Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Detekce a sběr dat z mobilních zařízení v budovách

Plzeň 2018 Matěj Lochman

Místo této strany bude zadání práce.

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 17. května 2018

Matěj Lochman

Abstract

Detection and data collection from mobile devices in buildings

This thesis focuses on possible ways of how to detect mobile devices within buildings, special attention is given to wireless networks. Apart from essential WiFi communication principles, the study looks into MAC address randomization and its flaws. Data collection is realized by collecting Probe request frames from unassociated devices. Recent legislative changes (GDPR) were taken into account. Observed results were used to design and implement a web application, which is capable of collecting data of nearby devices and presenting them via simple reports in a web browser.

Abstrakt

Tato diplomová práce se věnuje možným způsobům detekce mobilních zařízení v budovách a soustředí se zejména na technologii WiFi. Jsou zde probrány principy WiFi komunikace a také je prozkoumána randomizace MAC adres. Data jsou sbírána na základě WiFi Probe requestů. Kromě toho je také zohledněna základní legislativa v oblasti sběru dat, včetně nového nařízení GDPR. Prozkoumáno bylo několik metod detekce a identifikace zařízení a tyto poznatky byly využity při návrhu a následné implementaci výsledné aplikace. Výsledkem práce je funkční webová aplikace, obsahující několik nástrojů, které umožňují sledovat chování pozorovaných zařízení.

Obsah

1 Úvod

Cílem této diplomové práce je navrhnout a implementovat aplikaci pro vyhodnocení dat sesbíraných o detekovaných mobilních zařízeních v budově. Aplikace umožní prohledávání a filtrování dat uložených v databázi a poskytuje uživateli možnost zobrazení relevantních údajů o rozpoznaných zařízeních.

V teoretické části práce je obsažen přehled dostupných způsobů detekce mobilních zařízení v budově (přičemž důraz je kladen zejména na pasivní metody detekce a využití technologie WiFi). Dále je zkoumána randomizace MAC adres a několik různých metod pro identifikaci zařízení, které ji využívají. Pozornost je věnována i základní legislativě v oblasti sběru dat, a to včetně dlouho připravovaného Obecného nařízení o ochraně osobních údajů (GDPR), týkajícího se všech členských států EU, které vstupuje v účinnost 25. 5. 2018.

Samotné implementaci aplikace předchází analýza požadavků a pečlivý návrh architektury aplikace. Aplikace je testována na několika různých úrovních a celou práci uzavírá kapitola věnující se vyhodnocení výsledků.

2 Detekce mobilních zařízení

2.1 Způsoby detekce mobilních zařízení v budovách

Existuje několik způsobů detekce mobilních zařízení na základě různých technologií a postupů, které je možné rozdělit do dvou základních kategorií, a to bez ohledu na to, zdali se jedná o detekci uvnitř budov, či jinde. Tyto dva způsoby se liší podle přístupu detekovaného zařízení.

První z nich vyžaduje aktivní přístup zařízení či jeho uživatele a odtud nese název aktivní detekce. Druhý přístup naopak nevyžaduje aktivní přístup detekovaného zařízení a je z jeho pohledu transparentní, tedy uživatel zařízení nezaregistruje, že je pozorován.

2.1.1 Aktivní detekce

Aktivní detekce vyžaduje určitou spolupráci od detekovaného zařízení či jeho uživatele. Uživatel tedy o detekci ví a musí vykonat určité kroky k tomu, aby se detekce uskutečnila.

Jednou z možností je na detekované zařízení umístit aplikaci, která o něm sbírá vybrané informace. Může sbírat například informace o poloze na základě GPS (pokud jej zařízení má) v určitých časech, informace o stavu baterie zařízení a podobně. Sesbírané informace pak mohou být pravidelně odesílány na server, kde jsou později různými způsoby zpracovány a vyhodnoceny.

Celý proces lze automatizovat tak, aby uživatel nemusel aplikaci spouštět pokaždé, když dorazí do budovy, ve které má detekce probíhat. Toho lze docílit například tak, že data začnou být automaticky sbírána po připojení ke konkrétní WiFi síti nebo tehdy, kdy detekovaná poloha zařízení odpovídá určitým GPS souřadnicím. Přestože uživatel aplikaci ručně nespustil, jedná se o aktivní přístup, jelikož tuto aplikaci na své zařízení nainstaloval za účelem vykonávání kódu umožňujícího samotný sběr dat.

Další možností aktivní detekce zařízení je to, že se zařízení aktivně připojí na WiFi síť umístěnou na požadované lokaci. Následně o něm mohou být zjišťovány informace, jako je IP adresa zařízení a aktivní porty či služby, pomocí síťových nástrojů (např. nmap), tj. bez nutnosti instalace aplikace na zařízení.

Vzhledem k požadavků zadavatele se tato práce nebude dále zabývat aktivním způsobem detekce mobilních zařízení.

2.1.2 Pasivní detekce

Pasivní detekce, na rozdíl od aktivní detekce, nevyžaduje explicitní spolupráci od detekovaného zařízení. Tento způsob detekce je tedy z pohledu detekovaného zařízení transparentní.

Pasivní detekce může být založena například na odposlechu režijních informací ve WiFi nebo Bluetooth sítích. Například u WiFi probíhá výměna informací ještě před samotným připojením do sítě – zařízení tedy nemusí být připojeno k žádné síti, avšak musí mít WiFi zapnutou. S pomocí přístupového bodu mohou být tyto informace monitorovány a ukládány pro pozdější zpracování. Informace, které lze tímto způsobem o zařízení zjistit, jsou však omezeny pouze na ty, které jsou sdíleny prostřednictvím daného protokolu do okolí. Toto představuje hlavní nevýhodu pasivní detekce oproti aktivní, která nabízí daleko více možností a informací, které lze o zařízeních zjišťovat.

Jednou z nich je například informace o poloze, pasivní přístup totiž neumožňuje získat tak citlivou informaci, jakou je GPS poloha. Existují však způsoby, pomocí kterých je možné polohu zařízení odhadnout. Jednou z možností je triangulace na základě síly přenášených signálů, s jejíž pomocí je možné dosáhnout přesnosti odhadu polohy zařízení až na vzdálenost jednotek metrů [12].

Pro detekování zařízení pomocí WiFi a Bluetooth je důležité pochopit princip a detaily komunikace mezi zařízeními. V následujících kapitolách budou tyto technologie přiblíženy.

2.1.3 Lokalizace zařízení

Jednou z vlastností, které lze o mobilních zařízení zjistit, je jejich poloha, respektive odhad jejich reálné polohy. Pro určení polohy existuje mnoho různých technologií, metrik pro odhad vzdálenosti od měřícího bodu a algoritmů pro výpočet samotné polohy.

Používané technologie

Existuje několik technologií umožňujících detekci mobilních zařízení, z nichž každá využívá specifický hardware a má své výhody i nevýhody. Tyto technologie se dále dělí podle pasivního, či aktivního přístupu. Pro tvorbu lokalizačního systému lze využít například Global Positioning System (GPS), Radio

Frequency Identification (RFID), Ultra-Wideband (UWB), infračervené záření, bezdrátové sítě (WLAN), Bluetooth a další technologie [20][14][24][25].

GPS

Mezi nejznámější technologie lokalizace zařízení patří nepochybně technologie GPS, která umožňuje detekovat lokaci zařízení s přesností na 2–10 metrů [15]. Tato čísla odpovídají ideálním podmínkám ve venkovním prostředí, avšak užití této metody v budovách dosahuje méně přesných výsledků, což je způsobeno horším šířením satelitního signálu, který zprostředkovává lokalizaci.

RFID

Lokalizace pomocí technologie RFID je založena na radiofrekvenční komunikaci mezi RFID čtečkou a RFID tagem, který může být aktivní, nebo pasivní. Pro komunikaci na vzdálenosti větší než jeden metr jsou využívány elektromagnetické vlny ve velmi vysokých frekvencích Ultra High Frequencies (UHF) [16].

Aktivní tagy jsou založeny na technologii Tag Talks First, což znamená, že vysílají své údaje do okolí. K tomu však potřebují vlastní zdroj energie, který jim umožňuje větší vzdálenosti čtení (od 15 m až do 100 metrů u UHF), než u pasivních tagů. Nevýhodou lokalizace zařízení za použití aktivních tagů je jejich větší velikost, způsobená přiloženou baterií, s čímž souvisí i vyšší pořizovací cena zařízení a nutná údržba kvůli výměně baterie [16].

Pasivní tagy používají metodu Reader Talks First, kdy RFID čtečka vyšle energii pomocí magnetického nebo elektromagnetického pole do pasivního tagu, který čeká, dokud není čtečkou vyzván k odpovědi, a pak odpoví. Pasivní tagy tedy nemají vlastní zdroj energie, a proto jsou levnější, skladnější a nevyžadují takovou údržbu. Jejich nevýhodou je ale kratší vzdálenost přenosu (0.5 m až do 10 metrů u UHF) [16].

Za nevýhodu technologie RFID lze považovat to, že všechna zařízení, která mají být lokalizována, musí být označena RFID tagem.

UWB

Ultra-Wideband zakládá vysílání velmi krátkých přenosech, většinou méně, než jednu nanosekundu. ve velmi širokých pásmech. UWB signály jsou oproti RFID vysílány po kratší dobu a umožňují vysílání signálu na několika pásmech zároveň. UWB signály také spotřebovávají méně energie než ostatní

radiofrekvenční tagy. Tyto signály také dobře procházejí zdmi a oblečením, naopak jsou rušeny kovovými a tekutými materiály. [20].

Bezdrátové sítě WLAN

Další možností lokalizace mobilních zařízení je využití lokálních bezdrátových sítí WLAN (IEEE 802.11). Velkou výhodou této technologie je její velká rozšířenost a poměrně nízká cena přístupových bodů (AP) potřebných pro vybudování bezdrátové sítě. Při tvorbě lokalizačního systému je tedy možné využít stávající bezdrátové sítě bez dalších investic do specializovaného hardwaru. Přesnost lokalizace se pohybuje od 3 do 30 metrů [20].

Jednou z prvních úspěšných realizací lokalizačního systému pomocí 802.11 byl RADAR. Experiment byl prováděn v třípatrové budově na jednom z pater o rozměrech 43.5 m na 22.5 m. Na této ploše o velikosti 980 m^2 byly umístěny pouze 3 přístupové body. Systém se podařilo optimalizovat tak, že dosahoval přesnosti detekce na 2 až 3 metry [12].

Bluetooth

Bluetooth je v mnoha ohledech velmi podobný bezdrátové WiFi technologii. Oproti ní dosahuje menších přenosových rychlostí a funguje na kratší vzdálenosti (10-15m), ale za to je méně náročný na energii a co se týče rozšířenosti je na tom podobně [20].

Metriky pro odhad vzdálenosti

Pro odhad vzdálenosti mobilního zařízení od měřícího zařízení (ve WiFi a Bluetooth sítích přístupový bod, u RFID technologie čtečka) se využívá několik následujících metrik [20].

Time of Arrival čas příchodu zprávy od zařízení

Time Difference of Arrival rozdíl dob šíření signálu k různým měřícím zařízením

Received Signal Strength síla signálu přijaté zprávy

Roundtrip Time of Flight doba přenosu zprávy od pozorovaného zařízení k měřícímu a zpět

Tyto metriky jsou většinou použity při výpočtu triangulace polohy zařízení. Dalším způsobem je tzv. Analýza scény, která využívá otisků prostředí pomocí rádiových frekvencí ještě před samotnou detekcí, tomu se říká offline

fáze. Během online fáze se zjišťuje poloha zařízení porovnáváním aktuálních signálů s otisky získaných v offline fázi.

2.1.4 Vybraná technologie

Pro další zkoumání byla zvolena detekce pomocí WiFi, jelikož tato technologie je z uvedených technologií nejrozšířenější, je cenově dostupná a byla preferovanou volnou zadavatele práce.

2.2 Detekce pomocí WiFi

Standard IEEE 802.11 (bezdrátové LAN sítě WLAN) patří do spojové vrstvy v modelu ISO/OSI nebo z pohledu TCP/IP do vrstvy síťového rozhraní. Tato vrstva se dále dělí na dvě podvrstvy – Media Access Control (MAC) a Logical Link Control (LLC), která tvoří rozhraní mezi MAC vrstvou a vrstvou síťovou, resp. spojovou.

Pro pasivní sbírání dat o zařízeních je důležité pochopit základní principy fungování MAC vrstvy, v níž jsou definovány standardy pro zařízení v těchto sítích, ať už se jedná o stanice (Station, STA) nebo přístupové body (Access Point, AP).

Obsah této kapitoly je čerpán z [1] a [18].

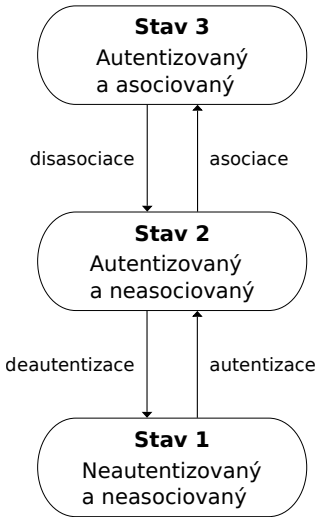
2.2.1 Proces navázání spojení

Hlavním účelem spojové vrstvy je přenos dat, s nímž souvisí i přenos režijních údajů zajišťujících vytvoření a šifrování spojení. Na rozdíl od LAN sítí je kvůli absenci fyzického propojení zařízení zapotřebí řešit nalezení a identifikování kompatibilních okolních sítí, a to již před samotným připojením (tzv. proces skenování, viz dále). Po jejich nalezení jsou zařízení vůči sobě autentizována a nakonec jsou spolu asociována. Zobrazení těchto stavů a přechodů mezi nimi je znázorněno na obrázku 2.1.

Každý přechod mezi těmito stavy je iniciován přenosem parametrů, které jsou předávány pomocí k tomu určených rámců, přičemž pro každý přechod existuje specifický typ rámce (viz. kapitola 2.2.2).

Skenování

Zařízení, které se chce připojit k nějaké síti, musí nejprve projít procesem skenování, při kterém dochází k identifikaci přístupových bodů v jeho okolí.



Obrázek 2.1: Proces navázání spojení

Proces skenování je na straně stanice, která hledá okolní sítě, definován několika parametry.

Mezi tyto parametry patří například BSSType, který určuje typ infrastruktury sítě, číslo kanálu, Basic Service Set Identifier (BSSID), SSID sítě, ScanType a další.

ScanType určuje přístup ke skenování okolních sítí a může být buď pasivní, nebo aktivní. Pasivní a aktivní skenování se liší v tom, který z konců iniciuje spojení, viz dále.

Pasivní skenování

Skenující zařízení při pasivním přístupu pouze periodicky naslouchá okolním přístupovým bodům na všech kanálech. Ty totiž opakovaně vysílají svoji přítomnost do okolí pomocí tzv. Beacon rámců. Stanice zpracovává přijaté parametry uchované uvnitř Beacon rámců od AP a zjišťuje tak kompatibilitu obou zařízení.

Z hlediska bezpečnosti je výhodou tohoto přístupu to, že po sobě zařízení nenechává žádnou stopu a nelze jej tak snadno sledovat jako při aktivním skenování. Nevýhodu však představuje delší doba trvání připojení, jelikož je závislá na dobách vysílání přístupového bodu. Také se může stát, že při přepínání na jiný kanál dojde k tomu, že stanice nezachytí některý z Beacon rámců [5].

Aktivní skenování

Aktivní skenování představuje opačný přístup, kdy skenující stanice aktivně periodicky vysílá svoji prezenci do okolí pomocí Probe Request rámců a umožňuje tak přístupovým bodům (nebo jiným zařízením v případě Ad-Hoc sítí[1]) odpovědět pomocí Probe Response rámců. Vysílání Probe Request rámců probíhá postupně na všech kanálech a může se dotazovat na jedno konkrétní, nebo všechna zařízení, která rámec obdrží. Cíl požadavku je stanoven pomocí identifikátorů sítě BSSID (MAC adresa cílového zařízení) nebo SSID (název cílové sítě), které určují jedno konkrétní zařízení, nebo všechna zařízení v dosahu s pomocí BSSID nastaveného na broadcastovou hodnotu (`FF:FF:FF:FF:FF:FF`). Na jeden požadavek tak může přijít více odpovědí od různých zařízení, dokonce i ze stejné sítě, a to v případě, že obsahuje více přístupových bodů.

Aktivní přístup je šetrnější na spotřebu baterie, protože naslouchání odpovědi probíhá jen krátce a také poskytuje rychlejší navázání kontaktu. Aktivní skenování zahlcuje síť režijními rámci více, než pasivní skenování. Výhodou je možnost připojit se ke skrytým sítím, které nevysílají svoje SSID do okolí pomocí Beacon rámců. Pro připojení k těmto sítím je zapotřebí explicitně poslat Probe Request s parametry požadované sítě.

Přístupové body podporují obě metody, což znamená, že vysílají Beacon rámce (pokud se nejedná o skryté sítě), aby mohly být detekovány pomocí pasivních zařízení a zároveň naslouchají Probe Request rámcům od aktivních zařízení. Jednotlivé stanice však používají různé přístupy, většinou v závislosti na operačním systému.

Autentizace

Autentizace je nutným krokem k ověření identity stanice a navázání spojení s přístupovým bodem. Standard 802.11 definuje autentizaci pomocí dvou základních přístupů – Open System (otevřený systém) a Shared Key (sdílený klíč).

Ověřování pomocí sdíleného klíče (neboli Wired Equivalent Privacy WEP) bylo dříve využíváno k šifrování spojení. Tato šifra však již byla prolomena a nadále se nepoužívá. Místo toho je používána autentizace na principu otevřeného systému pomocí rámců Authentication request a Authentication response. V této fázi tedy neprobíhá žádné šifrování. To je zajištěno bezprostředně po asociaci pomocí komplexnějších šifrovacích algoritmů, jako např. WPA2 [18].

[1] V Ad-Hoc sítích se jednotlivá zařízení spojí mezi sebou, přičemž jedno z nich zastupuje funkci AP.

Preautentizace

Jak již bylo řečeno autentizace je nutným krokem před asociací s přístupovým bodem, standard 802.11 však nic neříká o tom, že po autentizaci musí bezprostředně navazovat asociace. Stanice se mohou autentizovat vůči více přístupovým bodům zároveň, a pak když je vyžadována asociace, zařízení už se nemusí autentizovat. Tomuto procesu se říká preauntentizace.

Asociace

Asociace umožňuje udržet spojení mezi zařízením a přístupovým bodem tak, aby mohlo docházet k přenosu datových rámců. Každá stanice může být v jeden okamžik asociována pouze s jedním AP. Asociace nastává až po autentizaci a probíhá podobně pomocí požadavku a odpovědi, viz dále.

2.2.2 Typy 802.11 MAC rámců

Na úrovni MAC vrstvy jsou nositelem informace rámce, které se dělí do čtyř základních typů – Data, Control, Management a Reserved rámce. Každý z těchto typů se dělí na další podtypy, a to podle konkrétní funkce, kterou plní. Kompletní přehled všech typů a podtypů rámců je uveden v tabulce 2.1 [18].

Data rámce

Datové rámce slouží k přenosu samotných dat napříč bezdrátovou sítí. Většinou zapouzdřují protokoly vyšších vrstev TCP/IP zásobníku, jindy však mohou sloužit například k informování ostatních stanic o tom, že stanice nemá žádná data k vysílání.

Tyto rámce jsou posílány pouze za asociovaného stavu, nejsou tedy pro tuto práci podstatné a nebudou dále zkoumány.

Control rámce

Control rámce jsou nutnou režií potřebnou pro přenos ostatních rámců. Pomáhají doručovat Management rámce i Data rámce a jsou využívány jak stanicemi, tak přístupovými body. Slouží například k potvrzování rámců (Acknowledgement) po jejich přijetí a dále také poskytují prostředky pro šetření baterie přenosných zařízení díky Power Save-Poll (PS-Poll) rámcům. Tato zařízení totiž mohou šetřit baterii díky tomu, že na krátkou dobu vypnou své antény, pokud zrovna nechtějí vysílat. Po určité periodě se zařízení opět probouzí a vysílá zmíněný PS-Poll přístupovému bodu, který musí

Frame type	Frame subtype
Management frame	Association Request
	Association Response
	Reassociation Request
	Reassociation Response
	Probe Request
	Probe Response
	Beacon
	Announcement Traffic Indication Message (ATIM)
	Disassociation
	Authentication
	Deauthentication
	Reserved
Control frame	Reserved
	Power Save (PS)-Poll
	Request To Send (RTS)
	Clear To Send (CTS)
	Acknowledgement (ACK)
	Contention-Free (CF)-End
	CF-End + CF-Ack
Data frame	Data
	Data + CF-Ack
	Data + CF-Poll
	Data + CF-Ack + CF-Poll
	Null function (no data)
	CF-Ack (no data)
	CF-Poll (no data)
	CF-Ack + CF-Poll (no data)
	Reserved
Reserved frame	Reserved

Tabulka 2.1: Typy rámců 802.11

po dobu neaktivity stanice uchovávat rámce pro ni určené a probuzené stanici je poslat.

Management rámce

Management rámce umožňují stanicím a přístupovým bodům vyhledat další zařízení v okolí a navázat s nimi spojení. Právě tento proces je velmi důležitý při sbírání dat o blízkých mobilních zařízeních. Základní podtypy těchto rámců tvoří Beacon, Probe, Autentizační a Asociační rámce, viz tabulka 2.1.

Beacon rámce

Přístupové body periodicky upozorňují své okolí na svoji přítomnost pomocí tzv. Beacon rámců, které jsou vysílány na všech kanálech a ostatní stanice jsou schopné tyto rámce zachytit. Beacon rámce obsahují informace o parametrech a nastavení přístupového bodu, z nichž některé údaje jsou povinné.

Mezi povinné údaje v Beacon rámci patří:

Timestamp — časové razítko udávající dobu, po kterou je AP aktivní,

Beacon interval — interval, po kterém je Beacon vysílán, udává se v jednotkách Time Unit (1 TU = 1024 µs), výchozí hodnota bývá 100 ms,

Capability info — obsahuje bitové příznaky vlastností bezdrátové sítě,

SSID — Service Set Identifier – identifikátor bezdrátové sítě (u skrytých sítí, které nevysílají svoje SSID, je tato hodnota nastavena na nulu),

Supported rates — udává podporované přenosové rychlosti AP.

Kromě těchto povinných údajů mohou Beacon rámce obsahovat i řadu nepovinných údajů, jako například informaci o tom, zda zařízení podporuje Quality of Services (QoS), v jaké zemi vysílá (každá země má svá specifická omezení na frekvence a síly přenášených signálů) a jiné, většinou hardwarově specifické, informace.

Probe rámce

Probe rámce se dělí na Probe request a Probe response, první z nich slouží stanicím k objevení okolních přístupových bodů. Zařízení, které obdrží Probe request pak odpoví pomocí Probe response, ve které jsou informace o schopnostech daného AP.

Asociační rámce

Asociace nastává až po autentizaci a probíhá podobně jako u Probe rámců pomocí požadavku a odpovědi. Stanice vyšle rámec Association request konkrétnímu AP, ke kterému se chce připojit. Tento rámec udává SSID a další informace o síťové kartě. Přístupový bod přijme tento rámec a v případě úspěchu odpoví rámcem Association response. V opačném případě odpoví Disassociation rámcem a tím stanici odmítne. K přijetí dojde pouze pokud jsou obě zařízení kompatibilní a stanice již byla autentizována – v tomto případě AP přiřadí stanici asociační identifikátor a umožní jí přístup k síti.

Dalším typem asociačních rámců jsou tzv. Reassociation rámce, které jsou využívány především v sítích s více přístupovými body. Při pohybu stanice se mění i vzdálenosti od jednotlivých AP a tím se mění i síla přijímaného signálu. V určitý okamžik se proto vyplatí odpojit se od současného AP a připojit se k bodu se silnějším signálem, čehož je dosaženo právě pomocí Reassociation rámců, fungujících analogickým způsobem na základě požadavků a odpovědí. Celý tento proces je z pohledu uživatele transparentní [18].

2.2.3 Probe Request rámce

Pro účely této práce je důležitá zejména fáze před samotnou asociací zařízení, tedy informace, které jsou šířeny, než proběhne navázání spojení. Tyto informace jsou přenášeny pomocí Beacon rámců a Probe request a response rámců. Konkrétně lze pro detekci okolních zařízení využít jen Probe request rámce, jelikož Beacon rámce jsou v režii přístupových bodů a ty nejsou primárním cílem detekce.

Struktura rámce

Obsah rámce je popsán v tabulce 2.2, která je převzata z [1]. Pořadí prvků v této tabulce odpovídá pořadí, ve kterém by se měly informace v rámci vyskytovat.

Information	**Notes**
SSID	If dot11MeshActivated is true, the SSID element is the wildcard value
Supported rates	
Request information	The Request element is optionally present if dot11MultiDomainCapabilityActivated is true.

Information	Notes
Extended Supported Rates	The Extended Supported Rates element is present if there are more than eight supported rates, and is optionally present otherwise.
DSSS Parameter Set	The DSSS Parameter Set element is present within Probe Request frames generated by STAs using Clause 16, Clause 17, or Clause 19 PHYs if dot11RadioMeasurementActivated is true. The DSSS Parameter Set element is present within Probe Request frames generated by STAs using a Clause 20 PHY in the 2.4 GHz band if dot11RadioMeasurementActivated is true. The DSSS Parameter Set element is optionally present within Probe Request frames generated by STAs using Clause 16, Clause 17, or Clause 19 PHYs if dot11RadioMeasurementActivated is false. The DSSS Parameter Set element is optionally present within Probe Request frames generated by STAs using a Clause 20 PHY in the 2.4 GHz band if dot11RadioMeasurementActivated is false.
Supported Operating Classes	The Supported Operating Classes element is present if dot11ExtendedChannelSwitchActivated is true.
HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented attribute is true.
20/40 BSS Coexistence	The 20/40 BSS Coexistence element is optionally present when the dot112040BSSCoexistenceManagementSupport attribute is true.
Extended Capabilities	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero.
SSID List	The SSID List element is optionally present if dot11MgmtOptionSSIDListActivated is true.
Channel Usage	The Channel Usage element is optionally present if dot11MgmtOptionChannelUsageActivated is true.
Interworking	The Interworking element is present if dot11InterworkingServiceActivated is true.

Information	Notes
Mesh ID	The Mesh ID element is present if dot11MeshActivated is true.
Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements.

Tabulka 2.2: Obsah Probe Request rámce

2.2.4 Získání údajů z Probe Request rámců

Získání hodnot z Probe request rámců vyžaduje přístup k údajům z nízké úrovně komunikace bezdrátových zařízení.

2.2.5 Management Information Base

Jedním z nejčastějších způsobů, jak monitorovat síťová zařízení, je pomocí Management Information Base (MIB).

MIB je databáze, která kromě monitorování síťových prvků umožňuje i jejich správu. Databáze tvoří stromovou hierarchii, kde každý z objektů obsahuje jednoznačný identifikátor (OID). Na základě těchto OID jsou objekty z databáze čteny a spravovány pomocí protokolu Simple Network Management Protocol (SNMP). Tento protokol využívá ke správě zařízení dvě části. Jednou je agent, který se vyskytuje na straně pozorovaného zařízení a zprostředkovává informace pro druhou část (monitorovací stranu), která sbírá informace o spravovaném zařízení a umožňuje jej monitorovat.

IEEE 802.11 MIB

IEEE definuje 802.11 MIB, která uchovává informace o parametrech a vlastnostech vázajících se k vrstvě síťového rozhraní bezdrátových sítí. Tato databáze definuje vlastnosti[2], které zařízení musí podporovat, aby se mohlo účastnit komunikace v těchto sítích. Z této databáze lze získat informace o přenášených Beacon a Probe rámcích, jako například *dot11BeaconPeriod*, což je perioda, po které jsou rámce odesílány, nebo *dot11DesiredSSID*, což je požadované SSID uvedené v Probe request rámci.

Aby bylo možné přistoupit k těmto informacím přímo u detekovaných zařízení, musel by na nich být umístěn SNMP agent. To je však v rozporu s požadavky této práce, konkrétně s pasivní detekcí zařízení ve smyslu, že není potřeba je na detekci předem připravovat (viz kapitola 2.1.2).

[2]http://www.ieee802.org/11/802.11mib.txt

Potřebné informace tedy musí být sbírány z přístupových bodů. Výrobci jednotlivých zařízení většinou definují svoji MIB a poskytnuté údaje se mohou v určitých případech lišit. Záleží tedy na výrobci, zda umožňuje sbírat informace o Probe request rámcích z okolních zařízení.

MIB na přístupových bodech

Pro tuto práci budou využity routery Mikrotik[3], které poskytují tyto MIB:

- MIKROTIK-MIB
- MIB-2
- HOST-RESOURCES-MIB
- IF-MIB
- IP-MIB
- IP-FORWARD-MIB
- IPV6-MIB
- BRIDGE-MIB
- DHCP-SERVER-MIB
- CISCO-AAA-SESSION-MIB
- ENTITY-MIB
- UPS-MIB
- SQUID-MIB

Žádná z těchto MIB neumožňuje získat informace o příchozích probe request rámcích. Pro jejich získání tedy musí být využito jiných prostředků.

[3]https://wiki.mikrotik.com/wiki/Manual:SNMP#Management_information_base_.28MIB.29

2.2.6 Sniffování paketů

Vzhledem k tomu, že se informace o Probe request rámcích nevyskytují v MIB databázích zařízení, ani se jinak nelogují, je zapotřebí sesbírat vybrané informace přenášené přímo při komunikaci v bezdrátových sítích. Jednou z možností jak pozorovat komunikaci v sítích je použít analyzátor paketů známý jako paketový sniffer. Jedná se o program nebo hardware, který naslouchá probíhající komunikaci a zachytává všechny průchozí pakety, které může následně filtrovat nebo různými způsoby analyzovat a dále zkoumat.

Promiskuitní a monitorovací režim

Aby zařízení mohlo odposlouchávat komunikaci, musí mít síťový adaptér, který podporuje promiskuitní nebo monitorovací režim. Oba tyto režimy způsobí to, že jsou přijímány všechny pakety bez ohledu na jejich příjemce. V běžném režimu síťový adaptér akceptuje pouze pakety pro něj určené.

Promiskuitní režim je možné využít jak u běžných síťových adaptérů, tak u bezdrátových, a vyžaduje připojení k síti, ve níž odposlouchává komunikaci. Monitorovací režim je oproti tomu limitován jen pro bezdrátové karty a odposlouchává veškerou okolní komunikaci aniž by byl připojen k některému přístupovému bodu. Aby bylo možné zachytit nízkoúrovňové režijní zprávy, je zapotřebí právě monitorovacího režimu. Promiskuitní režim totiž může nesprávně překládat 802.11 hlavičky na ethernetové, kvůli čemuž pak dochází k odstranění informací specifických pro bezdrátové sítě [11].

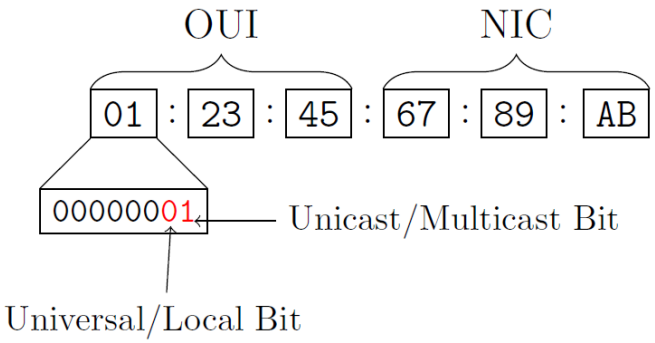
2.3 Identifikace zařízení

Pro sběr a analýzu metadat sledovaných zařízení je nezbytné získané údaje označit, aby bylo možné je vztáhnout k některému ze zařízení. Bez identifikátoru by například sledování pohybu jednoho zařízení, byť po omezenou dobu, nebylo možné. Tato kapitola se zabývá právě výběrem vhodného identifikátoru na základě údajů ze získaných Probe request rámců.

2.3.1 MAC adresa

V každém Probe request rámci je posílána MAC adresa zařízení v hexadecimálním tvaru, která identifikuje zařízení na linkové vrstvě. Je tvořena 48 bity a skládá se z několika částí. První tři oktety označují výrobce zařízení a označují se jako Organizationally Unique Identifier (OUI). Seznam těchto identifikátorů má na starost Institute of Electrical and Electronics Engineers (IEEE), který výrobcům umožňuje tyto identifikátory koupit a zaregistrovat.

Zbytek adresy může výrobce libovolně přiřadit za předpokladu, že nepoužije stejnou adresu pro dvě různá zařízení.



Obrázek 2.2: Struktura MAC adresy

OUI dále uchovává dva příznaky, a to v posledních dvou bitech prvního oktetu, viz obrázek 2.2 [21]. Důležitý je zejména první z nich, který rozlišuje, zda se jedná o globální, či lokální adresu, a nazývá se Universal/Local bit. Lokální adresy nemusí být unikátní a většinou se používají pouze po přechodnou dobu. Používají se například pro peer-to-peer sítě, mobilní hotspoty nebo mohou být použity pro randomizaci MAC adres.

Kromě OUI existuje také tzv. Company Identifier (CID), který si společnosti také registrují prostřednictvím IEEE. Na rozdíl od předchozího má tento identifikátor vždy nastavený local bit.

2.3.2 Randomizace MAC adres

V posledních letech vzniklo několik systémů umožňující sledování a profilování zařízení, případně jejich uživatelů. Tyto systémy se zakládají právě na identifikaci pomocí MAC adres. V extrémním případě masového nasazení takovýchto systémů se společnou databází by díky neměnnému unikátnímu identifikátoru mohl vzniknout profil uživatele, ze kterého by šlo vypozorovat současnou polohu a veškerou historii jeho pohybu. Výrobci operačních systémů se proto snaží adresy randomizovat a chránit tak soukromí svých uživatelů.

Randomizací MAC adres se v posledních letech zabývá několik studií [27], [23], [21]. Způsoby, jakými je řešena, se liší podle operačního systému, případně výrobce, či modelu zařízení. Je to zapříčiněno tím, že zatím neexistuje specifikace, která by deklarovala, jak má randomizace MAC adres probíhat.

Randomizace probíhá jen za neasociovaného stavu zařízení a týká se zejména Probe request rámců. Když zařízení aktivně vyhledává okolní přístupové body, je používána randomizovaná adresa s lokálním bitem, jakmile je však zařízení asociováno s přístupovým bodem, je opět používána globální adresa. Tím pádem nejsou ovlivněny ostatní síťové nástroje závislé na identifikaci zařízení podle MAC adresy (např. přístup do sítě na základě MAC) [21].

Android

Android přidal randomizaci MAC adres od verze 6.0[4]. Pro starší verze operačního systému existují aplikace, které randomizaci umožňují, avšak jen s právy superuživatele [27]. Dá se tedy předpokládat, že většina zařízení s verzí tohoto operačního systému menší než 6.0 randomizaci nepodporuje.

iOS

Prvním operačním systémem podporujícím randomizaci byl iOS, ve kterém je implementována od verze 8[5].

Windows

Windows podporuje randomizaci od verze 10 a na rozdíl od ostatních operačních systémů používá randomizovanou adresu i pro připojování k sítím. Pro každou síť je používána jen jedna randomizovaná adresa, aby mohlo být dodrženo ověřování přístupu do sítě pomocí MAC adresy [4].

Linux

Linux přidal randomizaci MAC adres od verze jádra 3.18 a randomizuje adresu pro každou skenovací iteraci[6].

2.3.3 Otisky zařízení

Jedním ze způsobů, jakým lze zařízení i přes randomizaci adres identifikovat, je na základě tzv. Information elements často také označovaných jako tagy.

[4]https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html

[5]https://support.apple.com/cs-cz/HT201395

[6]https://github.com/torvalds/linux/commit/effd05ac479b

Tyto informace (viz tabulka 2.2) obsahují podle studie [27] dostatečné množství náhodné informace, aby bylo možné na jejich základě zařízení částečně rozlišit. Postup, jakým vytvořit otisk zařízení, je uveden v [19].

Studie [21] však upozorňuje na nedostatky této metody, jako například to, že experimenty s otisky zařízení vznikly nad datasetem bez randomizace (kvůli možnosti ověření správnosti otisků), což poněkud zkresluje výsledky účinnosti této metody. Při vyhledávání sítě totiž zařízení posílá zprávy s různými hodnotami Information elements (většinou dvě konfigurace), a to při používání globální, ale i randomizované adresy. To znamená, že jedno zařízení vysílá více různých otisků. Jednou z konfigurací, která je zpravidla posílána, je zpráva obsahující minimální množství těchto elementů. To má za následek to, že většina zařízení posílá záznam s totožným otiskem.

Android od verze 8.0 Oreo[7] přistupuje k této problematice podobným způsobem a snaží se minimalizovat zanechávanou stopu tak, aby ho podobné systémy nedokázaly identifikovat (aktuálně jen pro zařízení Pixel, Pixel XL and Nexus 5x).

2.3.4 Zohlednění doby příchodů zpráv

Kromě tohoto přístupu zabývajícího se samotným obsahem posílaných zpráv se objevilo několik studií, které se věnují dobám příchodů zpráv. Ve studii [23] byla dokázána 75% úspěšnost rozpoznání zpráv z původního zařízení i přes to, že v nich byla využita randomizovaná adresa.

K rozlišení jednotlivých zařízení bylo využito dob mezi příchody zpráv v rámci tzv. dávky Probe requestů. Dávka je tvořena několika požadavky rychle vyslanými v krátké době (10 ms). Základem byl předpoklad, že se prodleva mezi těmito požadavky u jednotlivých zařízení liší. Zmíněné úspěšnosti se podařilo dosáhnout po vytvoření vektoru příznaků na základě této vlastnosti a použitím metod strojového učení nad dostatečně velkým datasetem.

Důležitým poznatkem je to, že jsou sledovány dávky, které jsou tvořeny po sobě jdoucími požadavky s každou ze sítí, které zařízení zná. Pokud tedy zařízení tento seznam neuchovává, nebo ho nevyužívá pro vyhledávání sítí, pak tato metoda nemusí fungovat.

[7] https://android-developers.googleblog.com/2017/04/changes-to-device-identifiers-in.html

2.3.5 Zohlednění pořadí příchodů zpráv

Posílané zprávy obsahují pořadí zprávy od daného zařízení, které však kvůli randomizaci nemusí nutně navazovat. Dalším z přístupů pro identifikaci zařízení je zkoumání posloupnosti těchto identifikátorů ve snaze najít nějaký vzor chování, podle kterého by se dalo zařízení rozlišit.

2.3.6 Zohlednění polohy zařízení

Dalším z možných přístupů je zohlednění polohy zařízení, kterou je možné získat na základě síly signálu od více AP. Dá se předpokládat, že záznamy pořízené v podobnou dobu na podobném místě pravděpodobněji souvisí se stejným zařízením, než záznamy pořízené na opačné straně.

Předpokladem pro tuto techniku je odladěný systém, který spolehlivě dokáže určit polohu zařízení a také dostatečně homogenní prostředí, kde se zařízení pohybují předvídatelnou rychlostí.

2.3.7 Nedostatky randomizace MAC adres

Randomizace MAC adres má kromě toho, že lze aplikovat výše zmíněné metody pro částečnou identifikaci zařízení, další nedostatky, a to zejména její rozšířenost.

Studie [22] z počátku letošního roku se zabývá rozšířeností randomizace a kromě jiného uvádí, že podle datasetu z roku 2016 randomizaci používají necelá 3 % zařízení. Dnes se dá již samozřejmě předpokládat o něco vyšší využití.

Podle [21] většina zařízení s operačním systémem Android ve verzi podporující randomizaci ji z nějakého důvodu nevyužívá, což může být způsobeno nepodporovaným chipsetem nebo firmwarem. Dále bylo potvrzeno, že jsou občas posílány globální adresy i u zařízení podporujících randomizaci, a to například pokud má zařízení rozsvícený displej a z nějakého důvodu i v okamžiku příchozího telefonátu. Toto chování bylo pozorováno napříč všemi telefony s operačním systémem Android, který tvoří majoritní podíl trhu chytrých telefonů[8].

Vzhledem k počtu zařízení podporujících randomizaci a zmíněným nedostatkům při otiskování zařízení není randomizace v této práci dále řešena a je jejím možným dalším rozšířením. Záznamy s randomizovanou adresou lze snadno určit pomocí bitu lokální alokace adresy a při jejich zpracování je vynechat, nebo je zahrnout pouze do reportů, u kterých to dává smysl.

[8] http://gs.statcounter.com/os-market-share/mobile/worldwide

3 Legislativa v oblasti sběru dat

V bezdrátových sítích o sobě zařízení šíří informace za tím účelem, aby byly schopné se mezi sebou dorozumět a navázat spojení. Tyto informace jsou periodicky vysílány do okolí. Většina poskytovaných údajů je ryze technického charakteru, například se může jednat o hardwarové parametry zařízení (frekvence, podporované rychlosti přenosu apod.), díky kterým komunikující zařízení zjistí vzájemnou kompatibilitu a další parametry následné komunikace.

Tyto informace nejsou šifrovány a mohou být kýmkoli pasivně odposlechnuty, aniž by o tom uživatel zařízení věděl. Za určitých podmínek tedy může být důsledkem šíření těchto informací zásah do soukromí držitele zařízení, zejména pokud jde o jednoznačný identifikátor Media Access Control (MAC) společně s polohou zařízení. Každé zařízení má tento identifikátor, na jehož základě je možné sledovat vzory chování jeho uživatele. Nutno dodat, že existují nástroje[1], které umožňují změnit MAC adresu zařízení.

Hlavní citlivou informací, kterou lze o mobilním zařízení zjistit, je jeho poloha, a to z toho důvodu, že je velmi často spjata s polohou jeho uživatele. Podle studie [6] má 87 % uživatelů chytrých telefonů své zařízení neustále u sebe. Podobný závěr lze vyvodit u chytrých hodinek, náramků a jiné nositelné elektroniky, která je již svojí podstatou svázána s polohou svého uživatele. Informace o poloze (či přibližné poloze) může být velkým zásahem do soukromí a lze ji zneužít.

Jednou z přenášených informací, která může napovědět o poloze a chování jedince, je jednoznačný identifikátor sítě Service Set Identifier (SSID). Zařízení si totiž mohou uchovávat seznam posledních připojených sítí, respektive jejich SSID, viz kapitola 2.2. Tento seznam je pak vysílán do okolí ve snaze najít preferovanou síť.

Vzhledem k současné rozšířenosti WiFi sítí, kdy většina veřejných míst, jako jsou obchodní centra, letiště, kavárny a jiné podniky, poskytuje připojení k síti, lze na základě tohoto seznamu vypozorovat, jaká místa uživatel navštěvuje. Název sítě totiž často souvisí s její lokací, ať už jde o název podniku, nebo přímo název lokace (např. *OC Plzen Plaza* nebo *prg.aero-free*). Ze samotného názvu sítě lze pomocí databáze WiFi sítí, jako například Wi-

[1]Například https://github.com/alobbs/macchanger, či https://technitium.com/tmac.

GLE[2], zjistit její polohu a tedy lokalitu kde se uživatel nacházel.

Aby nedocházelo ke zneužívání těchto informací, mohou být některé z nich prohlášeny za osobní údaje. Na definici, sběr, uchovávání a následné zpracování osobních údajů se vztahuje legislativa platná v dané zemi. V České republice se touto problematikou zabývá Úřad pro ochranu osobních údajů[3], který vydal Zákon o ochraně osobních údajů definující jak s těmito údaji zacházet.

3.1 Zákon o ochraně osobních údajů

V roce 2000 byl deklarován Zákon č. 101/2000 Sb., o ochraně osobních údajů, který je průběžně upravován a doplňován a jeho aktuální znění [7] je účinné od 1. července 2017.

Podle tohoto zákona se osobním údajem rozumí: *jakákoliv informace týkající se určeného nebo určitelného subjektu údajů. Subjekt údajů se považuje za určený nebo určitelný, jestliže lze subjekt údajů přímo či nepřímo identifikovat zejména na základě čísla, kódu nebo jednoho či více prvků, specifických pro jeho fyzickou, fyziologickou, psychickou, ekonomickou, kulturní nebo sociální identitu.*

3.1.1 Síťové identifikátory jako osobní údaje

Otázkou je, zdali jsou údaje zpracovávané v této práci osobními, či nikoliv. Nejblíže ke zmíněné definici má MAC adresa – ze své podstaty se jedná o jedinečný identifikátor. Jak již bylo zmíněno, existují nástroje, které ho umožňují upravit. Dokonce samotné operační systémy mobilních zařízení (např. Android a Apple iOS) se snaží zachovat anonymitu zařízení tím, že randomizují MAC adresy při vyhledávání okolních sítí proto, aby nebylo možné daná zařízení sledovat. Na základě těchto okolností by se mohlo zdát, že MAC adresa není jedinečným identifikátorem, který by umožnil identifikovat jedince, a tím pádem by se nejednalo o osobní údaj.

Přestože se randomizace MAC adres již využívá v praxi, jsou za osobní údaje považovány kromě MAC adres i síťové a jiné identifikátory, jako je například IP adresa nebo cookies. [26]

[2]https://wigle.net/

[3]https://www.uoou.cz/

3.1.2 Sběr a zpracování osobních údajů

Podle § 5 je správce povinen stanovit účel, ke kterému jsou osobní údaje zpracovány, a také prostředky a způsob jejich zpracování.

Správce může zpracovávat osobní údaje jen se souhlasem subjektu údajů. Bez tohoto souhlasu je může zpracovávat jen za podmínek, které nejsou relevantní pro tuto práci.

Uchovávání osobních údajů

Osobní údaje lze uchovávat jen po dobu nezbytně nutnou k účelu jejich zpracování. Po této době smí být uchovávány jen pro státní statistické služby, vědecké účely a účely archivnictví.

3.1.3 Anonymizace osobních údajů

Anonymní údaje nelze na rozdíl od osobních vztáhnout k identifikované či identifikovatelné osobě a nevztahuje se na ně zákon o ochraně osobních údajů. Mezi těmito údaji a subjektem údajů neexistuje vazba a nemůže být nikým za rozumných předpokladů obnovena. K anonymizaci osobních údajů dochází jen pokud jsou z nich odstraněny informace tak, aby byla splněna tato podmínka [26].

3.1.4 Pseudoanonymizace osobních údajů

Kromě anonymních údajů existují tzv. pseudoanonymizované údaje, u kterých existuje vodítko nebo způsob, jak znovu asociovat citlivé údaje s jejich subjektem. Na tyto údaje se musí nahlížet jako na osobní údaje a platí na ně zákon o ochraně osobních údajů.

O pseudoanonymizaci, neboli zdánlivou anonymizaci, se jedná například v případě, kdy jsou jména subjektů nahrazena identifikátorem, ale existuje klíč, pomocí kterého lze zjistit, jaké jméno souvisí s uvedeným identifikátorem.

Například častou snahou v některých systémech je anonymizovat MAC adresy pomocí hashovacích funkcí s tím předpokladem, že hashovací funkce je jednosměrná a z takto vygenerovaného klíče nelze zpět získat původní MAC adresu. Pokud je ale hashovací funkce známá, je možné k libovolné MAC adrese vygenerovat identifikátor a dohledat tak data týkající se zvolené adresy. Tímto hashováním tedy nedochází k anonymizaci, ale k pouhé pseudoanonymizaci, a tím pádem se pořád jedná o osobní údaj. Pro ověření

těchto informací byl přímo kontaktován Úřad pro ochranu osobních údajů, viz odpověď v příloze B.

Pseudoanonymizace je vnímána jako jedna z forem bezpečnostního opatření proti úniku údajů. V případě takového úniku totiž není možné subjekty údajů bez potřebného klíče identifikovat [26].

3.2 GDPR

General Data Protection Regulation (GDPR) neboli Obecné nařízení o ochraně osobních údajů je nová legislativa Evropské Unie, která vstupuje v účinnost od 25. května 2018.

Cílem této legislativy je modernizovat právní rámec ochrany osobních údajů v evropském prostoru tak, aby zohledňoval technologické pokroky posledních let a hájil práva občanů EU proti neoprávněnému zacházení s jejich daty [3]. Týká se všech firem, institucí i jednotlivců, kteří zpracovávají data uživatelů. Jsou nastavena přísnější pravidla, jejichž porušení může být trestáno vysokými pokutami[4].

3.2.1 Zásady zpracování osobních údajů

Obecné nařízení definuje tyto zásady zpracování osobních údajů: zákonnost, korektnost, transparentnost, omezení účelu, minimalizace údajů, přesnost, omezení uložení, integrita a důvěrnost. Za jejich dodržování zodpovídá správce údajů a je povinen toto na základě výzvy dokázat, jelikož byl zaveden princip zodpovědnosti správce [26].

Většina těchto zásad byla uvedena již ve stávajícím zákoně, s příchodem GDPR jsou však více rozpracovány. Součástí Obecného nařízení jsou i tzv. recitály, které obsahují důvody přijetí jednotlivých povinností a někdy i návody jak tato nařízení chápat. Například síťové identifikátory (IP a MAC adresa) byly považovány za osobní údaje již v zákoně o ochraně osobních údajů, jen to nebylo nikde explicitně vyřčeno [26].

Zásada zákonnosti je nejdůležitější ze zmíněných, jelikož říká, že zpracovávání a samotný sběr osobních údajů lze provádět pouze na základě právního důvodu [26].

3.2.2 Pověřenec pro ochranu osobních údajů

Nově je některým zpracovatelům osobních údajů nařízeno zajistit nezávislou kontrolní funkci tzv. pověřence pro ochranu osobních údajů (Data Pro-

[4] Až 20 000 000 EUR, resp. 4 % celkového ročního obratu[26]

tection Officer). Tato osoba je zodpovědná za monitorování souladu zpracování osobních údajů s povinnostmi vyplývajícími z Obecného nařízení, dále také za školení pracovníků a provádění auditů. Povinnost jmenovat tohoto pověřence nastává jen ve třech případech, jimiž jsou:

1. zpracování provádí orgán veřejné moci či veřejný subjekt (s výjimkou soudů),
2. hlavní činnosti správce nebo zpracovatele spočívají v operacích zpracování, které vyžadují rozsáhlé pravidelné a systematické monitorování občanů,
3. hlavní činnosti správce nebo zpracovatele spočívají v rozsáhlém zpracování zvláštních kategorií údajů nebo osobních údajů týkajících se rozsudků v trestních věcech a trestných činů.

Pro tuto práci je relevantní druhý bod. Pravidelné a systematické monitorování totiž jasně zahrnuje všechny formy sledování a profilování na internetu, například i pro účely behaviorální reklamy [3].

3.2.3 Právo na výmaz

Právo na výmaz dává správci osobních údajů povinnost vymazat osobní údaje subjektu bez zbytečného odkladu, pokud nastane jeden z důvodů [3]:

- Osobní údaje již nejsou potřebné pro účel, pro který byly shromažďovány nebo zpracovávány.
- Občan odvolá souhlas, pokud je zpracování založeno na souhlasu, a neexistuje žádný další právní důvod pro zpracování.
- Občan vznese námitku proti zpracování z důvodu oprávněných zájmů správce osobních údajů, jako je např. vedení záznamů o zaměstnancích.
- Osobní údaje byly zpracovány protiprávně.
- Pokud není dán rodičovský souhlas se zpracováním osobních údajů dětí.
- Právní povinnost stanovená právem Unie nebo členským státem.

3.3 WiFi sniffing

Dalším důležitým poznatkem je to, že WiFi sniffing je považován za odposlech (podobně jako odposlech telefonu) a je proto trestným činem, přestože jsou sbírána jen provozní data[5].

3.4 Právní důsledky

Závěrem této kapitoly tedy je, že MAC adresa je osobním údajem a zůstává osobním údajem i po její úpravě hashovací funkcí. Platí na ni tedy zákon o ochraně osobních údajů a dnes už téměř platné nařízení GDPR.

Při zpracování osobních údajů je nutné dodržovat zásady zmíněné v této kapitole. Mezi nejdůležitější zásady patří potřeba právního důvodu pro sběr dat, potřeba mít explicitní souhlas subjektu s podmínkami sběru těchto dat (účel, doba uchovávání atd.) a poskytovat subjektu možnost tyto údaje kdykoli prohlédnout, anonymizovat nebo vymazat.

Při reálném použití této aplikace musí být na tyto aspekty brán zřetel. Důležité je také neopominout fakt, že ke sběru dat je využíván odposlech síťové komunikace, což je protiprávní.

[5] http://www.itbiz.cz/sniffing-odposlech-datove-komunikace

4 Návrh aplikace

Aplikace byla navržena s ohledem na požadavky, které byly shromážděny na základě konzultací se zadavatelem práce.

4.1 Funkční požadavky

- Navržená aplikace bude detekovat zařízení a sbírat o nich základní veřejně dostupná metadata (přibližná lokace a jiné identifikátory). Tyto údaje budou ukládány do databáze.
- Aplikace se bude zaměřovat na pasivní detekci zařízení, tzn. detekovaná zařízení nejsou na detekci předem připravována.
- Aplikace umožní údaje v databázi prohledávat, filtrovat a zobrazovat relevantní údaje o detekovaných zařízeních.
- Aplikace bude poskytovat webové uživatelské rozhraní.
- Aplikace bude logovat svoje aktivity (ukládání, mazání záznamů z DB, aktivita plánovaných úloh apod.).
- Aplikace bude umožňovat konfiguraci parametrů pomocí souboru.

4.2 Podobné technologie

Existuje několik technologií, které se zabývají podobnou problematikou. Jedná se například o Cisco Meraki[1], které se zabývá monitorováním sítě, ale poskytuje i nástroje pro sběr informací z management rámců WiFi komunikace.

Další podobnou technologií je Meshlium[2], které nabízí kromě softwarového řešení i vlastní hardware připravený pro sběr dat.

Tyto nástroje jsou sice velmi propracované a mají mnoho využití, od toho se však odráží i jejich vysoká pořizovací cena. Vlastní řešení poskytuje kromě nízkých nákladů také možnosti aplikaci libovolně rozšířit.

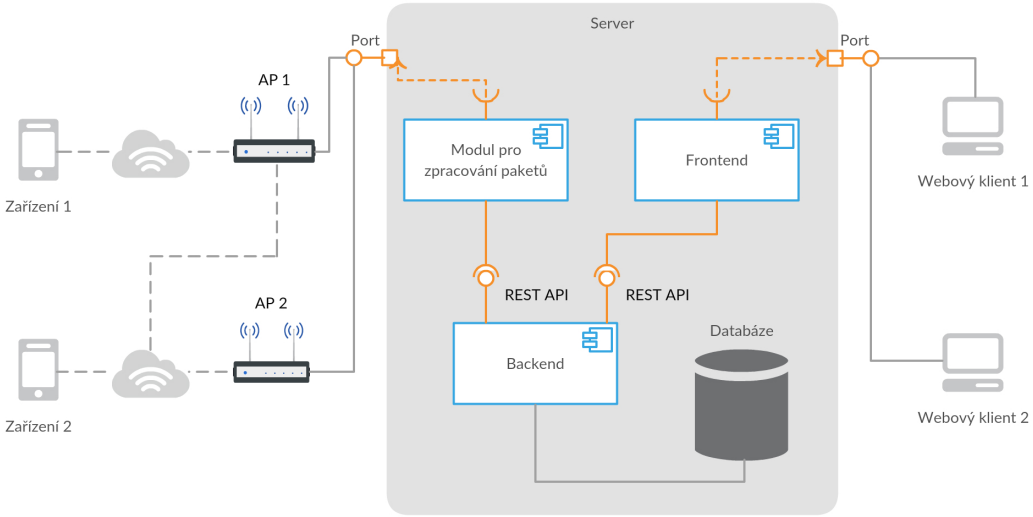
[1] https://meraki.cisco.com/solutions/location-analytics
[2] http://www.libelium.com/products/meshlium/smartphone-detection/

4.3 Architektura

Návrh architektury aplikace proběhl s důrazem na modularitu řešení tak, aby jednotlivé moduly byly technologicky nezávislé a bylo možné je v případě potřeby nahradit jinou implementací, například v jiném programovacím jazyce.

4.3.1 Architektura aplikace

Základní model aplikace je popsán obrázkem 4.1, který zobrazuje reálné použití aplikace.



Obrázek 4.1: Architektura aplikace

Mobilní zařízení se zapnutou WiFi vysílají do okolí Probe request pakety, které jsou zachyceny přístupovými body pracujícími v monitorovacím režimu. Přístupové body zachycují pouze režijní rámce, nikoli obsah přenášených dat. Jeden požadavek může být zachycen více přístupovými body ve stejný okamžik, což je vidět na obrázku 4.1 u Zařízení 2.

Zachycené rámce jsou streamovány na server s vyhrazeným portem, na kterém naslouchá modul pro sběr dat. Ten má na starost zpracování příchozího toku dat a po jejich zpracování je odesílá na hlavní aplikační modul pomocí RESTového (Representational State Transfer) rozhraní.

Aplikační modul zajistí persistentní uložení přijatých dat do databáze, ze které jsou data posléze dále vyhodnocována v podobě reportů. Tyto reporty jsou pak poskytovány opět přes RESTové rozhraní prezentační vrstvě aplikace, která se stará o jejich vykreslení. Přístup k aplikaci je umožněn přes webový prohlížeč.

4.3.2 Model-View-Controller

Současným trendem webových aplikací je třívrstvá architektura Model-View-Controller (MVC), nebo některé její variace. Model definuje datovou strukturu a zajišťuje základní práci s daty na nejnižší úrovni. View, neboli pohled, prezentuje uživateli data uchovávaná v modelu a umožňuje mu vykonávat akce na základě jeho pokynů. Tyto akce jsou delegovány do komponenty zvané Controller, neboli řadič. Ten zajišťuje reakce na uživatelské akce, ať už se jedná o úpravu modelu, či zobrazení jiného pohledu.

Výhodou této architektury je rozdělení zodpovědnosti jednotlivých částí aplikace, z čehož plyne, že je možné zaměnit implementaci jedné vrstvy nebo mít více implementací (např. jedna pohledová vrstva pro webovou aplikaci a jiná pro mobilní aplikaci). Z těchto důvodů byla vybrána pro tuto aplikaci právě MVC architektura. Model zastupuje backend aplikace společně s modulem pro zpracování požadavků. Controller a View je zastoupen na frontendu aplikace.

4.3.3 Přístupové body

Aby mohly být přístupové body použity s touto aplikací, musí splňovat následující požadavky.

WiFi sniffing

Prvním důležitým kritériem je podpora sniffování WiFi komunikace, včetně management paketů. Ideální je, když zařízení umožňuje vyfiltrovat pouze relevantní pakety, tzn. ty pakety, které obsahují Probe request. Tento typ rámců je tak jako tak filtrován v modulu pro zpracování paketů, ale je lepší, pokud jsou vyfiltrovány předem, aby došlo ke snížení provozu sítě a nemusely být posílány informace, které nejsou zapotřebí.

Streamování záznamu

Kromě samotného zachycování komunikace je zapotřebí přenést uchované informace na vzdálený server. Je vhodné, aby přístupový bod uměl streamovat zachycenou komunikaci na zvolený server na základě IP adresy a portu.

Pokud by tento požadavek nebyl splněn, existují i jiné alternativy pro přenos zachycených dat, která mohou být například ukládána do souboru přímo na AP. V tomto případě by bylo zapotřebí dopsat skript nebo aplikaci, která by periodicky odesílala obsah souboru na modul pro zpracování paketů, nebo napsat novou implementaci modulu pro zpracování paketů a odesílat požadavky rovnou na backend.

Výhodou RESTového rozhraní na backendu je zejména technologická nezávislost, která umožňuje napsat parsovací modul v téměř libovolném programovacím jazyce. Toto je jeden z důvodů, proč je modul pro zpracování paketů oddělen od backendu. Sběrné zařízení může být například Raspberry Pi s přidaným USB WiFi adaptérem a pomocí skriptu mohou být požadavky rovnou parsovány a posílány na backend v požadovaném tvaru.

Podpora více režimů operace

Pokud je zapotřebí, aby použitý přístupový bod zároveň pracoval v normálním režimu (např. režim AP nebo režim routeru) a poskytoval tedy možnost připojení k síti, musí disponovat dvěma síťovými kartami. Síťová karta může totiž pracovat jen v jednom režimu operace a pro sběr dat je potřebný monitorovací režim, viz kapitola 2.2.6. Další možností je virtualizace, která umožňuje přepínat mezi oběma režimy za cenu horšího výkonu zařízení.

4.3.4 Modul pro parsování paketů

Tento modul slouží pro sběr dat z přístupových bodů a splňuje pět základních rolí: příjem, filtrování, parsování, anonymizaci a posílání paketů na backend. Naslouchá podle parametrů specifikovaných v konfiguračním souboru (protokol a port) a přijímá příchozí pakety, ze kterých vyfiltruje jen Probe requesty. Každý příchozí paket naparsuje a zpracuje MAC adresu zařízení pomocí hashovací funkce tak, aby docházelo k anonymizaci dat co nejdříve. Ze zpracovaného paketu sesbírá pouze vybrané informace (viz. 4.3.5), a ty pak odesílá na backend k dalšímu zpracování.

Důvodem vyčlenění tohoto modulu z backendu je kromě zmíněné možnosti více různých implementací také snadné navýšení výkonu sběru dat. Pro sběr je možné spustit více instancí tohoto modulu, a tím rozložit výpočetní zátěž na více serverů. Například každý přístupový bod může komunikovat s jednou instancí tohoto modulu.

Průběh zpracování Probe request rámců

Zpracované a vyfiltrované pakety jsou ukládány do fronty podle času zachycení záznamu na AP. Tato fronta je po naplnění v jedné dávce odeslána na backend, kde jsou údaje dále zpracovány. Obsah fronty také může být odeslán po určitém časovém intervalu, záleží která z událostí nastane dříve. Tento parametr spolu s velikostí fronty je možné nastavit pomocí konfiguračního souboru modulu.

Fronta je použita z toho důvodu, že pozorovaná zařízení mohou vysílat velký počet zpráv a posílání každé zprávy zvlášť by RESTové rozhraní serveru zbytečně zahlcovalo. Podle studie [17], která se zabývá frekvencí posílaných Probe requestů, je průměrný počet posílaných zpráv 55 za hodinu. Dále ale ukazuje, že zařízení může vyslat za stejnou dobu například i kolem 4000 požadavků, zejména v případě, že má zapnutou obrazovku, nebo se v blízkosti vyskytuje síť, kterou zařízení zná. To je více než jeden požadavek za vteřinu, ale je nutné uvážit, že se jedná pouze o jedno zařízení a jeden přístupový bod. Ten samý záznam může být zachycen z více přístupových bodů zároveň, a tím se nápor na server navyšuje, nehledě na to, že se ve sledovaném prostoru pravděpodobně bude pohybovat větší počet zařízení.

4.3.5 Vybraná metadata

Z každého Probe requestu jsou vybrány a uchovávány následující informace:

OID (CID)	identifikátor výrobce zařízení z prvních tří oktetů MAC adresy před hashováním
ID zařízení	identifikátor zařízení tvořený hashem z MAC adresy
lokální bit	příznak, zda se jedná o lokálně alokovanou MAC adresu
časová značka	čas, kdy byl záznam zachycen pomocí AP
SSID	identifikátor vyhledávané sítě
síla signálu	uváděná v dB
frekvence	podporovaná zařízením v MHz
kanál	na kterém byla zpráva zachycena
velikost	Probe requestu v bytech
ID AP	identifikátor AP (MAC adresa), kterým byl záznam pořízen

Tyto údaje byly vybrány, aby bylo možné získat požadovaná data pro reporting viz. 4.3.7. Další údaje, které Probe request uchovává, jsou povětšinou technické parametry, které nepředstavují zajímavé údaje pro reportování.

Jedním z údajů, který by byl přínosný, ale není zpracováván, je parametr Wi-Fi Protected Setup (WPS) ze sekce nepovinných parametrů. Tento protokol usnadňuje spárování zařízení při připojování k síti. Součástí tohoto

parametru je identifikátor a často také model zařízení. Nebyl však zpracován, protože v Probe requestu není velmi často přítomen. Studie [27] zkoumá tři soubory dat, v nichž se tento parametr vyskytuje pouze v 8 %, 6 % a 4 % případů.

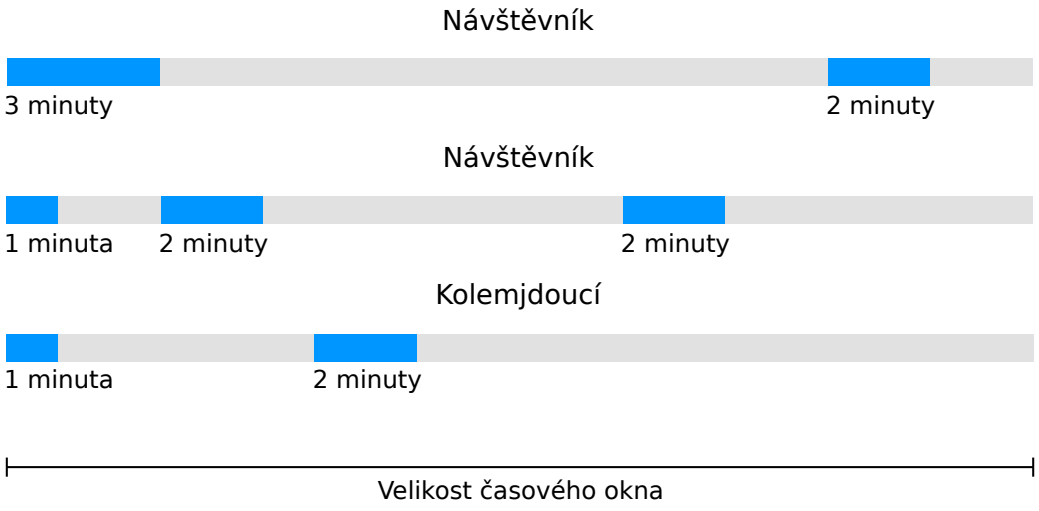
Všechny výše uvedené atributy kromě posledního jsou přístupné přímo z Probe requestu, pouze identifikátor AP musí být tímto modulem přidán. Na základě těchto informací je vytvořen objekt nazývaný CapturedData, který je odeslán na backend.

4.3.6 Backend

Backend představuje výpočetní jádro aplikace. Shromažďuje a ukládá získané informace do databáze, dále je zpracovává v podobě reportů a poskytuje je prezentační vrstvě.

Vyhodnocení přítomnosti zařízení

Přítomnost zařízení je vyhodnocována do tzv. návštěv, které reprezentují určitou dobu přítomnosti zařízení. Návštěvy jsou rozděleny do tří skupin, viz obrázek 4.2. První z nich rozlišuje dlouhodobější návštěvníky, kteří se vyskytují v pozorované oblasti alespoň po určitou dobu stanovenou prahem, druhá popisuje kolemjdoucí, což je zbytek uzavřených návštěv, které nebyly zařazeny do předchozí kategorie. Třetí stav definuje otevřenou návštěvu, která ještě nebyla uzavřena a rozdělena do jedné z předchozích kategorií.



Obrázek 4.2: Vyhodnocení délky návštěv

Pro rozlišní návštěvníků od kolemjdoucích je použit přístup jako v [2] se stejnou výchozí hodnotou prahů, tu je však možné pomocí konfigurace

měnit. Během zpracování návštěv je uchováváno časové okno (výchozí hodnota je 20 minut) a pokud doba přítomnosti zařízení přesáhne práh (v tomto případě 5 minut), je zařazen jako návštěvník, v opačném případě je označen za kolemjdoucího. V případě, že se po celou dobu okna neobjeví žádný záznam, je návštěva uzavřena.

Tento způsob udržování návštěv poskytuje robustnější řešení, než pouhé vyhodnocování rozdílu času aktuálního záznamu s časem posledního záznamu dané návštěvy.

Zpracování získaných metadat

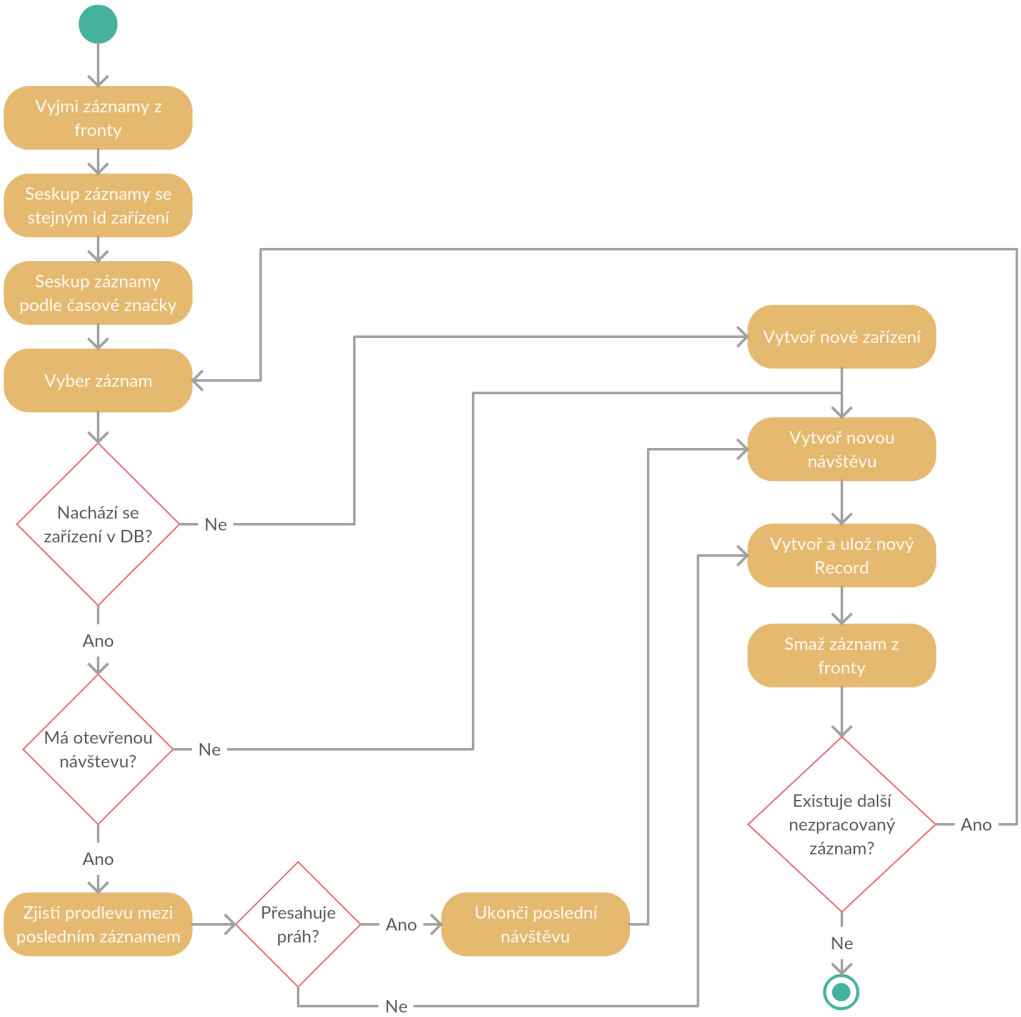
Po přijetí dávky od modulu pro zpracování paketů jsou všechny záznamy bezprostředně vloženy do persistentní fronty, jejímž účelem je seskupit paralelní záznamy z více přístupových bodů, které se týkají téhož zařízení. Podmínkou detekce takovýchto záznamů a správného fungování této funkcionality je samozřejmě časová synchronizace přístupových bodů. Do fronty jsou ukládány objekty typu *CapturedData* a jsou řazeny podle časové značky příchodu na AP.

Fronta je persistentní, aby nedošlo ke ztrátě všech jejích záznamů v případě pádu nebo pozastavení aplikace. V případě pádu backendu zůstane fronta uchována a sběrné moduly se po dobu výpadku několikrát pokusí opakovat pokus o poslání zpracované dávky. Pokud se tedy podaří backend znovu včas zprovoznit, nemusí dojít ke ztrátě dat.

Z takto vytvořené fronty jsou záznamy periodicky vybírány pomocí plánované úlohy, jejíž zjednodušený průběh je znázorněn na obrázku 4.3. Doba opakování úlohy a počet vybraných záznamů jsou volitelné pomocí konfiguračního souboru.

Záznamy vybrané z fronty jsou nejprve seskupeny podle identifikátoru zařízení a pro každý z nich jsou seřazeny a seskupeny podle časové značky. Při seskupení podle časové značky jsou objekty převedeny z typu CapturedData na objekt Record, který seskupuje paralelní záznamy z více AP. Každý objekt Record tedy kromě jiného uchovává časovou značku a pro každé AP, které tento záznam zachytilo, také sílu signálu.

Pak je iterováno nad všemi záznamy jednoho zařízení, což je zobrazeno úkonem vyber záznam a posledním větvením v diagramu. Nejprve je dohledáno právě zařízení související se zpracovávanými záznamy. Pokud se nenachází v databázi, je vytvořeno nové a spolu s ním i nová návštěva. Pokud se podaří zařízení najít, pak je vyhodnocena jeho poslední návštěva s nově příchozím požadavkem na základě postupu uvedeného v 4.3.6. Tím je vyhodnoceno, zda záznam již nepatří do této návštěvy a je potřeba ji uzavřít



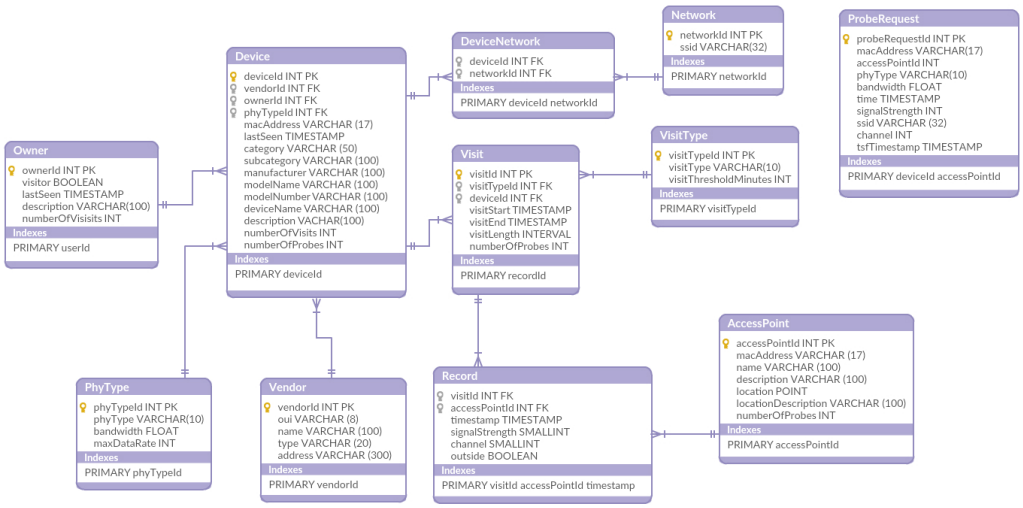
Obrázek 4.3: Návrh úlohy pro zpracování příchozích požadavků

a vytvořit novou, nebo je možné ho přidat do stávající návštěvy, čímž zůstane dále otevřena.

Návrh databáze

Návrh databáze proběhl na základě vybraných parametrů (viz 4.3.5), které jsou sbírány. Vzhledem k tomu, že obsah ukládaných dat má pevně danou strukturu, byla zvolena relační databáze, která dostačuje potřebám této aplikace. Databázový model je možné vidět na obrázku 4.4.

Tabulka Device reprezentuje konkrétní zařízení, o kterém jsou data sbírána. Každé zařízení má svého vlastníka, který je definován tabulkou Owner, slouží pro další analýzu a vyhodnocení dat. Entita PhyType určuje typ WiFi připojení podporovaný zařízením. Informace o výrobci zařízení jsou uchová-



Obrázek 4.4: Databázový model

vány v tabulce Vendor. Každé zařízení definuje množinu známých sítí, tento vztah je definován tabulkou Network a vazbou s kardinalitou M:N pomocí rozkladové tabulky NetworkDevice.

Ústředním nositelem dat je entita Record (záznam), která ukládá vybrané informace z příchozího Probe requestu. Každý záznam odkazuje na přístupový bod, ze kterého byl pořízen. Přístupový bod je uložen v tabulce AccessPoint. Tabulky Visit a VisitType slouží ke zpracování informací o záznamech, tak aby bylo možné vyhodnotit délku a typ návštěvy.

ORM mapování

Vytvoření databázové struktury je zajištěno pomocí objektově relačního mapování (ORM). Tento přístup umožňuje propojit databázové entity a objekty v programovacím jazyce, které tvoří reprezentaci databáze přímo v aplikaci. Data z databáze jsou při ukládání či načítání údajů mapována pomocí definovaných objektů a kromě toho umožňují ORM nástroje také vygenerování struktury tabulek a jejich vazeb.

RESTové rozhraní modulu pro zpracování paketů

Backend vystavuje RESTové rozhraní pro modul pro zpracování paketů. Výhodou tohoto přístupu je technologická nezávislost a snadná implementace, díky rozšířenosti této technologie totiž existuje mnoho knihoven a nástrojů. Další výhodou tohoto rozhraní je jednotný přístup a možnost záměny databáze v jednom modulu místo více změn v každém modulu, který s ní komunikuje.

Toto rozhraní definuje endpoint pro ukládání jednotlivých záznamů a také ukládání více záznamů naráz v podobě dávek.

Reporting

Součástí logiky aplikace je reporting, který zajišťuje zpracování a vyhodnocení záznamů tak, aby bylo možné je zobrazit formou reportů. Veškeré výpočty a statistiky probíhají v této části aplikace a jejich výsledky jsou posílány prezentační vrstvě, která se stará o jejich vykreslení. Více v kapitole 4.3.7.

RESTové rozhraní klienta

Za posledních pár let se RESTové rozhraní stalo de facto standardem mezi aplikační logikou a prezentační vrstvou aplikace. Ze stejných důvodů jako u rozhraní pro modul pro zpracování paketů je i zde využito právě RESTové rozhraní. Umožňuje předat data potřebná pro vykreslení stránky, a to včetně dat pro reporting.

4.3.7 Prezentační vrstva

Prezentační vrstva má na starost zobrazení sesbíraných a vyhodnocených údajů.

Pro návrh prezentační vrstvy byl vytvořen prototyp v podobě drátěného modelu (wireframe) za účelem rozmyšlení obsahu a rozložení prvků aplikace před vlastní implementací. Takto navržený prototyp byl konzultován s vedoucím práce a je součástí přiloženého CD.

Pohled na data

Data jsou zobrazována pomocí tabulek, grafů a jiných reportovacích prostředků. Tabulky umožňují přímý pohled na hodnoty v nasbíraných datech a poskytují možnost vyfiltrovat, vyhledat a seřadit údaje podle zvolených parametrů. Kromě tabulek poskytuje aplikace několik reportů v podobě grafu.

Reporting v podobě grafů

Během návrhu aplikace byly navrženy následující reporty:

- přehled počtu a dob trvání návštěv zařízení,
- počet zařízení aktuálně se pohybujících ve sledovaném prostoru,

- počet zařízení v konkrétních hodinách v průběhu dne,
- přehled výrobců zařízení a jejich četnosti,
- počet záznamů podle síly signálu z jednotlivých přístupových bodů,
- přehled uložených vyhledávaných sítí podle četnosti výskytů.

Pro většinu z těchto grafů lze vybrat období, za které se má graf zobrazit, případně přístupový bod, kterého se mají zobrazená data týkat.

5 Implementace aplikace

Při implementaci bylo dbáno na best practices zvolených technologií, rozdělení závislostí komponent a na programování proti rozhraní. Před vyčerpávající dokumentací pomocí komentářů, které nemusí vždy odpovídat naprogramované funkci, je kladen důraz na členitost kódu a vhodné pojmenování funkcí a proměnných. Ve všech modulech aplikace je používáno principů Dependency Injection a také několika návrhových vzorů, jako např. singleton, builder a fasáda.

V další části jsou popsány jednotlivé moduly aplikace spolu s nastavením přístupových bodů.

5.1 Přístupové body

Kapitola 4.3.3 již definovala požadavky na AP, které musí splňovat, aby mohl být použit pro tuto práci. Tato část popisuje nastavení přístupového bodu a konfiguraci, která byla použita pro sběr dat.

5.1.1 Simulace AP pomocí virtualizace

Jelikož nebyl k dispozici přístupový bod s požadovanými vlastnostmi, byl pro vývoj a debugování aplikace simulován pomocí virtualizace v nástroji VirtualBox[1]. Díky virtualizaci lze vyzkoušet libovolný operační systém routeru a s pomocí USB WiFi adaptéru lze simulovat nastavení jako při použití reálného AP. Je zapotřebí adaptér, který podporuje monitorovací režim. Pro tuto práci byl použit TP-LINK TL-WN722N.

Vyzkoušeny byly dva operační systémy pro routery, a to Mikrotik RouterOS 6.38.3 a OpenWrt[2]. První z nich se nepodařilo simulovat kvůli tomu, že nebyl způsob, jak virtualizovat WiFi kartu. RouterOS totiž nepodporuje externí USB adaptéry [10] a virtualizace síťové karty přes PCI (PCI passthrough) nebyla na použitém PC možná.

5.1.2 Použitá konfigurace

Pro vývoj byl tedy využit systém OpenWrt ve verzi 15.05, jehož výhodou je, že je založen na Linuxu a umožňuje tedy používat linuxové nástroje. Za další

[1] https://www.virtualbox.org/
[2] https://openwrt.org/

výhodu je možné považovat poměrně nízké hardwarové požadavky[3], díky kterým může být nahrán i na některý z domácích routerů.

Nastavení monitorovacího režimu

Monitorovací režim je možné dočasně nastavit pomocí nástroje *iwconfig* příkazem:

```
1   $ sudo iwconfig wlan0 mode monitor
```

Alternativou je permanentní nastavení pomocí souboru */etc/config/wireless*. Zde je ukázka použité konfigurace, která umožňuje režim ap a monitorovací režim zároveň:

```
config wifi-device 'wlan0' # název zařízení
    option type '80211'    # typ zařízení
    option channel '11'    # zvolený kanál
    option hwmode '11g'    # režim WiFi

config wifi-iface         # nastavení monitorovacího rozhraní
    option device 'wlan0' # asociované zařízení
    option ifname 'mon0'  # název rozhraní
    option mode 'monitor' # režim operace

config wifi-iface             # nastavení rozhraní ap
    option device 'wlan0'     # asociované zařízení
    option ifname 'wifi0'     # název rozhraní
    option network 'lan'      # asociovaná síť
    option mode 'ap'          # režim operace
    option ssid 'OpenWrtWiFi' # SSID sítě
    option encryption 'psk2'  # nastavení šifrování
    option key '123456789'    # heslo připojení
```

Zahájení sniffování

Pro sniffování WiFi komunikace existuje několik nástrojů, mezi nejpoužívanější patří Aircrack-ng, Kismet, Tcpdump a Wireshark. Může být použit libovolný z nich, v této práci byl pro zachycení komunikace využit Tcpdump, protože je snadný pro použití a dá se snadno nainstalovat na router se systémem OpenWrt pomocí balíčkovacího systému *opkg*.

Příkaz pro zahájení naslouchání:

[3]https://openwrt.org/supported_devices

```
1   $ tcpdump -s0 -U -i mon0 'type mgt subtype probe-req'
```

Obsah zpráv zachycený pomocí tcpdump může být přeposlán na server s modulem pro zpracování záznamů například pomocí nástroje netcat:

```
1   $ tcpdump -s0 -U -i mon0 'type mgt subtype probe-req'
2     -w - | nc 192.168.0.100 12345
```

V případě potřeby šifrovaného spojení je možné využít nástroj ssh.

5.2 Modul pro zpracování paketů

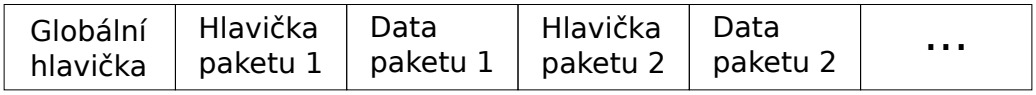
5.2.1 Příjem paketů

Příjem paketů je definován několika parametry, protokolem UDP či TCP a číslem portu. Modul po spuštění zajistí poslech na základě zvolených parametrů. Příjem paketů běží až do té doby, než je aplikace explicitně ukončena. Dalším parametrem je *sleeptime*, který určuje, na jak dlouhou dobu se má naslouchající vlákno uspat v případě, že zrovna nepřicházejí žádná data. Dále je možné nastavit velikost příchozího bufferu pomocí parametru *bufferSize*.

5.2.2 Parsování paketů

Přístupové body, které používají sniffing k zachytávání dat, ukládají všechny informace do *pcap*[4] formátu. Pro porozumění obsahu posílaných zpráv je nejprve zapotřebí je naparsovat. Tento formát se používá pro uchování údajů ze síťové komunikace. K zachyceným paketům jsou přidávány některé režijní informace, jako je např. verze pcap protokolu, typ sítě, maximální délka a časová značka přijetí paketu atd.

Formát *.pcap* souboru je rozdělen do několika částí. Nejprve je uvedena globální hlavička a po ní následuje libovolný počet párů hlavička paketu a data paketu (viz obr. 5.1). Každý blok má definované vlastnosti a velikost. Pro získání samotných dat uvnitř paketu musí být odstraněna globální hlavička a pak hlavička každého paketu.



Obrázek 5.1: Struktura *pcap* formátu

[4]https://wiki.wireshark.org/Development/LibpcapFileFormat

Mezi hlavní knihovny pracující s tímto formátem patří libpcap, která vznikla spolu s tímto formátem pro linux. Pro Windows pak existuje implementace zvaná WinPcap. Tyto dvě knihovny poskytují rozhraní pro čtení *pcap* formátu, které je využíváno řadou knihoven v různých programovacích jazycích. Mezi nejznámější patří např. Net::Pcap, Jpcap, python-libpcap, Ruby/Pcap atd.

Kromě *pcap* formátu existují i jiné způsoby obalení paketů. Například Mikrotik používá pro záznam paketů a následný přenos po síti protokol TaZman Sniffer Protocol (TZSP)[5]. Jedná se o protokol, který slouží k zapouzdření jiných protokolů pomocí UDP přenosu, a často se používá právě k přenesení zachyceného bezdrátového přenosu.

Tento protokol také definuje hlavičku, která obaluje každý paket a přidává informaci o verzi a typu přenášeného protokolu. Po hlavičce může následovat řada parametrů, které mohou nebo nemusí být definovány. Nejprve je definován typ parametru (1 B), pak délka (1 B) a samotná hodnota, jejíž délka závisí na typu parametru. Mezi tyto parametry patří například časová značka přijetí paketu, síla signálu při přijetí a číslo kanálu. Tato sekce parametrů je ukončena znakem TAG_END, který obsahuje samé jedničky. Nakonec následuje samotný obsah paketu.

Použité knihovny pro parsování

V Javě existují následující nástroje pro práci s pcap formátem – jpcap, jNetPcap, Jpcap, Pcap4J a pkts.io. Většina z nich je uzpůsobena přímo pro záznam, v tomto modulu však dochází pouze k parsování existujícího záznamu. Pro tento účel se nejvíce hodí knihovna, která podporuje streamy. Z těchto knihoven byly vybrány poslední dvě. Pcap4J byla vybrána z důvodu možnosti parsování Probe request rámců a kvůli aktivitě projektu. Bohužel ale nepodporuje streamy, a proto byla využita i knihovna pkts.io.

5.2.3 Filtrování paketů

Po zpracování paketů dochází k jejich filtrování. Další zpracování podstupují jen pakety obsahující Probe request rámce, ostatní pakety jsou zahozeny.

5.2.4 Konfigurace

Konfigurace tohoto modulu je umožněna pomocí souboru, v souladu s požadavky práce. Nastavení je možné pomocí souboru *application.properties*,

[5]https://wiki.mikrotik.com/wiki/Ethereal/Wireshark

ve kterém je možné nastavit následující parametry:

```
listener.hostname=0.0.0.0 # rozhraní na kterém aplikace na-
                          # slouchá, defaultně všechna rozhraní
listener.udp=false        # protokol true = UDP, false = TCP
listener.port=37009       # port
listener.tzspHeader=false # zda pakety obsahují TZSP hlavičku
listener.sleepTimeMillis=20000 # čas po kterou se vlákno uspí
                               # pokud nikdo nevysílá
listener.bufferSize=16384 # velikost bufferu v bytech
# MAC adresa přístupového bodu na který je modul napojený
apMac=905C445734AC
# URI RESTového rozhraní backendu, kam jsou posílány záznamy
rest.url=http://localhost:8080/packetanalyser/api/v1/data/re-
cords/batch
anonymizer.anonymize=true       # anonymizace MAC adres
anonymizer.hashType=SHA-256     # algoritmus hashování
timed-buffer.durationSeconds=20 # doba pro odeslání dávky
timed-buffer.capacity=100       # kapacita bufferu
# nastavení logování
logging.level.cz.zcu.dp=INFO    # úroveň logování
logging.file=app.log            # název souboru s logy
```

5.3 Backend

5.3.1 Použité technologie

Pro implementaci backendu je využita Java verze 1.8 kvůli její rozšířenosti právě pro backend aplikace. Existuje zde několik zaběhlých nástrojů, které mají podrobnou dokumentaci a mnoho problémů již bylo řešeno a popsáno. Verze 1.8 byla zvolena zejména kvůli podpoře Stream API.

Spring a Spring Boot

Hlavní z použitých knihoven je Spring. Důvodem použití je podpora injekce závislostí pomocí anotace *@Autowired*, usnadnění práce s databází díky JpaRepository, RESTovým rozhraním a serializací a v neposlední řadě snadná implementace plánovaných úloh.

Dále byl použit Spring Boot, který usnadňuje konfiguraci projektu a samotné spouštění, zjednodušuje správu knihoven, načítání parametrů ze souboru, načítání inicializačních dat do databáze a další.

JPA a Hibernate

Pro usnadnění tvorby databázových entit byla použita specifikace Java Persistence Api (JPA), která umožňuje ORM mapování, jako konkrétní implementace je použit Hibernate. Díky těmto nástrojům je snadné zaměnit použitou databázi, je to jen otázka nastavení v konfiguračním souboru.

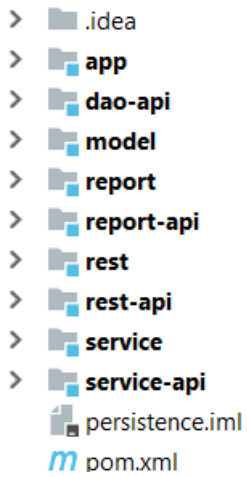
Lombok Project

Další použitou knihovnou je Lombok Project[6], umožnující značně zjednodušit kód pomocí anotací, které způsobí vygenerování opakujícího se kódu. Výsledné třídy obsahují méně kódu a jsou tím pádem daleko přehlednější.

5.3.2 Struktura

Aplikace je vytvořena jako maven pojekt s moduly. Tato konfigurace umožňuje rozdělení logiky do několika částí, ale zároveň usnadňuje správu verzí částí aplikace, závislých knihoven a také sestavení aplikace. V případě potřeby je možné vyčlenit některé z modulů a vytvořit z nich samostatnou aplikaci snadněji, než kdyby aplikace tvořila jeden velký celek.

Projekt se skládá z modulů, které je možné vidět na obrázku 5.2. Většina modulů poskytuje dvě části – jednu, která definuje rozhraní daného modulu a druhou se samotnou implementací. Důvodem pro oddělení rozhraní od implementace je zejména možnost více implementací jedné části a minimalizace závislostí knihoven.



Obrázek 5.2: Struktura backendu

[6]https://projectlombok.org/

Modul app

Tento modul je spouštěcím bodem aplikace, obsahuje třídu AppRunner s metodou main a třídu InitDataLoader pro načtění inicializačních dat do databáze.

Modul dao-api

Další modul definuje databázové rozhraní Data Access Object (DAO). Je implementováno pomocí rozhraní ze Spring Data, které zajistí vygenerování příslušného kódu na základě specifikovaného rozhraní. Takže není zapotřebí psát implementační třídu, proto neexistuje modul zvaný dao.

Modul model

Model obsahuje všechny databázové entity a utility s nimi svázané, např. enumy a konvertory, pomocí kterých je lze převést na řetězec do databáze.

Modul report

Tento modul obsahuje služby spojené s vytvářením reportů. Nachází se zde například plánované úlohy, které jsou volány z třídy ReportSheduler obsažené v tomto modulu.

Modul report-api

Report-api definuje rozhraní k předchozímu modelu reportů, je zde definováno rozhraní fasády ReportFacade, která umožňuje získat každý z vytvořených reportů.

Modul rest

RESTové rozhraní je uloženo v tomto modulu. Nachází se zde controllery, které jsou dále rozděleny na dvě skupiny. První slouží pro zisk surových dat z databáze a druhou kategorii tvoří controllery pro reporty.

Modul rest-api

Modul rest-api definuje rozhraní pro předchozí model. Jsou zde popsány hlavně objekty pro přenos tzv. Data Transfer Objekty (DTO) a další společné objekty související se serializací objektů, například map.

Modul service

Zde jsou definovány služby zejména pro práci s entitami, pro jejich korektní vytváření a přidávání vazeb mezi entitami.

Modul service-api

Service-api poskytuje rozhraní pro práci se službami. Každá ze služeb pracující s databázovými entitami dědí od společného rozhraní, které zajišťuje implementaci základních metod pro práci s danou entitou.

5.3.3 Plánované úlohy

Aplikace podporuje dvě plánované úlohy. První z nich je popsána na obr. 4.3 a zajišťuje zpracování příchozích záznamů z fronty. Druhá se stará o uzavírání návštěv, pro které se neobjevil další záznam po uplynutí časového okna. Kdyby nebyly takto uzavírány, pak by po zařízeních, která se v systému vyskytnou pouze jednou, zůstávaly neuzavřené návštěvy.

Obě úlohy jsou implementovány pomocí anotace *@Scheduled*, která zajistí opakované spouštění úlohy a vykonává vždy jen jednu instanci. Nemůže se tedy stát, že by se v případě dlouhého trvání úlohy spustila dvakrát paralelně. Tato anotace přijímá jako parametr dobu, po které se má úloha opakovat. Formát této doby je možné zapsat ve stylu zápisu nástroje cron. Struktura zápisu se skládá ze sedmi parametrů, přičemž poslední je volitelný:

```
<sekunda> <minuta> <hodina> <den v měsíci> <měsíc> <den
v týdnu> <rok>
```

Tento styl zápisu je velmi flexibilní, kromě číselných hodnot umožňuje zapsat i výčty, rozsahy a další. Například následující zápis nastaví úlohu, která se vykonává každou minutu od 13:00 do 13:05 a od 20:00 do 20:05 každý den:

```
0 0-5 13,20 * * ?
```

5.3.4 Konfigurace backendu

Nastavení backendu je možné pomocí souboru *application.properties*, ve kterém lze nastavit následující parametry:

```
server.contextPath=/packetanalyser     # kontext aplikace
server.port=8080                       # port
rest.version=1                         # verze REST api
rest.baseUrl=api/v${rest.version}      # základní URI
```

```
rest.data=${rest.baseUrl}/data          # endpoint pro data
rest.reports=${rest.baseUrl}/reports    # endpoint pro reporty
scheduler.data-aggregation-count=5000   # počet záznamů vybra-
                                        # ných z fronty
scheduler.data-aggregation-cron=0 0/20 * * * * # plán úlohy
          # pro zpracování záznamů (každých 20 minut)
scheduler.close-old-visits-cron=0 0/20 * * * * # plán úlohy
          # pro uzavírání starých návštěv (každých 20 minut)
#   VISIT SETTINGS
visit.presenceThresholdMin=2 # práh pro detekci přítomnosti
visit.visitorThresholdMin=5  # práh pro návštěvníky
visit.visitWindowMin=20      # velikost časového okna
#   DATABASE SETTINGS
spring.datasource.driverClassName=org.postgresql.Driver
spring.datasource.url=jdbc:postgresql://localhost:5432/packet-
analyser
spring.datasource.username=postgres
spring.datasource.password=postgres
spring.jpa.hibernate.ddl-auto=validate # způsob inicializace
                                       # DB tabulek
```

5.4 Vybraná databáze

V návrhu aplikace byl již odůvodněn výběr relační databáze a podle toho proběhl i návrh struktury tabulek. Mezi neplacené relační databáze patří např. MySQL, její odnož MariaDB, a PostgreSQL. Použita byla databáze PostgreSQL 9.6, jelikož podporuje možnost využít rozšíření pro prostorové a geolokační údaje PostGIS, které by bylo možné využít v případě rozšíření práce o prostorové údaje pozorovaných zařízení a přístupových bodů.

V implementaci však nebylo použito žádné vlastnosti týkající se přímo PostgreSQL a je tedy možné ji snadno zaměnit kvůli abstrakci, kterou poskytuje knihovna JPA.

5.5 Prezentační vrstva

5.5.1 Použité technologie

V současné době se v oblasti frontendu webových aplikací skloňují zejména technologie Angular, ReactJS a Vue.js[7]. Vzhledem ke stoupající popularitě těchto nástrojů a jejich podobnému zaměření se objevuje i řada jejich porovnání.[8][9][10][11]

Angular 5

Ze zmíněných technologií byl vybrán JavaScriptový framework Angular (5.2.1). Jak je uvedeno v článcích, poskytuje většinu potřebných funkcionalit bez další potřebné konfigurace. Jedná se například o routování, služby pro HTTP komunikaci, práci s formuláři a jejich validací a v neposlední řadě poskytuje nástroj angular-cli pro snadnou manipulaci s projektem. Také má přehledně definovanou strukturu komponent ve čtyřech oddělených souborech (šablona, soubor se styly, skripty s logikou a testy). Dalším důležitým faktorem je programovací jazyk TypeScript, který je nadstavbou JavaScriptu a kromě podpory rozhraní a abstraktních tříd umožňuje i statickou typovou kontrolu.

Všechny tyto funkce včetně možnosti psaní v TypeScriptu jsou umožněny i například v ReactJS, je však potřeba řešit další konfigurace.

Vzhled a pozicování

Pro usnadnění práce se vzhledem stránky a pozicováním jejích prvků byl vybrán Bootstrap 4. Alternativou k této technologii je například Material Design, který však neumožňuje pozicování prvků pomocí sloupců jako Bootstrap.

Tabulky

Zobrazení, filtrování a vyhledávání dat bylo již v prototypu navrženo pomocí tabulek. Existuje řada knihoven poskytujících tyto funkce, pro realizaci byla zvolena ng2-smart-table, která kromě požadovaných vlastností zvládá i řazení, editaci a přidávání záznamů.

[7] https://medium.com/codingthesmartway-com-blog/the-2018-roadmap-to-fullstack-web-development-8884ff02557a

[8] http://www.cuelogic.com/blog/angular-vs-react-vs-vue-a-2018-comparison/

[9] https://vuejs.org/v2/guide/comparison.html

[10] https://kruschecompany.com/blog/post/angular-vs-react-vs-vue

[11] https://codeburst.io/angular-vs-react-vs-vue-f470f5b74bf6

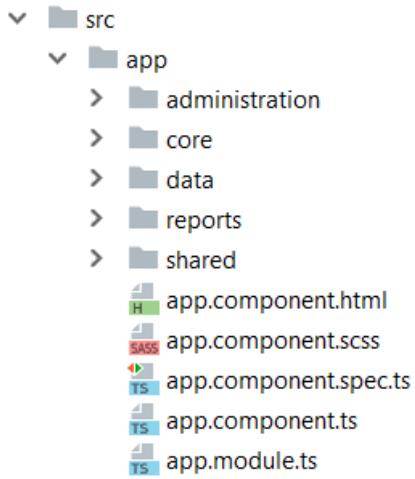
Grafy

V JavaScriptu existuje několik nástrojů pro vykreslování grafů, mezi nejznámější patří například Chart.js, C3, Echarts, TauCharts, Chartist, Plotly.js, NVD3 a další [8], [9].

Většina těchto nástrojů je aktivně aktualizována a poskytuje více či méně podobnou funkcionalitu. Vybrána byla knihovna Echarts, jelikož postačuje potřebám této práce, je velmi dobře dokumentována spolu se spoustou příkladů[12] a poskytuje online nástroj pro ladění grafů.

5.5.2 Struktura

Frontend aplikace je rozdělen do několika modulů (viz obr. 5.3), zejména kvůli přehlednosti, rozdělení zodpovědnosti a umožnění lazy loadingu modulů.



Obrázek 5.3: Základní struktura frontendu

Aplikace se skládá z jednoho centrálního modulu s názvem core, jednoho sdíleného shared, a pak z modulu pro každou ze základních funkcí. Komponenta app obsahuje router, který má na starost směrování a vykreslování požadovaných komponent.

Modul administration

Administrační modul je připraven pro komponenty ke konfiguraci aplikace, ale nakonec není využit kvůli dostatečné možnosti nastavení komponent pomocí souborů.

[12] https://ecomfe.github.io/echarts-doc/public/en/api.html#echarts

Modul core

Modul core uchovává hlavní prvky aplikace, např. komponenta s navigací stránky a objekty typu singleton jako jsou služby pro načítání a uchovávání dat a definice objektů s nimi spojených.

Modul data

Tento modul reprezentuje stránku s daty skládající se z menu po levé straně a tabulek s daty, kde si lze prohlížet a filtrovat záznamy z databáze.

Modul reports

Modul pro zobrazování reportů nad daty definuje všechny komponenty pro jejich vykreslování a objekty. Je zde uchován abstraktní report, od kterého dědí všechny ostatní reporty základní funkcionality. Také se zde nachází formulář pro konfiguraci jednotlivých reportů.

Modul shared

V této části aplikace se objevují drobnější komponenty a prvky, které lze opakovaně použít napříč aplikací. Nachází se zde například části formulářů.

5.5.3 Služby

Služby v aplikaci se dělí na dva základní typy–RESTové služby pro výměnu dat s backendem a služby pro ukládání těchto dat tzv. data store.

RESTové služby

RESTové služby stahují údaje z backendu na základě definovaných parametrů pomocí RESTových volání. Získaná data jsou posílána pomocí DTO objektů.

Ukázka DTO objektu seskupujícího veškeré informace nutné pro report návštěv za vybrané období:

```
1 export interface VisitReportDto {
2   frequencySum: VisitStats;
3   frequencyByDate: VisitByDate[];
4
5   binsType: string[];
6   binsLength: string[];
7   binsPresentTime: string[];
8   binsNumberOfProbes: string[];
9 }
```

```
10
11 interface VisitByDate {
12   key: string;
13   value: VisitStats;
14 }
15
16 interface VisitStats {
17   frequencyByType: KeyValuePair[];
18   frequencyByLength: KeyValuePair[];
19   frequencyByPresentTime: KeyValuePair[];
20   frequencyByNumberOfProbes: KeyValuePair[];
21 }
22
23 interface KeyValuePair {
24   key: string;
25   value: number;
26 }
```

VisitReportDto umožňuje přenést údaje o návštěvách z backendu, předávány jsou čtyři kategorie četností. První parametr představuje souhrnné četnosti za vybrané období pro všechny dané kategorie. Parametr frequencyByDate uchovává podobné statistiky, ale za každý den ve vybraném období. Objekty začínající slovem bins určují názvy kategorií a jsou použity pro správné vykreslení grafu. Tento objekt je po přijetí na frontend rozdělen do příslušných objektů, ze kterých mohou být dále čteny hodnoty pro přiřazené grafy.

Data store

Data jsou pomocí RESTových služeb načtena z backendu do těchto data storů, sloužících v principu jako cache paměť prezentační vrstvy. Tyto služby tvoří centrální prvek pro přístup k datům napříč frontendem. Jsou to singletony, mají tedy po dobu běhu aplikace jen jednu instanci a při jejich používání ve více komponentách tudíž nedochází k opakovaným dotazům na stejná data. Se spuštěním aplikace dojde k vytvoření této služby a jsou vyslány požadavky pro všechna data, která data store obsahuje. Tato akce je vykonána asynchronně. Všechna data v úložišti jsou dostupná pomocí Observable objektů, na které je možné zaregistrovat callback (funkce která bude zavolána po dokončení určité činnosti) metodou *subscribe()* a ten je volán v případě změny objektu.

Kromě objektu Observable jsou v úložišti použity tzv. Subjecty, které umí to samé co Observable, ale navíc přidávají funkci *next()*. Tato funkce umožňuje propagovat změny všem objektům, které jsou nad tímto objektem registrováni metodou *subscribe()*. Konkrétně jsou použity jsou objekty

BehaviorSubject, které oproti Subjectům navíc poskytují možnost definovat výchozí hodnotu objektu.

Registrací na objekty této služby tedy dochází k centrálnímu, asynchronnímu propagování dat napříč celým frontendem a data nemusí být dotazována opakovaně.

6 Testování aplikace

Pro ověření funkčnosti aplikace bylo využito několika následujících prostředků.

6.1 Unit testy

Aplikace obsahuje unit testy jen pro ověření některých funkcí, testováno je několik pozitivních i negativních scénářů.

6.1.1 Modul pro zpracování paketů

V modulu pro zpracování textů jsou Unit testy použity pro ověření správného zpracování paketů PacketParserTest a pro fungování časovaného bufferu TimedBufferTest pro shromažďování požadavků před odesláním na backend.

6.1.2 Backend

Na backendu byl vytvořen unit test například pro kontrolu správného načítání informací o výrobcích zařízení, které bylo implementováno RESTovou službou. Unit test nakonec pozbyl smyslu když byla tato metoda nahrazena načítáním výrobců z lokálního souboru, jelikož docházelo k zablokování přístupu kvůli velkému počtu požadavků.

6.2 Testovaní na virtualizovaném AP

Aplikace byla v průběhu vývoje testována pomocí virtualizovaného přístupového bodu viz kap. 5.1 a bylo ověřeno, že aplikace dokáže sbírat a zpracovávat odposlechnuté Probe request rámce.

6.3 Testovaní se dvěma AP

Pro odladění paralelního sběru dat s více AP byla testována konfigurace ještě s druhým sběrným bodem, kterým byl notebook s linuxem a se síťovou kartou podporující monitorovací režim. Při tomto nastavení byla pomocí debugovacího nástroje ve vývojovém prostředí sledována správnost agregace paralelních záznamů.

6.4 Testovaní pomocí pcap souboru

Výhodou použité architektury kromě živého sběru je možnost nahrát data do aplikace pomocí libovolného pcap souboru. Podobně jako při posílání dat z AP lze použít nástroj netcat do kterého je přesměrován obsah pcap souboru nástrojem cat. Ukázka příkazu:

```
1    $ cat capture.pcap | nc 192.168.0.100 12345
```

6.4.1 Použití existujícího datasetu

Vzhledem k legislativě týkající se sběru a zpracování osobních dat a důsledků, které z ní plynou (viz kap. 3.4) není jednoduché splnit podmínky při kterých by mohl být sběr prováděn. Aby mohla být prokázána funkčnost aplikace a nedošlo k neoprávněnému sběru osobních údajů byla aplikace testována na existujícím volně přístupném datasetu [13]. Jedná se o dataset, který byl pořízen v rámci zkoumání právě Probe requestů. Byl pořízen v roce 2013 a pochází z prostor univerzity. Údaje v tomto datasetu byly anonymizovány, zejména MAC adresy (výrobce zůstal zachován a každé zařízení má pořád jednu adresu) a SSID identifikátory.

7 Vyhodnocení výsledků

7.1 Výsledky nad použitým datasetem

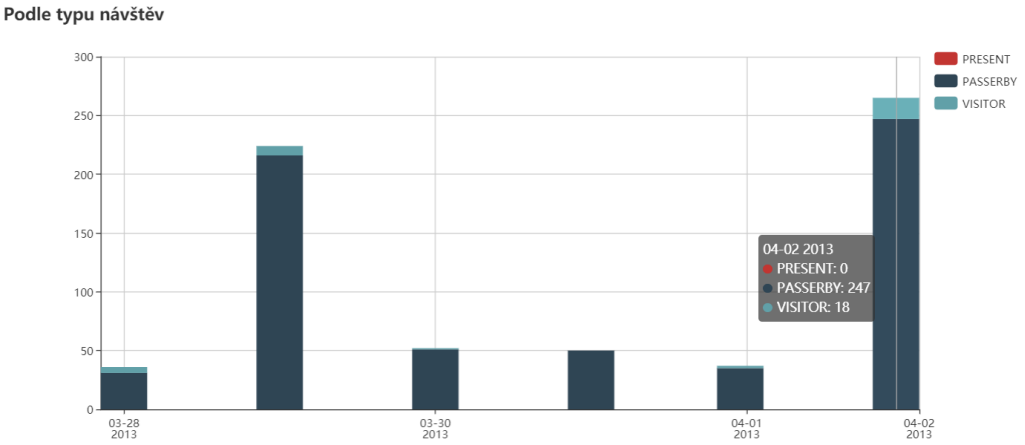
Tato podkapitola ukazuje výsledky reportů nad zpracovaným datasetem a znázorňuje a zkoumá grafy pocházející přímo z aplikace. Pozorováno je období šesti dnů od 28.3.2013 do 2.4.2013 včetně. První záznamy byl zachycen 28.03.2013 v 18:53:47.

7.1.1 Vyhodnocení návštěv

Prvním pohledem na data je vyhodnocení přítomnosti zařízení podle návštěv. Návštěvy jsou zkoumány podle jednotlivých dní a jsou rozděleny do čtyř následujících kategorií.

Podle typu návštěv

První kategorie rozděluje návštěvy podle typu návštěvníka na tři druhy – přítomný, kolemjdoucí a dlouhodobý návštěvník. Při prvním pohledu na



Obrázek 7.1: Report typu návštěv

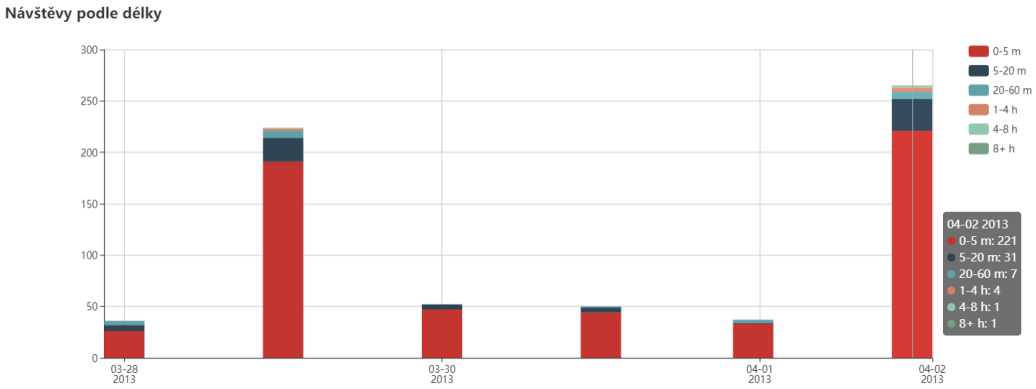
obr. 7.1 je možné si všimnout značných výkyvů v počtu návštěv za jednotlivé dny. Za zmínku stojí, že první den byl zaznamenán téměř až od sedmé hodiny večer, tím lze tedy vysvětlit menší počet návštěv. Při pohledu do kalendáře je možné zjistit, že 28.3.2013 je čtvrtek, další rázný pokles počtu návštěv (3. a 4. sloupec) je tedy způsoben tím, že se jedná o víkend. Zajímavé je, že během aprílového pondělí (1.4.) je počet návštěv dokonce menší než o víkendu,

mohlo se jednat o nějaký svátek nebo jiné volno (např. rektorské), to se ale nepodařilo dohledat. Největší počet návštěv je tedy zaznamenán v úterý.

Kromě počtu návštěv je možné si vypozorovat, že většina návštěv spadá do kategorie kolemjdoucích. Nízký počet dlouhodobých návštěv může být způsoben špatnou volbou hodnoty velikosti časového okna při zpracování návštěv. Počet aktuálně otevřených návštěv, tedy právě přítomných zařízení, při pohledu na datum nepřekvapí.

Podle délky

Dalším pohledem na návštěvy je jejich celková délka. Je vyhrazeno celkem šest kategorií do kterých může návštěva zapadat. Na obrázku 7.2 je vidět,



Obrázek 7.2: Report délky návštěv

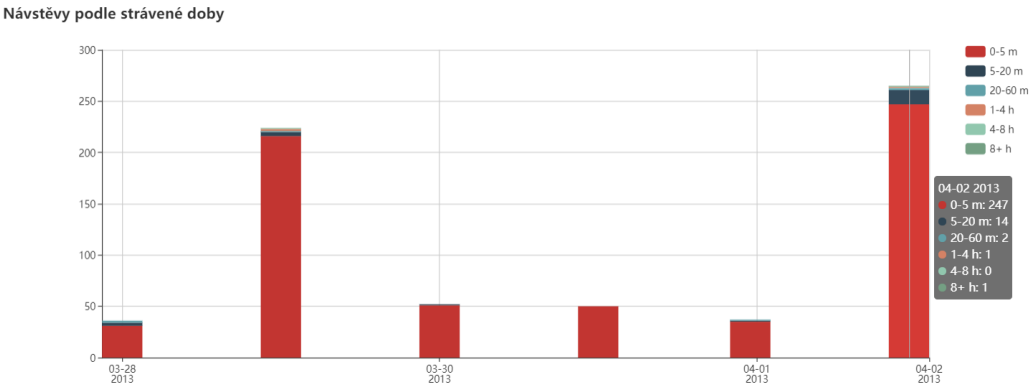
že velikost sloupců je pořád stejná jelikož se počet návštěv nemění. Nejvíce návštěv spadá do kategorie jednotek minut, což odpovídá i předchozímu grafu. Po dobu všech dní bylo zachyceny jen dvě návštěvy o délce 4-8 hodin a jen jedna trvající více jak osm hodin.

Při porovnání hodnot s předchozím grafem je možné si všimnout, že druhý sloupec ukazuje, že některé z návštěv trvající 20-60 minut spadají do kategorie kolemjdoucí. Naopak zas šestý sloupec zas řadí minimálně pět návštěv trvajících 5-20 minut do kategorie dlouhodobějšího návštěvníka. To je způsobeno tím, že délka návštěvy nereflektuje čas aktuálně strávený zařízením v pozorovaném prostoru, spíše dobu po kterou byla návštěva udržována jako otevřená.

Podle doby přítomnosti

Další kategorie zkoumá dobu reálně strávenou zařízením během návštěvy ve sledovaném prostoru. Jedná se jen o odhad, který ale poskytuje o něco přes-

nější pohled na strávený čas než předchozí kategorie. Přesnost této metriky je závislá na parametru prahu přítomnosti zařízení, který byl zvolen jako dvě minuty (jak již bylo zmíněno, průměrné zařízení vysílá Probe request zhruba za minutu a čtvrt). Přítomnost zařízení je navýšena o každý rozdíl dvou po sobě jdoucích hodnot spadající do tohoto prahu.



Obrázek 7.3: Report přítomnosti zařízení

Na obrázku 7.3 je vidět, že tato metrika opravdu lépe reflektuje strávený čas. Všechny návštěvy v posledním sloupci trvající déle než pět minut odpovídají osmnácti dlouhodobějším návštěvníkům podle prvního grafu.

Při seřazení návštěv podle délky se dá pomocí aplikace zjistit, že dvě nejdelší návštěvy má na starost stejné zařízení, viz obr. 7.4.

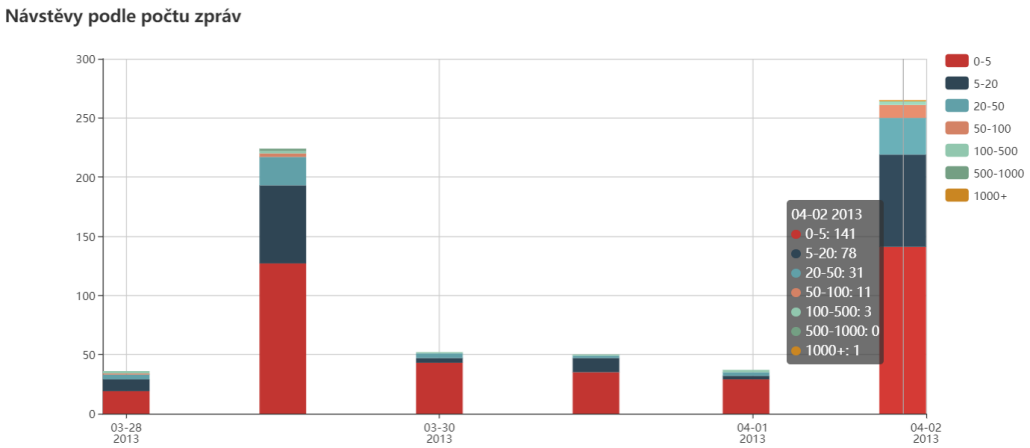
ID	ID zařízení	Typ	Začátek	Konec	Délka	Čas přítomnosti	Počet záznamů
ID	ID zařízení	Typ	Začátek	Konec	Délka	Čas příto	Počet zá
46133	27062958866F65C3357F2BD053D409718D9A6CE2A296037A3AE643C18627FFA0	VISITOR	02.04.2013 09:56:01.311	02.04.2013 18:19:51.157	08:23:49	08:23:49	1037
34301	27062958866F65C3357F2BD053D409718D9A6CE2A296037A3AE643C18627FFA0	VISITOR	29.03.2013 11:22:41.125	29.03.2013 18:13:35.771	06:50:54	06:50:54	862
40245	5A7E5FEE9A872EAEA56A3C1462547D213C8041E7D9D3EA875661F94371A65A98	VISITOR	02.04.2013 11:39:33.884	02.04.2013 16:05:49.034	04:26:15	02:01:10	158

Obrázek 7.4: Návštěvy delší než 4 hodiny

Také je možné si všimnout, že obě návštěvy mají stejnou délku a čas přítomnosti, to znamená, že zprávy od zařízení chodit minimálně každé dvě minuty po celou dobu návštěvy. Obě návštěvy dokazují, že zařízení v průměru vysílalo zhruba každou půl minutu.

Podle počtu zpráv

Poslední report návštěv je rozděluje podle počtu přijatých zpráv do několika kategorií.

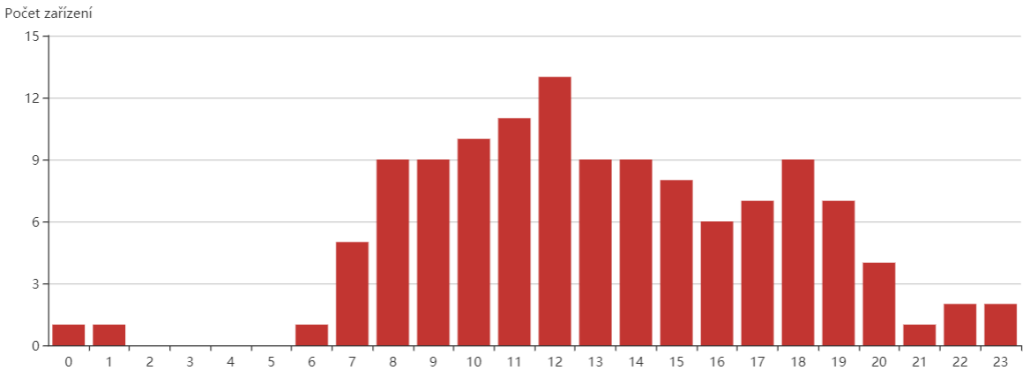


Obrázek 7.5: Report počtu zpráv během návštěv

Obrázek 7.5 ukazuje, že největší počet návštěv je tvořen jedním až čtyřmi záznamy, další poměrně velkou část tvoří návštěvy s 5-20 záznamy. V pravém sloupci je opět vidět zařízení s nejdelší návštěvou a největší aktivitou z obr. 7.4.

7.1.2 Počet zařízení v průběhu dne

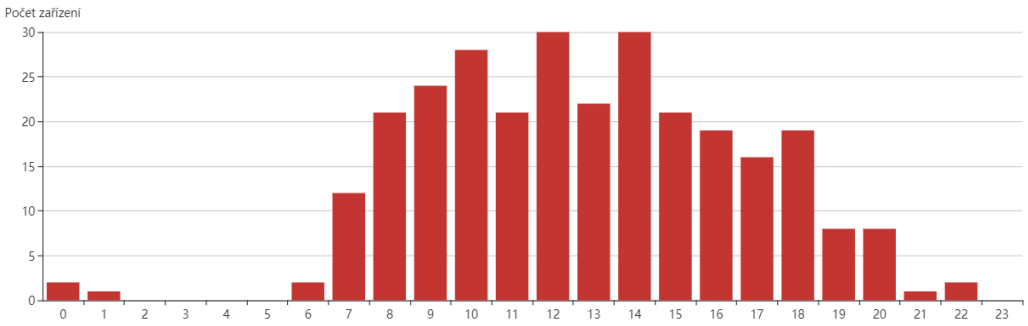
Dalším typem reportu je počet zařízení v průběhu dne za zvolené období. Obrázek 7.6 ukazuje průměrné hodnoty za celé sledované období.



Obrázek 7.6: Report průběhu dne za období 28.3. - 2.4.

Z tohoto grafu lze vypozorovat, že v nočních hodinách je počet přítomných zařízení minimální. Od šesté hodiny začíná počet zařízení stoupat a špička nastává kolem poledne, poté počet opět klesá. Zajímavá je rostoucí hodnota mezi šestou a sedmou hodinou večer, tu je možné vysvětlit tím, že většina lidí odcházela domů. Sběr totiž údajně probíhal u vchodu na

univerzitu, kde pravděpodobně zachytil kumulující počet odcházejících lidí. Také je možné si všimnout jednoho zařízení vyskytujícího se na univerzitě do pozdních nočních hodin.



Obrázek 7.7: Report průběhu dne ze 2.4.

Maximum kolem poledne lze odůvodnit přesunem na oběd, zajímavý je však spíš poměrně nízký počet nalezených zařízení který je způsoben průměrem hodnot ze všech dní. Při zobrazení všedního dne (obr. 7.7 z úterý 2.4) je možné vidět téměř dvakrát vyšší hodnoty.

7.1.3 Zařízení podle výrobců

Další report sleduje četnost výskytů výrobců zařízení. Na obrázku 7.8 je vidět podíl jednotlivých výrobců.

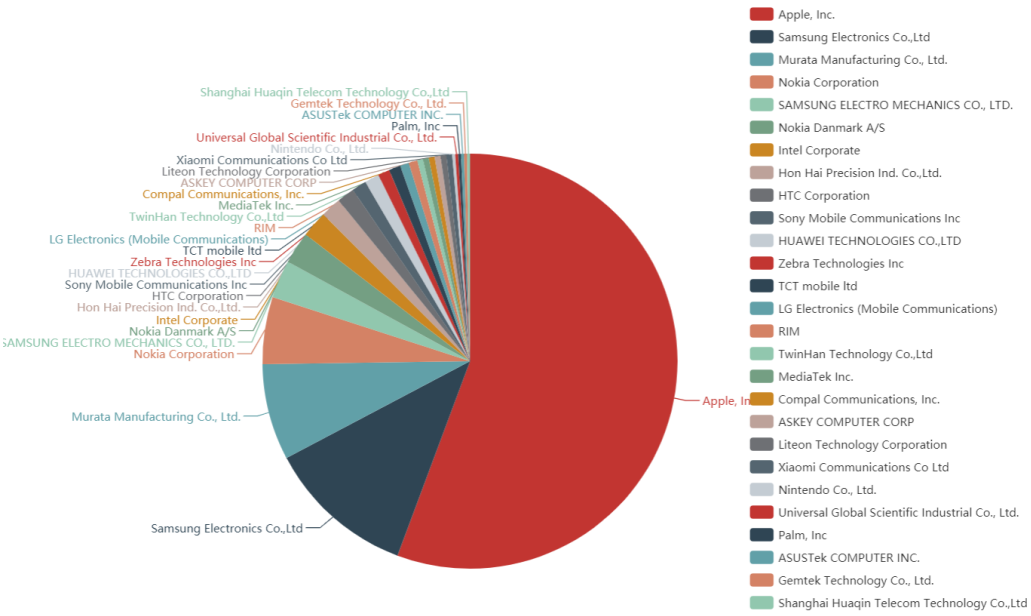
Velkých četností dosahují známé firmy jako je Apple, Samsung, Nokia, Intel, HTC atd. Je možné si všimnout, že databáze[1] výrobců získaná od IEEE obsahuje některé výrobce hned několikrát. Graf je zřejmě ovlivněn trhem mobilních zařízení, v místě ve kterém byl tento záznam pořizován.

7.1.4 Vyhledávané sítě

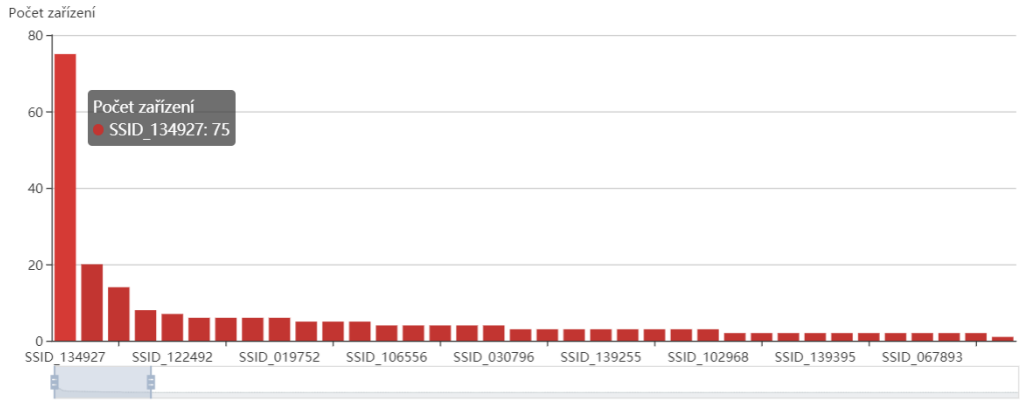
Další report se zaměřuje na sítě, které byly zařízeními explicitně vyhledávané, jsou to sítě, které má dané zařízení uložené. Obrázek 7.9 ukazuje anonymizované názvy sítí podle četnosti.

Nejvíce hledaná síť byla vyhledávána pětasedmdesáti zařízeními, mohlo by jít o univerzitní síť. Na obrázku je vidět jen část grafu, symbol dole pod grafem znázorňuje posuvník a při oddálení grafu, téměř nelze maximum spatřit kvůli mnoha sítím o jedné četnosti. Je zde možno pozorovat tzv. "dlouhý ocas", tedy že málo sítí je hledáno hodně zařízeními a je mnoho sítí vyhledávaných málo zařízeními.

[1] https://regauth.standards.ieee.org/standards-ra-web/pub/view.html#registries



Obrázek 7.8: Report výrobců zařízení

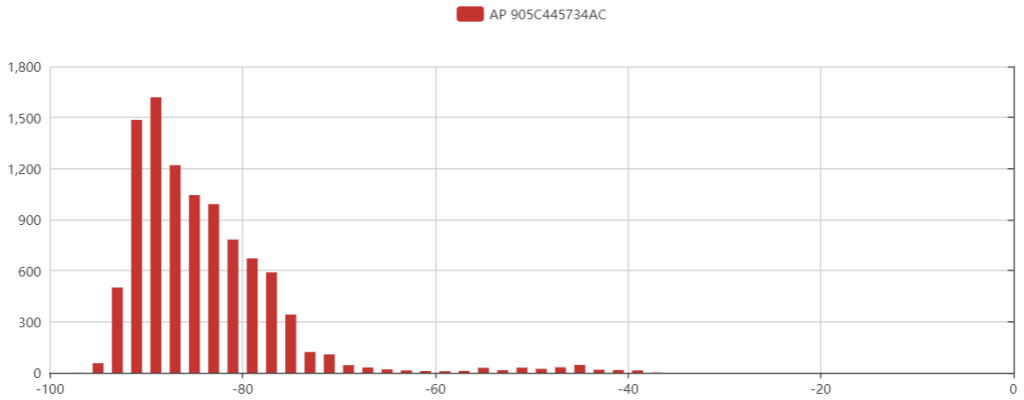


Obrázek 7.9: Report vyhledávaných sítí

7.1.5 Síla signálu podle AP

Poslední report ukazuje četnosti síly signálu jednotlivých AP. Má za cíl sledovat například vhodnost umístění přístupového bodu. Na obrázku 7.10 je jsou vidět hodnoty jen pro jedno AP, což je způsobeno tím, že z datasetu není možné u záznamů rozlišit z kterého AP byl pořízen.

Tento report tedy postrádá pro tento dataset smysl. Podobně na těchto datech nemá smysl ukazovat další report, který aplikace umožňuje, a to sledovat aktuální počet zařízení na jednotlivých AP, který je průběžně aktualizován každé dvě vteřiny.



Obrázek 7.10: Report síly signálu podle AP

7.2 Užitečnost a hodnota dat

Předchozí kapitola dokazuje použití aplikace na reálných datech a ukazuje, že je na základě reportů možné pozorovat určité trendy sledovaných zařízení.

Report s počty zařízení v průběhu dne umožňuje sledovat špičky návštěvnosti a reporty s návštěvami dokáží odhadovat délku trvání pobytu zařízení, či jeho nositele v pozorovaných prostorech. Také lze sledovat aktuální počet zařízení.

Kromě toho je možné sledovat četnosti výrobců zařízení a četnost pozorovaných sítí, ze kterých lze někdy podle názvů identifikovat (pokud nejsou anonymizována) veřejná místa, která jsou uživateli zařízení navštěvována.

7.3 Soulad s platnou legislativou

Vzhledem k tomu, že MAC adresa je osobním údajem a dochází k jejímu zpracování, a byť je ihned po získání upravena hashovací funkcí, platí na ni zákon o ochraně osobních údajů a dnes už téměř platné nařízení GDPR. Více o právních důsledcích v kap. 3.4.

Záznamy pořizované během vývoje a debugování této aplikace (vyjma pozorovaného datasetu, který obsahuje již anonymizovaná data, která nelze vztáhnout k žádným osobám) nebyly uchovávány a tedy nejsou nikde uloženy.

7.4 Možnosti rozšíření

Práci je možné rozšířit například o počítání polohy sledovaných zařízení pomocí sesbírané síly signálů z více přístupových bodů, které by mohlo být

použito třeba ke generování teplotních map, nebo přesnějšímu sledování pohybu zařízení.

Také by bylo možné implementovat způsob pro identifikaci zařízení, která randomizují svou MAC adresu pomocí technik popsaných v kapitole 2.3. Aby bylo možné zařízení spolehlivě rozlišit, bylo by vhodné nejprve odladit vyhodnocování polohy a tuto informaci při identifikaci zohlednit.

Další možností rozšíření je implementace dalších reportů, například zobrazení síly signálu zpráv v průběhu jedné návštěvy. Dále je možné navěsit reakce na různé události v systému, např. při objevení nebo zmizení některého ze zařízení.

8 Závěr

Výsledkem této diplomové práce je aplikace, která umožňuje sbírat data o okolních mobilních zařízeních pomocí WiFi, konkrétně Probe request rámců.

Práce obsahuje přehled dostupných způsobů detekce mobilních zařízení v budově se zaměřením na pasivní metody detekce, u kterých není vyžadována předchozí příprava detekovaného zařízení např. formou instalace aplikace, viz kapitola 2. Součástí práce je také přehled základní legislativy v oblasti sběru dat (kapitola 3).

Návrhu aplikace je věnována celá kapitola 4. Aplikace je navržena modulárně a splňuje funkční požadavky definované zadavatelem, viz kapitola 4.1. Aplikace ukládá detekovaná mobilní zařízení a jejich základní veřejně dostupná metadata (např. přibližná lokace v podobě nejbližšího přístupového bodu) do relační databáze PostgreSQL. Aplikace umožňuje sesbíraná data prohledávat, filtrovat a zobrazovat relevantní údaje o detekovaných zařízeních.

Aplikace byla implementována pomocí technologií Java 1.8, Spring, Spring Boot, JPA a Hibernate, Lombok (backendová část) a Angular 5, Bootstrap 4, ECharts (frontendová část). Testování aplikace proběhlo pomocí několika Unit testů a reálného datasetu z prostředí univerzity (viz kapitola 6).

Nad tímto datasetem bylo provedeno vyhodnocení výsledků, jemuž je věnována kapitola 7, kde byl zároveň vyhodnocen soulad aplikace s platnou legislativou.

Literatura

[1] IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*. Dec 2016, s. 1–3534. doi: 10.1109/IEEESTD.2016.7786995.

[2] Cisco Meraki Location Analytics. `https://documentation.meraki.com/MR/Monitoring_and_Reporting/Location_Analytics`. Accessed: 2018-3-09.

[3] GDPR, Obecné nařízení o ochraně osobních údajů prakticky. `https://www.gdpr.cz/gdpr/`. Accessed: 2018-3-09.

[4] How MAC address randomization works on Windows 10. `http://www.mathyvanhoef.com/2016/03/how-mac-address-randomization-works-on.html`. Accessed: 2018-3-09.

[5] Passive and active scanning. `https://www.wi-fi.org/knowledge-center/faq/what-are-passive-and-active-scanning`. Accessed: 2018-3-09.

[6] 2014 Zogby Analytics - Millennial Study. `https://www.miteksystems.com/sites/default/files/Documents/zogby_final_embargo_14_9_25.pdf`. Accessed: 2018-3-09.

[7] Zákon č. 101/2000 Sb., o ochraně osobních údajů. `https://www.uoou.cz/VismoOnline_ActionScripts/File.ashx?id_org=200144&id_dokumenty=21409`. Accessed: 2018-3-09.

[8] Compare best JavaScript chart libraries 2017. `https://blog.sicara.com/compare-best-javascript-chart-libraries-2017-89fbe8cb112d`, . Accessed: 2018-3-09.

[9] 9 best JavaScript charting libraries. `https://hackernoon.com/9-best-javascript-charting-libraries-46e7f4dc34e6`, . Accessed: 2018-3-09.

[10] Mikrotik USB Port with WiFi USB Dongle. `https://forum.mikrotik.com/viewtopic.php?t=92460`. Accessed: 2018-3-09.

[11] Wireshark WLAN Capture Setup. https://wiki.wireshark.org/CaptureSetup/WLAN. Accessed: 2018-3-09.

[12] Bahl, P. – Padmanabhan, V. N. RADAR: an in-building RF-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, 2, s. 775–784 vol.2, 2000. doi: 10.1109/INFCOM.2000.832252.

[13] Barbera, M. V. et al. CRAWDAD dataset sapienza/probe-requests (v. 2013-09-10). Downloaded from https://crawdad.org/sapienza/probe-requests/20130910, September 2013.

[14] Deak, G. – Curran, K. – Condell, J. A survey of active and passive indoor localisation systems. *Computer Communications*. 2012, 35, 16, s. 1939 – 1954. ISSN 0140-3664. doi: http://dx.doi.org/10.1016/j.comcom.2012.06.004. Dostupné z: http://www.sciencedirect.com/science/article/pii/S014036641200196X.

[15] Enge, P. K. The Global Positioning System: Signals, measurements, and performance. *International Journal of Wireless Information Networks*. 1994, 1, 2, s. 83–105. ISSN 1572-8129. doi: 10.1007/BF02106512. Dostupné z: http://dx.doi.org/10.1007/BF02106512.

[16] Finkenzeller, K. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*. Wiley Publishing, 3rd edition, 2010. ISBN 0470844027, 9780470665121.

[17] Freudiger, J. How Talkative is Your Mobile Device?: An Experimental Study of Wi-Fi Probe Requests. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '15, s. 8:1–8:6, New York, NY, USA, 2015. ACM. doi: 10.1145/2766498.2766517. Dostupné z: http://doi.acm.org/10.1145/2766498.2766517. ISBN 978-1-4503-3623-9.

[18] Gast, M. S. *802.11 Wireless Networks: The Definitive Guide, Second Edition*. O'Reilly Media, Inc., 2005. ISBN 0596100523.

[19] Gentry, D. – Pennarun, A. Passive Taxonomy of Wifi Clients using MLME Frame Contents. Technical report, Google, Inc., 2016.

[20] Liu, H. et al. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*

(Applications and Reviews). Nov 2007, 37, 6, s. 1067–1080. ISSN 1094-6977. doi: 10.1109/TSMCC.2007.905750.

[21] Martin, J. et al. A Study of MAC Address Randomization in Mobile Devices and When it Fails. *CoRR*. 2017, abs/1703.02874. Dostupné z: http://arxiv.org/abs/1703.02874.

[22] Matte, C. – Cunche, M. Spread of MAC address randomization studied using locally administered MAC addresses use historic. Research Report RR-9142, Inria Grenoble Rhône-Alpes, January 2018. Dostupné z: https://hal.inria.fr/hal-01682363.

[23] Matte, C. et al. Defeating MAC Address Randomization Through Timing Attacks. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '16, s. 15–20, New York, NY, USA, 2016. ACM. doi: 10.1145/2939918.2939930. Dostupné z: http://doi.acm.org/10.1145/2939918.2939930. ISBN 978-1-4503-4270-4.

[24] Pahlavan, K. – Li, X. – Makela, J. P. Indoor geolocation science and technology. *IEEE Communications Magazine*. Feb 2002, 40, 2, s. 112–118. ISSN 0163-6804. doi: 10.1109/35.983917.

[25] Pirzada, N. et al. Comparative Analysis of Active and Passive Indoor Localization Systems. *AASRI Procedia*. 2013, 5, s. 92 – 97. ISSN 2212-6716. doi: http://dx.doi.org/10.1016/j.aasri.2013.10.063. Dostupné z: http://www.sciencedirect.com/science/article/pii/S2212671613000644.

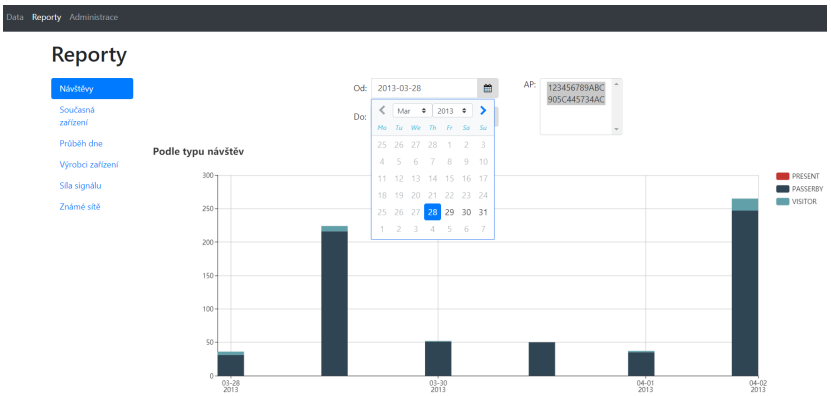
[26] Žůrek, J. J. *Praktický průvodce GDPR*. ANAG, 1st edition, 2017. ISBN 978-80-7554-097-3.

[27] Vanhoef, M. et al. Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. In *ACM AsiaCCS*, Xi'an, China, May 2016. doi: 10.1145/2897845.2897883. Dostupné z: https://hal.inria.fr/hal-01282900.

Přílohy

A Uživatelská příručka

Aplikace se dělí do dvou stránek, první pro zobrazování reportů a druhá pro prohlížení dat, nahoře je menu pro přepínání mezi nimi.

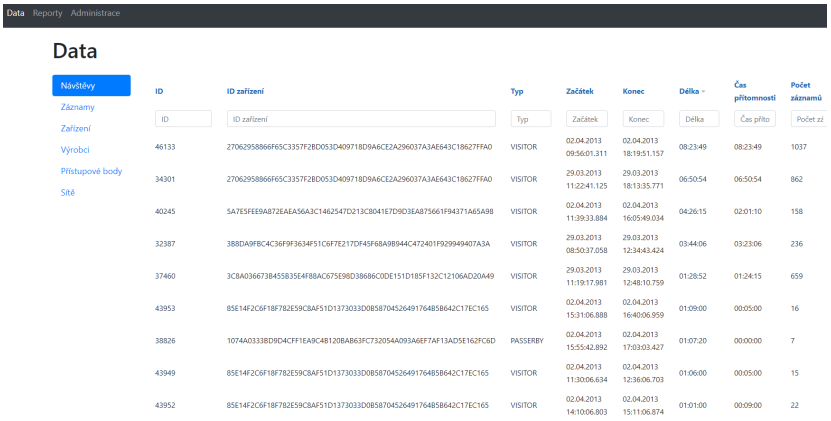


Obrázek A.1: Ukázka stránky reportů

Po kliknutí na stránku s reporty se vlevo objeví menu, které umožňuje vybrat konkrétní report A.1. Načítání dat probíhá na pozadí a může chvíli trvat, než se graf vykreslí.

Většina reportů umožňuje další nastavení pomocí zobrazeného formuláře. Kromě toho je možné volit AP ke kterému se vážou výsledky.

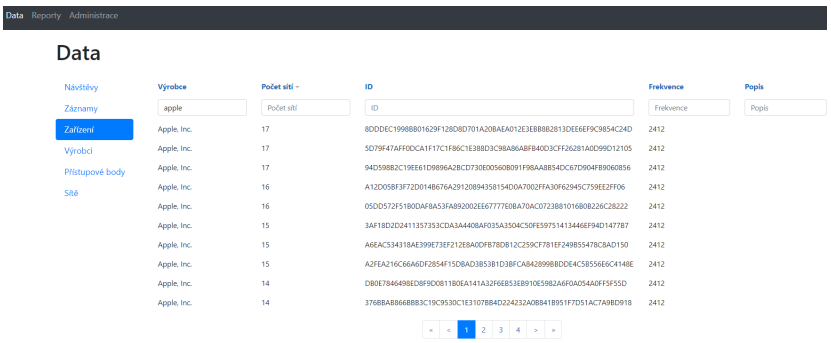
Při vybrání sekce data se zobrazí následující obrazovka s daty, viz obr. A.2.



Obrázek A.2: Ukázka stránky s daty

Na této stránce je možné si prohlížet data, řadit je a vyhledávat v nich pomocí políček nad jednotlivými sloupci. Na obrázku A.3 je vidět ukázka

vyfiltrování zařízení podle výrobce Apple a seřazení podle počtu známých sítí, které zařízení vyhledávalo.



Obrázek A.3: Ukázka filtrování a řazení záznamů

B Email od uoou.cz

ÚŘAD PRO OCHRANU OSOBNÍCH ÚDAJŮ
Pplk. Sochora 27, 170 00 Praha 7
tel.: 234 665 111, fax: 234 665 444
posta@uoou.cz, www.uoou.cz

Čj. UOOU-11670/17-2
Vyřizuje: Mgr. Andrea Čermáková

Vážený pan
Matěj Lochman
matej.lochman@gmail.com

Praha 25. ledna 2018

Wi-fi tracking

Úřad pro ochranu osobních údajů (dále jen „Úřad“) obdržel dne 28. 11. 2017 Váš dotaz ohledně zpracování osobních údajů v rámci wi-fi trackingu v souvislosti s přípravou diplomové práce. Uvádíte, že účelem je anonymní sběr statistik o okolních pohybujících se zařízení. K tomu Vám sděluji následující:

Při využití wi-fi trackingu za účelem statistického vyhodnocování návštěvnosti bez zpracování osobních údajů se Úřad kloní k názoru, že je možné ho provozovat bez souhlasu subjektu údajů, jelikož se nejedná o zpracování osobních údajů.

MAC adresu však považuje současná právní úprava, tedy zákon č. 101/2000 Sb., o ochraně osobních údajů a o změně některých zákonů, jakož i úprava účinná od 25. 5. 2018, tedy nařízení Evropského parlamentu a Rady (EU) 2016/679 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů a o zrušení směrnice 95/46/ES (obecné nařízení o ochraně osobních údajů) – dále jen „obecné nařízení“ za osobní údaj. Ani po hashování MAC adresy nelze hovořit o anonymním údaji.

Jsou-li zaznamenávány v rámci wi-fi trackingu osobní údaje, je nutné na takové jednání nahlížet jako na zpracování osobních údajů. To lze provádět na základě právních důvodů. Zákon o ochraně osobních údajů se jim věnuje v ustanovení § 5 odst. 2, v obecném nařízení jsou vyjmenovány v čl. 6. Případ wi-fi trackingu, pokud pomineme souhlas dotčených osob, by bylo možné provozovat jedině na základě právního důvodu „oprávněné zájmy správce“ dle ustanovení § 5 odst. 2 písm. e) zákona o ochraně osobních údajů, následně dle čl. 6 odst. 1 písm. f) obecného nařízení. Tomu však musí předcházet pečlivá analýza daného oprávněného zájmu v komparaci s právy a svobodami fyzických osob.

Pokud máte na mysli provozováním wi-fi trackingu *ve firemním prostředí* jeho využití na pracovišti, lze odkázat na stanovisko skupiny WP29 ke zpracování údajů na pracovišti dostupné na níže uvedených internetových stránkách:
https://www.uoou.cz/assets/File.ashx?id_org=200144&id_dokumenty=28294

Obrázek B.1: Korespondence s uoou.cz (1)

Nařízení ePrivacy by mělo chránit důvěrnost elektronických komunikací jak fyzických osob, tak právnických osob (ekvivalent listovního tajemství).

S pozdravem

Mgr. Ladislav Hejlík
vedoucí oddělení konzultací
podepsáno elektronicky

Obrázek B.2: Korespondence s uoou.cz (2)