledku by sa jednalo o niečo podobné ako implementovala aplikácia Mapy.cz, avšak nebolo by to tak presné, pretože monitorovacie zariadenie môže mať dosah desiatky metrov, zatiaľ čo Mapy.cz využívajú priamo polohu konkrétnych zariadení [44]. Presnejšiu polohu jednotlivých zariadení by bolo možné získať pomocou trilaterácie na základe sily zachyteného signálu.

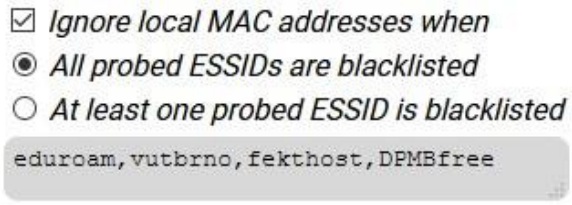
Pred využitím v reálnej prevádzke by však bolo najskôr potrebné implementovať potrebné náležitosti na splnenie podmienok GDPR, s ktorými riešenie práce nepočíta. Tiež by bolo potrebné zvýšiť celkovú bezpečnosť zariadení a webového rozhrania, čo spadá mimo rozsah tejto práce.

4 VÝSLEDKY MONITORINGU

Táto kapitola predstavuje vzorové spracovanie dát z monitoringu pomocou webového rozhrania a predstavuje výsledky jednotlivých algoritmov. Ďalej zahŕňa podrobnejšiu analýzu dát v MySQL databáze.

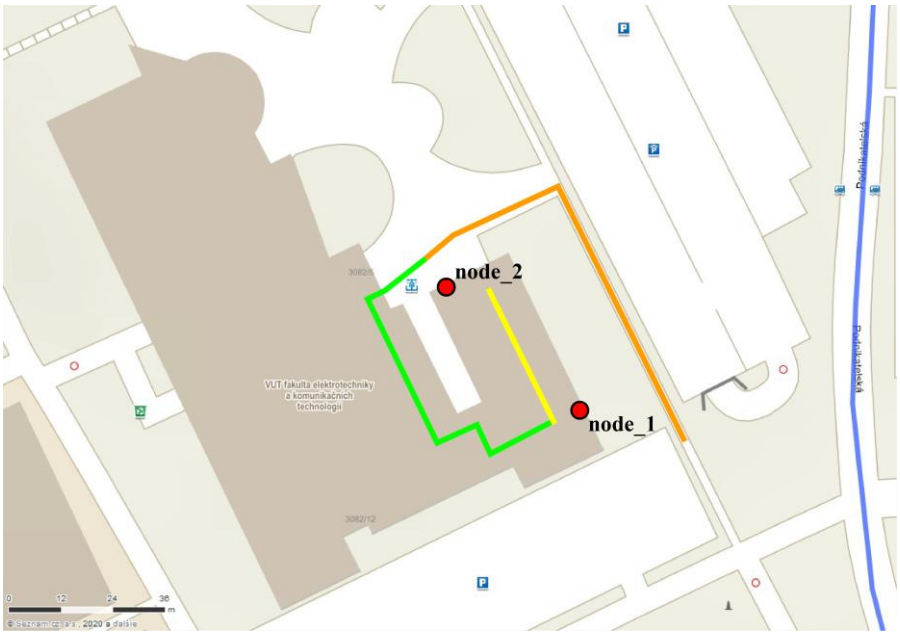
4.1 Spracovanie dát webovým rozhraním

V nasledujúcich podkapitolách sú zobrazené a opísané výstupy jednotlivých algoritmov, implementovaných v rámci webového rozhrania. Perióda importovania bola nastavená na 10 sekúnd a Bluelog *amnesia* režim na 10 min. V algoritmoch, kde je to možné, bol použitý nasledovný zoznam *blacklist*. Zabráni sa tým nežiadúcemu spájaniu časových známok zo zariadení, ktoré vyhľadávajú iba tieto, v danej oblasti bežné, ESSID.



Obr. 24 Nastavenie *Blacklisted Keys* algoritmov webového rozhrania

Zber dát, analyzovaných v nasledujúcich podkapitolách, prebiehal pomocou zariadení *node* rozmiestnených v priestoroch VUT na adrese Technická 12, ako je približne uvedené na obrázku nižšie. K zariadeniu *node_1* bol pripojený jeden 2,4 GHz USB Wi-Fi adaptér a k *node_2* jeden 2,4 GHz a jeden 5 GHz USB Wi-Fi adaptér.

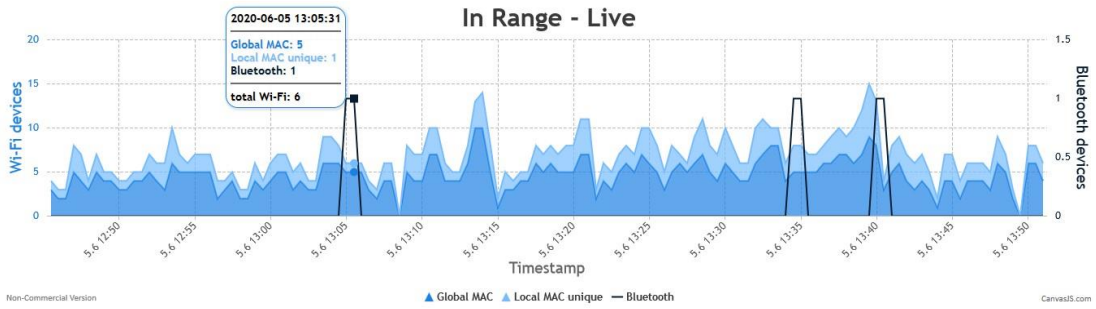


Obr. 25 Poloha monitorovacích zariadení počas testovania, Zdroj: "Mapy.cz"

Modrou farbou je v obrázku vyznačená trasa autobusov verejnej dopravy podniku DPMB a oranžovou, zelenou a žltou sú vyznačené najkratšie trasy medzi zariadeniami. Vzhľadom na to, že má budova viac poschodí a vchodov, skutočné trasy ľudí sa mohli výrazne líšiť od nákresu.

4.1.1 Algoritmus *In Range – Live data*

Na obrázkoch *Obr. 26* a *Obr. 27* nižšie je zobrazený výstup algoritmu po zhruba 60 minútach od spustenia, v časti *Text Output* sú vypísané štatistiky, ktoré sú aktualizované s každým spustením algoritmu (30 sekúnd) a zobrazujú hodnoty len pre aktuálny krok. Časová perióda, teda ako ďaleko do histórie sa algoritmus v každom kroku pozerá, bola nastavená na 1 minútu a oneskorenie na 30 sekúnd. Ako zdroj dát bolo nastavené zariadenie *node_1* z obrázku *Obr. 25* a časové rozpätie je viditeľné v grafe na ose x.



Obr. 26 Výstupný graf algoritmu *In Range – Live data*

Text Output

13:53:01 (5.6.2020)
Showing results of last 1 minute(s) updated every 30 second(s) delayed by 30 second(s)

Wi-Fi
Number of devices with global (unique) MAC address: 4
Number of identified local MAC address fingerprints: 3
Estimated total number of devices within reach: 7
Number of detected local (randomized) MAC addresses: 4
Bluetooth
Number of devices discovered: 0

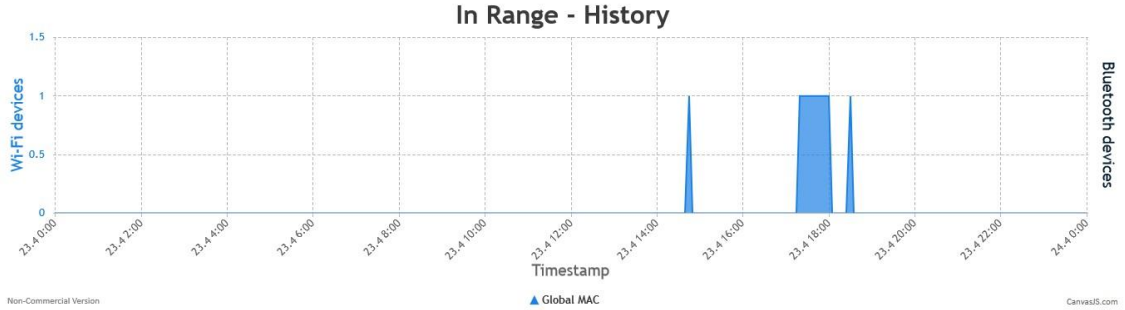
Obr. 27 Výstupný text algoritmu *In Range – Live data*

4.1.2 Algoritmus *In Range – History*

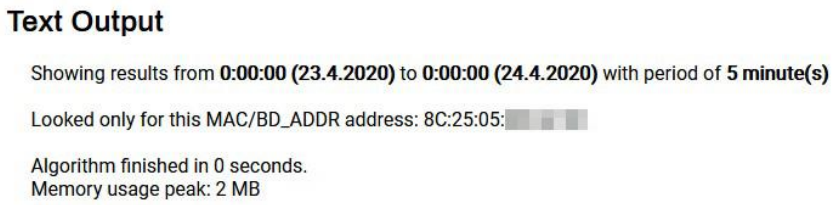
Výstup tohto algoritmu som sa rozhodol ukázať na jednom konkrétnom príklade. Graf na obrázku nižšie zachytáva jeden celý deň, z ktorého časť som strávil v priestoroch školy. Nastavenia algoritmu sú nasledovné:

- **Source Database(s):** rpi_mon_node_1
- **Time Range:** From 2020-04-23 00:00:00 To 2020-04-24 00:00:00
- **Time Step:** 5 Minute(s)
- **Time Period:** 5 Minute(s)
- **Show Data:** Wi-Fi
- **Wi-Fi Standards:** 2.4GHz (802.11b/g)
- **Specific MAC/BD_ADDR:** MAC adresa môjho smartfónu

Prvá krátka zaznamenaná prítomnosť časovo sedí s mojím príchodom do školy popri monitorovacom zariadení *node_1* z obrázku *Obr. 25* (v tom čase ešte nebolo spustené zariadenie *node_2*). Po príchode som dlhší čas strávil vo vzdialenej učebni a neskôr som sa zastavil na konzultáciu, pričom monitorovacie zariadenie je umiestené priamo v kancelárii pri okne. Dĺžke a času konzultácie odpovedá dlhšie zaznamenanie mojej prítomnosti. Po konzultácií som sa na krátko opäť zastavil vo vzdialenej učebni a potom som zamieril domov rovnakou trasou akou som prišiel, čomu odpovedá posledné zaznamenanie.

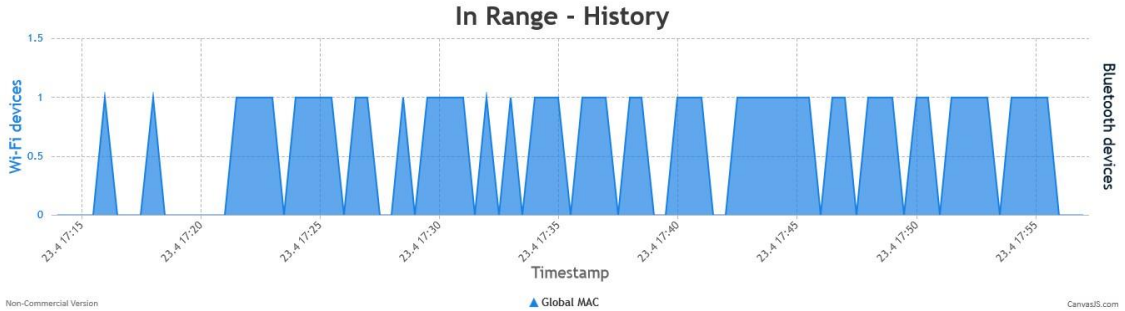


Obr. 28 Výstupný graf algoritmu *In Range – History*



Obr. 29 Výstupný text algoritmu *In Range – History*

Na nasledujúcom obrázku je graf priblížený a časový krok a perióda sú znížené na 1 minútu. Z výsledku vyplýva, že prítomnosť nie je v MySQL zaznamenaná spojito a je preto potrebné voliť dostatočne veľké periódy, aby sa dáta spojili. Hustotu dát ovplyvňuje najmä perióda importovania logov z monitoringu do MySQL databázy ale aj rýchlosť, akou monitorovacie programy tieto logy zapisujú. Predvolená hodnota programu airodump-ng je 5 sekúnd a v prípade programu Bluelog nie je definovaná, dá sa však ovplyvniť dĺžkou skenovacích cyklov.



Obr. 30 Demonštrácia nespojitých dát v MySQL databáze

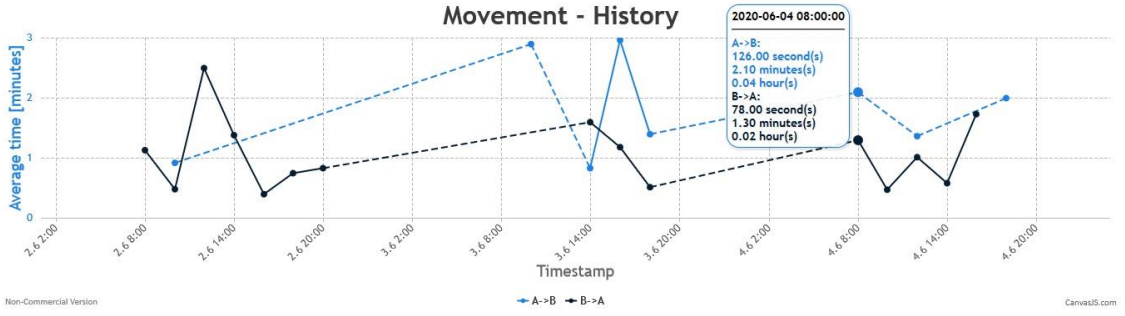
Je potrebné povedať, že aj napriek tomu, že som mal so sebou po celý čas smart hodinky so zapnutým Bluetooth rozhraním, ale neboli ani raz zachytené. Tým sa potvrdzuje nižšia spoľahlivosť takéhoto typu monitoringu.

4.1.3 Algoritmus *Movement – History*

Na obrázkoch nižšie je zobrazený výstup algoritmu s nasledujúcimi nataveniami:

- **Source Database as point A:** rpi_mon_node_1
- **Source Database as point B:** rpi_mon_node_2
- **Time Range:** From 2020-06-02 00:00:00 To 2020-06-05 00:00:00
- **Time Step:** 2 Hour(s)
- **Threshold:** 3 x 3
- **Absolute Maximum Threshold:** 5 Minute(s)
- **Power Limit:** -59 dBm
- **Timestamp Limit:** nie
- **Show Data:** Wi-Fi
- **Wi-Fi Standards:** 2.4GHz (802.11b/g)
- **Blaclisted Keys:** obrázok *Obr. 24*
- **Specific Keys:** nie

Vzdialenosť zariadení zobrazená na obrázku *Obr. 25* nie je ideálna pre použitie tohto algoritmu, pretože ich dosah sa navzájom prekrýva. Na riešenie takejto situácie vzniklo nastavenie *Power Limit*, to však výrazne redukuje počet časových známok s ktorými algoritmus pracuje. V tomto prípade bolo potrebné nastaviť hodnotu na -59 dBm, čo má za následok veľký pokles v počte časových známok zariadení. Analýzou MySQL databázy sa ukázalo, že v prípade zariadenia *node_1*, len 6,3% (2379) časových známok z daného časového obdobia spĺňa toto obmedzenie sily signálu. V prípade zariadenia *node_2* to je 26% (8633), čo môže byť spôsobené tým, že sa nachádza bližšie k trase, po ktorej sledované osoby chodili. Uvedené čísla platia pre analýzu všetkých zariadení pracujúcich v pásme 2,4 GHz z daného časového obdobia, nie len tie, ktoré sa našli v oboch databázach naraz, čo je ďalší filter, ktorý algoritmus používa.

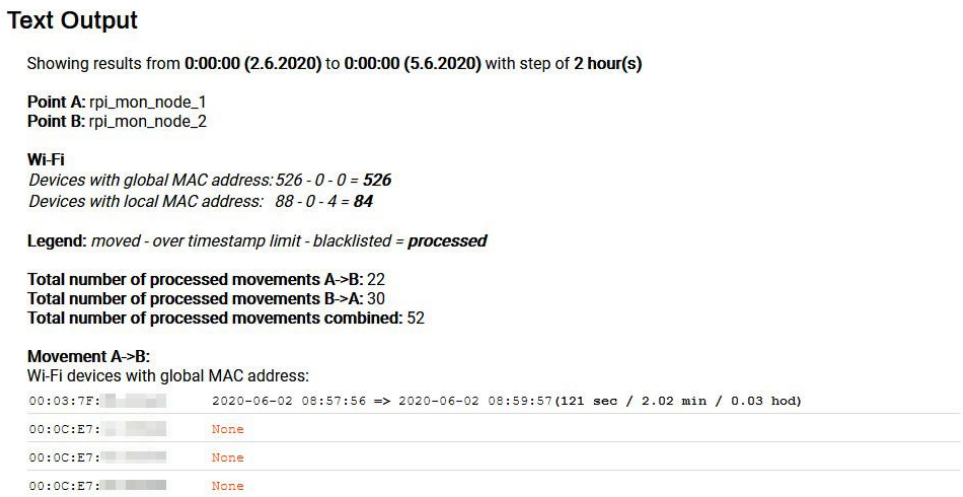


Obr. 31 Výstupný graf algoritmu *Movement – History*

Hodnota nastavenia *Absolute Maximum Threshold* bola nastavená tak, aby filtrovala tie presuny, ktoré neboli priame alebo sa osoba so sledovaným zariadením niekde po ceste zastavila na dlhšiu dobu. Bluetooth dáta nemohli byť použité, keďže bolo nastavené

filtrovanie podľa sily signálu. A taktiež nemohli byť použité ani 5 GHz Wi-Fi dáta, pretože zariadenie *node_1* nemalo žiadne 5 GHz Wi-Fi rozhranie.

Za povšimnutie stojí, že všetky spracované presuny boli zachytené počas dňa, čo sedí s predpokladom, že v tom čase sa ich zachytilo viac, a je teda väčšia šanca, že prejdú filtrami. Prerušované čiary spájajú body, ktoré majú medzi sebou jeden alebo viac krokov bez dát.



Obr. 32 Výstupný text algoritmu *Movement – History*

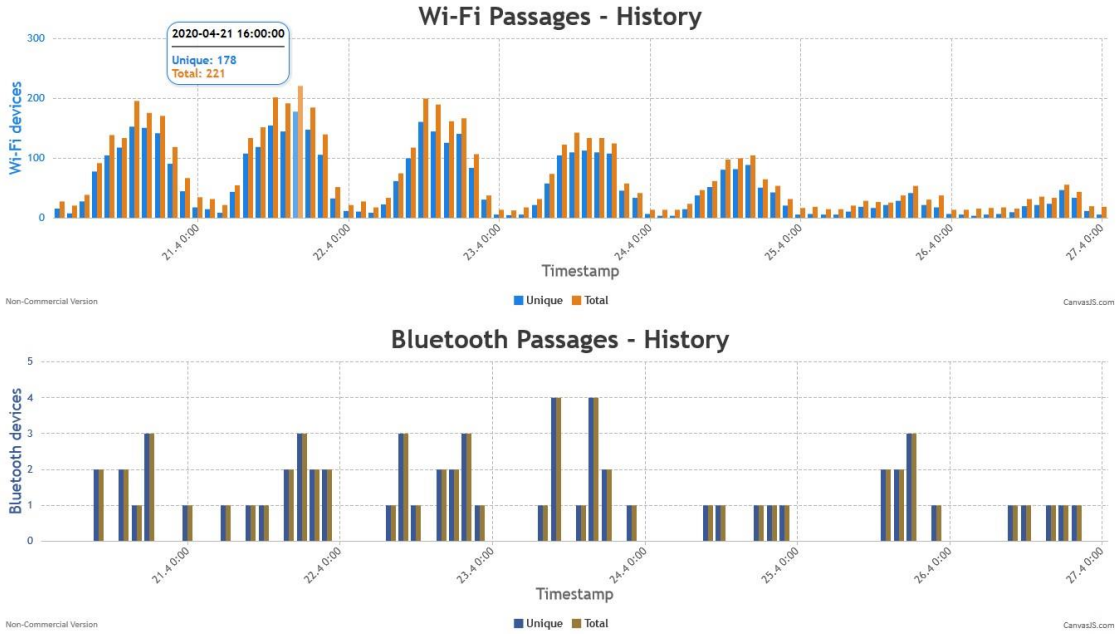
Na obrázku *Obr. 32* je zachytený úryvok výstupu v časti *Text Output*. Rovnakým spôsobom sú vypísané všetky spracované zariadenia. Výpis „*None*“ pri kľúči zariadenia znamená, že sa zariadenie síce našlo v databázach pre oba body, avšak nenašiel sa žiadny presun. V tomto prípade to bola pravda pre drvivú väčšinu zachytených zariadení, keďže bolo použité nastavenie *Power Limit*. Ako je možné vyčítať z obrázka, pri vyše 600 zachytených zariadeniach sa pri použitých nastaveniach našlo iba 52 presunov.

4.1.4 Algoritmus *Passages – History*

Na obrázkoch nižšie je zobrazený výstup algoritmu s nasledujúcimi nataveniami:

- **Source Database(s):** rpi_mon_node_1
- **Time Range:** From 2020-04-20 00:00:00 To 2020-04-27 00:00:00
- **Time Step:** 2 Hour(s)
- **Threshold:** 10 Minute(s)
- **Timestamp Limit:** nie
- **Show Data:** Wi-Fi a Bluetooth
- **Wi-Fi Standards:** 2.4GHz (802.11b/g)
- **Blaclisted Keys:** obrázok *Obr. 24*
- **Specific Keys:** nie

Grafy na obrázku *Obr. 33* zachytávajú počet unikátnych a celkových priechodov zachytených zariadení v dosahu zariadenia *node_1* z obrázku *Obr. 25* v priebehu jedného celého týždňa. V grafoch pre Wi-Fi zariadenia je vidieť výrazné navýšenie počtu priechodov zariadení počas pracovných dní a zároveň oveľa menší nárast počas víkendu. V piatok 24.4.2020 bolo zachytených zhruba o policu menej priechodov ako prvé tri dni v týždni. Z grafu pre Bluetooth zariadenia, taktiež na obrázku *Obr. 33*, ale aj z textového výstupu na obrázku *Obr. 34* je očividný priepastný rozdiel zachytených priechodov oproti Wi-Fi zariadeniam.



Obr. 33 Výstupné grafy algoritmu *Passages – History*

Text Output

Showing results from **0:00:00 (20.4.2020)** to **0:00:00 (27.4.2020)** with step of **2 hour(s)**

Wi-Fi
Devices with global MAC address: 1923 - 0 - 0 = ***1923***
Devices with local MAC address: 389 - 0 - 5 = ***384***
Bluetooth
Devices: 65 - 0 - 0 = ***65***

Legend: *passed - over timestamp limit - blacklisted =* ***processed***

Total number of devices passed: 2377

Wi-Fi devices with global MAC address:

```
00:00:00:      2020-04-20 15:04:30 |
00:03:7F:      2020-04-20 04:44:26 | 2020-04-20 10:38:12 | 2020-04-20 10:38:46 | 2020-04-21 05:17:24 | 2020-04-21 10:34:32 | 2020-04-22 04:45:26 |
               2020-04-22 10:38:16 | 2020-04-22 10:38:24 | 2020-04-22 10:38:50 | 2020-04-23 08:06:13 | 2020-04-23 08:18:47 | 2020-04-24 04:45:53 |
00:04:4B:      2020-04-23 16:49:08 | 2020-04-23 16:54:44 | 2020-04-23 16:55:08 | 2020-04-23 17:16:50 | 2020-04-23 17:16:51 |
00:07:AB:      2020-04-22 10:25:57 |
```

Obr. 34 Výstupný text algoritmu *Passages – History*

Na obrázku *Obr. 34* je zobrazený úryvok výstupu v časti *Text Output*. Rovnakým spôsobom je vypísaných všetkých 2377 spracovaných zariadení. Štatistiky spracovaných a nespracovaných zariadení je možné vyčítať z textu. V tomto prípade sa odfiltrovalo 5 zariadení s lokálnou MAC adresou, pretože ich zoznamy vyhľadávaných ESSID obsahovali iba ESSID zo zoznamu *blacklist* na obrázku *Obr. 24*.

4.2 Podrobnejšia analýza dát v MySQL databáze

Aj v týchto podkapitolách platí to, čo je napísané v úvode kapitoly *4.1 Spracovanie dát webovým rozhraním*.

4.2.1 Celkový počet identifikovaných zariadení

Monitoring prebiehal v dvoch obdobiach. Prehľad celkového počtu identifikovaných Wi-Fi a Bluetooth zariadení je uvedený v nasledujúcej tabuľke *Tab. 7*. V stĺpcoch Wi-Fi je uvedený súčet globálnych MAC adries a identifikovaných unikátnych PNL zoznamov s použitím nastavenia *blacklist* z obrázka *Obr. 24*. Tieto dáta bolo možné získať algoritmom *Passages – History* pri nastavenom časovom rozsahu na celé dané obdobie monitorovania.

Tab. 7 Prehľad celkového počtu zachytených zariadení

	2,4 GHz Wi-Fi		***5 GHz Wi-Fi***		***Bluetooth***	
	node_1	***node_2***	***node_1***	***node_2***	***node_1***	***node_2***
20.3.2020 – 29.4.2020	6129	-	-	-	228*	-
2.6.2020 – 5.6.2020	2266	1324	-	398	118	90
* monitorovanie Bluetooth bolo z neznámych príčin prerušené od 29.3 do 6.4						

Nižší počet zachytených zariadení pomocou *node_2* môže byť spôsobený tým, že sa nachádza ďalej od verejného chodníku. Avšak druhým dôvodom môže byť preťaženie Raspberry Pi z dôvodu použitia dvoch USB Wi-Fi adaptérov a zároveň použitia ako server. Pri preťažení sa perióda importovania logov do MySQL databázy predlžuje a nemusia sa tak zachytiť všetky zariadenia.

4.2.2 Zachytené prístupové body (AP)

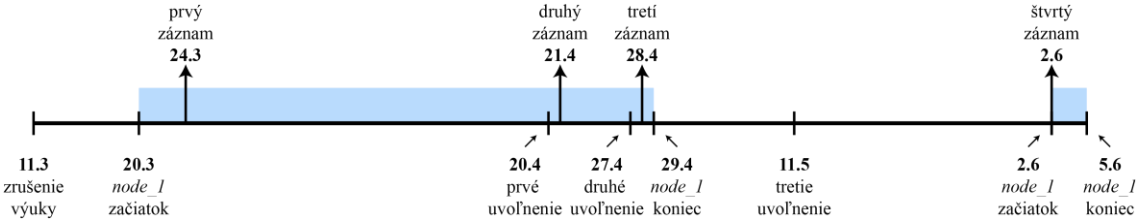
Aj keď prístupové body nie sú cieľ monitoringu predstaveného v tejto práci, ich zachytávanie prebiehalo rovnako ako zachytávanie iných zariadení. Vzhľadom na svoju trvalú prítomnosť v okolí monitorovacích zariadení, sú v MySQL databáze počty riadkov tabuľky `AccessPoints` väčšie ako v prípade tabuľky `Clients`. Konkrétne zaznamenalo zariadenie *node_1* počas oboch monitoringov vyše 850 tisíc riadkov v tabuľke `AccessPoints` zatiaľ čo v tabuľke `Clients` sa nazbieralo len vyše 326 tisíc riadkov.

Keďže tieto dáta nie sú nijako využívané v algoritmoch na webovom rozhraní a nie sú cieľom takéhoto typu monitoringu, bolo by v prípade reálne využitia vhodnejšie tieto dáta do MySQL databázy neimportovať. Hlavným prínosom by bolo ušetrenie výkonu zariadení a zároveň minimalizovanie zbieraných dát, čo sa v pravidlách GDPR považuje za pozitívne.

Zaujímavosťou v týchto dátach je 193 unikátnych globálnych MAC adries prístupových bodov s názvom „DPMBfree“. Ako je vyznačené v obrázku *Obr. 25*, zariadenie sa nachádzalo v blízkosti trasy dvoch liniek verejnej dopravy podniku DPMB.

4.2.3 Trendy spojené s pandémiou koronavírusu

Výhodou využitia univerzálneho JSON formátu na predávanie dát z algoritmov do grafov je, že tieto dáta sú prenosné do iných prostredí. Na spracovanie dát v tejto kapitole bol využitý MATLAB a venuje sa trendom v zachytených dátach vo vzťahu k aktuálnej pandémii koronavírusu SARS-CoV-2. Na obrázku nižšie je uvedená časová os epidemiologických opatrení a monitorovania [45].



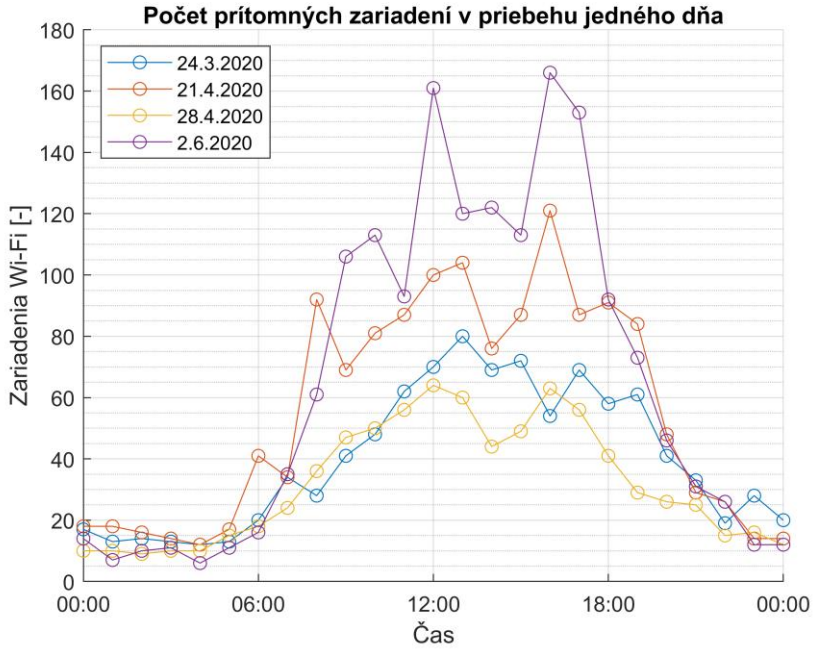
Obr. 35 Časová os epidemiologických opatrení a monitoringu

Nasledujúca tabuľka poskytuje krátky popis všetkých bodov na časovej osi [45]. Sadou dát sa myslí počet prítomných zariadení a taktiež počet unikátnych a celkových priechodov zariadení v priebehu jedného dňa. Vždy sa jedná o ten istý deň v týždni – utorok.

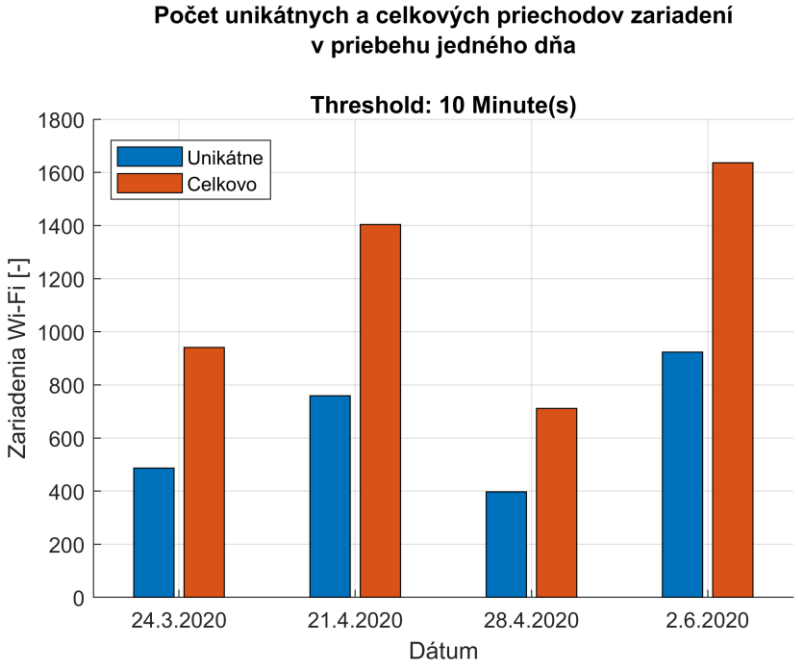
Tab. 8 Popis časovej osi epidemiologických opatrení a monitoringu

Dátum	***Udalosť***	***Popis***
11.3.2020	zrušenie výuky	Prezenčná výuka a prítomnosť študentov bola zakázaná
20.3.2020	*node_1* začiatok	Spustenie monitoringu zariadením *node_1*
24.3.2020	prvý záznam	Prvá sada dát pre túto kapitolu
20.4.2020	prvé uvoľnenie	Povolenie prítomnosti maximálne 5 osôb, iba záverečné ročníky
21.4.2020	druhý záznam	Druhá sada dát pre túto kapitolu
27.4.2020	druhé uvoľnenie	Povolenie prítomnosti maximálne 5 osôb, všetky ročníky
28.4.2020	tretí záznam	Tretia sada dát pre túto kapitolu
29.4.2020	*node_1* koniec	Ukončenie monitoringu zariadením *node_1*
11.5.2020	tretie uvoľnenie	Povolenie prítomnosti maximálne 15 osôb, všetky ročníky
2.6.2020	*node_1* začiatok	Opätovné spustenie monitoringu zariadením *node_1*
	štvrtý záznam	Štvrtá sada dát pre túto kapitolu
5.6.2020	*node_1* koniec	Ukončenie monitoringu zariadením *node_1*

Výsledky meraní sú uvedené na obrázkoch *Obr. 36* a *Obr. 37* nižšie. Pri hľadaní priechodov bolo použité nastavenie *Threshold* s hodnotou 10 minút. Vysvetlenie poklesu zachytených zariadení po druhom uvoľnení opatrení sa mi nepodarilo nájsť.



Obr. 36 Porovnanie prítomných zariadení počas pandémie



Obr. 37 Porovnanie priechodov zariadení počas pandémie

4.2.4 Komplikácie spojené so zoznamom PNL

Zdrojom dát v tejto podkapitole je monitoring zariadením *node_1* v období od 20.3.2020 do 29.4.2020. Pri bližšom pozorovaní zoznamov PNL pre lokálne MAC adresy sa ukázalo, že nielen že nemajú definované poradie jednotlivých ESSID, niektoré zoznamy sa pravdepodobne zachytia iba čiastočne. V nasledujúcej tabuľke je výpis z MySQL databázy, ktorá bola požiadaná o zobrazenie všetkých zoznamov PNL, ktoré obsahujú ESSID s názvom „`doma_krhov`“. Podľa názvu je možné usúdiť, že sa jedná o súkromný prístupový bod.

Tab. 9 Zoznam všetkých PNL pre ESSID `doma_krhov`

	Zoznam PNL
1.	`1e3420f4,doma_krhov,Blesk`
2.	`1e3420f4,doma_krhov,Blesk,vutbrno`
3.	`Blesk,doma_krhov`
4.	`Blesk,doma_krhov,1e3420f4`
5.	`doma_krhov,1e3420f4,Blesk`
6.	`doma_krhov,1e3420f4,Blesk,vutbrno`
7.	`vutbrno,1e3420f4,Blesk,doma_krhov`

Keďže algoritmy počítajú s tým, že zoznamy PNL nemusia byť v rovnakom poradí, budú riadky 1, 4 a 5 považovať za jedno zariadenie, tak isto ako riadky 2,6 a 7. Riadok 3 nemá anagram a bude považovaný za tretie zariadenie. Avšak pokiaľ sa na tento zoznam pozrie človek, môže usúdiť, že sa pravdepodobne jedná o jedno zariadenie, pričom algoritmy vyhodnotia rovnaký zoznam ako 3 rôzne zariadenia. Nie je možné s istotou povedať ktoré tvrdenie je pravdivé. Môže sa jednať o 3 zariadenia z jednej domácnosti, ktoré majú len trochu odlišné zoznamy PNL, ale taktiež sa môže jednať o 3 alebo dokonca 7 rôznych zariadení.

Tento zdroj neistoty v dátach by sa teda mohol pridať k zoznamu v kapitole *2.5.2 Komplikácie* a vytvára priestor na zlepšenie algoritmov, ktoré by mohli byť doplnené o ďalšie rozhodovacie mechanizmy.

4.2.5 Pomer globálnych a lokálnych MAC adries

V nasledujúcej tabuľke sú uvedené počty zaznamenaných MAC adries zariadením *node_1* v období od 20.3.2020 do 29.4.2020.

Tab. 10 Počet zachytených globálnych a lokálnych MAC adries

Typ MAC adresy	***Počet unikátnych záznamov***
Globálna	5086
Lokálna	54063

Aj napriek tomu, že globálne MAC adresy tvoria menej ako desatinu všetkých zachytených adries, stále tvoria hlavný zdroj dát v monitoringu, pretože z vyše 54 tisíc zachytených lokálnych MAC adries bolo pomocou zoznamu PNL identifikovaných len 1043 rôznych zariadení. Príčina nízkeho počtu identifikovaných zariadení s lokálnou MAC adresou je dôsledkom toho, že väčšina záznamov s lokálnou MAC adresou neobsahuje v zozname PNL žiadne ESSID. Podľa toho, čo je diskutované v predchádzajúcej kapitole *4.2.4 Komplikácie spojené so zoznamom PNL*, skutočný počet zariadení môže byť ešte menší.

Tab. 11 Počet ESSID v zozname PNL pre lokálne MAC adresy

Počet ESSID v zozname PNL	***Počet záznamov v MySQL databáze***
Nula	55122
Jeden a viac	5655
Ľubovoľný	*60777*

Pre porovnanie je uvedená rovnaká tabuľka aj pre globálne MAC adresy. Pri tých však algoritmy nepracujú s PNL ale priamo s MAC adresou.

Tab. 12 Počet ESSID v zozname PNL pre globálne MAC adresy

Počet ESSID v zozname PNL	***Počet záznamov v MySQL databáze***
Nula	123136
Jeden a viac	95980
Ľubovoľný	*219116*

Rozdiel v pomere unikátnych MAC adries a počtu záznamov v MySQL databáze pre globálne a lokálne MAC adresy je spôsobený tým, že pri lokálnej MAC adrese je len málo času na to, aby bola zachytená viackrát pričom pri globálnej MAC adrese tento problém nie je. Prispieva tomu tiež fakt, že trvalo prítomné zariadenia pripojené na prístupový bod vysielajú svoju globálnu MAC adresu a sú zachytené v podstate s každým importom dát do MySQL databázy.

5 ZÁVER

Hlavnou úlohou bakalárskej práce bolo vytvorenie monitorovacieho prostredia na platforme Raspberry Pi, ktoré dokáže zároveň zachytené dáta analyzovať. Zdrojom dát implementovaného monitoringu sú bezdrôtové komunikačné štandardy Wi-Fi a Bluetooth. Spoľahlivosť takéhoto monitoringu je diskutovaná v teoretickom úvode a na základe nadobudnutých skúseností a dát v kapitole *4 Výsledky monitoringu*. Vlastné riešenie sa delí na niekoľko častí a je implementované pomocou viacerých jazykov. Spolu ho tvorí 9525 riadkov kódu, z čoho jazyk PHP spolu s HTML zaberá 58,2% a jazyk Shell (*bash* skripty) zaberá 34,3%, ostatné riadky boli napísané v jazyku JavaScript alebo CSS.

Prvá časť, zachytávanie dát, prebieha automaticky a ovláda sa vytvorenými *bash* skriptami. Následne sú zachytené dáta ukladané do MySQL databázy, taktiež automaticky pomocou vytvorených *bash* skriptov. Poslednou časťou vlastného riešenia je spracovanie dát v MySQL databáze, ktoré prebieha pomocou vytvoreného webového rozhrania. Vzniklo tak komplexné monitorovacie prostredie s jednoduchým ovládaním, ktoré od užívateľa nevyžaduje znalosť jednotlivých použitých metód a súčastí. Interakcia s užívateľom prebieha výlučne pomocou vytvorených *bash* skriptov a webového rozhrania, ktoré obsahuje HTML formuláre na parametrizáciu výpočtových algoritmov. Monitorovacie zariadenie je tak možné ovládať kompletne na diaľku cez internet.

V rámci webového rozhrania boli implementované algoritmy na výpočet počtu zariadení v dosahu, priemerného času priechodu z bodu A do bodu B a počtu unikátnych a opakovaných priechodov zariadení v dosahu. Hlavným výstupom spracovania dát sú grafy, pričom niektoré detaily a štatistiky sú zobrazené v tabuľkách.

Monitorovacie prostredie bolo otestované v prevádzke po dobu 45 dní a výsledky monitoringu sú opísané v kapitole *4 Výsledky monitoringu*. Pri testovaní sa ukázalo, že použitá platforma Raspberry Pi 3B+ nemá dostatočný výkon na využitie maximálneho potenciálu, ktorý jej vytvorené monitorovacie prostredie dáva, a bolo by preto vhodnejšie použiť novšie, výkonnejšie modely.

Cesta, ktorou sa riešenie uberá sa ukázala ako schodná a po úpravách, aby spĺňala legislatívu Európskej Únie a bezpečnostné požiadavky, je jej využitie v praxi lákavé najmä kvôli nízkej cene platformy Raspberry Pi. V texte práce, tam kde je to relevantné, sú navrhované ďalšie vylepšenia ktoré by mohli byť implementované.

Literatúra

[1] MariaDB: Enterprise Open Source Database [online]. 699 Veterans Blvd, Redwood City, CA 94063, United States [cit. 2020-01-01]. Dostupné z: https://mariadb.com/

[2] PHP: Hypertext Preprocessor [online]. [cit. 2020-05-19]. Dostupné z: https://www.php.net/

[3] The Apache Software Foundation [online]. The Apache Software Foundation 401 Edgewater Place, Suite 600 Wakefield, MA 01880 U.S.A. [cit. 2020-05-19]. Dostupné z: https://www.apache.org/

[4] Airodump-ng [Aircrack-ng]. Aircrack-ng [online]. [cit. 2020-01-02]. Dostupné z: https://www.aircrack-ng.org/doku.php?id=airodump-ng

[5] NARDI, Tom. Bluelog. Github [online]. 22.6.2012 [cit. 2019-11-07]. Dostupné z: https://github.com/MS3FGX/Bluelog

[6] Installing operating system images. Raspberrypi.org [online]. Spojené kráľovstvo [cit. 2020-05-19]. Dostupné z: https://www.raspberrypi.org/documentation/installation/installing-images/README.md

[7] SSH (Secure Shell). Raspberrypi.org [online]. Spojené kráľovstvo [cit. 2020-05-19]. Dostupné z: https://www.raspberrypi.org/documentation/remote-access/ssh/

[8] Linux users. Raspberrypi.org [online]. Spojené kráľovstvo [cit. 2020-05-19]. Dostupné z: https://www.raspberrypi.org/documentation/linux/usage/users.md

[9] Understanding the Network Terms SSID, BSSID, and ESSID. Juniper Networks [online]. 23.1.2018 [cit. 2020-01-04]. Dostupné z: https://www.juniper.net/documentation/en_US/junos-space-apps/network-director3.3/topics/concept/wireless-ssid-bssid-essid.html

[10] LABUDA, Adam. Roaming ve WiFi sítích. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, 2018, 77 listů : ilustrace + 1 CD-ROM.

[11] Assigned Numbers. Bluetooth Technology Website [online]. [cit. 2020-01-04]. Dostupné z: https://www.bluetooth.com/specifications/assigned-numbers/

[12] Sanitized IEEE OUI Data. Linux Professionals [online]. [cit. 2020-01-04]. Dostupné z: https://linuxnet.ca/ieee/oui/

[13] Repozitár Samuell08/bachelors-thesis. GitHub [online]. [cit. 2020-05-21]. Dostupné z: https://github.com/Samuell08/bachelors-thesis

[14] Programming languages — C [PDF]. ISO/IEC 9899:201x. 2011 [cit. 2020-01-02]. Dostupné z: http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1570.pdf

[15] Mysql_secure_installation. MariaDB [online]. [cit. 2020-05-21]. Dostupné z: https://mariadb.com/kb/en/mysql_secure_installation/

[16] Repozitár aircrack-ng/rtl8812au. GitHub [online]. [cit. 2020-05-21]. Dostupné z: https://github.com/aircrack-ng/rtl8812au

[17] Predictable Network Interface Names. Systemd [online]. [cit. 2020-05-22]. Dostupné z: https://systemd.io/PREDICTABLE_INTERFACE_NAMES/

[18] Debian Wiki: NetworkInterfaceNames [online]. [cit. 2020-01-02]. Dostupné z: https://wiki.debian.org/NetworkInterfaceNames

[19] Alfa AWUS036ACH. ALFA Network Inc. [online]. [cit. 2020-05-22]. Dostupné z: https://www.alfa.com.tw/products_detail/1.htm

[20] PETRÁŠ, Samuel. Monitoring a analýza provozu v pásmu 2.4GHz. Brno, 2020. Dostupné také z: https://www.vutbr.cz/studenti/zav-prace/detail/123268. Semestrální práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Soběslav Valach.

[21] MariaDB Knowledge Base: mysqlimport. MariaDB: Enterprise Open Source Database [online]. [cit. 2020-01-02]. Dostupné z: https://mariadb.com/kb/en/mysqlimport/

[22] MariaDB Knowledge Base: mysqldump. MariaDB: Enterprise Open Source Database [online]. [cit. 2020-05-26]. Dostupné z: https://mariadb.com/kb/en/mysqldump/

[23] Industrial Scientific and Medical (ISM) Bands. Overview of Wireless Communication [online]. [cit. 2020-01-04]. Dostupné z: http://www.wirelesscommunication.nl/reference/chaptr01/dtmmsyst/ism.htm

[24] PISTOIA, G. Battery operated devices and systems: from portable electronics to industrial products. London: Elsevier, c2009. ISBN 978-0-444-53214-5.

[25] Wireless LAN (Wifi) Tutorial. Tutorial-Reports.com [online]. 18.2.2013 [cit. 2020-01-04]. Dostupné z: http://www.tutorial-reports.com/wireless/wlanwifi/index.php

[26] WiFi Logo. Wikimedia.org [online]. [cit. 2020-01-06]. Dostupné z: https://commons.wikimedia.org/wiki/File:WiFi_Logo.svg

[27] MARTIN, Jeremy, Travis MAYBERRY, Collin DONAHUE, Chadwick RIGGINS, Erik RYE a Dane BROWN. A Study of MAC Address Randomization in Mobile Devices and When it Fails. Proceedings on Privacy Enhancing Technologies [online]. 10.10.2017, 2017(4), 365–383 [cit. 2020-01-04]. DOI: https://doi.org/10.1515/popets-2017-0054. ISSN 2299-0984. Dostupné z: https://content.sciendo.com/view/journals/popets/2017/4/article-p365.xml

[28] ABDELRAHMAN, Ramia, Amin MUSTAFA a Ashraf OSMAN. A Comparison between IEEE 802.11a, b, g, n and ac Standards. IOSR Journal of Computer Engineering (IOSR-JCE) [online]. 2015, (Volume 17, 5, Ver. III), 26-29 [cit. 2020-01-04]. ISSN 2278-0661. Dostupné z: http://www.iosrjournals.org/iosr-jce/papers/Vol17-issue5/Version-3/D017532629.pdf

[29] FLICKENGER, Rob. Wireless Networking inthe Developing World: A practical guide to planning and building low-cost telecommunications infrastructure. Second Edition. Hacker Friendly, 2007.

[30] FARAHANI, Shahin. ZigBee wireless networks and transceivers. Boston: Newnes/Elsevier, c2008. ISBN 978-0-7506-8393-7.

[31] BAKKER, D. M. a Diane MCMICHAEL GILSTER. Bluetooth end to end. New York, NY: Newnes/Elsevier, 2002. ISBN 07-645-4887-5.

[32] BRUCE, Walter R. Wireless LANs end to end. New York: Hungry Minds Inc., c2002. ISBN 07-645-4888-3.

[33] Media Center. Bluetooth Technology Website [online]. [cit. 2020-01-03]. Dostupné z: https://www.bluetooth.com/media/

[34] SCHENKIRCH, Martin. What is Bluetooth Address (BD_ADDR). Bluetooth MAC Address Changer for Windows [online]. 8.2.2017 [cit. 2020-01-03]. Dostupné z: https://macaddresschanger.com/what-is-bluetooth-address-BD_ADDR

[35] OLIVEIRA, Luiz, Jano DE SOUZA, Daniel SCHNEIDER a Weiming SHEN. Mobile Device Detection Through WiFi Probe Request Analysis [online]. 2019 [cit. 2020-01-04]. Dostupné z: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8747391

[36] Check if Your Wireless Network Adapter Supports Monitor Mode & Packet Injection. Null Byte [online]. 11.12.2018 [cit. 2020-01-05]. Dostupné z: https://null-byte.wonderhowto.com/how-to/check-if-your-wireless-network-adapter-supports-monitor-mode-packet-injection-0191221/

[37] JONGERIUS, Silvan. WiFi-Tracking and Retail Analytics under the GDPR. In: TechGDPR [online]. 8.4.2019 [cit. 2020-05-29]. Dostupné z: https://techgdpr.com/blog/wifi-tracking-retail-analytics-gdpr/

[38] In and around the station. NS [online]. [cit. 2020-05-29]. Dostupné z: https://www.ns.nl/en/privacy/in-and-around-the-station.html

[39] CanvasJS [online]. [cit. 2020-05-31]. Dostupné z: https://canvasjs.com/

[40] DI LUZIO, Adriano, Alessandro MEI a Julinda STEFA. Mind your probes: De-anonymization of large crowds through smartphone WiFi probe requests. IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer

Communications [online]. IEEE, 2016, 2016, , 1-9 [cit. 2020-06-03]. DOI: 10.1109/INFOCOM.2016.7524459. ISBN 978-1-4673-9953-1. Dostupné z: http://ieeexplore.ieee.org/document/7524459/

[41] Bluetooth tools. BlackArch Linux [online]. [cit. 2020-06-04]. Dostupné z: https://blackarch.org/bluetooth.html

[42] Wireless tools. BlackArch Linux [online]. [cit. 2020-06-04]. Dostupné z: https://blackarch.org/wireless.html

[43] GEIER, Jim. Wireless LANs. Second Edition. 201 West 103rd St., Indianapolis, Indiana, 46290 USA: Sams Publishing, 2002. ISBN 0-672-32058-4.

[44] Aplikace Mapy.cz bojuje proti koronaviru COVID19 sdílením polohy. Seznam.cz [online]. [cit. 2020-06-04]. Dostupné z: https://www.seznam.cz/zastav-covid/

[45] Opatření přijatá vedením VUT v důsledku šíření koronaviru. Vysoké učení technické v Brně [online]. [cit. 2020-06-06]. Dostupné z: https://www.vutbr.cz/koronavirus

[46] Number of smartphone users worldwide from 2016 to 2021: (in billions). Statista [online]. 2019 [cit. 2020-01-05]. Dostupné z: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

[47] Aircrack-ng [online]. [cit. 2020-01-01]. Dostupné z: https://www.aircrack-ng.org/

[48] Kismet [online]. [cit. 2020-01-04]. Dostupné z: https://www.kismetwireless.net/

[49] Wireshark [online]. [cit. 2020-01-05]. Dostupné z: https://www.wireshark.org/

[50] Redfang. Penetration Testing Tools - Kali Linux [online]. [cit. 2020-01-05]. Dostupné z: https://tools.kali.org/wireless-attacks/redfang

Zoznam príloh

Príloha 1 - Štruktúra MySQL databázy

Tab. 13 Štruktúra tabuľky `AccessPoints`

Názov	*Dátový typ*	*Popis*
`BSSID`	`CHAR(17)`	Podľa konvencie je to MAC adresa prístupového bodu [9]
`first_time_seen`	`TIMESTAMP`	*Timestamp* prvého zaznamenania daného BSSID
`last_time_seen`	`TIMESTAMP`	*Timestamp* posledného zaznamenania daného BSSID
`channel`	`SMALLINT`	Kanál, na ktorom prístupový bod vysiela
`speed`	`SMALLINT`	Maximálna podporovaná rýchlosť prenosu v MB/s
`privacy`	`VARCHAR(20)`	Zabezpečenie (OPN = otvorený prístupový bod)
`cipher`	`VARCHAR(20)`	Šifrovací protokol
`authentication`	`VARCHAR(20)`	Overovací protokol
`power`	`TINYINT`	Zaznamenaná sila signálu. Hodnota -1 znamená, že bola zachytená len polovica komunikácie prístupový bod-klient bez odpovede Ak sú všetky hodnoty -1, znamená to, že použité rozhranie nepodporuje záznam sily signálu
`beacons`	`INT`	Počet zachytených rámcov *beacon frame*. Tieto rámce ohlasujú prítomnosť prístupového bodu a obsahujú informácie o ňom [10]
`IV`	`INT`	Počet zachytených paketov (v prípade WEP zabezpečenia počet inicializačných vektorov)
`LAN_IP`	`VARCHAR(15)`	IP adresa v rámci LAN siete
`ID_length`	`TINYINT`	Počet znakov ESSID
`ESSID`	`VARCHAR(127)`	Meno prístupového bodu [9] (prázdne pole značí skryté ESSID)
`passphrase`	`VARCHAR(127)`	Prelomené heslo pre daný prístupový bod (nepoužívané v tejto práci)
`standard`	`VARCHAR(3)`	Wi-Fi štandard 802.11*b,g* (2,4GHz) alebo 802.11*a* (5GHz)

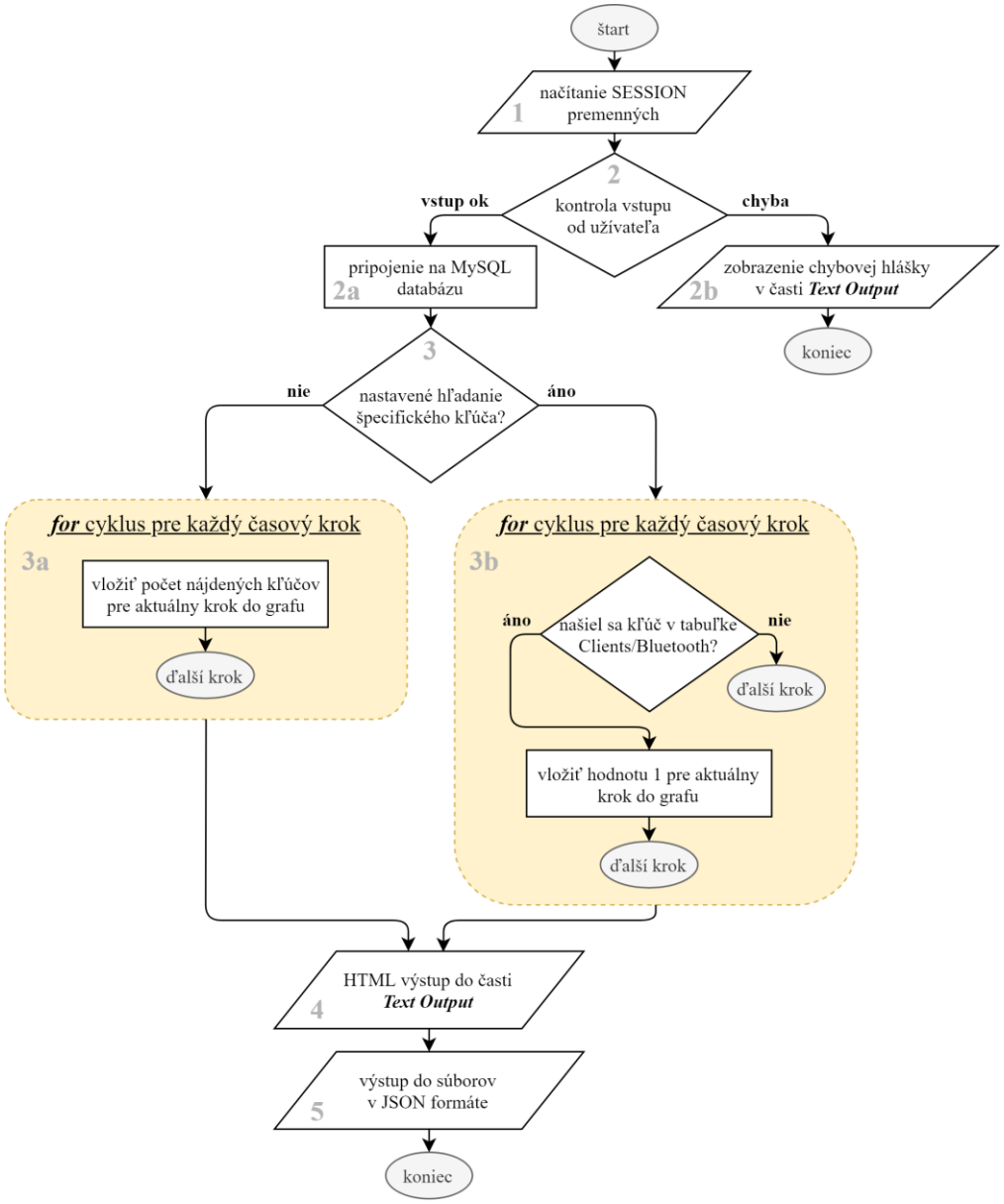
Tab. 14 Štruktúra tabuľky `Clients`

Názov	*Dátový typ*	*Popis*
`station_MAC`	`CHAR(17)`	MAC adresa klienta
`first_time_seen`	`TIMESTAMP`	*Timestamp* prvého zaznamenania danej MAC adresy
`last_time_seen`	`TIMESTAMP`	*Timestamp* posledného zaznamenania danej MAC adresy
`power`	`TINYINT`	Zaznamenaná sila signálu. Hodnota -1 znamená, že bola zachytená len polovica komunikácie klient-prístupový bod bez odpovede Ak sú všetky hodnoty -1, znamená to, že použité rozhranie nepodporuje záznam sily signálu
`packets`	`INT`	Počet zachytených paketov z daného zariadenia.
`probed_ESSIDs`	`VARCHAR(1023)`	Preferred Network List Prvá hodnota je MAC adresa prístupového bodu na ktorý je pripojené; `(not associated)` znamená, že nie je pripojené na žiadny prístupový bod
`standard`	`VARCHAR(3)`	Wi-Fi štandard 802.11*b,g* (2,4GHz) alebo 802.11*a* (5GHz)

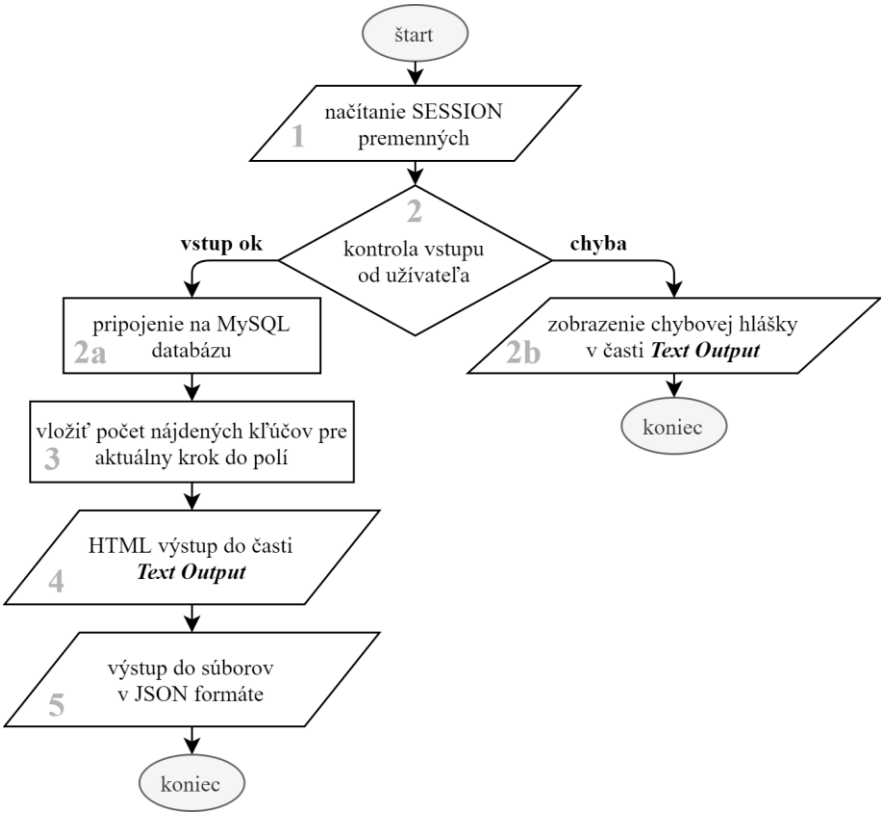
Tab. 15 Štruktúra tabuľky `Bluetooth`

Názov	*Dátový typ*	*Popis*
`last_time_seen`	`TIMESTAMP`	*Timestamp* posledného zaznamenania danej BD_ADDR adresy
`BD_ADDR`	`CHAR(17)`	BD_ADDR adresa zariadenia
`class`	`VARCHAR(127)`	Trieda zariadenia [11]
`class_detail`	`VARCHAR(127)`	Detail triedy zariadenia [11]
`OUI`	`VARCHAR(127)`	Výrobca zariadenia identifikovaný na základe BD_ADDR [12]
`device_name`	`VARCHAR(127)`	Užívateľom nastavené meno zariadenia

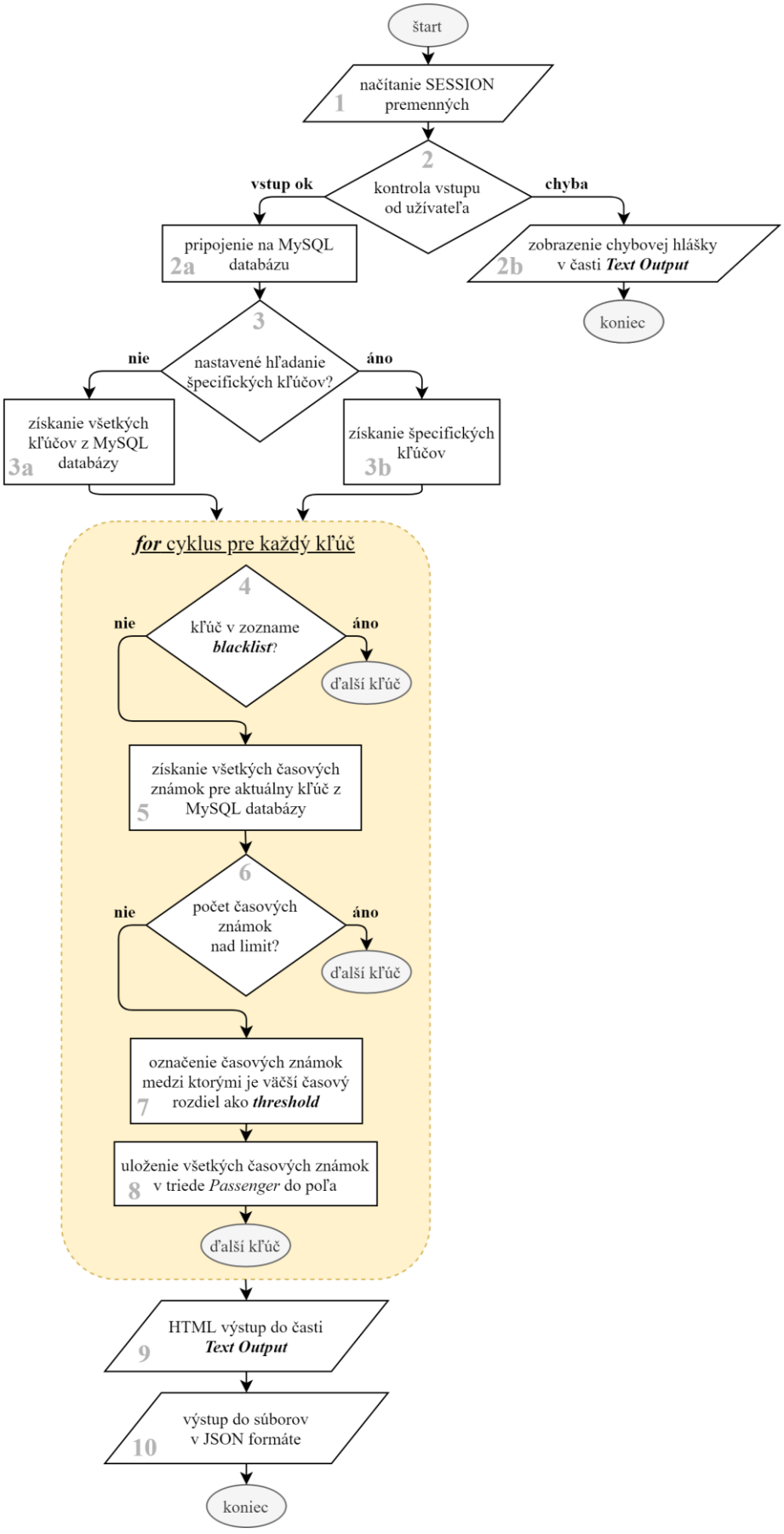
Príloha 2 - Vývojové diagramy algoritmov webového rozhrania



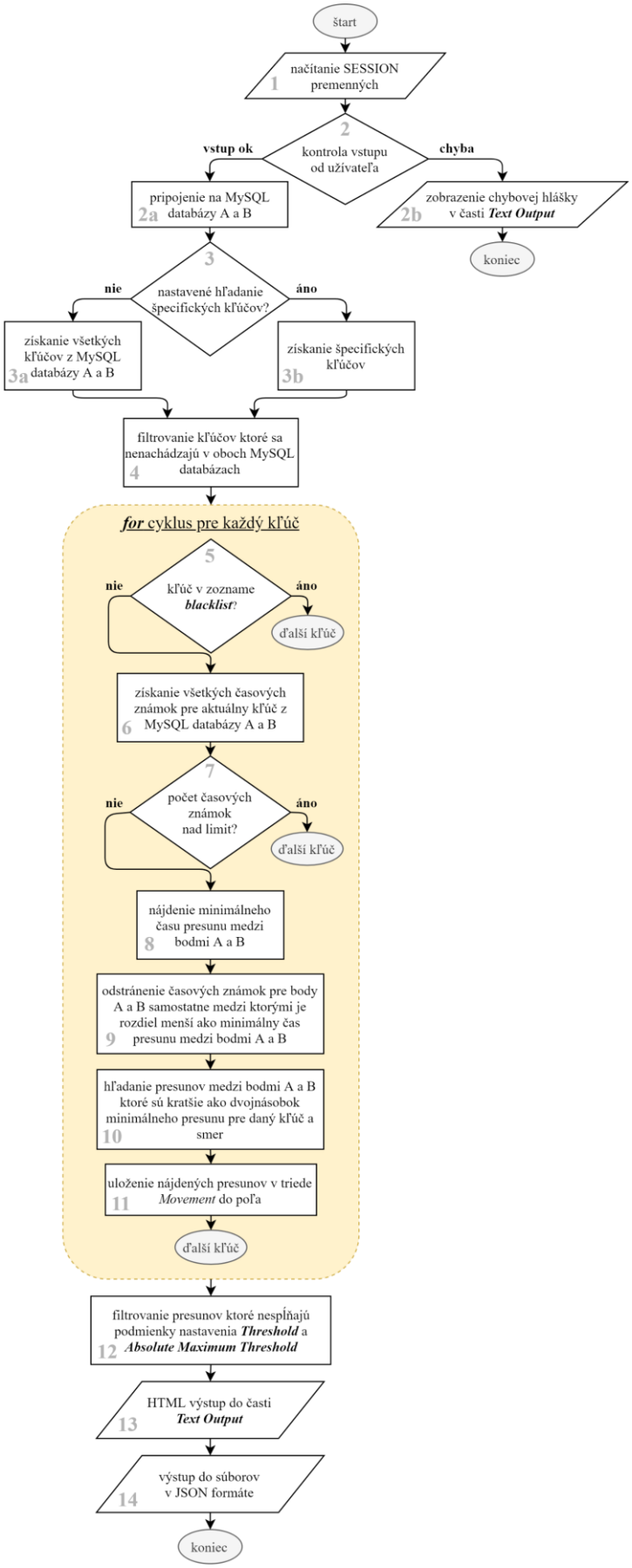
Obr. 38 Vývojový diagram algoritmu *In Range – History*



Obr. 39 Vývojový diagram algoritmu *In Range – Live data*



Obr. 40 Vývojový diagram algoritmu *Passages – History*



Obr. 41 Vývojový diagram algoritmu *Movement – History*

Príloha 3 - CD

Obsah CD:

BP_Petras.pdf	- Bakalárska práca
bachelors-thesis.zip	- Archív s verejným GitHub repozitárom k práci
monitoring	- Adresár obsahujúci definovaný adresárový strom