VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020 Zdeněk Bakó

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

ANALÝZA BEZDRÁTOVÉHO PROVOZU WI-FI
ANALYSIS OF WI-FI TRAFFIC

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Zdeněk Bakó**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Jan Pospíšil**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Telekomunikační a informační systémy**

Ústav telekomunikací

Student: Zdeněk Bakó ***ID:*** 197825
Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Analýza bezdrátového provozu Wi-Fi

POKYNY PRO VYPRACOVÁNÍ:

Bude provedena analýza protokolu IEEE 802.11 se zaměřením na současné standardy. Dále bude provedena evaluace open-source alternativních firmware, tj. např. OpenWRT a jeho alternativy, na vybraném hardware. Půjde o praktické srovnání konfigurovatelnosti a možnosti rozšíření jednotlivých firmware. Výsledkem bude výběr nejvhodnějšího řešení. V neposlední řadě pak budou zkoumány možnosti zachytávání rámců protokolu IEEE 802.11 a jejich následná analýza. Bude porovnán promiskuitní a monitorovací mód a možnosti jejich využití. V praktické části pak bude na základě zjištěných poznatků zprovozněn router s alternativním firmwarem doplněný o modul umožňující zachytávání provozu IEEE 802.11 v monitorovacím módu a dále budou rozebrány možnosti analýzy datového provozu v reálném čase. Výsledné zařízení tak bude schopno generovat statistiky provozu.

DOPORUČENÁ LITERATURA:

[1]GONG, Michelle, Brian HART a Shiwen MAO. Advanced Wireless LAN Technologies: IEEE 802.11AC and Beyond. GetMobile: Mobile Computing and Communications [online]. ACM, 2015, 18(4), 48-52 [cit. 2019-09-15]. DOI: 10.1145/2721914.2721933. ISSN 15591662.

[2]RIGELSFORD, Jon. 802.11 Wireless Networks: The Definitive Guide. Sensor Review [online]. Emerald Group Publishing Limited, 2003, 23(2) [cit. 2019-09-15]. DOI: 10.1108/sr.2003.08723bae.003. ISSN 0260-2288.

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

Vedoucí práce: Ing. Jan Pospíšil
Konzultant: Ing. Radek Fujdiak, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

ABSTRAKT

Tato bakalářská práce se zabývá zachytáváním a analýzou bezdrátového provozu Wi-Fi sítí. Cílem této práce je popsat možnosti zachytávání a analýzy provozu Wi-Fi sítí. Je zde popsán standard IEEE 802.11, jeho vlastnosti jako architektura a současné standardy. Je zde zahrnut výběr vhodného routeru, na kterém je zachytávání provozu a analýza realizována. Dále jsou porovnány některé open-source firmware určené pro routery a je vybrán nejvhodnější z nich. Porovnávány jsou takové nástroje určené pro zachytávání a analýzu provozu Wi-Fi sítí. V práci jsou popsány rozdíly mezi promiskuitním a monitorovacím režimem zachytávání a jejich možnostmi využití. V neposlední řadě jsou zde uvedeny možnosti analýzy zachycených rámců 802.11.

KLÍČOVÁ SLOVA

Wi-Fi, 802.11, zachytávání, monitorovací režim, promiskuitní režim, rámce, analýza, router, open-source, firmware, pakety, provoz,

ABSTRACT

This bachelor thesis deals with the capture and analysis of wireless traffic of Wi-Fi networks. The aim of this work is to describe the possibilities of capturing and analyzing the operation of Wi-Fi networks. It describes the IEEE 802.11 standard, its features such as architecture and current standards. A selection of a suitable router is performed on which traffic capture and analysis is performed. Some open-source firmware for routers is compared and the most suitable one is selected. Tools for capturing and analyzing Wi-Fi network traffic are compared and described. The differences between the promiscuous and monitoring capture mode and their possibilities of use are described. Finally, there are described options for analyzing captured 802.11 frames.

KEYWORDS

Wi-Fi, 802.11, capturing, monitor mode, promiscuous mode, frames, analyze, router, open-source, firmware, packets, traffic,

BAKÓ, Zdeněk. *Analýza bezdrátového provozu Wi-Fi*. Brno, 2020, 104 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Jan Pospíšil,

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Analýza bezdrátového provozu Wi-Fi“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Janu Pospíšilovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

Bezdrátové sítě prochází neustálým vývojem a jsou již téměř nedílnou součástí našich životů. Zejména bezdrátové sítě standardu IEEE 802.11, neboli technologie Wi-Fi („Wireless-Fidelity“), se těší obrovskému úspěchu a počet zařízení využívajících tuto technologii neustále roste. Aktuálně se počet aktivních zařízení pohybuje v řádu miliard, konkrétní odhad je přes 13 miliard aktivních zařízení po celém světě [1]. Dostupnost Wi-Fi sítí se rozšiřuje už i do veřejných prostranství měst, nebo do vozů městské hromadné dopravy. S tím vznikají nové možnosti jejich využití.

Wi-Fi sítě se však nejčastěji využívají pro připojení zařízení k síti Internet a komunikaci s ostatními zařízeními v síti jako jsou počítače, mobilní telefony, tiskárny, televize a další. Své využití nachází i v sítích IoT („Internet of Things“), které zažívají velký vzestup. S nárůstem počtu zařízení komunikujících přes Wi-Fi sítě, může být žádoucí provoz v těchto bezdrátových sítích monitorovat a analyzovat. Sledování a analyzování provozu může sloužit pro stanovení statistik o provozu, ke kontrole přenášených informací, ale zejména může pomoci odhalit bezpečnostní rizika a případné chyby v síti. Své uplatnění nachází i při návrhu a budování nových Wi-Fi sítí.

Tato bakalářská práce se zabývá možnostmi zachytávání bezdrátového provozu Wi-Fi sítí, možnostem jeho analýzy, a to za pomoci Wi-Fi routeru s použitím open-source firmware a open-source nástrojů. Jedná se o komplexní analýzu, která zahrnuje výběr vhodného hardware, software a klade si za cíl popsat problematiku zachytávání a analýzy Wi-Fi provozu. Teoretická část této práce se zabývá popisem vlastností standardu IEEE 802.11. Je zde krátce popsána jeho historie, architektura vrstev a techniky přenosu na fyzické vrstvě. Dále také formát rámců a přehled používaných standardů 802.11.

Praktická část začíná srovnáním a výběrem open-source firmware, určeného pro routery. Firmwary jsou porovnány na základě jejich vlastností, jako je konfigurovatelnost, možnosti jejich rozšíření a použití pro zachytávání provozu Wi-Fi sítí. Třetí kapitola se zabývá výběrem vhodného hardware, konkrétně Wi-Fi routeru, na kterém je zachytávání provozu realizováno. Porovnáváno je pět moderních Wi-Fi routerů, které jsou schopny provozu open-source firmware. Dále jsou popsány a porovnány režimy bezdrátových sítových karet, které umožňují zachytávaní paketů a rámců. Pozornost je věnována zejména tzv. monitorovacímu režimu. Pátá kapitola se věnuje nástrojům umožňujícím zachytávání provozu Wi-Fi sítí. Jsou zde popsány některé open-source nástroje pro monitorování, zachytávání a analýzu provozu v síti. V dalších kapitolách je zrealizováno zachytávání provozu na vybraném hardware a firmware. Jsou rozebrány možnosti a rozdíly zachytávání provozu v monitorovacím a promiskuitním režimu a možnosti jeho následné analýzy.

1 Standard IEEE 802.11

1.1 Historie standardu IEEE 802.11

Mezinárodní organizace IEEE („Institute of Electrical and Electronics Engineers“) založila roku 1990 pracovní skupinu s názvem 802.11. Tato skupina měla za úkol vytvořit standard pro bezdrátové lokální sítě, neboli WLAN („Wireless Local Area Network“), který by definoval tzv. bezdrátový Ethernet [2]. Tento standard, označený jako IEEE 802.11, byl schválen a publikován v roce 1997 a definuje přenosové schéma fyzické vrstvy a způsob řízení přístupu k médiu pro WLAN [3]. Později v roce 1999 vzniklo obchodní sdružení WECA („Wireless Ethernet Compatibility Alliance“), které certifikuje zařízení splňující standard IEEE 802.11, a vymyslelo pro tento standard obchodní název Wi-Fi. Sama organizace WECA byla později v roce 2002 přejmenována na Wi-Fi Alliance [4, 5]. Ve stejném roce bylo také vydáno první rozšíření původního standardu IEEE 802.11.

1.2 Architektura 802.11

Tak jako jiné standardy z rodiny IEEE 802, jako například IEEE 802.3 (Ethernet) nebo IEEE 802.5 (Token Ring), tak i 802.11 se zaměřuje na dvě nižší vrstvy modelu ISO/OSI: fyzickou a linkovou. Nad těmito dvěma vrstvami operuje podvrstva LLC („Logical Link Control“), která je zde specifikována stejně jako ve standardu 802.2. [3]

1.3 Fyzická vrstva 802.11

1.3.1 Frekvenční pásma

Komunikace ve Wi-Fi sítích probíhá na rádiových vlnách v bezlicenčních kmitočtových pásmech ISM („Industrial, Scientific, Medical“). Tato pásma mohou být využívána volně kýmkoliv za předpokladu dodržení určitých pravidel, která tyto pásma regulují. Nevýhodou těchto pásem je, že je využívají i další různé technologie a zařízení, např. pásmo 2,4 GHz (gigahertz) využívá také Bluetooth, RFID („Radio Frequency Identification“), mikrovlnné trouby, meteostanice nebo DECT („Digital Enhanced Cordless Telecomunications“) telefony. Z toho důvodu může v těchto pásmech docházet k častějšímu rušení. Wi-Fi sítě pracují ve frekvenčních pásmech 2,4 GHz, 3,7 GHz, 5GHz, 6 GHz a 60 GHz. Dále jsou rozebrána pouze dvě v Evropě nejčastěji využívaná pásma 2,4 a 5 GHz.

1.3.2 Pásmo 2,4 GHz

Pásmo začíná od 2,4 a končí v 2,4835 GHz, šířka pásma je tedy 83,5 MHz (megahertz). Maximální hodnota EIRP („Equivalent Isotropically Radiated Power“) je v ČR omezena na 100 mW. Toto pásmo je rozděleno do 14 kanálů o šířce 22 MHz, mezi kterými je vzájemný odstup 5 MHz, pouze kanál číslo 14 má odstup 12 MHz. Využití jednotlivých kanálů je regulováno v různých státech jinak. Kupříkladu v ČR je regulacemi ČTU (Český telekomunikační úřad) zakázáno využívání 14. kanálu. Některé varianty standardu 802.11 však používají jiné šířky kanálů než původní standard. Pouze tři ze čtrnácti kanálů se vzájemně nepřekrývají, v ČR a EU to jsou kanály č. 1, 7 a 13 [2, 7].

1.3.3 Pásmo 5 GHz

Toto pásmo je rozděleno na dva nesouvislé bloky, a to od 5,150 do 5,350 GHz a od 5,470 do 5,725 GHz. Maximální přípustná hodnota EIRP je zde 200 mW pro první 200MHz frekvenční blok, který lze však využít jen „indoor“, tedy uvnitř budov. Pro druhý 255MHz blok platí regulace EIRP na max. 1 W a může být využit i mimo budovy. Šířka jednoho kanálu v 5 GHz pásmu je 20 MHz a celkově je v Evropě k dispozici 19 kanálů, mezi kterými je rozestup 20 MHz [2, 7].

1.3.4 Podvrstvy

Fyzická vrstva je rozdělena na dvě podvrstvy [3, 6]:

- **PLCP** („Physical Layer Convergence Protocol“) - zajišťuje komunikaci mezi fyzickou podvrstvou PMD („Physical Medium Dependent“) a MAC („Media Access Control)“ podvrstvou. Komunikace je zajištěna díky mapování MPDU („MAC protocol data unit“) na PPDU („PLCP protocol data unit“), které jsou následně přenášeny po bezdrátovém médiu. Tato komunikace je zajištěna i opačným směrem, kdy PLCP doručuje příchozí rámce na MAC podvrstvu.

- **PMD** - je zodpovědná za odesílání a přijímaní datových jednotek fyzické vrstvy přes bezdrátové médium a také určuje přenosové schéma. Konkrétně provádí kódování, modulaci a demodulaci přenosových rámců.

Fyzická vrstva pomocí těchto dvou podvrstev zajišťuje komunikaci s vyšší vrstvou, ale hlavně definuje použitou modulaci a techniku rádiového přenosu. Právě technika, jakou je rádiový signál přenášen skrze bezlicenční pásmo, je důležitá z pohledu odolnosti vůči rušení a efektivity přenosu. Techniky přenosu používané u Wi-Fi

sítí jsou DSSS („Direct Sequence Spread Spectrum“), FHSS („Frequency Hopping Spread Spectrum“) a OFDM („Orthogonal Frequency Division Multiplexing“).

1.3.5 DSSS

Technika přímo rozprostřeného spektra spočívá v nahrazení jednotlivých bitů vysílaných dat, za sekvenci čipů, které dohromady tvoří symboly. Tyto symboly jsou pak vysílačem modulovány. To způsobuje umělé zvětšení šířky pásma a to v závislosti na délce sekvence chipů. Díky rozprostření signálu do širší části spektra je signál méně náchylný na rušení. Tato technika je využita u standardu IEEE 802.11b. [2, 8].

1.3.6 FHSS

Skupiny vzájemně komunikujících stanic vysílají data na jednom kmitočtu po velmi krátkou dobu a následně přeskočí na jinou frekvenci, kde v komunikaci pokračují. Doba, po kterou stanice vysílají na jednom kmitočtu se nazývá „dwell time“ a trvá 400 ms (milisekund). Stanice tedy změní kmitočet 2,5krát za vteřinu. Volba kmitočtu, na který stanice přeskočí je pseudonáhodná, ve skutečnosti jde totiž o pořadí kmitočtů, které je známé vysílající i přijímací stanici. Jiná skupina komunikujících stanic používá jiný pseudonáhodný seznam kmitočtů. Díky těmto přeskokům se minimalizuje šance, že více stanic bude vysílat na stejném kmitočtu zároveň.

Tato technika se používala pouze u původní verze standardu 802.11, novější verze standardu 802.11 ji již nepoužívají, využívá se však u technologie Bluetooth. [2, 3, 8]

1.3.7 OFDM

Ortogonální multiplex s frekvenčním dělením rozděluje kanály do několika subkanálů, po kterých jsou data přenášena v několika paralelních bitových tocích. Každý z těchto toků se používá pro modulaci jiné nosné. Konkrétními modulacemi mohou být jakékoliv typy digitální modulace jako QPSK („Quadrature phase shift keying“), 16-QAM („Quadrature amplitude modulation“) či 64-QAM. Díky paralelnímu vysílání je nižší možnost zkreslení signálu při přenosu různými cestami. Tuto techniku využívají standardy IEEE 802.11a/n/ac. [8, 11]

1.3.8 MIMO

SISO neboli „Single-Input Single-output“ využívá pouze jeden datový tok („stream“) skrze jednu anténu na vysílači a přijímači. SISO bylo využíváno v původním standardu IEEE 802.11 a jeho rozšířeních 802.11a/b/g. **MIMO** („Multiple-Input Multiple-output“) je technologie, která umožňuje dosáhnout vyšší spektrální účinnosti přená-

šeného signálu. MIMO využívá kombinace více antén na vysílači a přijímači, přičemž počet antén na obou stranách nemusí být stejný. Více antén umožňuje současně vysílat více signálů, jenž se šíří prostorem různými cestami po stejném kanálu. Těmto signálům se také říká prostorové streamy (anglicky „spatial streams“). Dále se využívá skutečnosti, že vysílané signály se nešíří pouze jedním směrem, ale odráží se od překážek v prostoru. To, v kombinaci s více anténami, způsobuje lepší propustnost a zlepšení síly signálu na přijímači využívajícím MIMO [8, 9]. V MIMO se využívá dvou technik:

- **Prostorová diverzita** (anglicky „spatial diversity“) využívá vícecestného šíření jednotlivých streamů v prostoru, kdy dochází k odrazům signálů od zdí a překážek. Jeden stejný datový stream je zároveň vysílán více anténami za použití časo-prostorového kódování. Digitální signálové procesory v přijímačích kombinují jednotlivé přijaté duplicitní datové streamy, a tím dochází ke zlepšení kvality přijatého signálu. Zároveň se také zvyšuje pravděpodobnost, že se alespoň jeden prostorový stream dorazí k přijímači. [10]
- **Prostorový multiplexing** (anglicky „spatial multiplexing“) umožňuje současně ve stejný čas vysílat více nezávislých signálů, neboli prostorových streamů, na stejném kanálu, přičemž každý stream pomocí jiné antény vysílače. Po každém streamu tedy lze vysílat rozdílná data. Při využití více prostorových streamů je tedy možno přenést více dat současně a tím se znatelně zvyšuje propustnost Wi-Fi sítě a efektivita využití pásma. Aby bylo možné využívat prostorových streamů, musí tuto technologii podporovat jak vysílající zařízení, tak zařízení jenž má signál přijímat. [6, 10]

Existují dvě varianty MIMO, a to SU-MIMO („Single User MIMO“) a MU-MIMO („Multi User MIMO“).

- **SU-MIMO** - umožňuje vysílat a přijímat více datových toků (streamů) zároveň, pouze do/od jediného zařízení, které podporuje MIMO a disponuje více anténami. V jednom čase tedy lze komunikovat pouze s jedním zařízením. Tuto variantu využívá standard 802.11n. [12]
- **MU-MIMO** - umožňuje vysílat více datových toků (streamů) zároveň, do více zařízení zároveň. To zvyšuje celkovou propustnost sítě a její kapacitu. Je implementováno od standardu 802.11ac, kde však funguje MU-MIMO pouze pro downstream, zatímco novější 802.11ax podporuje i upstream. V závislosti na počtu antén rozlišujeme implementaci MU-MIMO. Ty se označují třemi čísly, např. 2x2:2, kde první číslo reprezentuje počet vysílacích antén, druhé číslo počet přijímacích antén a třetí reprezentuje počet prostorových streamů, jenž zařízení podporuje. Například router disponující třemi anténami dokáže typicky pracovat v režimu MIMO 3x3:3, tedy dokáže vysílat tři datové toky

současně a obsloužit až tři klienty zároveň. Případně může vysílat dva datové toky do zařízení, které podporuje 2x2 MIMO a jeden stream do dalšího zařízení. MU-MIMO funguje pouze ve standardu IEEE 802.11ac a 802.11ax. [12, 13]

1.3.9 Beamforming

Beamforming je technika, která umožňuje řízení směrování rádiových vln v systémech podporujících MIMO. Vysílaným signálům může být upravena jejich fáze a amplituda tak, aby dorazili k umístění přijímače a mohli navzájem konstruktivně interferovat. Využívá se tedy konstruktivní interference mezi signály vysílanými z více antén. Analýzou vysílaného a přijímaného signálu lze šíření signálu z vysílače směřovat k přijímači. Výsledkem je lepší kvalita signálu při nižším nebo stejném vysílacím výkonu. Tato technika se využívá od standardu 802.11n, avšak standardizována a plně podporována je až od standardu 802.11ac. [14, 10]

1.4 Linková vrstva

Linková (spojová) vrstva je ve standardu 802.11 rozdělena na dvě podvrstvy:

- **LLC** -Podvrstva má za úkol multiplexování vysílaných protokolů, které jsou přenášeny MAC podvrstvou a jejich demultiplexování při příjmu. Dalším úkolem je detekce a kontrola chyb.

- **MAC** - Podvrstva řídící přístup k médiu. Řídí proces adresace a vysílání dat na bezdrátové médium. K tomuto účelu využívá přístupové metody zvané CSMA/CA.

1.4.1 Přístupová metoda CSMA/CA

Pro přístup ke sdílenému médiu se v 802.11 využívá metoda CSMA/CA („Carrier Sense Multiple Access/Collision Avoidance“), tedy metoda mnohonásobného přístupu s naslouchání nosné a vyvarování se kolizím.

Stanice v sítí před vlastním vysíláním naslouchají na aktuálně používaném přenosovém kanálu. Naslouchání je prováděno měřením signálu zachyceného na anténě stanice. Pokud stanice detekuje signál, jehož síla je větší než specifikovaný práh, tak kanál považuje za obsazený a své vysílání odloží. V opačném případě považuje rádiový kanál za volný.

V CSMA/CA se používají pro řízení přístupu k médiu dvě pomocné funkce a to DCF a PCF.

- **DCF** („Distributed Coordination Function“) - zajišťuje koordinaci přístupu k rádiovému kanálu bez podpory prioritního přístupu QoS („Quality of Service“) a negarantuje zpoždění ani šířku pásma. Je založena na přístupové metodě CSMA/CA. Pro zabránění vzniku kolizí v případě, kdy více stanic detekuje volný kanál a začne vysílat, se využívají časové mezery mezi vysílanými rámci IFS („InterFrame Space“) a tzv. „backoff“, neboli odklad vysílání. Stanice, která chce vysílat, naslouchá na kanálu a pokud jej detekuje jako volný, čeká po dobu intervalu DIFS („Distributed Coordination Function InterFrame Space“). Je-li kanál volný i po dobu intervalu povinného čekání DIFS, stanice začne vysílat. Pokud však v intervalu DIFS začne vysílat jiná stanice, musí čekající stanice své vysílání odložit (tzv. „backoff“). Dojde-li k opětovnému vysílání jiné stanice, doba odkladu se zdvojnásobuje. V případě že data byla příjemcem úspěšně přijata, čeká příjemce po dobu SIFS („Short InterFrame Space“) a pak příjem potvrdí zasláním příznaku ACK („Acknowledgement“) zpět na stanici. [8]

- **PCF** („Point Coordination Function“) - je volitelná koordinační funkce, která kombinuje metodu soutěžení o přístup k médiu (jako v DCF) a nesoutěžní metodu řízenou koordinátorem (tzv. „point coordinator“), kterým je přístupový bod. Koordinátor se postupně dotazuje jednotlivých stanic v síti, zda mají data k přenosu. Dotázaná stanice musí začít vysílat do intervalu SIFS. Pokud se tak nestane a stanice nereaguje, koordinátor pošle dotaz další stanici. Jestliže stanice nemá data k přenosu, vyšle prázdný rámec. Implementace PCF není příliš častá a lze ji využít pouze v infrastrukturních sítích s přístupovým bodem, nikoli v sítích ad-hoc. [2, 8]

Známým problémem metody CSMA/CA je tzv. skrytá stanice (anglicky „hidden station“). Tento problém nastává, když přístupový bod má ve svém rádiovém dosahu více stanic se kterými může komunikovat, ale tyto stanice se navzájem nedetekují. Důvodem je to, že jsou stanice od sebe příliš daleko, nebo za překážkou. Stanice nedetekují vysílání jiné stanice a mají kanál za volný a začnou s vlastním vysíláním. Přístupový bod pak detekuje kolizi a vysílaná data se musí posílat znovu. Ke kolizi může dojít téměř kdykoliv, protože stanice nepozná že právě probíhá vysílání jiné stanice.

Pro zamezení vzniku takových kolizí se problém skryté stanice řeší pomocí volitelného mechanismu RTS/CTS („Request To Send/Clear To Send“).

- **RTS/CTS** - volitelný mechanismus při kterém stanice, která chce vysílat, nejdříve odešle do okolí rámec RTS, kterým žádá o rezervaci kanálu pro vlastní vysílání. Pokud je kanál volný, přístupový bod zašle žádající stanici rámec CTS, kterým je uděleno právo vysílat. Všechny stanice, které zachytí rámce RTS a CTS si nastaví interní časovač, tzv. vektor přidělení sítě NAV („Network Allocation Vector“), na dobu trvání vysílání a budou kanál považovat za obsazený. Při použití mechanismu RTS/CTS klesá přenosová kapacita sítě. [3, 8]

1.4.2 Formát MAC rámce IEEE 802.11

Obecný formát MAC rámce se skládá z devíti polí o celkové délce až 2346 bajtů. Ne všechna pole jsou však využita všemi typy MAC rámců. Ve WLAN sítích standardu 802.11 se používají tři druhy rámců:

- **Řídící rámce** - slouží pro řízení přístupu k médiu a řízení výměny dat. Jde například o rámce ACK, které potvrzují úspěšný přenos a rámce RTS/CTS (viz. kapitola 1.5.1).

- **Management rámce** - slouží zejména pro proces asociace a autentizace při přístupu do sítě a opačně pro proces odpojení ze sítě. Patří sem rámec Beacon, který přístupový bod periodicky vysílá a jenž obsahuje parametry sítě, na jejichž základě se stanice může k síti připojit. Beacon rámec je rozebrán dále v textu. Dalšími management rámci jsou rámce Probe request, Probe response, Association request, Association response, Reassocation request, Reassocation response, Deauthentication, Authentication aj.

- **Datové rámce** - obsahuje přenášená data a informace o zdrojových a cílových adresách.

Význam jednotlivých polí v MAC rámci [17]:

- **Frame control** - 2 bajtové pole obsahuje informace o typu rámce.
  - Protocol version - informace o typu protokolu, aktuálně má hodnotu 0.
  - Type - 2 bity upřesňující zda se jedná o management (00), řídící (01) nebo datový rámec (10).
  - Subtype - 4 bity definující konkrétní typ rámce, například Beacon (1000).
  - To DS („distribution system“) - 1 bit určující zda je rámec určen pro distribuční systému.
  - From DS - 1 bit určující zda rámec pochází z distribučního systému.
  - More fragments - 1 bit informující o tom, zda jedná o poslední fragment nebo zda následují další fragmenty.
  - Retry - 1 bit pro určení zda se jedná o opakovaný rámec.
  - Power management - 1 bitové pole indikující zda stanice po vysílání přechází do úsporného režimu (1) nebo zůstává aktivní (0).
  - More data - 1 bitové pole poskytující informaci pro příjemce o tom, že odesílatel má uloženo více rámců k vysílání.
  - Protected - bit indikující použití zabezpečovacího mechanismu.
  - Order - 1 bitové pole, pokud je nastaveno na 1 příchozí rámce musí být zpracovány ve striktním pořadí.
- **Duration/ID** - 2 bajtové pole, nese informace o tom, jak dlouho bude stanice vysílat, tedy po jak dlouhou dobu bude médium zaneprázdněno pro ostatní stanice. Stanice si dle tohoto pole aktualizují NAV. Pokud se jedná o řídící rámec, nese toto pole tzv. „association ID“, které se používá pro identifikaci stanice, pro kterou jsou dostupné uložené rámce v době kdy je v úsporném režimu [15, 16].
- **Address 1-4** - čtyři 6 bajtová pole obsahující MAC adresy. Obsahují adresu zdroje (SA, „source address“), cílovou adresu (DA, „destination address“), adresu příjemce (RA, „receiver address“), přenášející stanice (TA, „transmitter address“) a BSSID („Basic Service Set Identifier“) [11]. Význam těchto adres je závislý na tom, zda byl rámec poslán z distribučního systému nebo do něj, tedy zda jsou nastaveny bity v poli To DS nebo From DS uvnitř Frame control [17].
- **Sequence control** - 2 bajtové pole obsahující dvě podpole - Sequence number a Fragment number. Obsahují čísla rámců a fragmentů a slouží k filtrování duplicitních rámců.
- **Data** - má variabilní délku 0 až 2312 bajtů. Obsahuje uživatelská data včetně bajtů pro šifrování.
- **CRC** („Cyclic redundancy check“) - 4 bajtové pole pro kontrolní součet rámce.

1.5 Autentizace a asociace

Proces autentizace a asociace, neboli přidružení k přístupovému bod v síti probíhá na základě skenování provozu [8]. Standard 802.11 definuje dva režimy skenování: pasivní a aktivní [3].

- **Pasivní skenování** - Stanice naslouchá na každém kanálu v rozsahu, který jí umožňuje fyzická vrstva. Stanice hledá konkrétní rámce zvané Beacon, které obsahují informace o dostupné síti. Zachycené Beacon rámce si ukládá do seznamu nalezených BSSID. [3]
- **Aktivní skenování** - Při aktivním skenování stanice vysílá na jednotlivých kanálech takzvané Probe rámce. Pokud tento rámec zachytí přístupový bod, odpovídá rámcem Probe response, který nese stejné informace jako Beacon rámec. Probe response rámce musejí být stanicí potvrzeny z důvodu integrity. [3]

Pro proces autentizace a asociace se používají následující management rámce:

- **Beacon** - rámec, který přístupový bod vysílá broadcastem v určitých intervalech. Obsahuje informace o parametrech sítě potřebné k asociaci. Konkrétně obsahuje: BSSID, SSID („Service Set Identifier“), parametry fyzické vrstvy, dostupné kanály, výkonové omezení, interval odeslání dalšího Beacon rámce. Stanice na základě tohoto rámce také mohou určit sílu signálu. [3]
- **Probe request** - pomocí tohoto rámce stanice hledá dostupné přístupové body, ke kterým je možné se asociovat. Obsahuje informace o tom, jaké frekvence a standardy stanice podporuje. [18]
- **Probe response** - Probe response - odpověď, kterou přístupový bod odpovídá na rámec Probe request, pokud má stanice kompatibilní parametry. Obsahuje stejné informace jako Beacon. Je nutné jej využít pokud má Wi-Fi síť skryté SSID a tedy nevysíla Beacon rámce. [18]
- **Authentication** - ověření identity a šifrování.
- **Deauthentication** - slouží pro deautentizaci.
- **Association request** - žádost o asociaci stanice k vybranému přístupovému bodu. Obsahuje například podporované kmitočty a SSID.
- **Association response** - odpověď na žádost o asociaci.
- **Reassociation request** - žádost o opětovnou asociaci.
- **Reassociation response** - odpověď na žádost o opětovnou asociaci.
- **Disassociation** - odpojení od sítě.

1.5.1 Autentizace pomocí WEP

WEP („Wired Equivalent Privacy“) je základní zabezpečovací mechanismus autentizace a šifrování, který je podporován ve standardech 802.11a/b/g [8]. WEP používá proudovou šifru RC4 s klíči o délce 40 nebo 104 bitů, ke kterým se přidává 24 bitový inicializační vektor [19]. V současné době je již WEP naprosto nevyhovujícím zabezpečením, které může být prolomeno během několika minut.

Proces autentizace pomocí WEP je následující. Na základě skenování je vybrán přístupový bod, ke kterému se má stanice připojit. Stanice vyšle Authentication rámec, kterým žádá o ověření identity, kterou je MAC adresa stanice. Přístupový bod odpovídá rovněž rámcem Authentication. Pro WEP existují dvě metody autentizace:

- **Open system authentication** - Neprovádí se defakto žádná autentizace. Stanice vysílá Authentication rámec na jehož základě obdrží od přístupového bodu odpověď. Každá stanice je autentizována [20].
- **Shared key authentication** - Stanice se autentizují pomocí sdíleného klíče WEP. Provádí se ve čtyřech krocích. Stanice zašle přístupovému bodu žádost o autentizaci. Přístupový bod odpoví rámcem, kterým obsahuje čitelný text, tzv. „challange text“. Stanice tento text zašifruje pomocí WEP klíče a odešle autentizační rámec na přístupový bod. Přístupový bod text dešifruje a pokud se text shoduje s odeslaným textem, odpovídá rámcem o úspěšné nebo neúspěšné autentizaci.

1.5.2 Autentizace pomocí WPA/WPA2

V roce 2003 byl nedostatečný WEP nahrazen novým řešením WPA („Wi-Fi Protected Access“). WPA je zpětně kompatibilní s WEP a dopředně s jeho nástupcem WPA2. WPA používá pro utajení dat protokol dynamicky měnící se klíče TKIP („Temporal Key Integrity Protocol“), který sice stejně jako WEP používá šifru RC4 má však delší 48 bitový inicializační vektor. Pro kontrolu integrity zpráv používá mechanismus MIC („Message-Integrity Check“) místo CRC. Dnes již je WPA překonáno a nahrazeno WPA2 [8].

WPA2 přišlo v roce 2004 s doplňkem standardu 802.11i, který rozšiřuje původní standard. Protokol TKIP, který se ukázal jako nedostatečný byl nahrazen protokolem CCMP („Counter-mode Cipher Block Chaining Message Authentication Code Protocol“). Tento protokol zajišťuje šifrování a zajištění integrity a nahrazuje tedy mechanismus MIC. Protokol CCMP implementuje šifrování pomocí AES („Advanced Encryption Standard“) a dynamicky generuje 128 bitové klíče [8, 21].

WPA/WPA2 zavadí dva režimy autentizace :

- **WPA(2) Enterprise** - probíhá identifikace a autentizace uživatele, který přístupovému bodu poskytne autentizační údaje (jméno, heslo či ceriftikát). Přístupový bod si údaje ověří u autentizačního serveru (např. RADIUS nebo Kerberos) [21, 22].
- **WPA(2)-PSK („Pre-Shared Key“)** - identifikuje a autentizuje se stanice. Autentizaci zde provádí přístupový bod na základě znalosti sdíleného klíče (hesla). Pro každou stanici, která zná heslo se odvozují různé klíče. V tomto režimu není zapotřebí autentizační server [22].

1.6 Přehled standardů IEEE 802.11

Standard IEEE 802.11 od svého vzniku prochází neustálým vývojem. V současné době existuje přes 25 specifikací a dodatků tohoto standardu. V této podkapitole je seznam těch nejzásadnějších. Přehled všech popsaných standardů je v tabulce 1.1.

Tab. 1.1: Přehled standardů IEEE 802.11 [10].

Standard	**Označení**	**Rok vydání**	**Frekvenční pásmo [GHz]**	**Šířka pásma**	**Max. přenosová rychlost [Mbit/s]**	**Max. počet streamů**	**Technika fyzické vrstvy**
802.11	-	1997	2,4	22	2	1	FHSS, DSSS
802.11a	Wi-Fi 1	1999	5	20	54	1	OFDM
802.11b	Wi-Fi 2	1999	2,4	22	11	1	DSSS
802.11g	Wi-Fi 3	2003	2,4	20	54	1	OFDM
802.11n	Wi-Fi 4	2009	2,4/5	20/40	600	4	OFDM, MIMO
802.11ac	Wi-Fi 5	2014	5	20/40/80/160	6933	8	OFDM, MU-MIMO
802.11ax	Wi-Fi 6	2019	2,4/5	20/40/80/160	9607.8	8	OFDMA, MU-MIMO

1.6.1 IEEE 802.11a

Jedna z nejstarších specifikací standardu, jenž byla uvedena roku 1999. Pracuje v bezlicenčním frekvenčním pásmu 5 GHz, jehož výhodou je nižší vytížení a větší počet kanálů. Přenos na fyzické vrstvě je řešen pomocí OFDM. Nabízí teoretickou přenosovou rychlost až 54 Mbit/s (megabit za sekundu) [8]. Dnes je tento standard označován též jako Wi-Fi 1 [6].

1.6.2 IEEE 802.11b

Roku 1999 byl schválena specifikace IEEE 802.11b, která pracuje ve frekvenčním pásmu 2,4 GHz a využívá metodu rozprostřeného spektra DSSS s modulací CCK („Complementary Code Keying“). Díky tomu nabízí výrazné vyšší rychlosti, než původní standard 802.11. Maximální teoretická přenosová rychlost činí 11 Mbit/s. Implementuje techniku „dynamic rate shifting“, díky které umožňuje přepínání mezi čtyřmi rychlostmi - 1 Mbit/s, 2 Mbit/s, 5,5 Mbit/s a 11 Mbit/s. Připojená stanice se mezi rychlostmi přepíná v závislosti na síle signálu [8, 23]. Dnes je tento standard označován též jako Wi-Fi 2 [6].

1.6.3 IEEE 802.11g

Specifikace z roku 2003 využívající frekvenční pásmo 2,4 GHz s přenosovo metodou OFDM, díky které dosahuje vyšší rychlosti, než předchůdce. Maximální teoretická rychlost je 54 Mbit/s, tedy stejná jaké je dosaženo ve specifikaci 802.11a v 5 GHz pásmu. Nabízí celkem 8 rychlostí a to v závisloti na použité modulaci. 802.11g je zpetně kompatibilní s 802.11 díky implementaci ochraných mechanismů jako RTS/CTS. Nevýhodou je však nižší propustnost sítě v případě, že jsou připojeny stanice, které podporují pouze 802.11b [8]. Dnes je tento standard označován jako Wi-Fi 3 [6].

1.6.4 IEEE 802.11n

Spoustu změn přinesl standard 802.11n uveden v roce 2009. Tento standard pracuje ve frekvenčních pásmech 2,4 GHz i 5 GHz. I nadále je použita technika OFDM, která byla vylepšena pro dosažení vyšších rychlostí. Pro vyšší rychlosti se zde využívá technologie MIMO (viz. kapitola 1.3.8) a volitelná šířka kanálu 20 nebo 40 MHz. V PPDU 802.11n se nově objevuje pojem HT („High Throughput“), který popisuje standard fyzické vrstvy. Přenosová rychlost může být od 72 Mbit/s (šířka kanálu 20 MHz a 1x1 MIMO) až po 600 Mbit/s (šírka kanálu 40 MHz a 4x4 MIMO). Je zpětně kompatibilní se staršími standardy 802.11a/b/g [24, 11]. Tento standard je dnes označován jako Wi-Fi 4 [6].

1.6.5 IEEE 802.11ac

Standard schválený v roce 2014 jenž využívá pouze 5 GHz frekvenční pásmo. Fyzická vrstva využívá techniku OFDM a přidává podporu MU-MIMO, Beamforming a modulace 256-QAM. PPDU obsahuje pole VHT („Very High Throughput“), jenž značí velmi vysokou propustnost. Šířka kanálu se zvýšila na 80 nebo 160 MHz. Díky

tomu lze dosahovat teoretických přenosových rychlostí až 6933 Mbit/s, při použití přístupového bodu s 8 anténami a stanice se 4 anténami s šířkou kanálu 160 MHz. Při použití 1x1 MIMO a 80MHz kanálu je přenosová rychlost 433 Mbit/s [10]. Standard je označován jako Wi-Fi 5 [6].

1.6.6 IEEE 802.11ax

Posledním vydaným standardem je 802.11ax. Standard využívá frekvenční pásma 2,4 GHz a 5 GHz. Přenosové rychlosti jsou až čtyrnásobné oproti 802.11ac. Toho je docíleno vylepšenou modulací 1024-QAM, podporou up-link směru pro MIMO a MU-MIMO. Na fyzické vrstvě je použita vylepšená technika OFDM, a to OFDMA (Orthogonal Frequency-Fivision Multiple Access). V PPDU se značí jako HE („High-Efficiency“). V tomto standardu je také implementováno nové zabezpečení pomocí WPA3 [25]. Standard 802.11ax je označován jako Wi-Fi 6 a zatím není příliš rozšířen [6].

2 Open-source firmware pro routery

Firmware je specifický typ software, který provádí řízení hardwaru nějakého vestavěného systému [26]. Většina výrobců routerů dodává na trh zařízení s uzavřeným proprietárním řešením firmware. Výhodou těchto proprietárních firmwarů je většinou jednoduché uživatelské ovládání, disponující grafickým rozhraním a přednastavenými základními funkcemi. Firmware takového typu se hodí například pro použití v domácích nebo menších firemních sítích, kde uživatel nemusí disponovat znalostmi pro nastavení počítačových sítí a není potřeba pokročilé funkce jako VLAN („Virutal Local Area Network“), VPN („Virtual Private Network“), vlastní firewall, monitoring provozu a jiné. Nevýhodou takových uzavřených firmware je pravě nemožnost rozšířit je o nové funkce nebo detailně měnit nastavení sítě a routeru.

Alternativou k proprietárním firmware jsou firmwary typu open-source. Jako open-source (česky označováno jako „otevřený software“) se označuje software, který splňuje kritéria dle definice Open Source Initiative [27]. K takovému softwaru je tedy k dispozici zdrojový kód, který lze upravovat (například přidávat nové funkce) a volně jej šířit. Většinou je také možné využít open-source firmware na různém hardwaru či platformách.

V následujících podkapitolách jsou uvedeny nejrozšířenější open-source firmwary pro routery. Následně jsou tyto firmwary porovnány a je vybrán ten, který se nejvíce hodí pro realizaci praktické části této bakalářské práce.

Pro porovnání firmware byl použit vybraný router, jehož výběrem se zabývá následující kapitola 3, a jednodeskový počítač Raspberry Pi 3 Model B+. Bohužel bylo zjištěno, že jak na vybraném routeru, tak na Raspberry Pi 3 Model B+ není možné nainstalovat a provozovat všechny tyto open-source firmwary určené pro routery, a to z důvodu jejich nekompatibility, respektive nepodpory ze strany firmwaru. Jediný oficiálně podporovaný firmware je OpenWrt. Při pokusu o nahrání nepodporovaného firmware by mohlo dojít k zablokování, neboli "bricknutí"(anglicky "bricking"), což je stav při kterém je zařízení nepoužitelné a ve většině případů i neopravitelné. Důvodem je nahrání špatného firmware, nebo výpadek napájení během aktualizace či nahrávání nového firmware. Proto budou vybrané open-source firmwary popsány a porovnány alespoň teoreticky dle získaných informací.

2.1 DD-WRT

DD-WRT je alternativní open-source firmware pro WLAN routery založený na Linuxovém jádru s licencí GNU GPL („GNU General Public License“) [28]. Vydán byl v roce 2005 a je nadále vyvíjen. [29]. V současnosti je dle databáze routerů na

stránkách DD-WRT [30] podporováno přes více než 800 zařízení. Pro každý router je firmware modifikován v závislosti na použitém hardwaru routeru. DD-WRT disponuje velkou komunitou uživatelů a vývojářů, kteří se zapojují do vývoje. Díky tomu je dostupná uživatelská podpora na oficiálním DD-WRT fóru a Wiki, kde jsou k nalezení návody. Poslední stabilní verzí je verze v24 SP1 z roku 2008. Tato verze je však nadále upravována a používána do dnes.

2.1.1 Prostředí DD-WRT

DD-WRT lze ovládat ze dvou rozhraní. Skrze webové rozhraní, které je dostupné na IP adrese routeru přes, nebo pomocí příkazové řádky dostupné po nastavení SSH („Secure Shell“) či Telnet. Ve webovém rozhraní jsou jednotlivé části nastavení a funkce přehledně rozděleny do sekcí, které dále obsahují podsekce. V každé podsekci lze upravovat konkrétní nastavení routeru a sítě. Konkrétní popis webového rozhraní DD-WRT je k nalezení na Wiki stránkách DD-WRT [31].

Příkazový řádek v DD-WRT je dostupný po konfiguraci SSH skrze webové rozhraní. Syntaxe a příkazy jsou tedy stejné jako na jiných Linuxových distribucích [32].

2.1.2 Konfigurovatelnost DD-WRT

Konfigurovatelnost DD-WRT je na velmi vysoké úrovni a lze tedy detailně konfigurovat veškeré parametry routeru a sítě. Samozřejmostí jsou základní nastavení jako vytvoření Wi-Fi sítí, DHCP („Dynamic Host Configuration Protocol“) a DNS („Domain Name Server“) serveru, podrobné nastavení Firewallu nebo přesměrování a blokování portů. DD-WRT je dostupné v několika verzích (např. Mini, Mikro, Mega), které se liší velikostí a dostupnými funkcemi. Pokročilejší konfigurace se odvíjí zejména od použitých rozšíření, která jsou pro tento firmware dostupná. S nimi lze router s DD-WRT použít například jako webový server, FTP („File Transport Protocol“) server, NAS („Network Attached Storage“) nebo používat VPN, VLAN, QoS a spoustu dalších. Přehled možností využiti DD-WRT naskytuje i webová stránka s tutoriály [33].

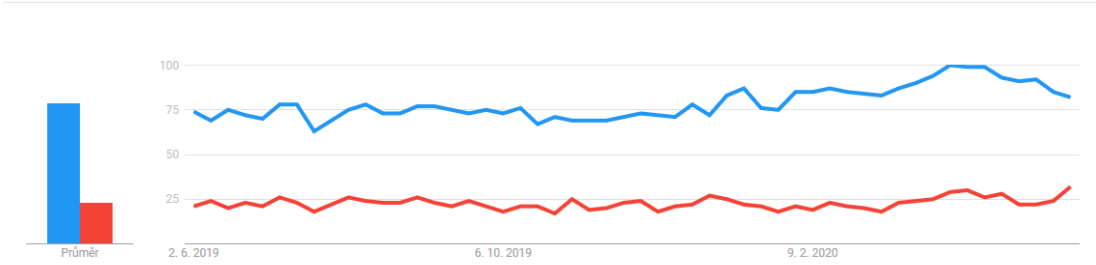
2.1.3 Rozšířitelnost DD-WRT

Vzhledem k linuxovému základu, lze DD-WRT rozšiřovat pomocí balíčků (anglicky packages). Každý jeden balíček většinou reprezentuje jeden software, který se po spuštění příslušného příkazu doinstaluje. Pro firmware DD-WRT je celkem k dispozici více než 3000 balíčků [33].

Pro účely praktické části bakalářské práce, tedy monitorování provozu a zachytávání rámců, je důležité aby firmware podporoval potřebné nástroje. Pro DD-WRT jsou k tomuto účelu k dispozici Tcpdump, Kismet, Aircrack-ng a Horst.

2.2 OpenWrt

OpenWRT je open-source GNU/Linux firmware pro routery a vestavěné systémy. Vyvíjen je od roku 2004 a zatím poslední stabilní verzí je verze 19.07.3 z května 2020 [34]. Je tedy novější a častěji aktualizovaný oproti jiným firmware. Firmware nabízí podporu více než 1500 zařízení, to je téměř dvakrát více než nabízí DD-WRT. OpenWrt je vyvíjen velkou komunitou, díky které nabízí podporu na komunitním fóru. Dále na oficiálních stránkách OpenWrt je k dispozici velké množství návodů. OpenWrt je oproti podobnému DD-WRT rozšířenější a častěji využívaný, svědčí o tom například statistika vyhledávání pomocí vyhledávače Google. Statistika je znázorněna v grafu 2.1.



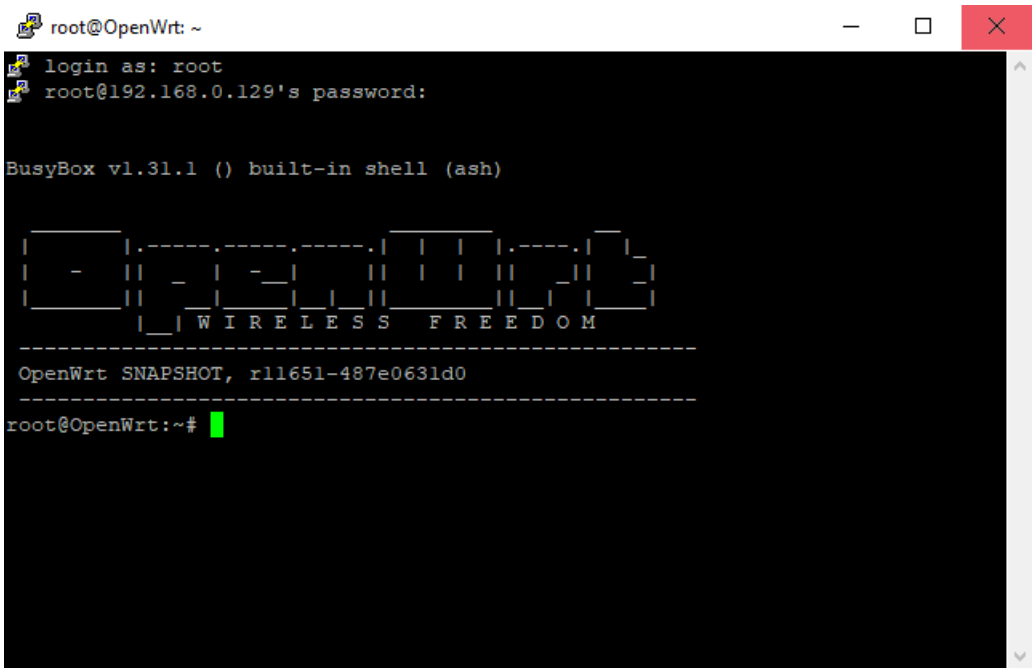
Obr. 2.1: Statistika vyhledávání termínu "openwrt" (modrá) a "ddwrt" (červená) vyhledávačem Google [35].

2.2.1 Prostředí OpenWrt

OpenWrt nabízí ve výchozím stavu prostředí příkazové řádky, tak jak je na obrázku 2.2. Pomocí balíčků však lze doinstalovat jednoduché webové rozhraní. Příkazový řádek je linuxový ash shell, stejně jako v případě DD-WRT. Syntax a příkazy jsou tedy stejné jako na jiných linuxových distribucích.

2.2.2 Konfigurovatelnost OpenWrt

I v případě OpenWrt platí že možnosti konfigurace jsou obrovské. Možnosti konfigurace se odvíjí i od výkonu použitého hardwaru. Na OpenWrt lze provozovat webový server, emailový server, DLNA server, VPN a spoustu dalších služeb. Pro představu,



Obr. 2.2: Prostředí příkazové řádky OpenWrt.

co vše konfigurace OpenWrt nabízí, lze nahlédnou do online uživatelské příručky [36]. Většinu nastavení je třeba provádět skrze příkazový řádek, to z OpenWrt dělá pro běžného uživatele složitý nástroj.

2.2.3 Rozšířitelnost OpenWrt

Rozšiřovat OpenWrt o další software a funkce lze pomocí balíčků z repozitářů. K dispozici je něco přes 10000 balíčků.

Pro monitorování a zachytávání provozu Wi-Fi sítě je k dispozici hned několik nástrojů. Z těch nejznámějších je to Tcpdump, T-shark, Kismet, Horst a Aircrack-ng.

2.3 Tomato

Tomato je open-source firmware založen na linuxovém jádru. Tomato, jenž byl uveden roku 2006, navazuje na starší firmware HyperWRT. Určen je pro routery používající chipy od společnosti Broadcom. Počet podporovaných zařízení je tedy značně omezen. [37, 38]. Tomato firmware není tolik rozšířen a proto nedisponuje velkou komunitou uživatelů jako třeba DD-WRT nebo OpenWrt. Poslední verze 1.28 původního Tomato firmware pochází z roku 2010. Existují však modifikované firmwary

Tomato, tzv. módy, které na původní firmware navazují. Nejnovější mód Tomato firmware je FreshTomato, jehož poslední verze pochází z roku 2019 [39].

2.3.1 Prostředí Tomato

Tomato klade důraz na stabilitu, rychlost a efektivitu. Disponuje přehledným a jednoduchým webovým rozhraním rozděleným na panel s menu nabídkou a oknem zobrazující obsah vybrané položky z menu. Kromě webového rozhraní je k dispozici i příkazový řádek a to po připojení přes SSH. [39]

2.3.2 Konfigurovatelnost Tomato

Tomato firmware není tolik konfigurovatelný jako OpenWrt a DD-WRT i přesto nabízí mnohem více možností než většina proprietárních firmware. Funkce a možnosti nastavení se liší v závisloti na použitému módu firmwaru Tomato. Firmware umožňuje základní nastavení parametrů sítě, DHCP server, pokročilé QoS, zobrazení statistik provozu sítě a mnoho dalších funkcí [39].

2.3.3 Rozšířitelnost Tomato

Původní Tomato firmware nepodporuje rozšíření pomocí balíčků, avšak některé jeho módy, jako například TomatoUSB, ano.

2.4 Srovnání

Vzhledem k nemožnosti praktického porovnání výše jmenovaných open-source router firmwarů, bylo provedeno alespoň porovnání na základě zjištěných dostupných informací.

Z pohledu počtu podporovaných zařízení je na tom nejlépe OpenWrt, které podporuje více než 1500 zařízení, což je téměř dvojnásobek oproti počtu zařízení podporovaných DD-WRT. Nejhůře je na tom Tomato, a to z důvodu podpory pouze některých zařízení postavených na chipsetech značky Broadcom.

Ve srovnání konfigurovatelnosti a rozšíření, tedy možnosti nastavení a podpory funkcí, jsou na tom nejlépe OpenWrt a DD-WRT. Tyto firmwary jsou si velmi podobné a jsou i založeny na stejném linuxovém základu. Oba firmwary nabízí velmi detailní možnosti konfigurace hardwaru i instalovaného softwaru. Jako lepší se opět jeví OpenWrt a to zejména díky tomu, že nabízí více rozšiřujících balíčku. Tomato v tomto ohledu zaostává, jelikož jeho účelem je být jednoduchý a efektivní. U Tomato velmi záleží na použitém módu, protože ne všechny módy podporují rozšířitelnost pomocí balíčků. Přehled srovnání jednotlivých firmware je v tabulce 2.1.

Tab. 2.1: Přehled a srovnání open-source Firmware DD-WRT, OpenWrt a Tomato.

Firmware:	**DD-WRT**	**OpenWrt**	**Tomato**
Založeno na:	Linux	Linux	Linux
Podporovaná zařízení:	800+	1500+	30+
Ovládací rozhraní:	Web-GUI CLI	Web-GUI CLI	Web-GUI CLI
Možnosti konfigurace:	široké	široké	omezené
Dostupná rozšíření:	3000+	10000+	závisí na módu
Aktuální verze:	v24 SP1 (2008)	19.07.3 (2020)	1.28 (2010)
Podpora:	Wiki, Fórum	Wiki, Fórum	-

3 Výběr vhodného routeru

Tato kapitola se zabývá výběrem vhodného routeru pro provoz open-source firmware s možnostmi zachytávání provozu Wi-Fi sítě. Je provedeno porovnání pěti vybraných routerů, a to na základě podpory open-source firmware, výkonnosti hardwaru, podporovaných standardů, technologií a ceny. Pro porovnání byly vybrány tyto routery:

- **Turris Omnia** - kapitola 3.1
- **Linksys WRT1900AC v2** - kapitola 3.2
- **Netgear Nighthawk X4S R7800 AC2600** - kapitola 3.3
- **Zyxel NBG6817 Armor Z2 AC2600** - kapitola 3.4
- **Linksys WRT3200ACM** - kapitola 3.5

3.1 Turris Omnia

Turris Omnia je router od sdružení CZ.NIC, který se vyznačuje výkonným hardware a open-source firmwarem založeným na OpenWrt. Hardware je založen na chipsetech Marvell a Qualcomm Atheros. Nabízí dvoujádrový procesor o taktu 1,6 GHz, 2 GB (gigabajtů) RAM („Random Access Memory“) a 8 GB velké úložiště.

O drátové připojení k sítí se stará jeden gigabitový WAN („Wide Area Network“) port spolu s pěti gigabitovými LAN („Local Area Network“) porty. Jako jediný nabízí také připojení optického kabelu díky SFP („Small form-factor pluggable“) konektoru. O bezdrátové připojení k Wi-Fi síti se stará dvojice chipů Atheros, které dohromady podporují standardy 802.11a/b/g/n/ac a technologie 2x2 a 3x3 MIMO s podporou až tří prostorových streamů. Přenosové rychlosti jsou až 1300 Mbit/s v 5GHz pásmu a až 300 Mbit/s v 2,4GHz pásmu.

Turris Omnia nabízí širokou škálu využití a možnost dalšího rozšíření pomocí dvou mini PCIe slotů a jednoho mini PCIe/mSATA („mini-Serial Advanced Technology Attachment“) slotu. Dva z těchto slotů jsou však již zabrané výchozími Wi-Fi moduly. Dále nabízí slot pro SIM kartu a USB („Universal Serial Bus“) 3.0 port. Lze tak například přidat DVB-T („Digital Video Broadcasting – Terrestrial“) tuner a sdílet v síti televizní vysílání, nebo přidat zvukovou kartu a využít router jako internetové rádio. Další možností je rozšíření o LTE („Long Term Evolution“)/UMTS („Universal Mobile Telecommunications System“)/GSM („Global System for Mobile Communications“) modul do mini PCIe. Do mSATA slotu lze připojit třeba SSD („Solid-State disk“) disk jako rychlé úložiště, ze kterého lze skrze DLNA („Digital Living Network Alliance“) streamovat filmy v síti. Turris Omnia lze přestavět i na NAS s vlastními disky. Možnosti funkčních rozšíření tohoto routeru jsou opravdu velké a bylo by zajímavé prozkoumat, pro co vše by bylo možné jej využít.

Velkou výhodou je podpora velké komunity a samotného výrobce. Z open-source firmware je podporováno OpenWrt (které je i výchozím firmware). Oba WLAN chipsety v Turris Omnia podporují monitorovací režim. Parametry routeru Turris Omnia jsou uvedeny v tabulce 3.1 [40, 41].

Tab. 3.1: Parametry routeru Turris Omnia [40].

CPU:	Marvell 88F6820 1.6 GHz dual core
RAM:	2 GB DDR3
Uložiště:	8 GB eMMC
WAN:	1x Gbit port, Eth. chip: Marvell 88F6820
LAN:	5x Gbit port. Switch: Marvell 88E6176
WLAN:	Chipset1: Atheros AR9287 – 802.11b/g/n, 2,4 GHz, 2x2:2 MIMO, až 300 Mbit/s
	Chipset2: Qualcomm Atheros QCA9880 v2 – 802.11an+ac, 5 GHz, 3x3:3 MIMO, až 1300 Mbit/s
SFP:	Ano
FW/OS:	Turris OS (upravené OpenWrt)
Podporované FW/OS	OpenWrt
Monitorovací režim:	Chipset1: ANO
	Chipset2: ANO
Cena:	7799 Kč

3.2 Linksys WRT1900AC v2

Domácí router od společnosti Linksys je kompletně založen na chipsetech od Marvell. Nabízí stejný dvoujádrový procesor jako Turris Omnia, zaostává však ve velikosti operační paměti RAM, kde nabízí 512 MB a úložiště, které má kapacitu 128 MB.

Router má jeden gigabitový WAN port a čtyři gigabitové LAN porty. Bezdrátové připojení zajišťují dva chipy Marvell, jeden pro 5GHz standardy an+ac a druhý pro 2,4GHz b/g/n, oba s podporou technologie 4x4:3 MIMO. Kombinovaná teoretická přenosová rychlost činí až 1900 Mbit/s, z toho 1300 Mbit/s připadá pro 5GHz pásmo a 600 Mbit/ss pro pásmo 2,4 GHz.

Výrobce oficiálně uvádí podporu open source firmware jako OpenWrt a DD-WRT, ovšem bez oficiální podpory z jeho strany. Nenabízí možnosti rozšíření skrze PCIe/mSATA sloty, ale disponuje alespoň USB 3.0 a eSATA portem [42, 43]. Použitý chipset Marvell sice podporuje monitorovací režim, ale funkčnost pod OpenWrt není zaručena. Někteří uživatele totiž hlásí problémy s tímto režimem. Parametry routeru jsou uvedeny v tabulce 3.2 [42].

Tab. 3.2: Parametry routeru Linksys WRT1900AC v2 [42].

CPU:	Marvell 88F6820 1.6 GHz dual core
RAM:	512 MB DDR3
Uložiště:	128 MB eMMC
WAN:	1x Gbit port, Eth. chip: Marvell 88F6820
LAN:	4x Gbit port. Switch: Marvell 88E6176
WLAN:	Chipset1: Marvell 88W8864 – 802.11an+ac, 5 GHz, 4x4:3 MIMO, až 1300 Mbit/s
	Chipset2: Marvell 88W8864 – 802.11b/g/n, 2,4 GHz, 4x4:3 MIMO, až 600 Mbit/s
SFP:	Ne
FW/OS:	Linksys Linux based
Podporované FW/OS	OpenWrt, DD-WRT
Monitor mode	Chipset1: ANO (ale funkčnost nezaručena)
	Chipset2: ANO (ale funkčnost nezaručena)
Cena:	6229 Kč

3.3 Netgear Nighthawk X4S R7800 AC2600

Domácí „herní“ router od Netgear kompletně založen na chipsetech Qualcomm Atheros. Nabízí dvoujádrový procesor o taktu 1.7 GHz, 512 MB RAM a 128 MB velké uložiště. K dispozici je zde také dvojice USB 3.0 portů.

Co se týče konektivity, k dispozici je jeden gigabitový WAN port a čtyři gigabitové LAN porty. O bezdrátové wifi připojení se stará dvojice chipů, jeden pro 5GHz pásmo a standardy an+ac a druhý pro 2,4GHz pásmo a standardy b/g/n. Rychlost přenosu ve 2,4GHz pásmu může být až 800 Mbit/s a v 5GHz až 1733 Mbit/ss. Router umí využívat až 160MHz šířky kanálu. Oba chipy podporují technologii 4x4:4 MIMO, tedy až 4 prostorové streamy.

Na routeru běží proprietární firmware od Netgear, lze na něj však nainstalovat open source firmware, konkrétně podporované jsou OpenWrt, DD-WRT a Voxel [44, 45]. Monitorovací režim by měl být pod OpenWrt podporován za použití vhodného modulu ovladače ath10k. Parametry routeru Netgear Nighthawk X4S R7800 AC2600 jsou uvedeny v tabulce 3.3

Tab. 3.3: Parametry routeru Netgear Nighthawk X4S R7800 AC2600 [44].

CPU:	Qualcomm IPQ8065 1.7 GHz dual core
RAM:	512 MB RAM DDR3
Uložiště:	128 MB eMMC
WAN:	1x Gbit port, Eth. chip: Qualcomm IPQ8065
LAN:	4x Gbit port, Switch: QCA8337
WLAN:	Chipset1: Qualcomm Atheros QCA9984 – 802.11an+ac, 5 GHz, 4x4:4 MIMO, až 1733 Mbit/s
	Chipset2: Qualcomm Atheros QCA9984 – 802.11b/g/n, 2,4 GHz, 4x4:4 MIMO, až 800 Mbit/s
SFP:	Ne
FW/OS:	Netgear Linux based
Podporované FW/OS	OpenWrt, DD-WRT, Voxel
Monitor mode	ANO (při použití vhodného modulu ovladače ath10k)
	ANO (při použití vhodného modulu ovladače ath10k)
Cena:	5399 Kč

3.4 Zyxel NBG6817 Armor Z2 AC2600

Výkonný domácí „herní“ router od firmy Zyxel je také založen na chipsetech Qualcomm Atheros jako Netgear. To znamená dvoujádrový procesor na 1.7 GHz, 512 MB RAM a 4 GB eMMC úložiště. K dispozici jsou také dva USB porty, jeden verze 2.0 a druhý verze 3.0.

Připojení zajišťuje jeden gigabitový WAN port a čtyři taktéž gigabitové LAN porty. WiFi připojení je možné v 2,4GHz a 5GHz pásmu a starají se o něj dva stejné chipy, které dohromady podporují standardy 802.11b/g/n/ac a technologii 4x4:4 MIMO. Teoretická rychlost přenosu v 2,4GHz pásmu je až 800 Mbit/s a až 1733 Mbit/s v pásmu 5GHz. I tento routeru umí využívat 160 MHz šířku kanálu [46].

Firmware routeru je proprietární řešení od Zyxel. Z open-source firmwarů je podporováno OpenWrt. Vzhledem ke stejnému hardware, jako má předchozí Netgear Nighthawk X4S R7800 AC2600, tak i zde platí že monitorovací režim by měl být podporován. Parametry routeru Zyxel NBG6817 Armor Z2 AC2600 jsou uvedeny v tabulce 3.4.

Tab. 3.4: Parametry routeru Zyxel NBG6817 Armor Z2 AC2600 [46].

CPU:	Qualcomm IPQ8065 1.7 GHz dual core
RAM:	512 MB RAM DDR3
Uložiště:	4 GB eMMC
WAN:	1x Gbit port, Eth. chip: Qualcomm IPQ8065
LAN:	4x Gbit port, Switch: QCA8337
WLAN:	Chipset1: Qualcomm Atheros QCA9984 – 802.11an+ac, 5 GHz, 4x4:4 MIMO, až 1733 Mbit/s
	Chipset2: Qualcomm Atheros QCA9984 – 802.11b/g/n, 2,4 GHz, 4x4:4 MIMO, až 800 Mbit/s
SFP:	Ne
FW/OS:	Proprietární Zyxel
Podporované FW/OS	OpenWrt
Monitor mode	ANO (při použití vhodného modulu ovladače ath10k)
	ANO (při použití vhodného modulu ovladače ath10k)
Cena:	5373 Kč

3.5 Linksys WRT3200ACM

Velmi výkonný router od Linksys z řady WRT stojí kompletně na chipsetech Marvell. Dvoujádrovému procesoru na 1,8 GHz zde sekunduje 512 MB RAM a 256 MB velké uložiště. Za zmínku také stojí existence dvou USB portů (2.0 a 3.0) a jednoho eSATA portu.

Router disponuje celkem pěti gigabitovými porty – jeden pro WAN a čtyři pro LAN. Dále disponuje třemi chipy pro Wi-Fi připojení, z nichž jeden slouží pro 5GHz pásmo a standardy an+ac, druhý pak pro 2,4 GHz a b/g/n. Třetí WLAN chip umí jak 2,4GHz pásmo, tak 5GHz, je však pomalejší a slouží primárně pro technologii DFS („Dynamic Frequency Selection“) v pásmu 5 GHz. Kromě Wi-Fi, umí tento chip i technologi Bluetooth 5.0. Přenosová rychlost v 5GHz pásmu, může u tohoto routeru teoreticky dosahovat rychlosti až 2600 Mbit/s a to díky 160MHz šířce kanálu. V nižším 2,4GHz pásmu je přenosová rychlost až 600 Mbit/s. To z něj dělá nejrychlejší router v tomto srovnání [47, 48].

Výchozí firmware routeru je proprietární řešení od Linksys pravděpodobně založeno na Linuxovém jádru. Výrobce však přímo uvádí podporu a funkčnost Open Wrt a DD-WRT, avšak bez jeho oficiální podpory. Monitorovací režim by měl být podporován na dvou WLAN chipech Marvel 88W8964. Parametry routeru Linksys WRT3200ACM jsou uvedeny v tabulce 3.5.

Tab. 3.5: Parametry routeru Linksys WRT3200ACM [47].

CPU:	Marvell 88F6820 1.8 GHz dual core
RAM:	512 MB DDR3
Uložiště:	256 MB eMMC
WAN:	1x Gbit port, Eth. chip: Marvell 88F6820
LAN:	4x Gbit port. Switch: Marvell 88E6352
WLAN:	Chipset1: Marvell 88W8964 – 802.11an+ac, 5 GHz, 4x4:3 MIMO, až 2600 Mbit/s
	Chipset2: Marvell 88W8964 – 802.11b/g/n, 2,4 GHz, 4x4:3 MIMO, až 600 Mbit/s
	Chipset3: Marvell 88W8887 – 802.11ac 5 GHz, Bluetooth 5.0, až 433 Mbit/s
SFP:	Ne
FW/OS:	Linksys Linux based
Podporované FW/OS	OpenWrt, DD-WRT
Monitor mode	Chipset1: ANO
	Chipset2: ANO
Cena:	6760 Kč

3.6 Srovnání routerů

Výkon:

Z pohledu výkonnosti hardwaru jsou všechny srovnávané routery dostatečně výkonné pro provoz open-source firmware a plnění účelu zachytávání provozu v síti, a to díky doujádrovým procesorům a moderním DDR3 pamětem. Pokud by měl být vybrán ten nejvýkonnější, byl by to router Turris Omnia a Linksys WRT3200ACM. Ostatní routery jsou z pohledu výkonu těsně za nimi. Výkon však není pro toto porovnání tím nejdůležitějším parametrem.

Konektivita:

Všechny srovnávané routery nabízí obdobný počet gigabitových ethernetových portů. Turris Omnia navíc umožňuje připojení optického kabelu. Bezdrátové Wi-Fi připojení umožňují všechny routery v pásmech 2,4 a 5 GHz s podporou většiny 802.11 standardů, konkrétně tedy 802.11a/b/g/n/ac. Rozdíly jsou patrné v maximálních teoretických přenosových rychlostech. Zde se jako nejlepší jeví Linksys

WRT3200ACM, který nabízí až 3200 Mbit/s kombinované přenosové rychlosti (2600 Mbit/s + 600 Mbit/s). Hned za ním jsou rotery Zyxel NBG6817 Armor Z2 AC2600 a Netgear Nighthawk X4S R7800 AC2600 s rychlostí až 2600 Mbit/s (1733 Mbit/s + 800 Mbit/s) kombinovaně. Naopak jako přenosově nejpomalejší vychází Turris Omnia, a to s kombinovanou přenosovou rychlostí 1600 Mbit/s (1300 Mbit/s + 300 Mbit/s). Turris Omnia také jako jediný nenabízí 4x4 MIMO ale jen 3x3:3 a 2x2:2.

Podpora open-source firmware:
Na všech pěti routerech lze provozovat nějaký open-source firmware. Nejčastěji je tímto firmwarem OpenWrt, který nabízí největší počet podporovaných routerů (viz. kapitola 2.4). Výchozí firmware routeru Turris Omnia TurrisOS je na OpenWrt přímo založen, na rozdíl od všech ostatních routerů, které používají vlastní proprietární firmware. To je velká výhoda Turrisu Omnia, protože odpadá instalace jiného firmwaru a navíc je jisté, že vše bude fungovat. Linksys u obou routerů sice oficiálně uvádí podporou open-source firmware OpenWrt a DD-WRT, ovšem bez technické podpory výrobce. Netgear a Zyxel oficiálně open-source firmware nepodporují, ale i přesto na nich lze zprovoznit OpenWrt, DD-WRT nebo Voxel, jsou totiž na seznamu podporovaných zařízení těchto firmwarů. V tomto ohledu se jako nejlepší jeví Turris Omnia, jelikož jeho firmware je jen modifikovaná verze OpenWrt, je tedy zaručena podpora tohoto firmware.

Podpora monitorovacího režimu:

Monitorovací režim je popsán v kapitole 4.2.2 této práce. Jde o zásadní funkcionalitu pro zachytávání rámců ve Wi-Fi síti a tedy i pro účel této bakalářské práce. Zjistit, zda daný router, respektive jeho WLAN chipset, podporuje tento režim je poněkud problematické. Velmi také záleží na použitém ovladači pro daný WLAN chipset. Chipset totiž může monitorovací režim zvládat, ale pokud není implementován v ovladači, tak jej nelze zapnout. Informace na internetu a datasheety výrobců jsou na tyto konkrétní údaje skoupé. Z informací, které se podařilo získat to vypadá, že všechny srovnávané routery by tento režim měly podporovat. Otázkou ovšem zůstává funkčnost tohoto režimu s ovladači dostupnými pro dané chipsety pod OpenWrt. Dle oficiální podpory a také komunity Turrisu Omnia, by tento router měl monitorovací režim podporovat a je možno jej pro daný účel využít, je však potřeba použít vhodný ovladač pro WLAN chipset ath10k a ath9k. To se nakonec po pořízení a zprovoznění Turrisu potvrdilo a monitorovacím režim dokonce fungoval bez nutnosti aktualizace ovladače. U ostatních routerů v tomto srovnání, založených na chipsetech Qualcomm Atheros (Zyxel a Netgear), by tomu mělo být obdobně. Nejistá funkčnost je u routerů Linksys, zejména u Linksys WRT1900AC. Větší Linksys WRT3200ACM monitorovací režim podporuje, ale není informace ově-

řující jeho funkčnost.

Poměr cena/výkon:

V poměru cena/výkon je na posledním místě Linksys WRT1900AC, který toho za cenu blížící se vyššímu modelu WRT3200ACM příliš nenabízí. Velmi dobrým poměrem ceny k výkonu však nabízí Zyxel NBG6817 Armor Z2 AC2600 a Netgear Nighthawk X4S R7800 AC2600, které jsou postaveny na totožném hardware, Zyxel však vyniká 4 GB uložištěm a nižší cenou. Nejdražšími routery tohoto výběru jsou Turris Omnia a Linksys WRT3200ACM. U Turris Omnia, ač není nejvýkonnějším zařízením, je vysoká cena dána jeho unikátností a šířkou možnosti jeho využití a rozšíření, ale i velká podpora výrobce. Linksys WRT3200ACM svoji vyšší cenu může ospravedlnit nejvyššími přenosovými rychlostmi ze všech pěti routerů a třemi WLAN chipy.

Závěr výběru:

Nejdůležitějšími kritérii pro výběr routeru byly v tomto případě podpora open-source firmware a možnost zachytávání v monitorovacího režimu, nikoliv maxilmální výkon. Vzhledem k tomu byl vybrán router Turris Omnia. Důvody jsou zejména jistá podpora open-source firmware (konkrétně OpenWrt), podpora monitorovacího režimu (viz. [49] a emailová komunikace s podporou Turris). Výkonnostně je Turris Omnia naprosto dostačující a podporuje moderní standardy 802.11 (kromě 802.11ax) včetně MIMO a 3 prostorových streamů. Navíc nabízí i možnosti hardwarového rozšíření.

4 Zachytávání provozu Wi-Fi sítí

4.1 Motivace

Důvodů pro zachytávání a analýzu provozu ve Wi-Fi sítích je spousta. Dochází-li ve Wi-Fi síti ke krátkodobým nebo i častým problémům, jako je rušení signálu, časté kolize, snížení rychlosti přenosu a nebo špatné směrování, může analýza zachyceného provozu pomoci při detekci příčiny těchto problémů. Může tedy posloužit při diagnostice poruch v síti.

Další využití se nabízí pro potřebu stanovení statistik o probíhajícím provozu, například ve firemní Wi-Fi síti. Předmětem analýz a následných statistik může být objem přenášených dat, nebo třeba jaký typ dat, jak často a kam je přenášen.

Samotnou kapitolou jsou důvody bezpečnostní. Na rozdíl od pevného kabelového připojení, lze rádiové sítě jako je Wi-Fi odposlouchávat. To znamená určité bezpečnostní riziko, jakým může být právě odposlouchávání a zachytávání komunikace ve Wi-Fi síti. Na základě zachyceného provozu a za použití správných nástrojů může útočník prolomit zabezpečení a dešifrovat tak přenášená data, nebo v případě nezabezpečených Wi-Fi sítí provést útoky typu Man in the Middle, MAC spoofing a jiné.

Naopak z druhého pohledu je možné zachytávání a následnou analýzu Wi-Fi provozu využít k detekci útočníka, který se pokouší o odposlouchávání bezdrátového kanálu.

4.2 Režimy bezdrátových karet

Bezdrátové síťové karty, respektive jejich chipset, pro standard 802.11 tedy pro Wi-Fi sítě, mohou operovat v několika různých režimech. Tyto režimy určují, jak se bude bezdrátové síťové rozhraní chovat. Mezi tyto režimy patří: Master nebo AP (rozhraní se chová jako přístupový bod), Managed (klientský mód, někdy také jako station), Ad-hoc nebo P2P (použití v sítích typu Ad-hoc), Mesh (použití v sítích typu Mesh), Repeater (opakovač signálu), Monitor (monitorovací režim) a Promiscuous (promiskuitní režim). Právě poslední dva jmenované režimy se využívají pro zachytávání rámců. Ne všechny jmenované režimy jsou bezdrátovými chipsety standardně podporovány, vždy záleží na hardwaru a zejména na použitém ovladači [50].

4.2.1 Promiskuitní režim

V tomto režimu může bezdrátová síťová karta zachytávat veškeré rámce, které pochází ze sítě, ke které je asociována. Je tedy nutné, znát heslo pro přihlášení do sítě.

Wi-Fi adaptér tedy pracuje na stejné frekvenci a šířce kanálu jako síť poskytovaná AP (Access Point). V promiskuitním režimu síťová karta přijímá i rámce, jenž nebyly adresovány této síťové kartě. Rámce, které obsahují SSID jiné sítě nejsou zpracovávány. Pro použití promiskuitního režimu musí chipset bezdrátové síťové karty tento režim podporovat a stejně tak i použitý ovladač [50].

Bezdrátová síťová karta v promiskuitním režimu [51]:

- Je připojena k AP.
- Filtruje rámce na základě SSID.
- Nefiltruje na základě MAC adres.
- Mění 802.11 rámce na formát rámců pro Ethernet (IEEE 802.2).
- Nenastavuje se ručně frekvence kanálu.
- V zachycených rámcích jsou vidět vyšší i vrstvy TCP/IP (nejsou šifrované).

4.2.2 Monitorovací režim

Monitorovací režim, označovaný jako RFMON („Radio Frequency Monitor“) je režim, ve kterém je bezdrátová síťová karta schopna zachytávat veškeré rámce na předem specifikovaném rádiovém kanálu, bez nutnosti asociace k síti (nefiltruje se dle SSID). Jsou zachytávány celé rámce ovšem v zašifrované formě. Je tedy vidět pouze informace o fyzické vrstvě. Síťová karta v tomto režimu nekontroluje kontrolní součet rámce (CRC) a lze tedy zachytit i poškozené rámce. V tomto režimu zpravidla nemůže bezdrátová síťová karta sama vysílat. Existují ovšem ovladače, které i vysílání (tzv. „frame injection“) v monitorovacím režimu umožňují. Této možnosti se často využívá při různých typech síťových útoků. Monitorovací režim není standardně podporován většinou Wi-Fi chipsetů a jejich ovladači. V některých případech je potřeba využít neoficiální ovladač vyvíjen komunitně, případně si zkompilovat vlastní. Jak jsem sám mohl zjistit, tak není vůbec snadné nalézt informaci o tom, zda konkrétní chipset tento režim podporuje. V datasheetech tato informace chybí a dohledat lze jen na různých neoficiálních webech a fórech, a to ne vždy [50].

Bezdrátová síťová karta v monitorovacím režimu [51]:

- Není připojena k AP a zpravidla nevysílá žádné rámce .
- Nefiltruje rámce na základě SSID.
- Nefiltruje na základě MAC adres.
- Je nutné ručně nastavit frekvenci a šířku kanálu, na kterém se má naslouchat.
- Zachytávají se celé rámce, ale obsah je šifrován.
- Lze vyčíst informace o 802.11 fyzické vrstvě.

4.2.3 Nastavení režimu Wi-Fi karty

Zkontrolovat zda používaná Wi-Fi karta, potažmo její čipset, podporuje monitorovací režim (nebo jakýkoliv jiný) je na Linuxových distribucích možné v terminálu pomocí nástroje iw [52]. Následující příkaz a jeho zkrácený výpis ukazuje, jak zjistit dostupné režimy Wi-Fi karty.

Výpis 4.1: Zkrácený výpis dostupných režimů bezdrátové síťové karty

```
root@turris:~# iw list
Wiphy phy0
        ...
        Supported interface modes:
                 * IBSS
                 * managed
                 * AP
                 * P2P-client
                 * P2P-GO
                 * P2P-device
        ...
```

Na operačním systému Windows, lze podporované režimy zjistit zadáním příkazu „`netsh wlan show wirelesscapabilities`“ do příkazového řádku.

Změnit režim Wi-Fi karty na Linuxových distribucích je snadné. Nejdříve je potřeba dané síťové rozhraní vypnout. Následně lze provést nastavení režimu a rozhraní opět zapnout. Celý proces lze provést pomocí příkazů, jenž jsou uvedeny ve výpisu 4.2, a které uvedou Wi-Fi adaptér do monitorovacího režimu. Poslední příkaz slouží k ověření změny režimu. Kromě toho lze zjistit další důležité informace o parametrech síťového rozhraní, zvláště důležitý je údaj o kanálu a frekvencích.

Výpis 4.2: Sled příkazů pro změnu režimu a výpis pro kontrolu režimu.

```
root@turris:~ $ ip link set wlan1 down
root@turris:~ $ iw dev wlan1 set type monitor
root@turris:~ $ ip link set wlan1 up
root@turris:~ $ iw dev wlan1 info
Interface wlan1
        ifindex 4
        wdev 0x100000001
        addr 00:1d:0f:b0:e7:c9
        type monitor
        wiphy 1
        channel 6 (2437 MHz), width: 20 MHz (no HT),
           center1: 2437 MHz
        txpower 20.00 dBm
```

Alternativně lze použít další nástroje a příkazy pro manipulaci se síťovým rozhraním jako `iwconfig` [53] a `ifconfig` [54].

5 Nástroje pro zachytávání provozu Wi-Fi

Nástroj, který umožňuje odposlouchávat, zachytávat a analyzovat pakety a rámce v sítí, bývá označován jako „sniffer“ nebo paketový analyzátor. V této kapitole jsou rozebrány některé open-source nástroje („sniffery“), určené pro zachytávání paketů a rámců ve Wi-Fi sítích. Nástroje byly vybrány takové, které jsou dostupné pro open-source firmware OpenWrt, jenž byl vybrán jako nejvhodnější. Všechny nástroje pro svoje fungování využívají knihovnu `Libpcap` [55], bez které by zachytávání rámců nebylo možné. K vyzkoušení zde zmíněných nástrojů byl použit jednodeskový počítač Raspberry Pi Model 3B+ a Turris Omnia.

5.1 Tcpdump

Open-source paketový analyzátor a „sniffer“, jenž je dostupný pro většinu Linuxových distribucí. Jeho původní verze vznikla již roku 1988 a s menšími změnami je vyvíjen do dnes, kdy poslední verze 4.9.3 je ze září 2019. Tcpdump nedisponuje grafickým rozhraním, běží pouze v příkazové řádce, díky tomu je však rychlý a nenáročný na hardware [56].

Ovládání Tcpdump je jednoduché. Probíhá skrze příkazovou řádku zadáním příkazu `tcpdump`, za nějž se dopisují jednotlivé parametry, které nastavují funkce tohoto nástroje. Parametrů, které lze zadat a tím měnit parametry zachytávání je spousta a jmenovat každý z nich by bylo nad rámec této práce, uvedeny proto budou jen některé z nich. Kompletní seznam dodatečných parametrů pro nástroj Tcpdump je k dispozici na manuálové stránce [57].

Pomocí Tcpdump lze zachytávat pakety a rámce z dostupných síťových rozhraní, tedy například ze síťové karty pro ethernet, bezdrátové Wi-Fi karty a dokonce i Bluetooth. Umožňuje zachytávání v promiskutiním i monitorovacím režimu. Pro výpis dostupných síťových rozhraní slouží následující příkaz:

Výpis 5.1: Výpis dostupných síťových rozhraní pro Tcpdump.

```
root@OpenWrt:~# tcpdump -D
1.eth0 [Up, Running]
2.wlan0 [Up, Running]
3.lo [Up, Running, Loopback]
4.any (Pseudo-device that captures on all interfaces) [Up
   , Running]
```

Samotné zachytávání rámců lze spustit příkazem `tcpdump` s parametrem `-i` (input), za kterým následuje logický název síťového rozhraní, na kterém chceme zachytávat. Pomocí následujícího příkazu ve výpisu 5.2 začne Tcpdump vypisovat do konzole všechny zachycené pakety na vybraném rozhraní.

Výpis 5.2: Příkaz pro spuštění zachytávání na síťovém rozhraní wlan0.

```
root@OpenWrt:~# tcpdump -i wlan0
```

Pomocí parametrů lze zachytávaný provoz filtrovat a to na základě mnoha filtrů. Filtrovat lze na základě IP adres, portů, protokolů nebo i velikosti paketů. Například následujícím příkazem ve výpisu 5.3 budou zachytávány pakety na všech dostupných síťových rozhraních, které přicházejí nebo odcházejí ze sítě 192.168.0.0/24.

Výpis 5.3: Příklad použití filtrování v Tcpdump.

```
root@OpenWrt:~# tcpdump -i any net 192.168.0.0/24
```

Kombinacím se zde meze nekladou, takže lze nastavit zachytávání dle vlastních požadavků. Výsledky zachytávání je možné také uložit do souboru. K tomu slouží parametr `-w`, za kterým následuje cesta a název souboru, do kterého se bude ukládat. Soubor nese příponu .pcap. Příklad je uveden ve výpisu 5.4.

Výpis 5.4: Příkaz pro ukládání zachytávání z Tcpdump do souboru.

```
root@OpenWrt:~# tcpdump -i any net 192.168.0.0/24 -w /
   mnt/sda1/zachytavani.pcap
```

Takto vytvořený PCAP („Packet Capture“) soubor je možné otevřít například v programu Wireshark a následně jej analyzovat. Soubor lze ale otevřít také přes Tcpdump, a to pomocí parametrů *-r* a zadáním cesty k souboru, jak ukazuje výpis 5.5.

Výpis 5.5: Příkaz pro otevření souboru v Tcpdump.

```
root@OpenWrt:~# tcpdump -r zachytavani.pcap
```

Pokud je potřeba zachytávat pouze pakety a rámce určitého protokolu přidá se za parametry název protokolu. Například pro zachytávání pouze ICMP („Internet Control Message Protocol“) paketů lze použít tento příkaz ve výpisu 5.6.

Výpis 5.6: Příkaz pro filtrování paketů na základě protokolu.

```
root@OpenWrt:~# tcpdump -i wlan0 icmp
```

Stejným způsobem lze filtrovat pakety jakýchkoliv dalších protokolů, jako například HTTP/S, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) a jiné.

Filtrovat lze i na zákaldě IP adres a portů. U IP adres lze zadat zdrojovou (*src*) nebo cílovou (*dst*) adresu. Pro zachycení paketů určených pro určitý port, lze zadat parametr *port* a číslo portu, jak ukazuje výpis 5.7.

Výpis 5.7: Příkaz pro filtrování na základě IP adres.

```
root@OpenWrt:~# tcpdump -i wlan0 src 192.168.0.129 port
   80
```

Pomocí některých parametrů lze přizpůsobovat výpis zachycených paketů a rámců. Takové a další parametry zachycuje následující tabulka 5.1.

Tab. 5.1: Některé další vybrané parametry nástroje Tcpdump.

Parametr:	**Význam:**
-X	Zobrazí data každého paketu v hex a ASCII formátu bez záhlaví linkové vrstvy.
-XX	Zobrazí data každého paketu v hex a ASCII formátu včetně záhlaví linkové vrstvy.
-A	Zobrazí data každého paketu ASCII formátu bez záhlaví linkové vrstvy.
-tttt	Vypíše časové značky v čitelném formátu, hodiny, minuty, sekundy a datum.
-vv	Podrobnější výpis obsahu.
-nn	Nepřevádět adresy a porty na jména.
-c	Zachytí pouze přesný počet paketů, definovaný číslem za parametrem.
-s	Nastavuje velikost každého zachyceného paketu v bajtech.
-e	Zobrazuje záhlaví linkové vrstvy.
-E	Dešifruje IPSEC poskytnutým klíčem.

Ve výchozím stavu by měl Tcpdump provádět zachytávání v promiskuitnim režimu. Přepnutí Wi-Fi karty do monitorovacího režimu je tedy potřeba provést buď

před spuštěním Tcpdump, nebo příkazem ve výpisu 5.8. Pokud není monitorovací režim podporován, vrátí Tcpdump chybu. Z vlastní zkušenosti však mohu doporučit změnit režim síťového rozhraní před samotným zachytáváním pomocí programu `iw`. Některé ovladače totiž nedovolují jiným nástrojům měnit stav síťového rozhraní.

Výpis 5.8: Spuštění Tcpdump v monitorovacím režimu.

```
root@OpenWrt:~# tcpdump -i wlan0 --monitor-mode
```

5.2 Kismet

Kismet je „sniffer“ a detekční systém pro bezdrátové sítě. Pracuje s Wi-Fi i Bluetooth rozhraními a některými softwarově definovanými rádii. Je multiplatformní, takže je dostupný pro Linux, OS X i Windows. Uveden byl v roce 2001 a je vyvíjen dodnes, kdy poslední verze je 2019-09-R1 [58].

Kismet je možné na Linuxových distribucích nainstalovat buď z balíčkových repozitářů, nebo zkompilovat zdrojový kód dostupný na GitHubu [59]. Zvolena byla druhá metoda, protože instalace z repozitáře z nějakého důvodu nefungovala správně a navíc se nejednalo o nejnovější verzi Kismet. Kismet funguje pouze s bezdrátovými kartami, které podporují monitorovací režim.

Na rozdíl od Tcpdump, Kismet disponuje grafickým rozhraním a to jak v terminálu (pouze starší verze Kismet), tak nově ve webovém rozhraní. Webové rozhraní je po instalaci dostupné na localhost adrese s portem 2051 (http://localhost:2501) stroje, kde je Kismet spuštěn. V případě použití na zařízení, které nedisponuje webovým prohlížečem (např. router), je možné se do webového rozhraní dostat z jiného zařízení (např. z PC), a to na IP adrese zařízení, kde je Kismet spuštěn.

Kismet je komplexnější a ne tak přímočarý nástroj jako Tcpdump, je však náročnější na hardware a zabírá i více místa, zejména kvůli spoustě závislostem na dalších balících, které musí být instalovány.

Chceme-li Kismet spustit stačí do příkazové řádky zadat příkaz `kismet`. Tím je Kismet spuštěn a zpřístupní se webové rozhraní, nedochází však k zachytávání a monitoringu sítí. K tomu je potřeba Kismetu předat informaci, na kterém síťovém rozhraní má naslouchat, to je provedeno příkazem uvedeným ve výpisu 5.9.

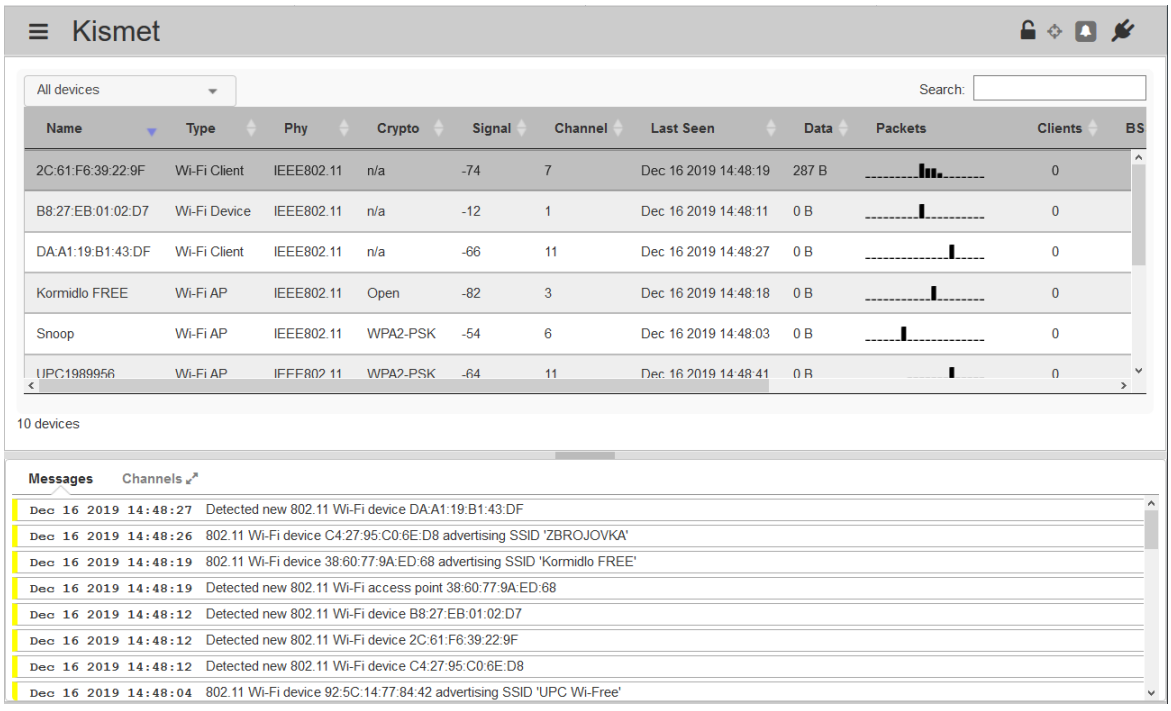
Výpis 5.9: Spuštění Kismet s volbou síťového rozhraní.

```
pi@raspberrypi:~ $ kismet -c wlan1
```

Nutno zopakovat, že bezdrátové síťové rozhraní musí podporovat monitorovací režim. Jakmile je Kismet na daném síťovém rozhraní spuštěn, je možné se podívat

do webového rozhraní na výsledky monitorovaní a zachytávání. Při prvním přístupu do webového rozhraní je potřeba vytvořit profil zadáním jména a hesla. Tyto přihlašovací údaje jsou následně potřeba při každém dalším přístupu. Na úvodní (a také hlavní) stránce webového rozhraní (obrázek 5.1) je v horní části vidět seznam všech zařízení, od kterých byly zachyceny rámce pasivním zachytáváním. Zobrazují se jak přístupové body, tak jejich klienti a stanice. Zobrazená zařízení lze filtrovat na základě jejich typu (AP, 802.11 devices, Bluetooth devices aj.). V přehledu lze vyčíst jméno a MAC adresu zařízení, nebo SSID vysílaných sítí a to na základě vysílaných Beacon a Probe rámců. Dále použité zabezpečení (u AP), úroveň signálu, velikost přenesených dat, využívaném kanálu a čas, kdy bylo zařízení naposledy zachyceno.

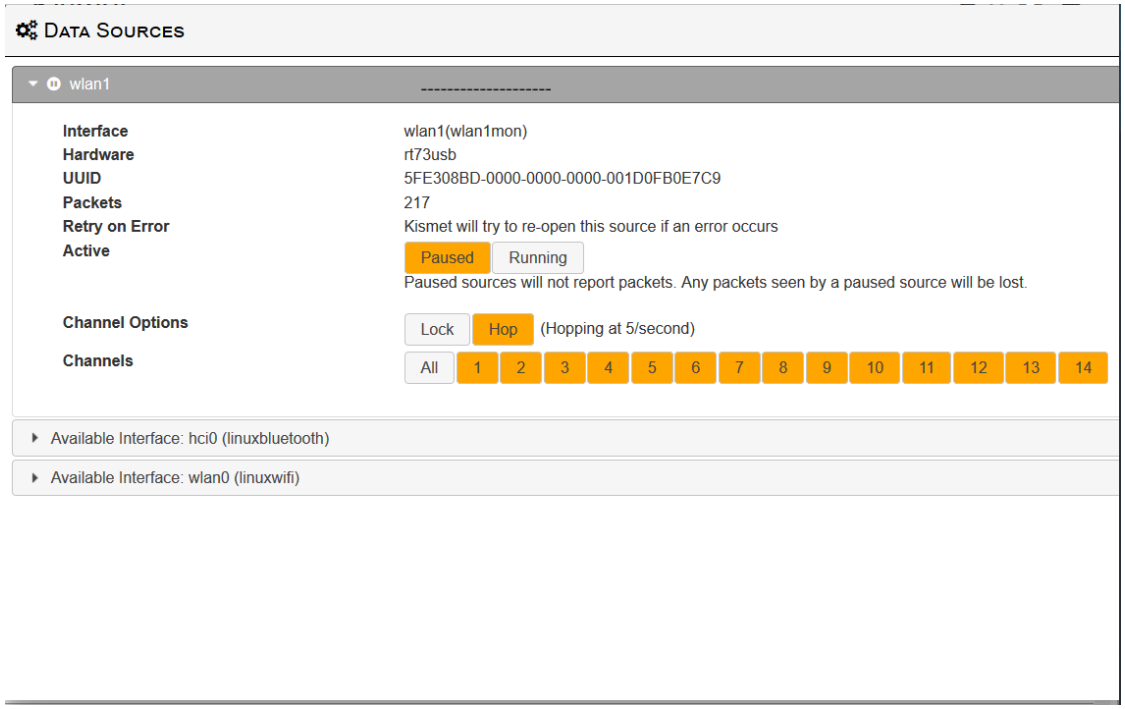
Pro zachycení co nejvíce rámců a objevení zařízení v blízkém okolí, využívá Kismet tzv. „channel hopping“ (přeskakování kanálů). Princip je takový, že Kismet mění po určitém intervalu kanál, na kterém zachytává rámce. Interval změny kanálu lze nastavit dle potřeb. Výhodou je zachycení paketů a rámců na více kanálech, nevýhodou je však to, že během přeskakování kanálu mohou uniknout některé vysílané rámce. Ve spodní části hlavní stránky jsou zobrazeny zprávy o právě detekovaných zařízeních. V kartě „Channels“ se zobrazuje graf s aktuálním počtem zařízení na jednotlivých kanálech. Tento údaje lze sledovat i zpětně.



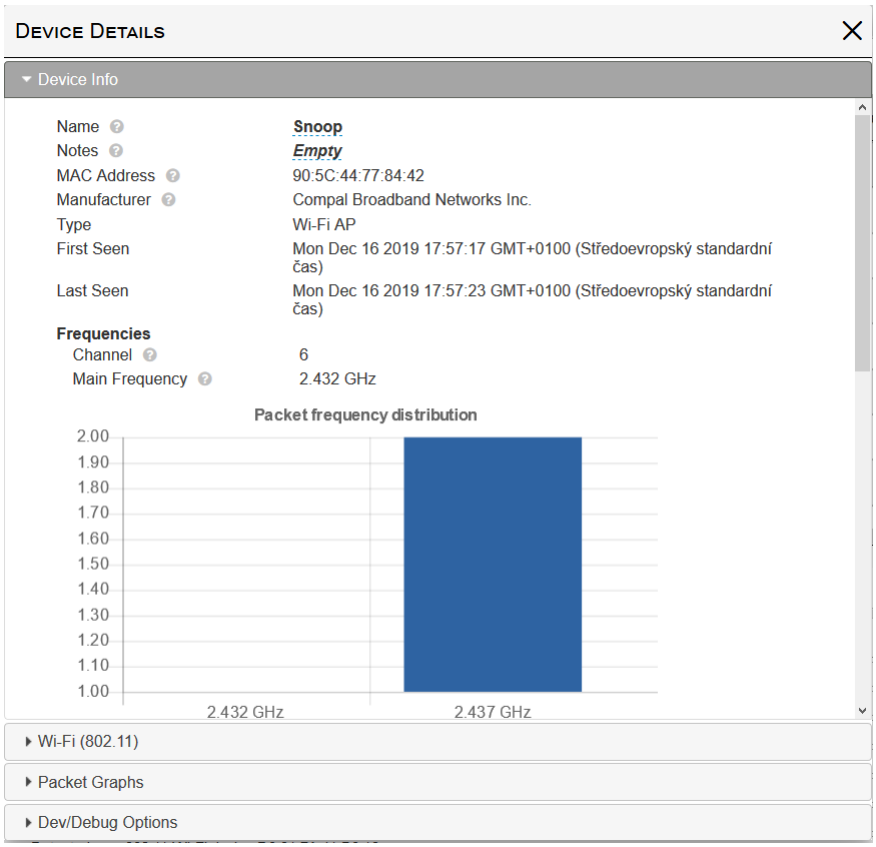
Obr. 5.1: Hlavní stránka webového rozhraní Kismet.

V právem horním rohu se nachází menu, které skrývá nastavení, kde je možné si přizpůsobit co vše se bude na hlavní stránce zobrazovat. Dále je zde nabídka „Data source“ (obrázek 5.2) pro volbu a nastavení síťového rozhraní, na kterém se mají rámce a pakety zachytávat. Právě zde lze nastavit interval změny kanálu, vybrat jen konkrétní kanál, případně zafixovat rozhraní pouze na jeden kanál. Lze i pozastavit zachytávání tlačítkem „Pause“.

Pro zobrazení informací o nalezených zařízeních, stačí na vybrané zařízení kliknout. Zobrazí se okno „Device detail“, které je rozděleno na další čtyři skrývající okna. V okně „Device info“ (obrázek 5.3) jsou informace o zařízení jako jméno, MAC adresa, výrobce, typ, frekvence a kanál, na kterém zařízení komunikuje. V okně „Wi-Fi (802.11)“ jsou informace o SSID, BSSID, počtu a typu zaslaných paketů včetně grafů, jak je vidět na obrázku 5.3. Velmi podrobné informace v textovém formátu obsahuje poslední okno „Dev/Debug option“.



Obr. 5.2: Nastavení síťového rozhraní v Kismet.



Obr. 5.3: Okno s informacemi o zařízení.

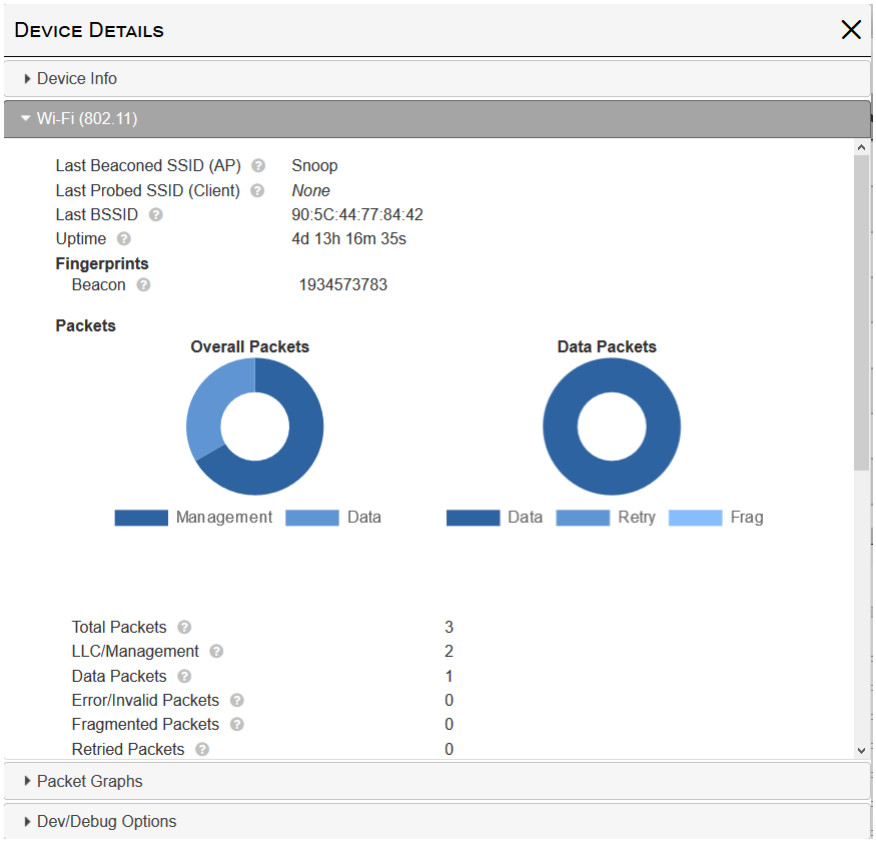
Po každém spuštění Kismet vytvoří log ve formátu *.kismet*, do kterého se ukládají zachytávané pakety. Uložené logy lze v Kismet znovu otevřít, slouží k tomu následující příkaz uveden ve výpisu 5.10.

Výpis 5.10: Příkaz otevření logu ze zachytávání v Kismet.

```
pi@raspberrypi:~ $ kismet -c /cesta_k_souboru/
   nazevsouboru.kismet
```

Pokud je potřeba otevřít log se zachycenými pakety pomocí Kismet, například ve Wireshark, je nejdříve nutné převést jej z formátu *.kismet* na *.pcap*. K tomu je zapotřebí doinstalovat Python modul, který je k dispozici zde [60]. Pakety lze filtrovat upravením filtrovacích pravidel v jednom z konfiguračních souboru pro Kismet.

Kismet lze rozšířit o další funkce pomocí různých pluginů. Celkově je Kismet komplexní nástroj s mnohými možnostmi nastavení.



Obr. 5.4: Okno s informacemi o přenesených paketech.

5.3 HORST

„Highly Optimized Radio Scanning Tool“ (HORST) je malý nenáročný program pro zachytávání a analýzu rámců. Funkčně je podobný jako Tcpdump i Kismet, je však jednodušší na ovládání a nenabízí tolik možností a nastavení [61]. HORST na rozdíl od Kismet pracuje pouze s Wi-Fi. Dostupný je skrze repozitáře pro většinu Linuxových distribucí. Horst vyžaduje Wi-Fi kartu s monitorovacím režimem.

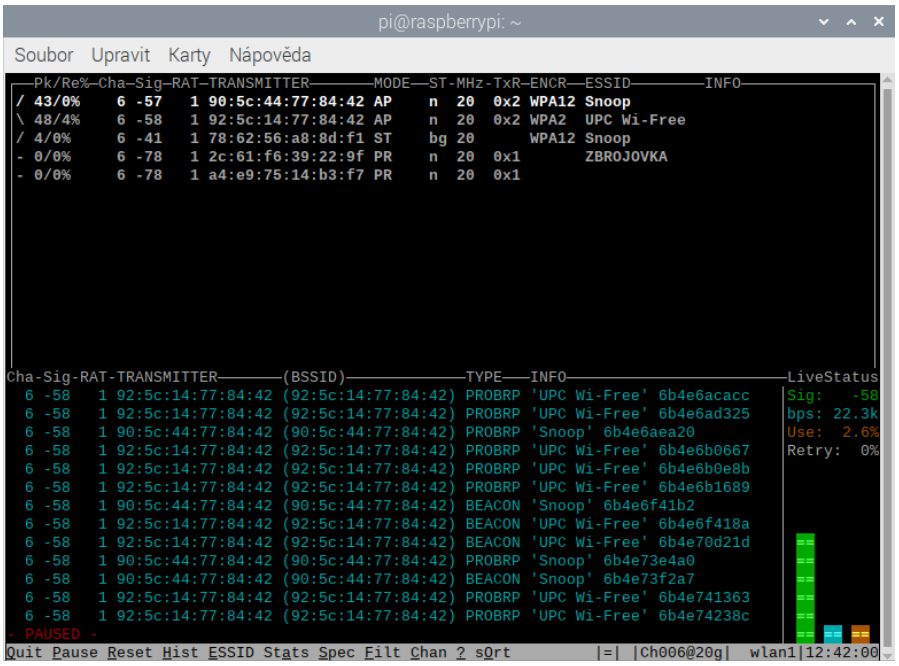
HORST pracuje v příkazové řádce s textovým grafickým prostředím. Pro spuštění je zapotřebí definovat na jakém síťovém rozhraní má naslouchat. To je provedeno následujícím příkazem:

Výpis 5.11: Spuštění HORST s volbou síťového rozhraní.

```
pi@raspberrypi:~ $ sudo horst -i wlan1
```

Hned po tom se v terminálu zobrazí textové rozhraní HORST tak, jak je vidět na obrázku 5.5. V horní části okna jsou zobrazeny informace nalezených zařízení.

Těmito informacemi jsou například počet vysílaných rámců, kanál, úroveň signálu, MAC adresa, typ zařízení, standard 802.11, šířka kanálu, použité zabezpečení a ESSID („Extended Service Set Identifier“). Ve spodní polovině okna jsou údaje o tom, jaké zařízení vysílalo a o jaký druh paketu se jedná. V pravo dole je menší statistika o přenesených paketech.



Obr. 5.5: Textové prostředí nástroje HORST.

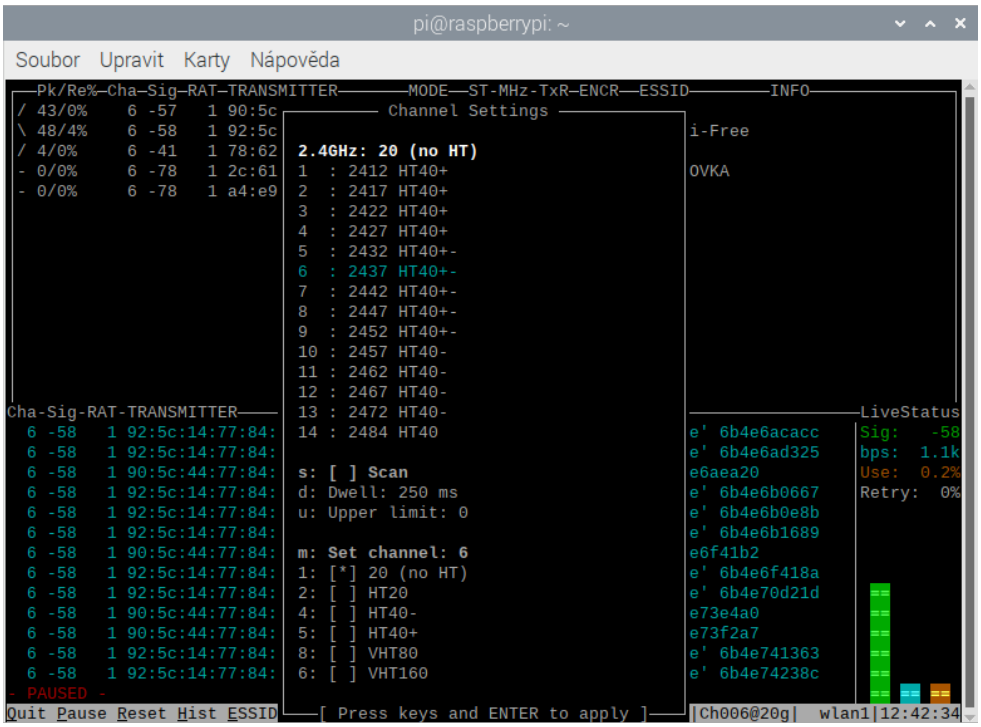
Úplně dole je umístěn panel s menu nabídkou, pomocí které lze HORST ovládat. Kromě ukončení, zastavení a resetování zachytávání, je zde nabídka pro volbu kanálu (obrázek 5.6), na který má být síťová karta naladěna a jeho šířka. Dále nabídka pro nastavení filtrování paketů a statistiky o provozu. Užitečnou funkcí je analyzátor spektra, který je zde také dostupný.

Zachycené pakety je možné ukládat do souboru pomocí příkazu ve výpisu 5.12, bohužel tento soubor není formátu *.pcap*, ale jde o obyčejný textový soubor. Soubor generovaný pomocí HORST tedy není možné otevřít ve Wireshark.

HORST má stejně jako Kismet podporu klient/server. Lze se tedy připojit ze vzdálené stanice a sledovat zachytávaný provoz vzdáleně.

Výpis 5.12: Příkaz pro uložení výsledků zachytávání do souboru.

```
pi@raspberrypi:~ $ sudo horst -i wlan1 -o horst_log
```



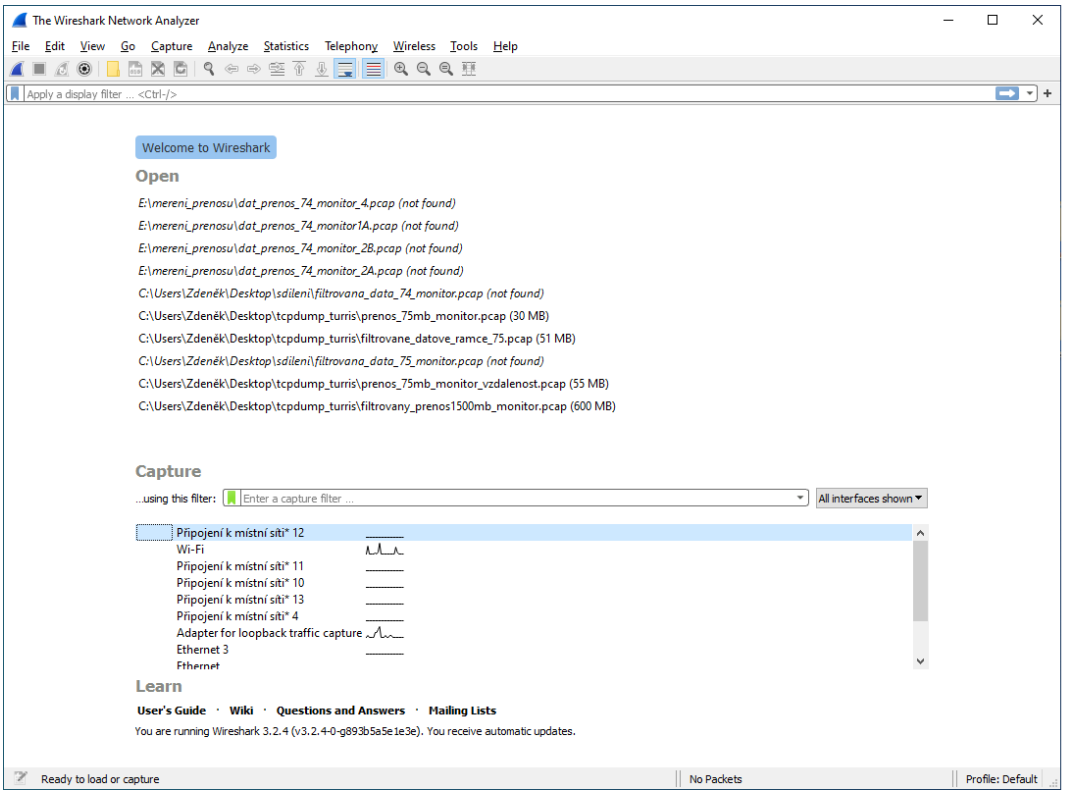
Obr. 5.6: Výběr kanálu v HORST.

5.4 Wireshark a T-shark

Wireshark je pravděpodobně nejznámější a zřejmě i nejpoužívanější paketový analyzátor a „sniffer“. Je multiplatformní a tedy dostupný pro operační systémy Windows, Linux a Mac OS. Oproti předchozím zmíněným softwarům je nejkomplexnější, protože nabízí nejvíce nástrojů nejen pro zachytávání komunikace, ale také pro její analýzu. Zároveň je však výkonově nejvíce náročný. Wireshark je velmi podobný jednoduchému Tcpdump, má ale navíc grafické uživatelské rozhraní a spoustu integrovaných řadících a filtrovacích voleb, které jsou jeho silnou stránkou. [62]

Wireshark umožňuje zachytávat komunikaci po kabelu (ethernet), z bezdrátových sítí (IEEE 802.11, Bluetooth) ale například i provoz na USB. Díky grafickému rozhraní umožňuje zobrazení zachycených rámců a jejich okamžitou analýzu. Zachycené rámce lze filtrovat podle mnoha filtrů [63], přičemž filtrování je možné již přímo během zachytávání. Samozřejmostí je ukládání do souboru v PCAP formátu a jejich čtení. Podporované jsou ale i další formáty.

Wireshark umožňuje i zachytávání v promiskuitním a monitorovacím režimu, přičemž vybrané síťové rozhraní umí do těchto režimů uvést. Ne vždy tato možnost však funguje a velmi záleží na ovladači daného adaptéru. Opět tedy platí, že je lepší síťové rozhraní do těchto režimů uvést ručně. Hned na úvodní obrazovce (obrázek 5.7) se dole zobrazují rozhraní, na kterých je možné zachytávat přenos.



Obr. 5.7: Grafické uživatelské rozhraní Wireshark.

Wireshark podporuje i pluginy, které mohou být použity pro analýzu nových protokolů.

T-shark je verze Wireshark určená pro běh v příkazové řádce a podporuje stejné možnosti jako samotný Wireshark. Je vhodný tedy zejména tam, kde není k dispozici grafické rozhraní (např. router), nebo tam, kde není žádoucí. Ovládá se tedy z příkazové řádky, ve které také umí zobrazovat výsledky a filtrovat je. [64] V podstatě je T-shark velmi podobný Tcpdump a i některé jeho parametry jsou totožné.

6 Realizace prvotního zachytávání

Tato kapitola pojednává o zprovoznění prvního zachytávání provozu Wi-Fi sítí na vybraném hardwaru a softwaru a ukazuje zásadní rozdíly mezi promiskuitním a monitorovacím režimem.

6.1 Použitý hardware

Pro zachytávání v monitorovacím režimu byl použit již zmiňovaný Turris Omnia, jehož parametry lze nalézt v kapitole 3.1. Turris byl připojen ethernetovým kabelem k notebooku, ze kterého byl přes SSH ovládán. Dalším síťovým hardwarem použitým pro zachytávání tentokrát v promiskuitním režimu je výkonný USB Wi-Fi adaptér Alfa Awus1900. Jeho parametry jsou uvedeny v tabulce 6.1. Tento USB adaptér byl připojen k druhému notebooku.

Tab. 6.1: Parametry USB Wi-Fi adaptéru Alfa Awus1900 [65].

Chipset:	Realtek RTL8814AU
Frekvenční pásmo:	2,4 GHz, 5 GHz
Standardy:	802.11a/b/g/n/ac
Šířka kanálu:	20/40/80 MHz
MIMO:	4x4:3
Přenosová rychlost	Max. 1300 Mbit/s
Monitorovací režim:	Ano
Promiskuitní režim:	Ano
Rozhraní:	USB 3.0 (3.1 Gen. 1)

6.2 Použitý software

Na routeru Turris Omnia běží firmware TurrisOS 4.0.5, jenž je založen na firmware OpenWrt verze 18.06 [66]. Ovladače pro Wi-Fi chipsety na Turrisu jsou původní předinstalované „ath9k“ a „ath10k“ a nebylo potřeba je nijak modifikovat. Notebook, ke kterému je Turris pevně připojen, disponuje operačním systémem Windows 10. Notebook s USB adaptérem „Alfa Awus1900“ má operační systém „Kali Linux“, což je linuxová distribuce odvozená od Debianu, navržená pro digitální forenzní analýzu a penetrační testy [67]. Tato distribuce má již ve svém základu předinstalované různé nástroje pro zachytávání a analýzu síťového provozu.

Pro zprovoznění promiskuitního a monitorovacího režimu na USB Wi-Fi adaptéru Alfa Awus1900 bylo nutné, nainstalovat upravený ovladač pro tento chipset. Jedná se o ovladač vyvíjený komunitou, jenž stojí za vývojem sady nástrojů Aircrack-ng [68], která slouží pro testování bezpečnosti Wi-Fi sítí a jejich prolamování. Ovladač je k dispozici zde [69]. Pro zachytávání byly použity programy Tcpdump a Wireshark.

6.3 Zachytávání v promiskuitním režimu

Pro promiskuitní režim byl použit „Alfa Awus1900“ a Wireshark. Nejdříve bylo nutné zjistit logický název síťového rozhraní, na kterém má být zachytávání spuštěno. To lze provést příkazem `ifconfig` nebo `iwconfig`, který byl použit pro zobrazení jen bezdrátových síťových rozhraní.

Výpis 6.1: Výpis z informací o síťových rozhraních pomocí `iwconfig`.

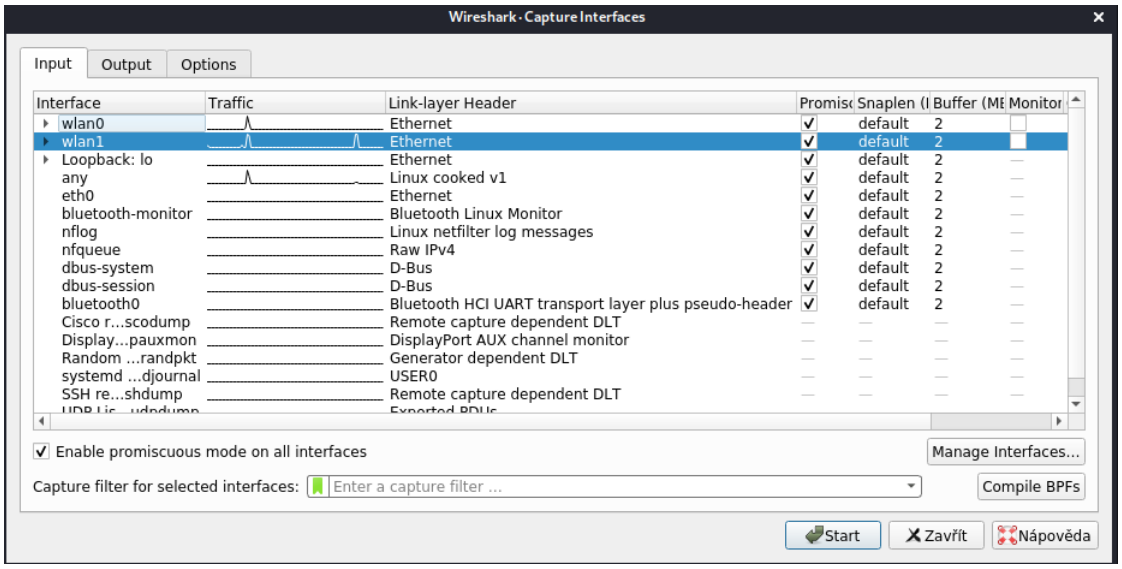
```
zdenek@kali:~$ sudo iwconfig
[sudo] heslo pro zdenek:
lo        no wireless extensions.

eth0      no wireless extensions.

wlan1     IEEE 802.11AC  ESSID:"Snoop"  Nickname:"<
   WIFI@REALTEK>"
          Mode:Managed  Frequency:5.18 GHz  Access Point:
              90:5C:44:77:84:35
          Bit Rate:1.3 Gb/s   Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key
             :****-****-****-****-****-****-****-****
             Security mode:open
          Power Management:off
          Link Quality=96/100  Signal level=-57 dBm
             Noise level=0 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx
             invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed
              beacon:0
```

Logický název rozhraní byl tedy „wlan1“ a z výpisu bylo vidět, že již bylo asociováno k bezdrátové síti s SSID „Snoop“. S touto informací bylo možné přejít do programu Wireshark a spustit zachytávání.

Po spuštění Wiresharku byla z horního menu otevřena nabídka „*Capture*“ a v ní položka „*Options*“. Zobrazilo se okno pro nastavení rozhraní pro zachytávání, jak je ukázano na obrázku 6.1.

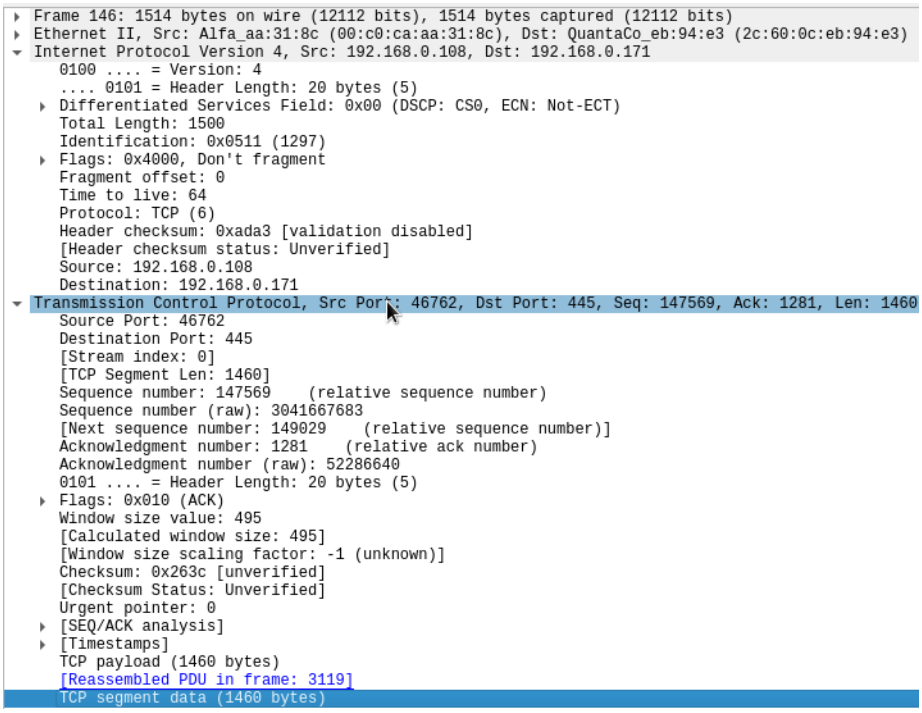


Obr. 6.1: Okno nastavení rozhraní pro zachytávání ve Wireshark.

Wireshark je schopen uvést rozhraní do promiskuitního režimu, stačí, aby bylo z vybraného rozhraní zaškrtnuto políčko „*Promiscuous*“. Po kliknutí na tlačítko „*Start*“ bylo spuštěno zachytávání provozu v promiskuitním režimu. Zachycené rámce se vždy ihned zobrazují v okně Wiresharku. Po zastavení zachytávání je možné výsledky uložit do souboru.

Jak je částečně vidět z obrázku 6.2, tak v promiskuitním režimu byly zachyceny pouze pakety, které byly adresovány pro a nebo ze zařízení, jenž patří do sítě, ke které je zachytávající síťové rozhraní asociováno. Potvrdilo se tedy to, že v promiskuitním režimu byl zachytáván pouze provoz patřící do dané Wi-Fi sítě, ke které je „sniffer“ připojen. Jakékoliv další rámce z okolních sítí byly zahozeny na základě filtrování dle SSID. V tomto režimu tedy není možné zachytávat provoz cizích Wi-Fi sítí, například Wi-Fi síť v sousedním bytu, bez asociace k nim.

Zachycené rámce byly dešifrované a kromě pozměněné hlavičky fyzické vrstvy (změna z 802.11 na ethernet) v nich byly viditelné i vyšší vrstvy, tedy síťová, transportní a aplikační. Ze zachycených rámců bylo možné vyčíst údaje o IP adresách ze síťové vrstvy, použitém protokolu a číslu portu na transportní vrstvě a také informace přenosu na aplikační vrstvě.



Obr. 6.2: Obsah rámce zachyceného v promiskuitním režimu.

6.4 Zachytávání v monitorovacím režimu

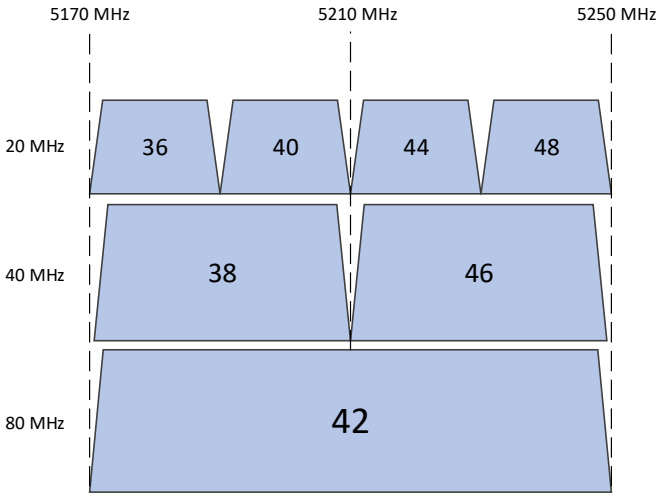
Pro zachytávání v monitorovacím režimu byl jako „sniffer“ použit router Turris Omnia nástroj Tcpdump. Opět bylo nejdříve nutné, nechat pomocí `iwconfig` vypsat všechna dostupná Wi-Fi rozhraní a zjistit jejich logický název. V případě Turrisu to bylo rozhraní „wlan0“ na kterém jsem se rozhodl zachytávat provoz v 5 GHz frekvenčním pásmu. Následně bylo toto rozhraní uvedeno do monitorovacího režimu pomocí příkazů uvedených ve výpisu 6.2

Výpis 6.2: Příkazy pro uvedení Wi-Fi rozhraní do monitorovacího režimu.

```
root@turris:~# ip link set wlan0 down
root@turris:~# iw dev wlan0 set type monitor
root@turris:~# ip link set wlan0 up
root@turris:~# iw dev wlan0 set freq 5180 80 5210
```

První příkaz Wi-Fi rozhraní vypnul, aby bylo možné měnit jeho režim. Druhý příkaz uvedl Wi-Fi rozhraní do monitorovacího režimu a třetí jej opět zapnul. Velmi důležitý je poslední uvedený příkaz, který nastavil frekvenci řídícího kanálu, jeho šířku a středovou frekvenci na které bude Wi-Fi karta operovat. To znamená, že zařízení nyní naslouchá na frekvenci 5180 MHz (kanál 36) se šířkou kanálu 80 MHz,

jehož středová frekvence je 5210 MHz (kanál 42). Tato šířka kanálu pokrývá kanály 36 až 48, jak je znázorněno na obrázku 6.3.



Obr. 6.3: Kanály IEEE 802.11 5GHz pásma pokryty aplikovaným nastavením.

Toto nastavení bylo vybráno z toho důvodu, že ve stejném pásmu se stejnou šířkou kanálu operoval domácí Wi-Fi router a jím distribuovaná Wi-Fi síť. Bylo tedy téměř jisté, že bude možné zachytit nějaký provoz. Je vhodné nejdříve zjistit, zda ve vybraném pásmu vysílají nějaká zařízení. Pokud by ve vybraném pásmu nevysílala žádná Wi-Fi zařízení, tak „sniffer“ by nezachytil žádné rámce. Šířka pásma 80 MHz byla vybrána i proto, aby bylo vyzkoušeno, zda je Wi-Fi chipset schopný zachytávat i při této šířce kanálu. Větší, tedy 160 MHz šířka pásma nebyla podporována.

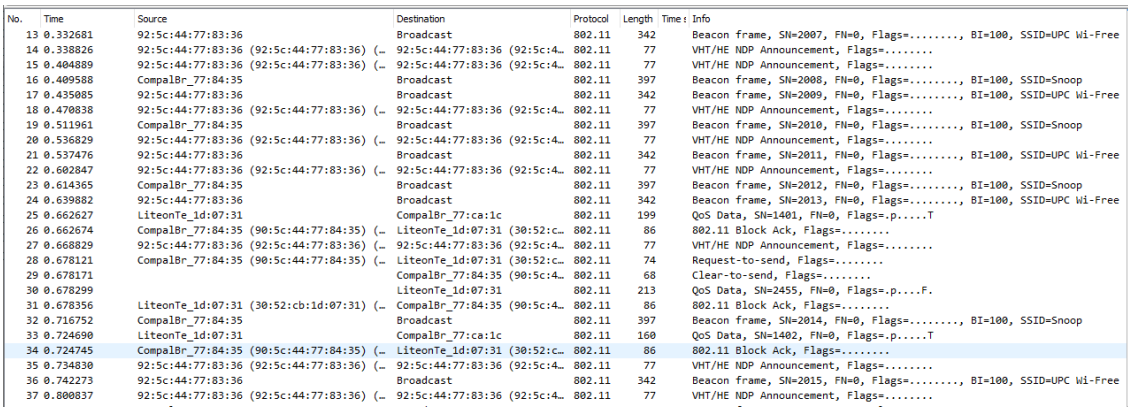
Následně byl spuštěn Tcpdump s několika argumenty, jak ukazuje výpis 6.3. Nebyly použity žádné filtry, takže byl zachytáván všechen provoz v daném pásmu.

Výpis 6.3: Příkaz pro spuštění zachytávání pomocí Tcpdump.

```
root@turris:~# tcpdump -i wlan0 -s0 -w /mnt/sda1/
   monitorovaci_rezim_turris.pcap
```

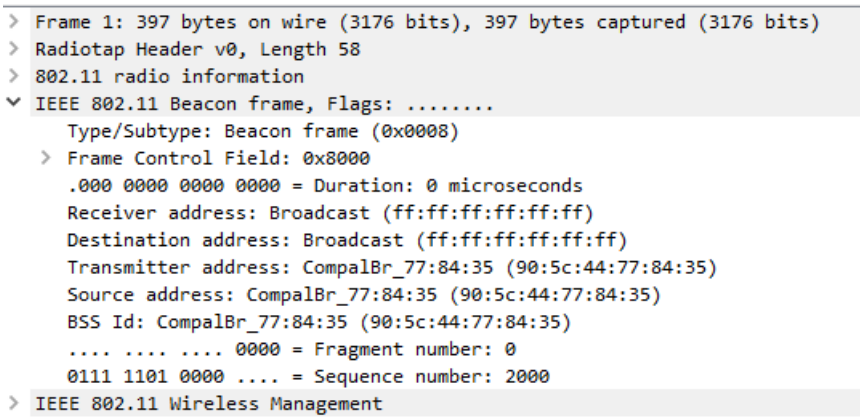
- **-i** Argument interfaces, za kterým následuje název síťového rozhraní.
- **-s0** Argument snaplen, nastaví maximální velikost (262144 bajtů) rámců pro zachycení.
- **-w** Argument write, uloží zachycené rámce do souboru.

S těmito parametry nebyly výsledky vypisovány přímo do konzole, ale ukládány do souboru, takže je následně po ukončení zachytávání bylo možné otevřít na notebooku ve Wireshark. Jak vypadaly zachycené rámce je možné vidět na následujících obrázcích 6.4 a 6.5. Je vidět, že byly zachyceny různé typy rámců 802.11 a místo IP adres jsou viditelné pouze MAC adresy.



No.	Time	Source	Destination	Protocol	Length	Time	Info
13	0.332681	92:5c:44:77:83:36	Broadcast	802.11	342		Beacon frame, SN=2007, FN=0, Flags=........, BI=100, SSID=UPC Wi-Free
14	0.338826	92:5c:44:77:83:36 (92:5c:44:77:83:36) (…	92:5c:44:77:83:36 (92:5c:4…	802.11	77		VHT/HE NDP Announcement, Flags=........
15	0.404889	92:5c:44:77:83:36 (92:5c:44:77:83:36) (…	92:5c:44:77:83:36 (92:5c:4…	802.11	77		VHT/HE NDP Announcement, Flags=........
16	0.409588	CompalBr_77:84:35	Broadcast	802.11	397		Beacon frame, SN=2008, FN=0, Flags=........, BI=100, SSID=Snoop
17	0.435085	92:5c:44:77:83:36	Broadcast	802.11	342		Beacon frame, SN=2009, FN=0, Flags=........, BI=100, SSID=UPC Wi-Free
18	0.470838	92:5c:44:77:83:36 (92:5c:44:77:83:36) (…	92:5c:44:77:83:36 (92:5c:4…	802.11	77		VHT/HE NDP Announcement, Flags=........
19	0.511961	CompalBr_77:84:35	Broadcast	802.11	397		Beacon frame, SN=2010, FN=0, Flags=........, BI=100, SSID=Snoop
20	0.536829	92:5c:44:77:83:36 (92:5c:44:77:83:36) (…	92:5c:44:77:83:36 (92:5c:4…	802.11	77		VHT/HE NDP Announcement, Flags=........
21	0.537476	92:5c:44:77:83:36	Broadcast	802.11	342		Beacon frame, SN=2011, FN=0, Flags=........, BI=100, SSID=UPC Wi-Free
22	0.602847	92:5c:44:77:83:36 (92:5c:44:77:83:36) (…	92:5c:44:77:83:36 (92:5c:4…	802.11	77		VHT/HE NDP Announcement, Flags=........
23	0.614365	CompalBr_77:84:35	Broadcast	802.11	397		Beacon frame, SN=2012, FN=0, Flags=........, BI=100, SSID=Snoop
24	0.639882	92:5c:44:77:83:36	Broadcast	802.11	342		Beacon frame, SN=2013, FN=0, Flags=........, BI=100, SSID=UPC Wi-Free
25	0.662627	LiteonTe_1d:07:31	CompalBr_77:ca:1c	802.11	199		QoS Data, SN=1401, FN=0, Flags=.p....T
26	0.662674	CompalBr_77:84:35 (90:5c:44:77:84:35) (…	LiteonTe_1d:07:31 (30:52:c…	802.11	86		802.11 Block Ack, Flags=........
27	0.668829	92:5c:44:77:83:36 (92:5c:44:77:83:36) (…	92:5c:44:77:83:36 (92:5c:4…	802.11	77		VHT/HE NDP Announcement, Flags=........
28	0.678121	CompalBr_77:84:35 (90:5c:44:77:84:35) (…	LiteonTe_1d:07:31 (30:52:c…	802.11	74		Request-to-send, Flags=........
29	0.678171		CompalBr_77:84:35 (90:5c:4…	802.11	68		Clear-to-send, Flags=........
30	0.678299		LiteonTe_1d:07:31	802.11	213		QoS Data, SN=2455, FN=0, Flags=.p....F.
31	0.678356	LiteonTe_1d:07:31 (30:52:cb:1d:07:31) (…	CompalBr_77:84:35 (90:5c:4…	802.11	86		802.11 Block Ack, Flags=........
32	0.716752	CompalBr_77:84:35	Broadcast	802.11	397		Beacon frame, SN=2014, FN=0, Flags=........, BI=100, SSID=Snoop
33	0.724690	LiteonTe_1d:07:31	CompalBr_77:ca:1c	802.11	160		QoS Data, SN=1402, FN=0, Flags=.p....T
34	0.724745	CompalBr_77:84:35 (90:5c:44:77:84:35) (…	LiteonTe_1d:07:31 (30:52:c…	802.11	86		802.11 Block Ack, Flags=........
35	0.734830	92:5c:44:77:83:36 (92:5c:44:77:83:36) (…	92:5c:44:77:83:36 (92:5c:4…	802.11	77		VHT/HE NDP Announcement, Flags=........
36	0.742273	92:5c:44:77:83:36	Broadcast	802.11	342		Beacon frame, SN=2015, FN=0, Flags=........, BI=100, SSID=UPC Wi-Free
37	0.800837	92:5c:44:77:83:36 (92:5c:44:77:83:36) (…	92:5c:44:77:83:36 (92:5c:4…	802.11	77		VHT/HE NDP Announcement, Flags=........

Obr. 6.4: Zachycené rámce v monitorovacím režimu.



```
> Frame 1: 397 bytes on wire (3176 bits), 397 bytes captured (3176 bits)
> Radiotap Header v0, Length 58
> 802.11 radio information
v IEEE 802.11 Beacon frame, Flags: ........
    Type/Subtype: Beacon frame (0x0008)
  > Frame Control Field: 0x8000
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: CompalBr_77:84:35 (90:5c:44:77:84:35)
    Source address: CompalBr_77:84:35 (90:5c:44:77:84:35)
    BSS Id: CompalBr_77:84:35 (90:5c:44:77:84:35)
    .... .... .... 0000 = Fragment number: 0
    0111 1101 0000 .... = Sequence number: 2000
> IEEE 802.11 Wireless Management
```

Obr. 6.5: Zachycené rámce v monitorovacím režimu a jejich hlavičky.

Na rozdíl od promiskuitního režimu zde byly zachyceny celé rámce, jenž pocházejí ze zařízení, která pracují ve stejném frekvenčním pásmu jako „sniffer“. Potvrdilo se tedy, že nedochází k filtrování na základě SSID ani MAC adres, které jsou viditelné. V monitorovacím režimu tedy bylo možné zachytit rámce 802.11 bez pozměněné hlavičky. Zachyceny byly kontrolní a management rámce typu „Beacon“, „Probe“, „RTS“, „CTS“ či „ACK“ a jiné. Dále byly zachyceny i datové rámce, které v sobě nesou šifrovaná data. Zachyceny byly také poškozené, nebo opakované rámce. To vše bez nutnosti asociace k síti.

Obsah zachycených rámců je odlišný od obsahu rámců z promiskuitního režimu zejména tím, že obsahují detailní informace o fyzické vrstvě. Tyto informace nelze v promiskuitním režimu zachytit, kvůli nahrazení původní hlavičky. Informace o fyzické vrstvě jsou obsaženy v hlavičkách „Radiotap header v0“ a „802.11 radio information“. Informace v nich obsažené jsou předmětem analýz v kapitole 8.2. Vyšší vrstvy, jako síťová nebo transportní, zde nejsou viditelné z důvodu šifrování.

Monitorovací režim byl stejným způsobem vyzkoušen i na USB adaptéru „Alfa Awus1900“. Zda nastal problém se „zafixováním“ na vybraný kanál a to i přesto, že byl použit totožný postup jako výše popsaný. Po spuštění zachytávání docházelo k tzv. „channel hopping“, tedy k přeskakování adaptéru po různých kanálech 2,4GHz a 5GHz frekvenčního pásma. Funguje to tak, že Wi-Fi adaptér vydrží vždy pouze chvíli na jedné frekvenci a následně je přeladěn na jinou. Tato funkce může být žádoucí při průzkumu spektra a vytíženosti jednotlivých kanálu, ale ne pokud je potřeba dlouhodobě odposlouchávat jeden kanál. Bylo zjištěno, že tento jev byl způsoben programem „NetworkManager“, který je na „Kali Linux“ přítomen. Jde o program pro poskytování detekce a konfigurace automatického připojování k síti, který řídí připojování k drátovým a bezdrátovým sítím v systému [70]. K vyřešení tohoto problému byl použit nástroj ze sady nástrojů „Aircrack-ng“, konkrétně „Airmon-ng“ [71], který je určen pro uvedení Wi-Fi rozhraní do monitorovacího režimu. Tento nástroj umí detekovat, zda ke konkrétnímu bezdrátovému Wi-Fi rozhraní nepřistupuje nějaký další proces, který by jej ovlivňoval. K tomu slouží příkaz, jenž je uveden v následujícím výpisu 6.4. Následně bylo možné vypsané procesy ukončit.

Výpis 6.4: Příkaz pro detekci procesů přistupujících k síťovému rozhraní.

```
zdenek@kali:~$ sudo airmon-ng check
[sudo] heslo pro zdenek:

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng␣check␣kill' before putting
the card in monitor mode, they will interfere by changing
    channels
and sometimes putting the interface back in managed mode

    PID Name
    608 NetworkManager
    712 wpa_supplicant
```

6.5 Srovnání promiskuitního a monitorovacího režimu

Praktickou zkouškou zachytávání Wi-Fi provozu v obou zmíněných režimech se potvrdila teorie uvedená v kapitole 4.2. Zde je shrnutí zjištěných informací:

- V **promiskuitním režimu** je Wi-Fi karta asociovaná k Wi-Fi síti, ze které následně „sniffuje“ (zachytává) veškerý provoz patřící do této sítě. Přijímá i pakety, které nejsou adresovány přímo jí a zároveň může vysílat. Pracuje tedy na frekvenci a šířce kanálu, jenž určuje AP. Provoz je filtrován na základě SSID, případné rámce zachycené z jiných sítí jsou tedy zahozeny. Hlavička fyzické vrstvy 802.11 je zahozena a nahrazena ethernetovou hlavičkou, čímž dochází ke ztrátě informací o bezdrátovém přenosu. Rámce obsahují hlavičky všech síťových vrstev a lze tedy zjistit, komu byly pakety adresovány, jaké protokoly byly použity pro přenos, jaký typ informací je přenášen a pro jakou aplikaci byl určen.
  Promiskuitní režim je vhodný pro monitoring a analýzu provozu v daném síťovém segmentu. Může být využit pro diagnostiku sítě a detekci problémů v síti. Zároveň však může být zneužit útočníky pro odposlouchávání komunikace v síti a k následným útokům.
- V **monitorovacím režimu** není Wi-Fi karta asociována k žádné síti a zpravidla nemůže sama vysílat. Je nutné ji naladit na konkrétní kanál či frekvenci vybraného frekvenčního pásma. Jsou zachytávány rámce vysílané stanicemi, jenž operují na stejném frekvenčním kanálu a zároveň jsou v dosahu příjmu „snifferu“. Neprobíhá tedy filtrování na základě SSID ani MAC adres. Zároveň musí „sniffer“ podporovat stejné nebo vyšší standardy 802.11. Nelze tedy například zachytávat rámce standardu IEEE 802.11ac s adaptérem podporujícím nanejvýš IEEE 802.11g. Rámce mají původní hlavičku 802.11 fyzické vrstvy a jsou tedy dostupné informace o parametrech bezdrátového spojení. Je možné zachytávat kontrolní, management a i datové rámce, které jsou však šifrovány. Zachytávány jsou i poškozené rámce se špatným kontrolním součtem. Vzhledem k tomu není možné vidět obsah vyšších vrstev TCP/IP.
  Monitorovací režim je možné využít při navrhovaní nových Wi-Fi sítí. Lze jím zjistit počet zařízení využívající daný kanál a vyhnout se tak rušení. Dále může sloužit pro studijní účely a pro lepší pochopení toho, jak funguje komunikace ve Wi-Fi sítích. Další využití může být analýza provozu a zejména pasivní odposlouchávání a útoky na Wi-Fi sítě.

7 Možnosti zachytávání

Tato kapitola se zabývá možnostmi zachytávání rámců IEEE 802.11 v monitorovacím režimu. V zásadě se jedná o zachytávání lokální a vzdálené.

- Lokální zachytávání - zachytávání je spuštěno přímo ze zařízení, jenž zachytávání vykonává a výsledky jsou ukládáno nebo zobrazeny lokálně na tomto zařízení.
- Vzdálené zachytávání - zachytávání je na zařízení jej vykonávající inicializováno ze vzdáleného klienta a výsledky jsou odesílány a ukládány do vzdáleného klienta.

7.1 Vzdálené zachytávání

Zachytávání Wi-Fi provozu je možné spustit buď přímo ze zařízení, které zachytávání provádí („sniffer“), nebo vzdáleně z jiného počítače jenž se ke „snifferu“ připojí. Vzdálené zachytávání („remote capture“) je výhodné v tom, že ke „snifferu“ je možné se připojit téměř odkudkoliv a monitorovat tak provoz bez nutnosti mít „sniffer“ fyzicky u sebe. „Sniffer“ však musí být připojen k internetu, to znamená že musí disponovat minimálně dvěma síťovými adaptéry.

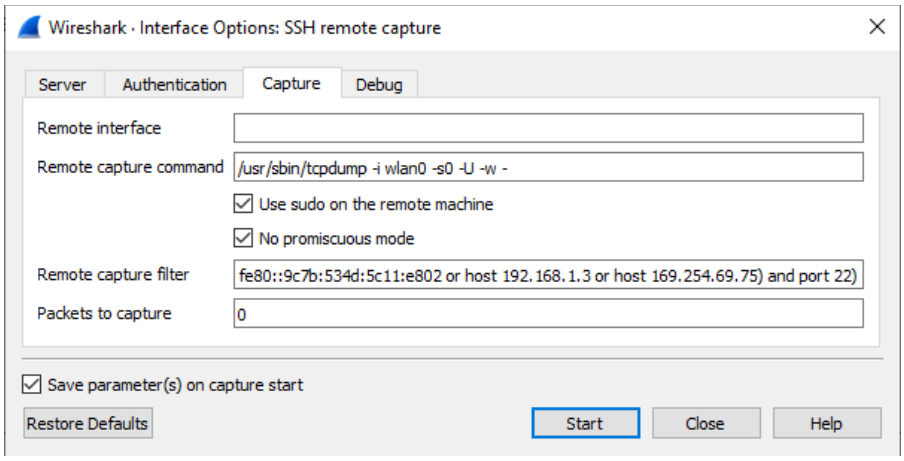
Při prozkoumávání vzdáleného zachytávání jsem zjistil dvě možnosti, jak jej zprovoznit. První z nich je nástroj „**SSHdump**“, jenž je součástí Wireshark a druhou je program „**RPCAPd**“.

7.1.1 SSHdump

Jedná se o nástroj, jenž je součástí Wireshark [72]. Při instalaci Wireshark je však nutné explicitně zatrhnout možnost instalace „SSHdump“ (Secure shell dump). Jedná o nástroj umožňující zachytávat síťový provoz přes SSH spojení. Na straně „snifferu“ je tedy nutné mít povolen a zapnut „SSH server“, ke kterému se vzdálený klient připojí.

Po spuštění Wiresharku je v seznamu dostupných síťových rozhraní dostupné rozhraní s názvem „SSH remote capture“. Kliknutím na ikonu ozubeného kola, vedle názvu rozhraní, se otevře nastavení (viz. obr 7.1).

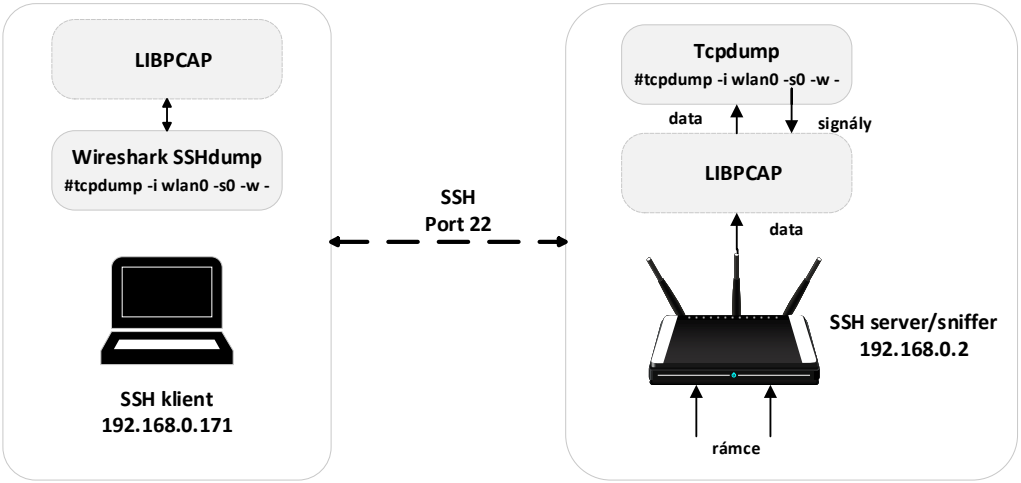
V okně „Server“ bylo nutné vyplnit IP adresu a port SSH serveru, v tomto případě adresa routeru Turris Omnia. V okně „Authentication“ bylo vyplněno jméno uživatele a heslo pro připojení klienta k SSH serveru. V předposledním okně „Capture“ se nastavují parametry zachytávání. Zde je pole „Remote capture command“, do kterého se zapíše příkaz pro spuštění programu pro zachytávání síťového provozu. Pole bylo vyplněno příkazem: „`/usr/sbin/tcpdump -i wlan0 -s0 -U -w -`“. Ten



Obr. 7.1: Nastavení nástroje „SSHdump“ ve Wireshark.

na straně serveru spustí z daného umístění Tcpdump s uvedenými parametry. Pomlčka za parametrem `-w` znamená, že zachycené rámce se ihned vypisují ve Wireshark. Dále bylo potřeba zaškrtnou dvě políčka, jenž jsou vidět na obrázku. Pole „Remote capture filter“, jenž slouží pro nastavení filtrování provozu, bylo ponecháno ve výchozím stavu.

Před samotným spuštěním bylo samozřejmě nutné, uvést Wi-Fi adaptér na Turris Omnia do monitorovacího režimu. Po spuštění se rámce zachycené na „snifferu“ zobrazují ve Wireshark na vzdáleně připojeném notebooku. Zachycené rámce pak bylo možné ve Wireshark uložit do souboru.

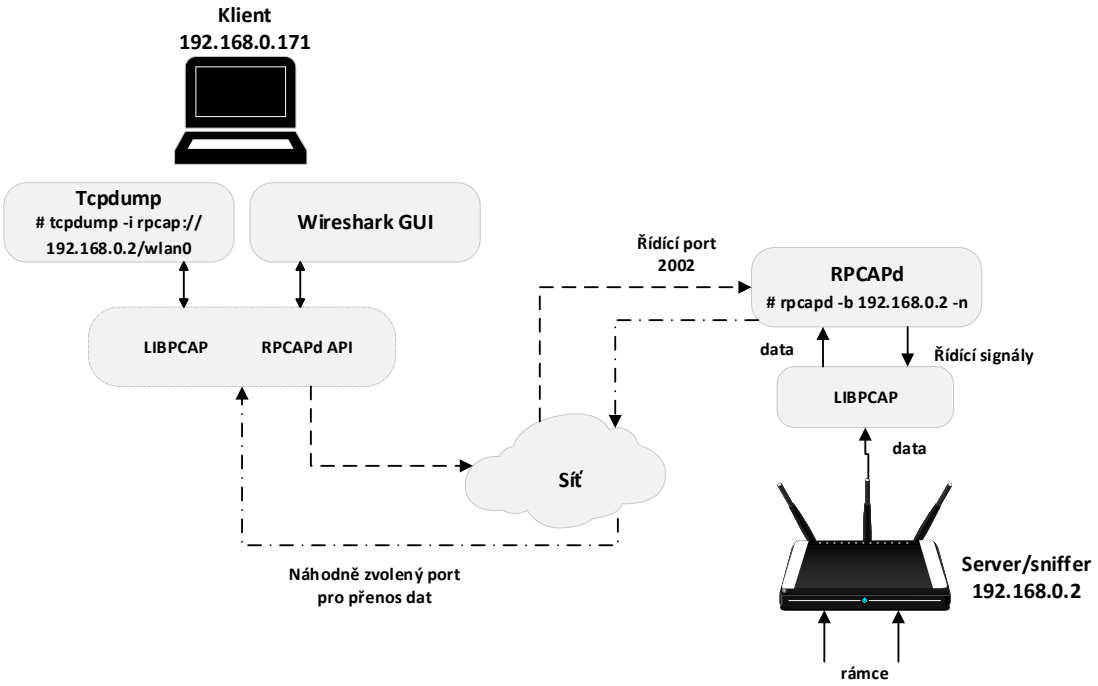


Obr. 7.2: Vzdálené zachytávání rámců pomocí SSHdump.

7.1.2 RPCAPd

RPCAPd („Remote packet capture deamon“) je pomocný program nebo spíše „deamon“ umožňující vzdálené zachytávání. Zpřístupňuje rámce z lokálních síťových rozhraní „snifferu“ vzdáleným klientům. Klientem může být jakýkoliv program pro zachytávání rámců založený na knihovně Libpcap. „RPCAPd“ je součástí této knihovny a to od verze 1.9.0.

Princip je zobrazen na obrázku 7.3. Na straně „snifferu“ je spuštěn RPCAPd, který přidá nové síťové rozhraní. K tomuto síťovému rozhraní pak vzdáleně přistupuje klient přes Tcpdump nebo Wireshark. Pro přenos řídících signálu je použit port 2002 a pro přenos dat je vybrán náhodně zvolený port.



Obr. 7.3: Vzdálené zachytávání rámců pomocí RPCAPd.

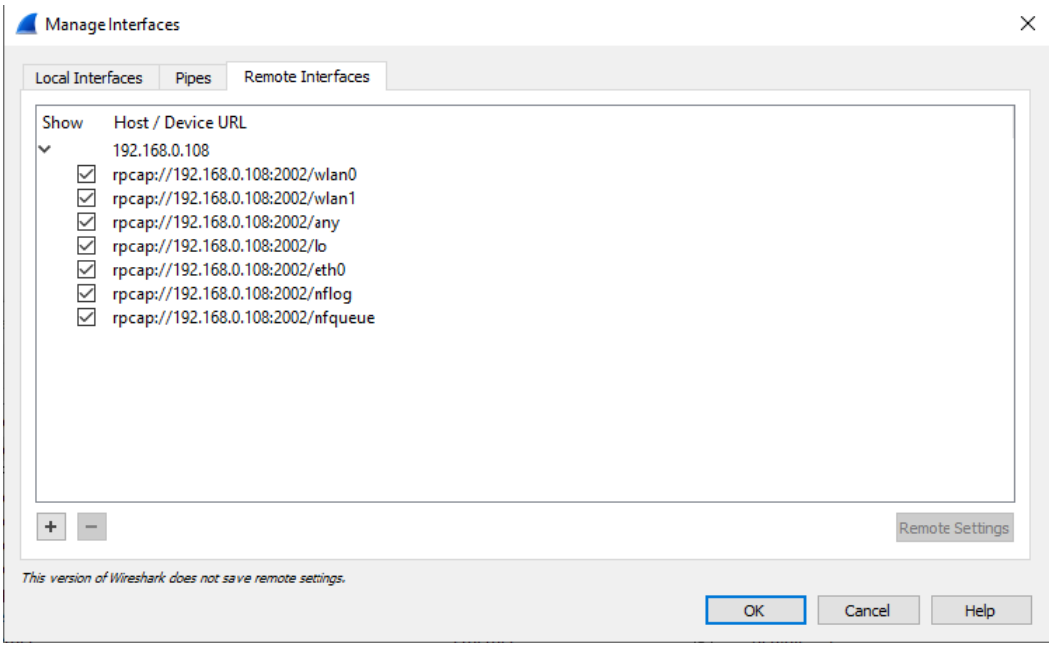
Pro zprovoznění RPCAPd bylo nutné zkompilovat a nainstalovat novou verzi knihovny Libpcap. Při kompilaci bylo nutné přidat parametr „`-enable-remote`“ za příkaz „`./configure`“. Bohužel vzdálené zachytávání pomocí RPCAPd nebylo možné vyzkoušet na routeru Turris Omnia. Důvodem bylo, že v době psaní této kapitoly se v TurrisOS 4.0.5 vyskytoval „bug“ [73], který znemožňoval instalaci kompilátoru „GCC“ (*GNU Compiler Collection*), jenž je nutný pro instalaci knihovny Libpcap. Vzdálené zachytávání bylo tedy zprovozněno na notebooku s Kali Linux a Wi-Fi adaptérem Alfa Awus1900. Postup byl následující. Pro spuštění RPCAPd stačí napsat následující příkaz 7.1.

Výpis 7.1: Příkaz pro spuštění RPCAPd.

```
zdenek@kali:~$ sudo rpcapd -b 192.168.0.108 -n
```

Parametr „`-b`“ zde určuje na jaké IP adrese je RPCAPd dostupný a „`-n`“ udává, že není potřeba autentizace.

Následně bylo možné ověřit funkčnost vzdáleného zachytávání ve Wireshark na klientském notebooku. Ve Wireshark bylo potřeba otevřít nabídku „Capture -> Options -> Manage Interfaces“ a poté v nově otevřeném okně „Remote Interfaces“. Zde bylo třeba přidat vzdálená rozhraní zadáním IP adresy RPCAPd serveru.
Poté se zobrazila všechna dostupná síťová rozhraní na vzdáleném zařízení, tak jak ukazuje obrázek 7.4.



Obr. 7.4: Přidání vzdáleného síťového rozhraní ve Wireshark.

Vybráním jednoho nebo více vzdálených síťových rozhraní, se tato rozhraní přidala do seznamu rozhraní, na nichž je možné ve Wireshark spustit zachytávání. Zachycený provoz se zobrazil ve Wireshark a bylo možné jej uložit do souboru. Pro zachytávání v monitorovacím režimu je opět nutné před samotným spuštěním zachytávání uvést patřičné Wi-Fi rozhraní do monitorovacího režimu.

8 Analýza provozu

Pro analýzu zachyceného provozu se používají tzv. paketové analyzátory, jenž umožňují interpretovat obsah zachycených rámců a paketů pro uživatele v čitelné formě. Tradičními a nejpoužívanější zástupci jsou Wireshark a Tcpdump, jenž byly již zmíněny v kapitole 5.

Pro analýzu rámců IEEE 802.11 jsem využil právě program Wireshark, ale i několik vlastních skriptů napsaných v jazyce Python. Zaměřil jsem se zejména na zachytávání a analýzu rámců novějšího standardu IEEE 802.11ac. Bylo zkoumáno, jaké informace lze získat o přenosu různých typů rámců na fyzické vrstvě. Zvláštní pozornost byla věnována rámcům využívajícím technologii MIMO a tomu, zda je možné komunikaci přenášenou za využití MIMO a prostorových streamů zachytit.

8.1 Možnosti analýzy

Analýzu je možné provést buď:

- Zpětně - načtením souboru se zachyceným provozem. Hodí se pro dlouhodobé zachytávání.
- V reálném čase - ihned po zachycení rámců, tzv. „live capture“. Ideálně v kombinaci s filtry, kdy předem vím, co hledám.

Právě při analýze mají velký význam filtry, které poskytuje knihovna Libpcap a které je možné využít téměř ve všech paketových analyzátorech. Pomocí filtrů je možné vyfiltrovat téměř jakýkoliv údaj obsažený v rámci. Pro následující ukázky analýzy byl použit provoz zachycen na Turris Omnia v monitorovacím režimu.

8.2 Analýza fyzické vrstvy

Jak již bylo uvedeno, v monitorovacím režimu lze kromě informací o typu rámců získat i informace o bezdrátovém přenosu na fyzické vrstvě IEEE 802.11. Nejdříve jsem se zaměřil na to, jaké konkrétní parametry spojení lze z rámců IEEE 802.11 získat. Dále mě zajímalo, zda je možné prokázat, že při komunikaci bylo využito technologie MIMO a prostorových streamů. Otázkou také bylo, zda je možné zachytit jednotlivé prostorové streamy. Pro analýzu jsem vybral standard IEEE 802.11ac, tedy Wi-Fi 5.

Pro zachytávání byl použit Turris Omnia, jehož Wi-Fi adaptér byl nastaven na odposlech 5 GHz pásma, konkrétně na kontrolní kanál 36 s šířkou kanálu 80 MHz a středovou frekvencí 5210 MHz. Opět se jedná o stejné pásmo, jenž využívala zařízení

v mé domácí Wi-Fi síti. Samozřejmě byl zachycen i provoz jiných zařízení operujících ve stejném pásmu.

8.2.1 Radiotap header

Nejdříve je potřeba se zmínit o hlavičce Radiotap. Tato hlavička není součástí standardního rámce 802.11, ale je poskytována ovladačem Wi-Fi adaptéru, jenž provádí zachytávání [76]. Je nutné dodat, že ne všechny ovladače Radiotap poskytují, opět je nutné se poohlédnout po chipsetu a jeho ovladači, jenž jsou podporovány. Radiotap poskytuje dodatečné informace o bezdrátovém přenosu, jako jsou informace o použité frekvenci a kanálu, modulaci signálu, přenosové rychlosti, síle signálu na anténách atd. Počet informací je však závislý na použitém Wi-Fi adaptéru a jeho ovladači. Tyto informace se propisují i do další hlavičky zvané „802.11 radio information“, kde je možné najít téměř totožné informace, jako v Radiotap hlavičce.

Wi-Fi adaptér routeru Turris Omnia, potažmo chipset QCA9880 a jeho ovladač ath10k Radiotap podporují, takže bylo možné ze zachycených rámců analyzovat parametry přenosu na fyzické vrstvě.

8.2.2 Parametry přenosu

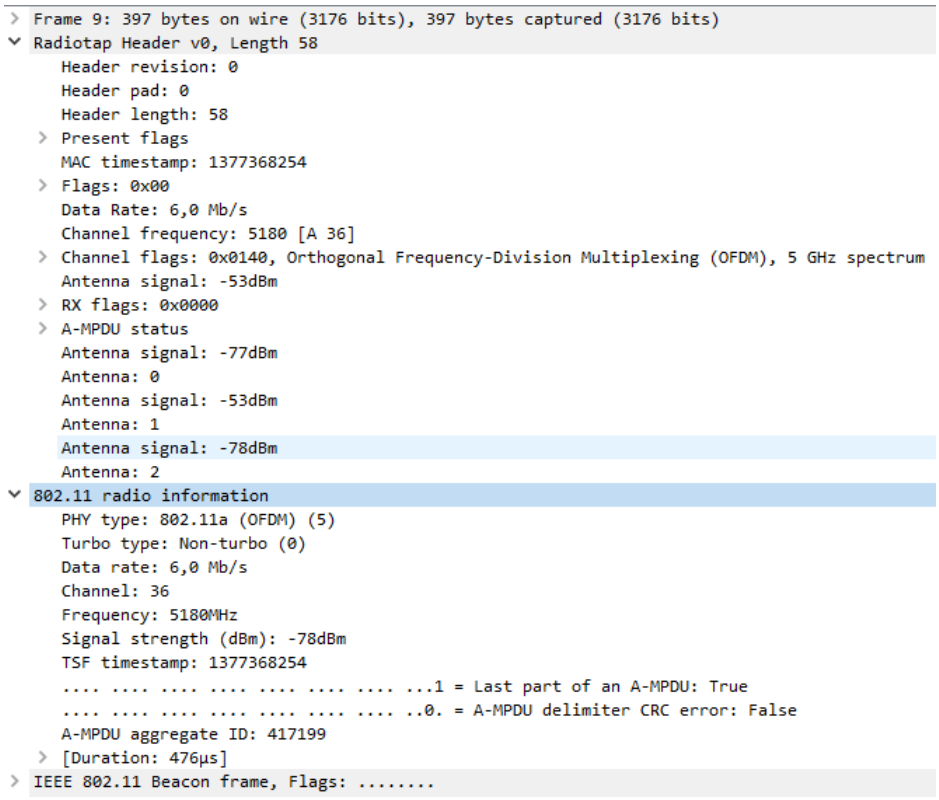
Zachytávání probíhalo pomocí Tcpdump a výsledky byly ukládány do souboru. Ten byl následně analyzován ve Wireshark na notebooku. Na obrázku 8.1 je vidět zachycený Beacon rámec s hlavičkami „Radiotap header“ a „802.11 Radio information“, ze kterých byly získány parametry přenosu.

Je vidět že tento rámec byl přenášen nízkou rychlostí 6 Mb/s. Přenos probíhal na frekvenci 5180 MHz, tedy kanál 36 a byla použita modulace OFDM. Již z tohoto šlo odvodit, že byl použit standard IEEE 802.11a, 802.11n nebo 802.11ac.

Dále bylo možné zjistit sílu přijatého signálu na všech třech anténách routeru. V další hlavičce je pole „PHY type“, které udává použitý 802.11 standard na fyzické vrstvě. Zde je uvedeno 802.11a, ačkoliv přístupový bod, vysílající tento Beacon rámec, používá v jím distribuované síti standard 802.11ac. Znamenalo to tedy, že Beacon rámec využívá staršího standardu 802.11a, a to z důvodu zpětné kompatibility. Dále se již jen opakují údaje již obsažené v Radiotap hlavičce.

Zajímavé je ovšem srovnání s informacemi o bezdrátovém přenosu obsažených v datových rámcích. Ukázalo se, že datové rámce obsahují více parametrů než rámce kontrolní a management. Ukázka přenosových parametrů v datovém rámci je na obrázku 8.2. Datový rámec obsahuje pole „VHT informations“, kde VHT znamená „Very High Throughput“. Toto pole je obsaženo v hlavičce PPDU pouze v rámcích 802.11ac a informuje o implementaci vylepšení na fyzické vrstvě, jimiž standard

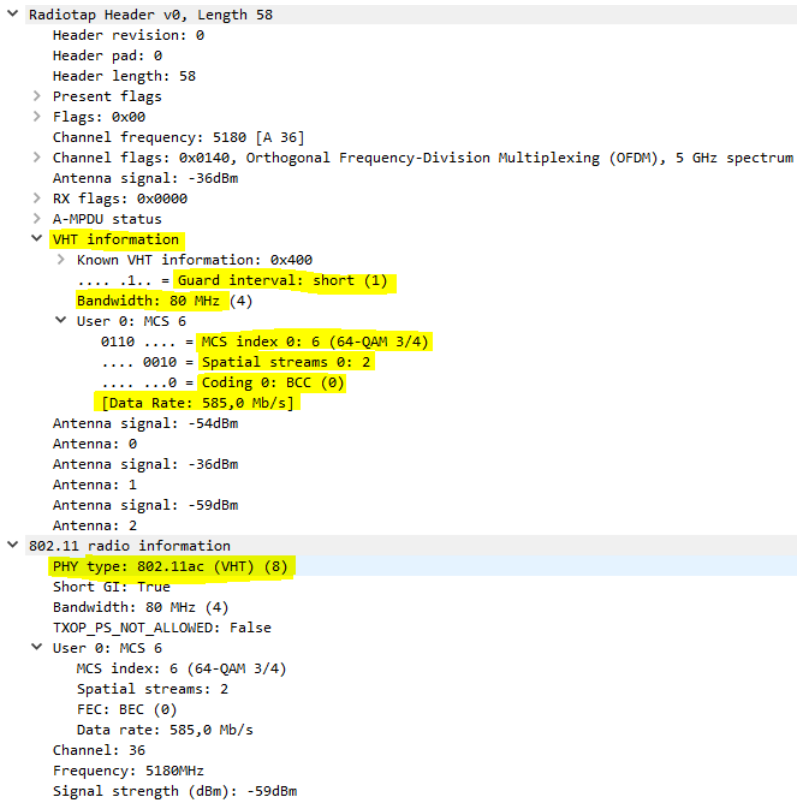
802.11ac disponuje [77]. Pro porovnání, předchozí standard IEEE 802.11n obsahuje pole HT tedy „High throughput“.



Obr. 8.1: Parametry bezdrátového přenosu získané z Radiotap hlavičky Beacon rámce.

V tomto případě pole VHT obsahovalo následující informace:

- **Bandwidth: 80 MHz** - Šířka kanálu tedy byla 80 MHz.
- **PHY type: 802.11ac** - Použitý standard 802.11 na fyzické vrstvě.
- **Short GI: True** - Značí že byl použit krátký (400 ns) ochranný interval („Guard Interval“). Ochranný interval se používá jako ochrana proti ztrátě, rušení a degradaci signálu při vícecestném šíření signálu díky odrazům [78].
- **MCS index: 6 (64-QAM 3/4)** - Modulační a kódové schéma („Modulation code scheme“). MCS udává kombinaci použité modulace, počtu prostorových streamů a kódovacího poměru. Společné s délkou ochranného intervalu a šířkou kanálu je podle něj možné určit přenosovou rychlost [79].
- **Spatial streams: 2** - Udává počet prostorových streamů, použitých pro přenos rámce.
- **Data Rate: 585,0 Mb/s** - Rychlost. kterou byl rámec přenášen. Odpovídá MCS indexu.



Obr. 8.2: Parametry bezdrátového přenosu získané z Radiotap hlavičky datového rámce.

Ukázalo se tedy, že ze zachyceného provozu je možné získat poměrně detailní informace o přenosu rámců 802.11. Potvrdilo se, že je možné zachytit i rámce vysílané pomocí více prostorových streamů. Nebylo však možné zachytit tyto streamy jednotlivě, nebo jednotlivé částí rámců vysílané individuálními streamy. Nelze tedy přesně učit, zda se jedná o duplicitní stream obsahující stejná data, nebo zda byla data rozdělena do více streamů pro rychlejší přenos. To si vysvětluji tím, že ačkoliv Wi-Fi adaptér příjme více streamů, přenášejících související části dat, streamy jsou signálovým procesorem zpracovány a složeny dohromady. Ty jsou pak odeslány jako kompletní rámec ovladači Wi-Fi adaptéru, který jej pomocí Libpcap knihovny interpretuje v analyzátoru. Ovladač tedy má informaci, že rámec byl složen z více streamů, avšak jednotlivými datovými segmenty přenášenými těmito streamy nedisponuje. Z analýzy lze také soudit, že kontrolní a management rámce nejsou vysílány za použití MIMO a více prostorových streamů. Tyto rámce byly také přenášeny pouze na nízkých přenosových rychlostech. To lze vysvětlit tím, že u těchto typů rámců je zapotřebí, aby byly zachyceny vícero stanicemi. Příkladem mohou být rámce CTS, RTS či Beacon. Pokud by tyto rámce byly vysílány pomocí více streamů, tak stanice jenž nemohou více streamů přijímat, by tyto rámce nezachytily.

8.3 Zjištění přítomnosti přístupových bodů

Ze zachyceného provozu v monitorovacím režimu bylo možné zjistit informace o tom, jaké přístupové body na daném kanálu vysílají. Tato informace může být užitečná při návrhu nové Wi-Fi sítě, ve snaze vyhnout se rušení v pásmu. Přístupové body, neboli „access pointy“ vysílají pravidelně management rámce typu „Beacon“. Tyto rámce byly již zmíněny v kapitole 1.4.2. Ze zachyceného provozu bylo možné tyto rámce ve Wireshark vyfiltrovat pomocí filtru uvedeném ve výpisu 8.1.

Výpis 8.1: Filtrování Beacon rámců.

```
wlan.fc.type_subtype == 8
```

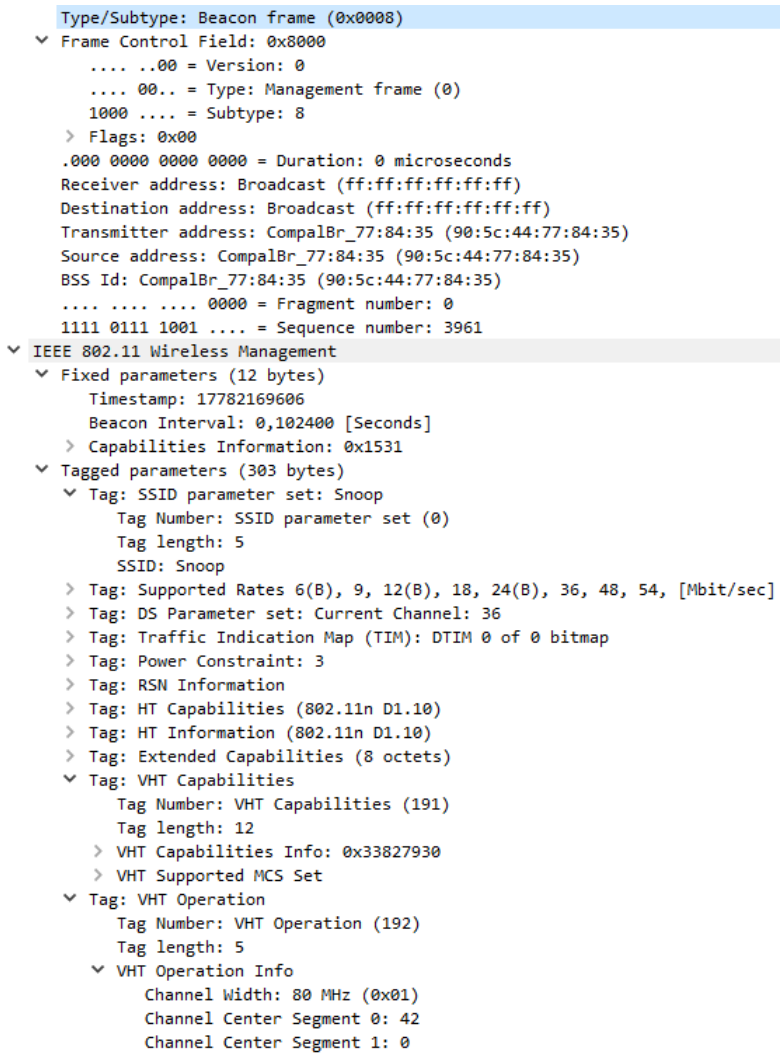
Před rovnítkem je název pole v němž se nachází informace o podtypu rámce. Číslo 8 pak v decimálním tvaru identifikuje právě rámce Beacon [6].

Z analýzy Beacon rámců bylo možné zjistit velké množství informací o přístupovém bodu a potažmo i o síti, kterou distribuuje. Část obsahu Beacon rámce zobrazuje obrázek 8.4. Jsou zde vidět některé zásadní údaje jako například vysílané SSID, interval opakování Beacon rámce, MAC adresa přístupového bodu, podporované přenosové rychlosti, aktuálně využívaný kanál a jeho šířka, použité zabezpečení a mnohem více.

V tomto případě bylo přítomno pole „VHT capabilites“ (obrázek 8.3), kde bylo možné najít informace o schopnostech, kterými přístupový bod disponuje. Kupříkladu podpora 160 MHz šířky kanálu, beamforming, prostorových streamů, MCS indexů atd.

```
▾ VHT Capabilities Info: 0x33827930
    .... .... .... .... .... .... .... ..00 = Maximum MPDU Length: 3 895 (0x0)
    .... .... .... .... .... .... .... 00.. = Supported Channel Width Set: Neither 160MHz nor 80+80 supported (0x0)
    .... .... .... .... .... .... ...1 .... = Rx LDPC: Supported
    .... .... .... .... .... .... ..1. .... = Short GI for 80MHz/TVHT_MODE_4C: Supported
    .... .... .... .... .... .... .0.. .... = Short GI for 160MHz and 80+80MHz: Not supported
    .... .... .... .... .... .... 0... .... = Tx STBC: Not supported
    .... .... .... .... .... .001 .... .... = Rx STBC: 1 Spatial Stream Supported (0x1)
    .... .... .... .... .... 1... .... .... = SU Beamformer Capable: Supported
    .... .... .... .... ...1 .... .... .... = SU Beamformee Capable: Supported
    .... .... .... .... 011. .... .... .... = Beamformee STS Capability: 4 (0x3)
    .... .... .... .010 .... .... .... .... = Number of Sounding Dimensions: 3 (0x2)
    .... .... .... 0... .... .... .... .... = MU Beamformer Capable: Not supported
    .... .... ...0 .... .... .... .... .... = MU Beamformee Capable: Not supported
    .... .... ..0. .... .... .... .... .... = TXOP PS: Not supported
    .... .... .0.. .... .... .... .... .... = +HTC-VHT Capable: Not supported
    .... ..11 1... .... .... .... .... .... = Max A-MPDU Length Exponent: 1 048 575 (0x7)
    .... 00.. .... .... .... .... .... .... = VHT Link Adaptation: No Feedback (0x0)
    ...1 .... .... .... .... .... .... .... = Rx Antenna Pattern Consistency: Supported
    ..1. .... .... .... .... .... .... .... = Tx Antenna Pattern Consistency: Supported
    00.. .... .... .... .... .... .... .... = Extended NSS BW Support: 0x0
› VHT Supported MCS Set
```

Obr. 8.3: Pole „VHT capabilites“ v Beacon rámci.



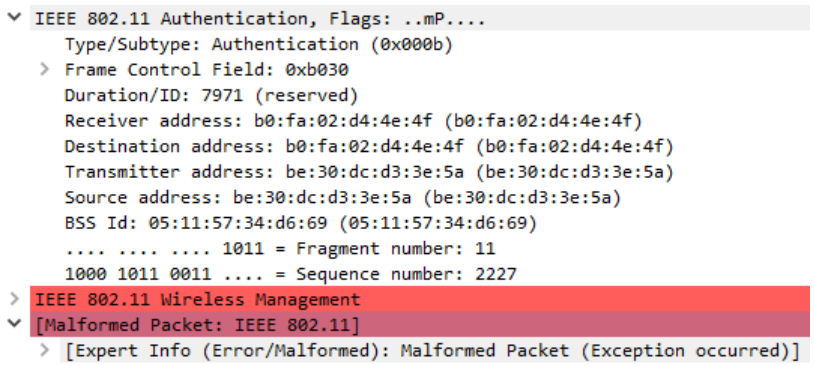
Obr. 8.4: Obsah Beacon rámce.

8.4 Poškozené a opakované rámce

Na rozdíl od promiskuitního režimu bylo v monitorovacím režimu možné zachytit i poškozené rámce jenž by byly zahozeny. Tyto rámce poškozená data, která zpravidla nelze dekódovat a proto i informace obsažené v těchto rámcích jsou buď nečitelné, nebo nepravdivé. Pro vyfiltrování poškozených rámců byl použit filtr uvedený ve výpisu 8.2. Část obsahu poškozeného rámce je zobrazena na obrázku 8.5.

Výpis 8.2: Filtrování poškozených rámců.

```
_ws.malformed
```

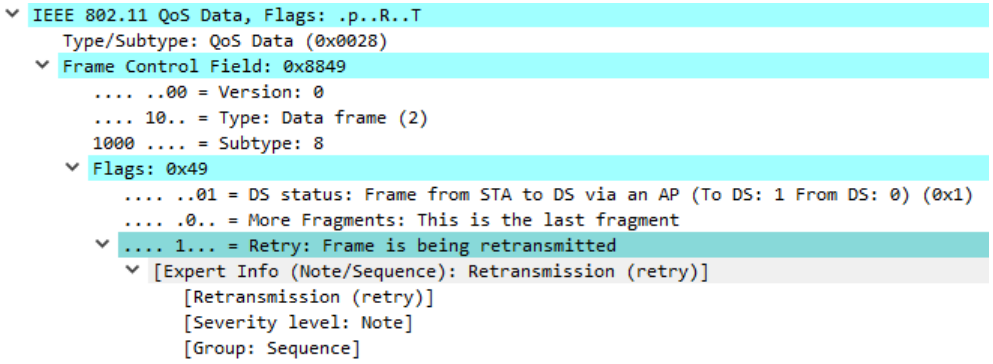


Obr. 8.5: Poškozený rámec.

Je vidět, že Wireshark automaticky detekoval a označil hlavičku, ve které došlo k poškození dat. Dále byly zachyceny i opakované rámce, tedy rámce u nichž bylo od cílové stanice vyžádáno jejich znovu zaslání. Pro zobrazení opakovaných rámců slouží následující filtr. Rámec s vyznačenou informací o opakovaní je zobrazen na obrázku 8.6.

Výpis 8.3: Filtrování opakovaných rámců

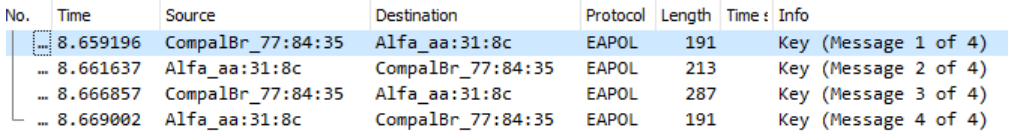
```
wlan.fc.retry == 1
```



Obr. 8.6: Opakovaný rámec.

8.5 Zachycení autentizace a dešifrování rámců

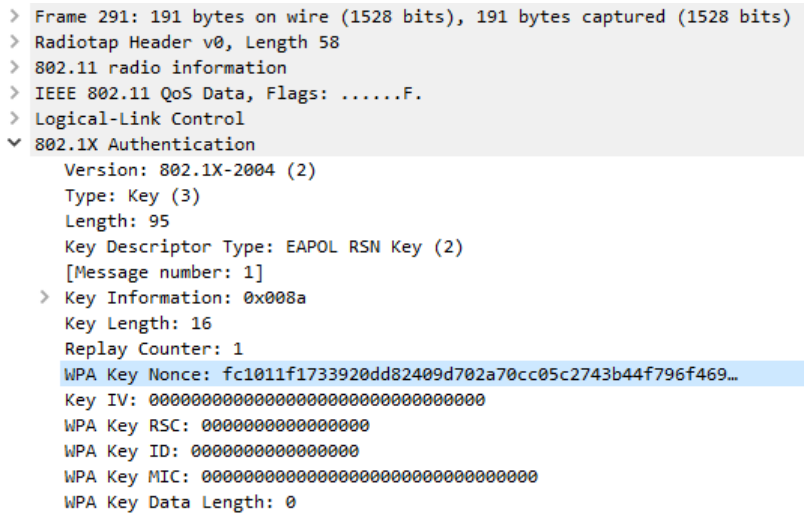
Při asociaci klienta k Wi-Fi sítí je provedena autentizace a výměna klíčů pomocí protokolu EAPOL (Extensible Authentication Protocol over LAN). Tomuto procesu se říká čtyřcestný handshake („4-way handshake“), V monitorovacím režimu bylo možné handshake zachytit a analyzovat.



No.	Time	Source	Destination	Protocol	Length	Time	Info
…	8.659196	CompalBr_77:84:35	Alfa_aa:31:8c	EAPOL	191		Key (Message 1 of 4)
…	8.661637	Alfa_aa:31:8c	CompalBr_77:84:35	EAPOL	213		Key (Message 2 of 4)
…	8.666857	CompalBr_77:84:35	Alfa_aa:31:8c	EAPOL	287		Key (Message 3 of 4)
…	8.669002	Alfa_aa:31:8c	CompalBr_77:84:35	EAPOL	191		Key (Message 4 of 4)

Obr. 8.7: Zachycený čtyřcestný handshake.

V první zprávě posílá přístupový bod klinetovi EAPOL rámec s „ANonce“ (Access Point nonce), což je náhodné číslo (viz. obrázek 8.8). Na základě tohoto čísla, MAC adresy přístupového bodu a a vlastní MAC adresy může klient vytvořit klíč PTK (Pairwise Transient Key). Tento klíč slouží ke generování dalších klíčů. [74]



```
> Frame 291: 191 bytes on wire (1528 bits), 191 bytes captured (1528 bits)
> Radiotap Header v0, Length 58
> 802.11 radio information
> IEEE 802.11 QoS Data, Flags: ......F.
> Logical-Link Control
v 802.1X Authentication
    Version: 802.1X-2004 (2)
    Type: Key (3)
    Length: 95
    Key Descriptor Type: EAPOL RSN Key (2)
    [Message number: 1]
  > Key Information: 0x008a
    Key Length: 16
    Replay Counter: 1
    WPA Key Nonce: fc1011f1733920dd82409d702a70cc05c2743b44f796f469…
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000
    WPA Key MIC: 00000000000000000000000000000000
    WPA Key Data Length: 0
```

Obr. 8.8: Obsah prvního EAPOL rámce.

Druhou zprávu posílá klient a obsahuje „SNonce“ (Station nonce), číslo které potřebuje přístupový bod pro vytvoření PTK. Zároveň tato zpráva obsahuje kontrolní hash MIC (Message integrity check) pro kontrolu, zda nedošlo k poškození nebo modifikaci zprávy (viz. obrázek 8.9). [74]. Na obrázku 8.10 zachycující třetí zprávu, je vidět, že většina polí se změnila ze stavu „Not set“ na „Set“.

```
> Frame 294: 213 bytes on wire (1704 bits), 213 bytes captured (1704 bits)
> Radiotap Header v0, Length 58
> 802.11 radio information
> IEEE 802.11 QoS Data, Flags: .......T
> Logical-Link Control
v 802.1X Authentication
    Version: 802.1X-2001 (1)
    Type: Key (3)
    Length: 117
    Key Descriptor Type: EAPOL RSN Key (2)
    [Message number: 2]
  v Key Information: 0x010a
      .... .... .... .010 = Key Descriptor Version: AES Cipher, HMAC-SHA1 MIC (2)
      .... .... .... 1... = Key Type: Pairwise Key
      .... .... ..00 .... = Key Index: 0
      .... .... .0.. .... = Install: Not set
      .... .... 0... .... = Key ACK: Not set
      .... ...1 .... .... = Key MIC: Set
      .... ..0. .... .... = Secure: Not set
      .... .0.. .... .... = Error: Not set
      .... 0... .... .... = Request: Not set
      ...0 .... .... .... = Encrypted Key Data: Not set
      ..0. .... .... .... = SMK Message: Not set
    Key Length: 0
    Replay Counter: 1
    WPA Key Nonce: f424e8a30f021db58753fb3b579418494e477ec8b6ec38d5…
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000
    WPA Key MIC: 5c7e8d86a308c10c65400fb056d87b8e
    WPA Key Data Length: 22
  > WPA Key Data: 30140100000fac020100000fac040100000fac028000
```

Obr. 8.9: Obsah druhého EAPOL rámce.

```
> Frame 296: 287 bytes on wire (2296 bits), 287 bytes captured (2296 bits)
> Radiotap Header v0, Length 58
> 802.11 radio information
> IEEE 802.11 QoS Data, Flags: ......F.
> Logical-Link Control
v 802.1X Authentication
    Version: 802.1X-2004 (2)
    Type: Key (3)
    Length: 191
    Key Descriptor Type: EAPOL RSN Key (2)
    [Message number: 3]
  v Key Information: 0x13ca
      .... .... .... .010 = Key Descriptor Version: AES Cipher, HMAC-SHA1 MIC (2)
      .... .... .... 1... = Key Type: Pairwise Key
      .... .... ..00 .... = Key Index: 0
      .... .... .1.. .... = Install: Set
      .... .... 1... .... = Key ACK: Set
      .... ...1 .... .... = Key MIC: Set
      .... ..1. .... .... = Secure: Set
      .... .0.. .... .... = Error: Not set
      .... 0... .... .... = Request: Not set
      ...1 .... .... .... = Encrypted Key Data: Set
      ..0. .... .... .... = SMK Message: Not set
    Key Length: 16
    Replay Counter: 2
    WPA Key Nonce: fc1011f1733920dd82409d702a70cc05c2743b44f796f469…
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000
    WPA Key MIC: b10c38e8ce6ae68568e809a22b110a45
    WPA Key Data Length: 96
    WPA Key Data: 51c67f182e022bd042188c49309a470d8ef2a7c0edef9d73…
```

Obr. 8.10: Obsah třetího EAPOL rámce.

Třetí zpráva je odpovědí od přístupového bodu a obsahuje informaci o vytvoření klíče PTK, dále klíč GTK (Group Temporal Key) vytvořený na základě údajů v předešlé zprávě a opět MIC pro kontrolu. Čtvrtou a poslední zprávu posílá klient a potvrzuje v ní, že byly přijaty a instalovány všechny klíče. [74]

Po zachycení všech čtyř zpráv, znalosti SSID a hesla Wi-Fi sítě, je možné dešifrovat rámce vysílané klientem, jehož handshake byl zachycen. Pro dešifrování byl použit Wireshark, ve kterém byla povolena možnost dešifrování IEEE 802.11 rámců a bylo zadáno SSID a heslo Wi-Fi sítě. Dešifrované rámce pak vypadají stejně, jako by byly zachyceny v promiskuitním režimu, ale obsahují navíc původní hlavičku 802.11. Bylo možné vidět i obsah síťové a transportní vrstvy (viz. obrázek 8.11).

```
… 31.083535                    LiteonTe_1d:07:31   802.11   1634      QoS Data, SN=352, FN=0, Flags=.p....F.
… 31.083536 LiteonTe_1d:07:31 … CompalBr_77:84:35 … 802.11     86      802.11 Block Ack, Flags=........
… 31.083604 Alfa_aa:31:8c (00:… CompalBr_77:84:35 … 802.11     74      Request-to-send, Flags=........
… 31.083627                    Alfa_aa:31:8c (00:… 802.11     68      Clear-to-send, Flags=........
… 31.084620 192.168.0.199      192.168.0.234       TCP      1608 0.0… [TCP Previous segment not captured] 53
… 31.084622 192.168.0.199      192.168.0.234       TCP      1608 0.0… 53620 → 445 [ACK] Seq=4145429 Ack=2987
… 31.084624 192.168.0.199      192.168.0.234       TCP      1608 0.0… 53620 → 445 [ACK] Seq=4146889 Ack=2987
… 31.084627 192.168.0.199      192.168.0.234       TCP      1608 0.0… 53620 → 445 [ACK] Seq=4148349 Ack=2987
… 31.084629 192.168.0.199      192.168.0.234       TCP      1608 0.0… 53620 → 445 [PSH, ACK] Seq=4149809 Ack
… 31.084631 192.168.0.199      192.168.0.234       TCP      1608 0.0… 53620 → 445 [ACK] Seq=4151269 Ack=2987
… 31.084633 192.168.0.199      192.168.0.234       TCP      1608 0.0… 53620 → 445 [ACK] Seq=4152729 Ack=2987
… 31.084635 192.168.0.199      192.168.0.234       TCP      1608 0.0… 53620 → 445 [ACK] Seq=4154189 Ack=2987
Frame 7896: 1608 bytes on wire (12864 bits), 1608 bytes captured (12864 bits)
Radiotap Header v0, Length 58
802.11 radio information
IEEE 802.11 QoS Data, Flags: .p.....T
Logical-Link Control
Internet Protocol Version 4, Src: 192.168.0.199, Dst: 192.168.0.234
   0100 .... = Version: 4
   .... 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
   Total Length: 1500
   Identification: 0xf42f (62511)
 > Flags: 0x4000, Don't fragment
   Fragment offset: 0
   Time to live: 64
   Protocol: TCP (6)
   Header checksum: 0xbdea [validation disabled]
   [Header checksum status: Unverified]
   Source: 192.168.0.199
   Destination: 192.168.0.234
Transmission Control Protocol, Src Port: 53620, Dst Port: 445, Seq: 4145429, Ack: 2987, Len: 1460
   Source Port: 53620
   Destination Port: 445
   [Stream index: 0]
   [TCP Segment Len: 1460]
   Sequence number: 4145429    (relative sequence number)
   Sequence number (raw): 2203685899
   [Next sequence number: 4146889    (relative sequence number)]
   Acknowledgment number: 2987    (relative ack number)
   Acknowledgment number (raw): 1333150959
```

Obr. 8.11: Dešifrovaný rámec IEEE 802.11.

Další možností dešifrování je využití nástroje Airdecap-ng, jenž je součástí balíčku Aircrack-ng [75]. Tomuto nástroji stačí pro dešifrování rámců pouze dvě zprávy EAPOL (zpráva 2 a 3 nebo 3 a 4). Aidecap-ng se spustí příkazem uvedeným níže (výpis 8.4), kde „SSID“ bylo nahrazeno SSID konkrétní Wi-Fi sítě, „heslo“ bylo nahrazeno heslem Wi-Fi sítě, za kterým následuje název PCAP souboru, jenž měl být dešifrován. Rámce, které nebylo možné dešifrovat, byly odstraněny a byl vytvořen nový PCAP souboru s pouze dešifrovanými rámci.

Výpis 8.4: Příkaz pro spuštěný nástroje Airdecap-ng.

```
airdecap-ng -e 'SSID' -p 'heslo' název_pcap_souboru.pcap
```

8.6 Analýza provozu v jazyce Python

Program Wireshark není jedinou možností, jak analyzovat zachycený síťový provoz. Alternativním řešením je využití vlastních skriptů napsaných v jazyce Python. K tomuto účelu slouží balíček **Pyshark**.

Pyshark je balíček pro Python, který umožňuje zachytávat síťový provoz, parsovat rámce a pakety z PCAP souborů a zachytávání v reálném čase. K tomu využívá program T-shark a jeho filtry [80]. Pyshark dále umožňuje i dešifrování rámců 802.11, přičemž zde platí stejné podmínky dešifrování jako při dešifrování ve Wireshark. Výhodou tohoto řešení je, že data z rámců a paketů lze dále zpracovávat dle vlastních potřeb. Vytvořil jsem proto dva malé skripty, jako ukázku možností jeho využití.

8.6.1 Analýza PCAP souboru

Pro ukázku možnosti analýzy PCAP souboru jsem vytvořil skript, jenž jsem nazval „`ChanAnalyze.py`“. Tento skript získá a vynese do grafu nebo terminálu, počet zařízení vysílajících na jednotlivých kanálech. Pro lepší ukázku výsledků jsem použil ještě jeden skript, který provádí „channel hopping“, tedy změnu kanálu Wi-Fi adaptéru. Původně jsem chtěl použít vlastní skript pro změnu kanálu, ale na GitHub jsem objevil skript, jenž činí přesně to, co bylo potřeba pro otestování a demonstraci funkčnosti vlastního skriptu. Skript „`chanhop.sh`“ má licenci GPLv2 a je vystaven na GitHub pod tímto odkazem [81]. Nejdříve byl spuštěn skript „`chanhop.sh`“, který dokola střídal kanály 1 až 13, přičemž na každém setrval po dobu 4 sekund. Během toho byl zachytáván provoz pomocí Tcpdump do souboru.

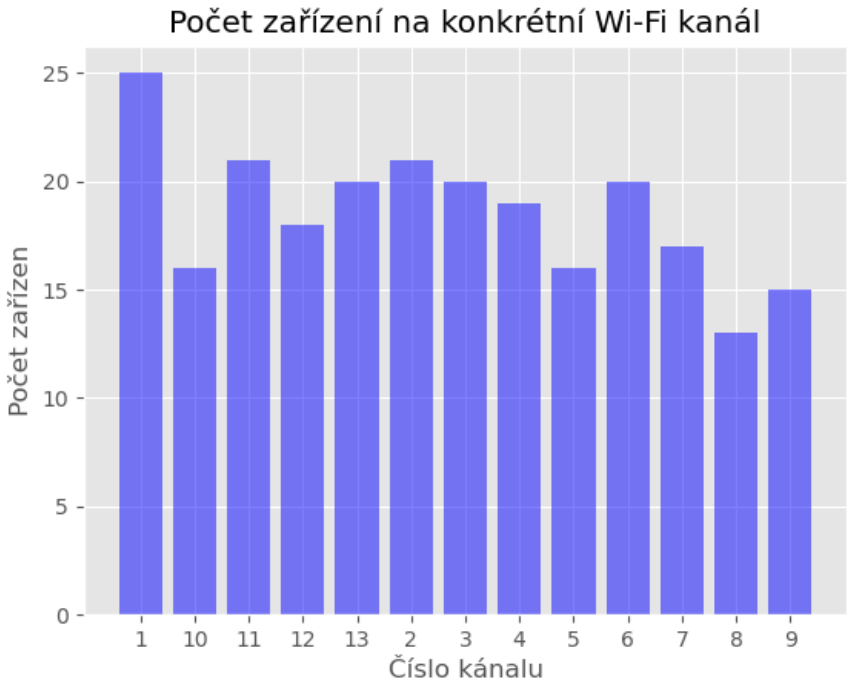
Pro analýzu souboru byl použit skript „`ChanAnalyze.py`“, jenž přijímá tyto argumenty:

- `-f` - Cesta k PCAP souboru, jenž má být analyzován.
- `-t` - Vypíše výsledky do terminálu.

Výpis 8.5: Spuštění skriptu ChanAnalyze.py.

```
zdenek@kali:~python3 ChanAnalyze.py -f /home/zdenek/
   Plocha/channels_test.pcap
Analyzing file /home/zdenek/Plocha/channels_test.pcap..
```

Skript prochází všechny rámce v PCAP souboru a z každého, pomocí filtrů, získá informaci o MAC adrese vysílající stanice a čísle kanálu. Informace o páru MAC adresa + číslo kanálu je uložena do listu, který je následně procházen a zbaven duplicitních párů. Pokud by se tedy v listu již nacházel stejný pár MAC adresa + kanál, je do nového listu uložen pouze jednou a jeho duplicity jsou ignorovány. Dále je list seřazen podle čísel kanálů a poté spočten počet unikátních MAC adres pro každý kanál. Tím bylo docíleno získání počtu zařízení vysílajících na jednotlivých kanálech. Výsledky jsou vyneseny do sloupcového grafu, který je vidět na obrázku 8.12. Graf byl také uložen ve formátu PNG (Portable Network Graphic).



Obr. 8.12: Graf s výsledky analýzy PCAP souboru pomocí skriptu v jazyce Python.

V případě přítomnosti argumentu `-t`, se výsledky vypíší do terminálu v podobě textového horizontálního grafu, jak ukazuje obrázek 8.13. Tato možnost se hodí zejména při analýze na zařízení, jenž nedisponuje grafickým rozhraním, tedy například právě router.

Obr. 8.13: Výsledky analýzy PCAP souboru vypsané textové podobě do terminálu.

Jak je vidět z výsledků uvedených výše, pomocí tohoto skriptu bylo možné určit vytížení jednotlivých kanálů ve 2,4GHz pásmu. **Bylo zjištěno, že v dané lokalitě je nejvytíženějším kanálem, co se počtu zařízení týče, kanál číslo 1**. Zdrojový kód použitého skriptu je uveden v příloze A.1. Obdobně je možné v Pythonu pomocí Pyshark provést analýzu téměř libovolných parametrů přenosu.

8.6.2 Analýza v reálném čase

Pro ukázku analýzy provozu v reálném čase v jazyce Python, jsem vytvořil skript „`LiveCap.py`“. Tento skript spouští zachytávání provozu a rovnou vynáší do grafu, případně terminálu, informaci o typu zachycených rámců a jejich počtu. Nejdříve bylo nutné uvést Wi-Fi adaptér do monitorovacího režimu a poté spustit skript. Skript přijímá následující argumenty:

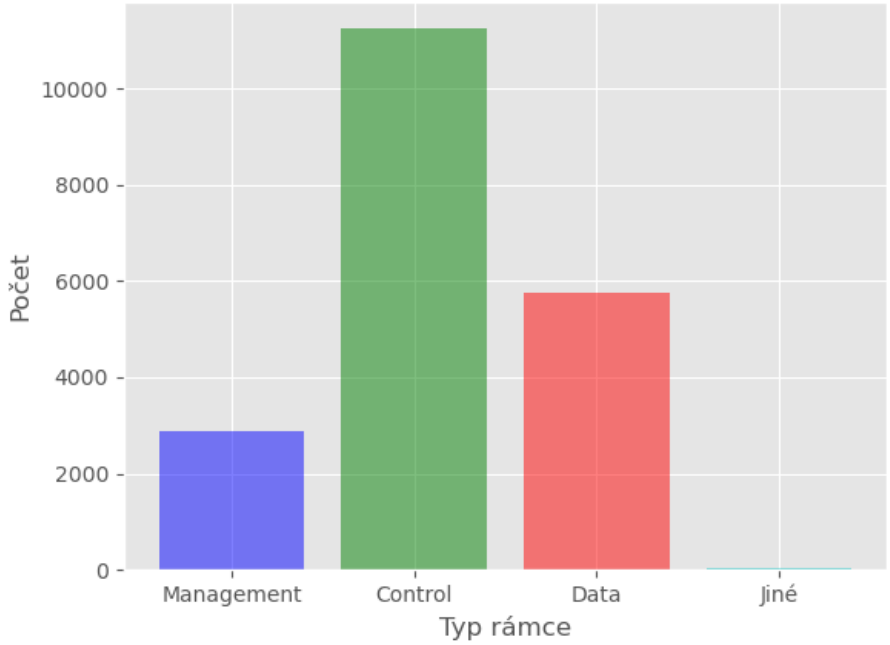
- `-i` - Síťové rozhraní, které má zachytávat provoz.
- `-t` - Vypíše výsledky do terminálu.
- `-mac` - Přepne na analýzu počtu vyslaných rámců připadajících na MAC adresu.

Příkaz pro spuštění skriptu je ve výpisu 8.6 .

Výpis 8.6: Spuštění skriptu LiveCap.py.

```
zdenek@kali:~python3 LiveCap.py -i wlan0
```

Po spuštění skriptu začalo zachytávání provozu a zároveň zobrazování výsledku v grafu, jenž se automaticky aktualizoval. Graf s výsledky je zobrazen na obrázku 8.14. Alternativně bylo možné výsledky nechat vypisovat do terminálu (obrázek 8.15), a to při použití argumentu `-t`. **Pomocí tohoto skriptu bylo možné v reálném čase získat informace o typu přenášených rámců a jejich počtu.** Obdobně je možné analyzovat v reálném čase libovolné parametry provozu. Zdrojový kód tohoto skriptu je uveden v příloze B.1.



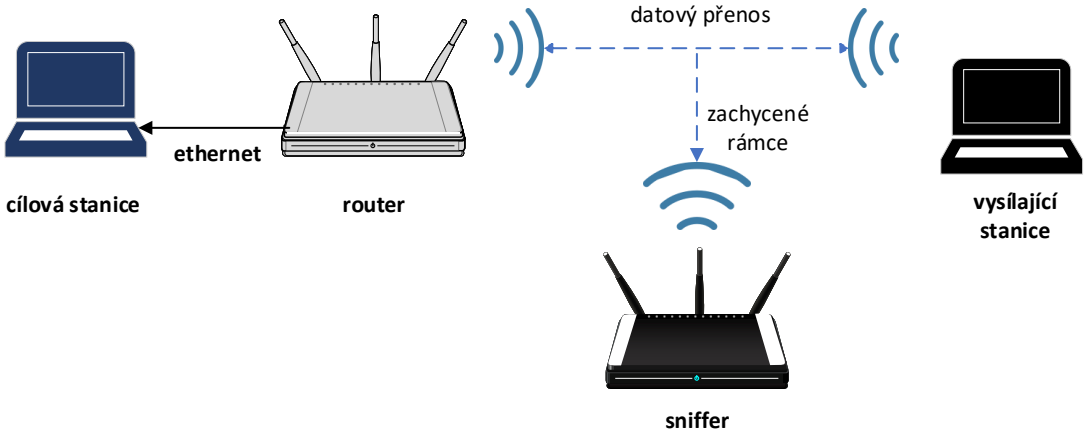
Obr. 8.14: Výsledky analýzy zachytávání v reálném čase.

```
2020-06-04 12:27:09
Management  [ 2871]  ***********
Control     [11241]  ****************************************
Data        [ 5768]  ********************
Jiné        [   28]  *
```

Obr. 8.15: Výsledky analýzy zachytávání v reálném čase zobrazeny v terminálu.

8.7 Analýza spolehlivosti zachytávání dat v monitorovacím režimu

Jak bylo zjištěno v kapitole 6.4, v monitorovacím režimu lze zachytit datové rámce v šifrované podobě. Také bylo zjištěno, že je možné zachytávat rámce vysílané více prostorovými streamy, jak je uvedeno v podkapitole 8.2.2. Další otázkou tedy bylo, zda je možné v monitorovacím režimu zachytit veškerý datový přenos, tedy datové rámce, které jsou přenášeny při přenosu souboru pomocí Wi-Fi. Schéma na obrázku 8.16 ukazuje scénář a zapojení, pro přenos dat mezi stanicemi a jejich zachytávání „snifferem“.



Obr. 8.16: Schéma zapojení scénáře pro zachytávání přenosu dat.

K notebooku, ze kterého byl soubor přenášen, byl připojen USB Wi-Fi adaptér Alfa Awus1900 (viz. kapitola 6.1). Soubor byl kopírován do sdílené složky, která byla umístěna v druhém notebooku, jenž byl do lokální sítě připojen pomocí ethernetového kabelu přes Wi-Fi router. Specifikace použitého Wi-Fi routeru jsou k nalezení zde [82]. Vysílající stanice, v tomto případě notebook s Alfa Awus1900, posílal soubor přes Wi-Fi do routeru, jenž data přeposílal přes ethernet do cílové stanice, tedy druhého notebooku. Po celou dobu přenosu bylo prováděno zachytávání provozu „snifferem“, kterým byl Turris Omnia (viz. 3.1). Ten byl nastaven pro naslouchání na stejném kanálu i jeho šířce, jako síť distribuovaná Wi-Fi routerem. Parametry Wi-Fi sítě byly: frekvence kanálu 5180 MHz (kanál č. 36), šířka kanálu 80 MHz a středová frekvence 5210 MHz (kanál č. 42). Pro kontrolu nastavení byl použit nástroj `iw`, tak jak je vidět ve výpisu 8.7, kde poslední řádek zobrazuje parametry nastaveného kanálu. Mimo jiné je zde na šestém řádku vidět, že Wi-Fi adaptér je v monitorovacím režimu.

Výpis 8.7: Kontrola nastavení síťového rozhraní na Turris Omnia.

```
root@turris:~# iw dev wlan0 info
Interface wlan0
        ifindex 14
        wdev 0x1
        addr 04:f0:21:42:11:03
        type monitor
        wiphy 0
  channel 36 (5180 MHz), width: 80 MHz, center1: 5210 MHz
```

Kromě zachytávání v monitorovacím režimu bylo během přenosu, spuštěno také zachytávání v promiskuitním režimu na Wi-Fi adaptéru Alfa Awus1900. Důvodem byla možnost porovnání skutečně odeslaných dat s daty zachycenými v monitorovacím režimu. Provoz zachycený tomto režimu tedy sloužil jako referenční. Přenášeným souborem bylo videosoubor o velikosti 77,37 MB (megabajtů). Vysílající stanice (notebook s Alfa Awus1900) byl umístěn v jiné místnosti než Wi-Fi router, který data přijímal a přeposílal do cílového notebooku. „Sniffer“, tedy Turris Omnia, byl umístěn zhruba metr od vysílajícího adaptéru. Všechna použitá zařízení disponovala vlastností vysílat a přijímat až tři prostorové streamy. Nemohlo tedy dojít k tomu, že by bylo použito více než tří streamů, které by „sniffer“ nebyl schopen zachytit. Na straně „snifferu“ byl pro zachytávání použit program TCPdump a Wireshark na straně vysílající stanice. Výsledky byly ukládány do PCAP souborů. Po ukončení přenosu byla obě zachytávání ihned ukončena.

Vzhledem k tomu, že v monitorovacím režimu byly rámce zašifrované a v promiskuitním nikoliv, ukázalo se obtížné navzájem zachycený provoz porovnat. Prvně zvažovaným postupem pro analýzu bylo dešifrování zachycených 802.11 rámců, vyfiltrování pouze datových paketů pocházejících od vysílající stanice a adresovaných do cílové. Následně porovnat tyto pakety s pakety z promiskuitního režimu podle identifikátoru „IP.ID“. Tento identifikátor je 16-bitová hodnota, která je unikátní pro daný datagram s danou zdrojovou adresou, cílovou adresou a protokol [83]. Ukázalo se však, že některé rámce se nepodařilo dešifrovat a také, že hodnota 16-bitového identifikátoru IP.ID není dostačující a po přetečení se začala opakovat. To znemožnilo analýzu, která by přesně určila, jaké pakety se podařilo zachytit a jaké naopak chybí.

Rozhodl jsem se proto pro jednodušší, ale dostačující analýzu. Porovnal velikost skutečně vyslaných a zachycených dat. Nejdříve byly v PCAP souboru z monitorovacího režimu vyfiltrovány pouze datové rámce, jenž měly zdrojovou MAC adresu vysílajícího adaptéru, cílovou MAC adresu adaptéru cílového notebooku a velikost přenášených dat větší než 1000 bajtů. Dále byly z PCAP souboru z promiskuit-

ního režimu vyfiltrovány pakety s protokolem TCP, jenž nesly IP adresu zdrojového adaptéru, IP adresu cílového adaptéru a zároveň jejich „payload“ byl větší než 1000 bajtů. K tomuto filtrováni byl použit program Wireshark s aplikovanými filtry, jenž jsou uvedeny ve výpisu 8.8 a 8.9.

Výpis 8.8: Filtr aplikovaný na rámce z monitorovacího režimu v programu Wireshark.

```
(wlan.sa == 00:c0:ca:aa:31:8c and wlan.da == 2c:60:0c:eb:94:e3)
   and frame.len > 1000 and wlan.fc.type_subtype == 40
```

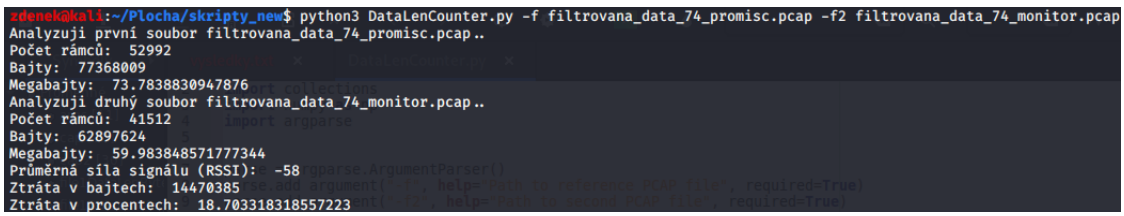
Výpis 8.9: Filtr aplikovaný na rámce z promiskuitního režimu v programu Wireshark.

```
(ip.src == 192.168.0.108 and ip.dst == 192.168.0.171) and frame.
   len > 1000 and tcp
```

Vyznám filtrů:

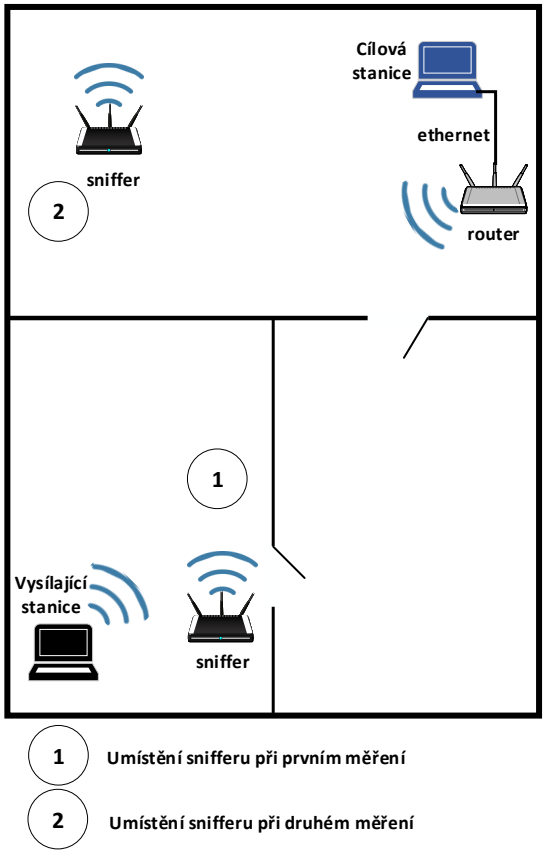
- **wlan.sa** - Zdrojová MAC adresa.
- **wlan.da** - Cílová MAC adresa.
- **frame.len** - Velikost rámce.
- **wlan.fc.type_subtype** - Identifikace typu rámce.
- **ip.src** - Zdrojová IP adresa.
- **ip.dst** - Cílová IP adresa.
- **tcp** - Pouze protokol TCP.

V Pythonu jsem vytvořil skript „`DataLenCounter.py`“, který rámce a pakety v obou filtrovaných PCAP souborech prošel, zjistil velikost přenášených dat v každém rámci a paketu a sečetl je. Data z promiskuitního režimu byla vzata jako referenční. Do konzole pak skript vypsal, celkovou velikost dat, jenž byla zachycena v obou režimech, v bajtech i megabajtech a vypočítal ztrátovost v bajtech a v procentech. Výstup skriptu je zobrazen na obrázku 8.17. Skript je uveden v příloze C.1.



Obr. 8.17: Výstup ze skriptu DataLenCounter.py.

Dále také vypsal průměrnou hodnotu síly přijatých signálů, tedy RSSI („Received Signal Strength Indication“) v dBm (decibel vztažený na 1 miliwatt), která činila -58 dBm. Bylo změřeno, že referenčním PCAP souboru bylo 73,78 MB dat, zatímco v monitorovacím režimu bylo zachyceno 59,98 MB dat. Ztráta tedy činila 18,7%. Ačkoliv se tedy „sniffer“ nacházel ve stejné místnosti jako vysílající stanice a jen metr od něj, nepodařilo se zachytit veškerá přenášená data souboru. Zajímal mě tedy vliv vzdálenosti a úrovně RSSI na ztrátovost. Bylo provedeno ještě jedno stejné měření, kdy však „sniffer“ byl umístěn v jiné místnosti za dvěma zdmi a zhruba 3 metry od Wi-Fi routeru. Scénáře umístění „snifferu“ zobrazuje obrázek 8.18.



Obr. 8.18: Umístění „snifferu“ při prvním a druhém měření.

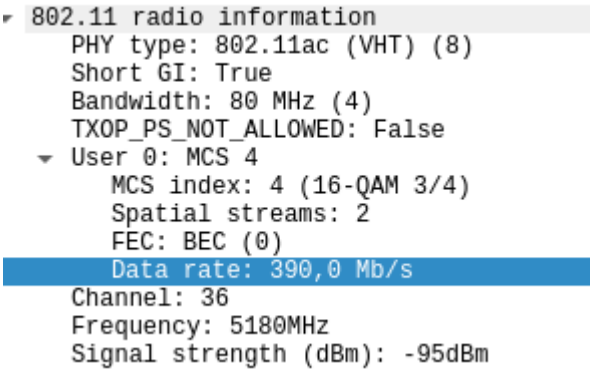
Ukázalo se, že v druhém případě spolehlivost zachytávání prudce klesla. Ztráta zachycených dat v monitorovacím režimu činila 95,78% a zachyceno bylo pouze 3,14 MB dat. Klesla také průměrná úroveň RSSI a to na -94 dBm. Z toho bylo možné usoudit, že pro co nejspolehlivější zachytávání přenosu je potřebné umístit zachytávající zařízení co nejblíže k vysílači, případně přímo mezi dvě komunikující stanice tak, aby úroveň přijatého signálu byla co nejlepší.

Vliv na výsledky mohl mít Beamforming, ale také citlivost Wi-Fi adaptéru Turrisu Omnia, potažmo jeho antén a jejich poloha. Výsledky z obou měření jsou zobrazeny v tabulce 8.1.

Tab. 8.1: Výsledky analýzy spolehlivosti zachytávání přenosu dat.

	Výsledky 1. měření	**Výsledky 2. měření**
Velikost skutečně přenášených dat [MB]	73,78	74,41
Počet skutečně přenesených rámců:	52992	53438
Počet zachycených rámců:	41512	2173
Průměrná úroveň RSSI [dBm]:	-58	-94
Zachycená data [MB]:	59,98	3,14
Ztrátovost dat [%]:	18,70	95,78

Ani v jednom z případů se tedy nepodařilo zachytit veškerý datový přenos, který probíhal při přenosu souboru přes Wi-Fi. Dle očekávání byl lepší výsledek zaznamenám při umístění „snifferu“ blíže k vysílači, kde ztráta dat činila 18,70%. Pro odposlouchávání datové komunikace tedy velmi záleží na umístění zařízení zachytávajícího provoz. Ukázalo se také, že porovnat rámce 802.11 zachycené v monitorovacím režimu s těmi v promiskuitním, není snadné kvůli šifrování rámců 802.11. Zajímavé je, že ačkoliv všechna použitá zařízení podporují až tři prostorové streamy, pro přenos byly použity pouze dva, jak mimo jiné ukazuje obrázek 8.19.



Obr. 8.19: Parametry přenosu souboru získané z pole VHT.

Závěr

V této práci bylo uskutečněno praktické seznámení s problematikou zachytávání a analýzy provozu bezdrátových Wi-Fi síti. Cílem bylo vybrat vhodný hardware a software, na kterém byly prozkoumány možnosti zachytávání a analýzy bezdrátového provozu Wi-Fi sítí. Tento cíl se podařilo naplnit. Byly porovnávány open-source firmwary určené pro routery. Zde bylo zjištěno, že většina těchto firmwarů nabízí jen malou podporu kompatibilních zařízení. Vybrán byl proto firmware OpenWrt, který nabízel nejširší podporu zařízení a zároveň nejvíce splňoval požadavky pro zachytávání provozu Wi-Fi sítí. Těmito požadavky byla zejména konfigurovatelnost a možnost rozšíření o nástroje, jenž umožňují zachytávání provozu.

Dále bylo provedeno srovnání Wi-Fi routerů a výběr takového, který nejlépe splňoval požadavky pro provoz open-source firmware a nástrojů pro zachytávání provozu. Již během tohoto výběru bylo zjištěno, že velmi záleží na použitém Wi-Fi chipsetu. Ne každý Wi-Fi chipset totiž podporuje režimy, jako je promiskuitní a monitorovací režim, které jsou potřebné pro zachytávání provozu Wi-Fi. Ukázalo se, že jistit informace o podpoře těchto režimů není vůbec snadné. Zároveň také bylo zjištěno, že ačkoliv Wi-Fi chipset daného zařízení tyto režimy podporuje, tak velmi záleží na použitém ovladači tohoto chipsetu. Pokud nejsou v ovladači tyto režimy implementovány, nelze na zařízení zachytávat bezdrátový Wi-Fi provoz, ačkoliv by to hardware umožňoval. Byl proto vybrán Wi-Fi router Turris Omnia, který nejlépe splňoval všechny požadavky pro zachytávání provozu. Navíc tento router využívá firmware TurrisOS, který je jen upravenou verzí OpenWrt a je na něm založen. Popsány a porovnány byly také open-source nástroje umožňující zachytávání a analýzu Wi-Fi provozu. Vybrány byly takové nástroje, které byly kompatibilní s vybraným firmware.

Po výběru vhodného hardware, firmware a nástrojů bylo zrealizováno prvotní zachytávání provozu, na kterém byly demonstrovány rozdíly mezi promiskuitním a monitorovacím režimem. Bylo zjištěno, že pro zachytávání v promiskuitním režimu je nutné, aby bylo zařízení vykonávající zachytávání, asociováno k Wi-Fi síti. Zároveň se potvrdilo, že v tomto režimu zachytává Wi-Fi adaptér i rámce, které nejsou adresovány pro něj. Neprovádí se tedy filtrování na základě MAC adres, ale pouze na základě SSID. V promiskuitním režimu, také dochází k nahrazení původních hlaviček 802.11, hlavičkami ethernetovými. Tím dochází ke ztrátě informací o přenosu na fyzické vrstvě. V promiskuitním režimu byl však zachycených rámcích vidět obsah vyšších vrstvy TCP/IP. Oproti tomu v monitorovacím režimu byly rámce šifrované, a tím pádem nebyly vyšší vrstvy viditelné. Důvodem je, že v monitorovacím režimu není zachytávající zařízení asociováno k síti. Z rámců v monitorovacím režimu bylo možné zjistit informace o parametrech přenosu na fyzické vrstvě. Podařilo se tak

například zjistit, jakou rychlostí byly rámce přenášeny, jaký byl použit standard 802.11, ale také kolika prostorovými streamy byl rámec přenášen.

Dále byly prozkoumány možnosti zachytávání v monitorovacím režimu, to konkrétně možnost vzdáleného zachytávání. Konkrétně popsány dvě možnosti vzdáleného zachytávání a to pomocí programů „SSHdump“ a „RPCAPd“. Poslední kapitoly se věnovaly analýze zachyceného provozu. Prozkoumáno bylo jaké rámce je možné v monitorovacím režimu zachytit. Byla provedena analýza parametrů přenosu na fyzické vrstvě. Zde bylo kupříkladu zjištěno, že pouze datové rámce jsou přenášeny za pomocí více prostorových streamů. Ostatní typy rámců se přenášejí na základních rychlostech, bez využití prostorových streamů, a to z důvodů zpětné kompatibility. Dále bylo předvedeno, ze je možné zachytit tzv. čtyřcestný handshake, na jehož základě lze rámce 802.11 dešifrovat. Následně byly popsány možnosti analýzy provozu za pomocí programovacího jazyka Python. Jako ukázka byly vytvořeny dva skripty, jenž umožňují analýzu zachyceného provozu ze souboru a v reálném čase. V úplně poslední část se věnovala spolehlivosti zachytávání dat. Bylo provedeno zachytávání přenosu souboru, které bylo následně analyzováno. Bylo zjištěno, že nebylo možné zachytit veškerá přenášená data a to i přesto, že zachytávající zařízení se nacházelo v blízkosti vysílače. Při větší vzdálenosti od vysílače byla ztráta ještě vyšší.

Podařilo se tedy splnit všechny stanovené cíle této práce a zjistit mnoho nových poznatků o možnostech zachytávání a analýzy provozu Wi-Fi sítí.

Literatura

[1] Wi-Fi at 20: The internet-s most important tech is about to grow up. *Android authority* [online]. 2019 [cit. 2019-11-03]. Dostupné z: https://www.androidauthority.com/wi-fi-anniversary-1001274/

[2] PETERKA, Jiří. Počítačové sítě, verze 4.0. *EArchiv: Archiv článků a přednášek Jiřího Peterky* [online]. c2015 [cit. 2019-11-10]. Dostupné z: https://www.earchiv.cz/l226/slide.php3?l=16&me=20

[3] WALKE, Bernhard H., Stefan MANGOLD a Lars BERLEMANN. *IEEE 802 Wirless Systems: Protocols, Multi-hop Mesh/Relaying, Performance and Spectrum Coexistnece*. Chichester: Wiley, c2006. ISBN 139780470014394.

[4] Who We Are: History. *Wi-Fi Alliance* [online]. Austin: Wi-Fi Alliance, c2020 [cit. 2020-06-07]. Dostupné z: https://www.wi-fi.org/who-we-are/history

[5] GRIFFITH, Eric. WECA becomes Wi-Fi Alliance. *Internet News: RealTime IT News* [online]. c2020, 2.9.2002 [cit. 2020-06-07]. Dostupné z: http://www.internetnews.com/wireless/article.php/1474361/WECA-becomes-Wi-Fi-Alliance.htm

[6] WESTCOTT, David A., David. D COLEMAN, Peter MACKENZIE a Ben MILLER. *Certified Wireless Analysis Professional: Official Study Guide (Exam PW0-270)* [online]. Indianapolis, Indiana: Wiley Publishing, c2011. ISBN 978-0-470-76903-4.

[7] Využívání vymezených rádiových kmitočtů. *Český telekomunikační úřad* [online]. c2018 [cit. 2019-11-13]. Dostupné z: https://www.ctu.cz/vyuzivani-vymezenych-radiovych-kmitoctu

[8] PUŽMANOVÁ, Rita. *Moderní komunikační sítě od A do Z: [technologie pro datovou, hlasovou i multimediální komunikaci]*. 2., aktualiz. vyd. Brno: Computer Press, 2006. ISBN 80-251-1278-0.

[9] KRUMBEIN, Adam. Understanding the Basics of MIMO Communication Technology. *RMFM* [online]. San Diego, (CA): Southwest Antennas, c2016 [cit. 2020-06-05]. Dostupné z: https://www.rfmw.com/data/swa-mimo-basics.pdf

[10] LEA, Perry. *IoT and Edge Computing for Architects: Implementing edge and IoT systems from sensors to clouds with communication systems, analytics, and security*. 2nd ed. Birmingham: Packt Publishing, 2020. ISBN 1839218878.

[11] DURČÍK, Václav. *Monitoring a analýza provozu bezdrátových sítí* [online]. Ostrava, 2015 [cit. 2019-11-14]. Dostupné z: http://hdl.handle.net/10084/108831. Bakalářská práce. Vysoká škola báňská - Technická univerzita Ostrava

[12] MU-MIMO vs SU-MIMO Wi-Fi. *TechGenix* [online]. Valletta: TechGenix Malta Limited C/O Ganado Services Limited, 2015, 28.9.2015 [cit. 2019-11-17]. Dostupné z: http://techgenix.com/mu-mimo-vs-su-mimo-wi-fi/

[13] EDWARDS, Robert. 802.11ac wireless: Channel Bonding, MIMO, Spatial Streams, and Beamforming. *Source One Technology* [online]. 2016, 26.8.2016 [cit. 2020-05-24]. Dostupné z: https://www.sourceonetechnology.com/802-11ac-wireless-channel-bonding-mimo-spatial-streams-and-beamforming/

[14] FRUHLINGER, Josh. Beamforming explained: How it makes wireless communication faster. *Networkworld* [online]. Los Angeles, 2019, 15.10.2019 [cit. 2019-11-17]. Dostupné z: https://www.networkworld.com/article/3445039/beamforming-explained-how-it-makes-wireless-communication-faster.html

[15] Understanding IEEE* 802.11 Authentication and Association. *Intel* [online]. 2019, 25.3.2019 [cit. 2019-11-25]. Dostupné z: https://www.intel.com/content/www/us/en/support/articles/000006508/network-and-io/wireless-networking.html

[16] WLAN MAC protocol | WLAN MAC frame format | 802.11 wifi MAC. *RF Wireless World* [online]. c2012 [cit. 2019-11-25]. Dostupné z: https://www.rfwireless-world.com/Articles/WLAN-MAC-layer-protocol.html

[17] IEEE 802.11 Mac Frame. *GeeksforGeeks: A computer science portal for geeks* [online]. Noida (Uttar Pradesh) [cit. 2019-11-25]. Dostupné z: https://www.geeksforgeeks.org/ieee-802-11-mac-frame/

[18] WLAN Probe Request Frame: WLAN Probe Response Frame. *RF Wireless World* [online]. c2012 [cit. 2019-11-26]. Dostupné z: https://www.rfwireless-world.com/Terminology/WLAN-probe-request-and-response-frame.html

[19] PETERKA, Jiří. Počítačové sítě, verze 4.0: Zabezpečení sítí dle 802.11. *EArchiv: Archiv článků a přednášek Jiřího Peterky* [online]. 2015 [cit. 2019-12-03]. Dostupné z: https://www.earchiv.cz/l226/slide.php3?l=17&me=25

[20] PETERKA, Jiří. Počítačové sítě, verze 4.0: IEEE 802.11i. *EArchiv: Archiv článků a přednášek Jiřího Peterky* [online]. 2015 [cit. 2019-12-03]. Dostupné z: https://www.earchiv.cz/l226/slide.php3?l=17&me=26

[21] PETERKA, Jiří. Počítačové sítě, verze 4.0: autentizace dle IEEE 802.11i. *EArchiv: Archiv článků a přednášek Jiřího Peterky* [online]. 2015 [cit. 2019-12-03]. Dostupné z: https://www.earchiv.cz/l226/slide.php3?l=17&me=29

[22] ŠKODÁK, Jaroslav. *Zabezpečení bezdrátových sítí IEEE 802.11* [online]. Brno, 2008 [cit. 2019-12-03]. Dostupné z: http://hdl.handle.net/11012/18797. Diplomová práce. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací. Vedoucí práce Martin Koutný.

[23] PETERKA, Jiří. Počítačové sítě, verze 4.0: přenos dat v IEEE 802.11b. *EArchiv: Archiv článků a přednášek Jiřího Peterky* [online]. 2015 [cit. 2019-12-04]. Dostupné z: https://www.earchiv.cz/l226/slide.php3?l=16&me=23

[24] PETERKA, Jiří. Počítačové sítě, verze 4.0: Standard 802.11n. *EArchiv: Archiv článků a přednášek Jiřího Peterky* [online]. 2015 [cit. 2019-12-04]. Dostupné z: https://www.earchiv.cz/l226/slide.php3?l=16&me=30

[25] IEEE 802.11ax Wi-Fi. *Electronics notes* [online]. 2019 [cit. 2019-12-04]. Dostupné z: https://www.electronics-notes.com/articles/connectivity/wifi-ieee-802-11/802-11ax.php

[26] FISHER, Tim. What Is Firmware?: A definition of firmware and how firmware updates work. *Life Wire* [online]. New York: Life Wire, c2020, 28.2.2020 [cit. 2020-06-08]. Dostupné z: https://www.lifewire.com/what-is-firmware-2625881

[27] The Open Source Definition. *Open Source Initiative* [online]. Palo Alto, 2007, 22.3.2007 [cit. 2019-11-28].
Dostupné z: https://opensource.org/docs/osd

[28] What is DD-WRT? *DD-WRT* [online]. 2019, 15.4.2019 [cit. 2019-12-01]. Dostupné z: https://wiki.dd-wrt.com/wiki/index.php/What_is_DD-WRT%3F#Which_build_to_flash_on_my_router

[29] What is DD-WRT? *DD-WRT* [online]. c2020, 1.6.2020 [cit. 2020-06-08]. Dostupné z: https://forum.dd-wrt.com/wiki/index.php/What_is_DD-WRT%3F

[30] Router Database. *DD-WRT* [online]. c2019 [cit. 2019-12-01]. Dostupné z: https://dd-wrt.com/support/router-database/

[31] Web interface. *DD-WRT* [online]. 2017, 13.11.2017 [cit. 2019-12-01]. Dostupné z: https://wiki.dd-wrt.com/wiki/index.php/Web_Interface

[32] Telnet/SSH and the command line. *DD-WRT* [online]. 2018, 18.6.2018 [cit. 2019-12-01]. Dostupné z: https://wiki.dd-wrt.com/wiki/index.php/Telnet/SSH_and_the_Command_Line#SSH

[33] Tutorials. *DD-WRT* [online]. 2019, 27.2.2019 [cit. 2019-12-01]. Dostupné z: https://forum.dd-wrt.com/wiki/index.php/Tutorials

[34] OpenWrt Project. *OpenWrt: Welcome to the OpenWrt Project* [online]. c2020, 12.5.2020 [cit. 2020-06-08]. Dostupné z: https://openwrt.org/

[35] *Google Trends: Porovnání* [online]. Mountain View, California: Google, 2020 [cit. 2020-05-27]. Dostupné z: https://trends.google.cz/trends/explore?q=openwrt,dd-wrt

[36] User Guide. *OpenWrt: Wireless Freedom* [online]. 2019, 10.7.2019 [cit. 2019-12-02]. Dostupné z: https://openwrt.org/docs/guide-user/start

[37] Tomato Firmware. *Polar Cloud* [online]. [cit. 2020-06-08]. Dostupné z: https://www.polarcloud.com/tomato

[38] Router List. *Tomato by Shibby* [online]. c2012-2017 [cit. 2020-06-08]. Dostupné z: https://tomato.groov.pl/?page_id=69

[39] Features. *Fresh Tomato* [online]. c2020 [cit. 2020-06-08]. Dostupné z: https://freshtomato.org/features.html

[40] Více než jen router. Open source centrum vašeho domova: Vlastnosti. *Turris* [online]. [cit. 2019-12-07]. Dostupné z: https://www.turris.cz/cs/omnia/#vlastnosti

[41] Turris Omnia. In: *Wikidevi* [online]. San Francisco (CA): Wikimedia Foundation, 2019, 5.7.2019 [cit. 2019-12-07]. Dostupné z: https://wikidevi.wi-cat.ru/index.php/Turris_Omnia

[42] Linksys WRT1900AC v2. In: *Info Depot Wiki* [online]. San Francisco (CA): Wikimedia Foundation, 2019, 27.10.2019 [cit. 2019-12-07]. Dostupné z: http://en.techinfodepot.shoutwiki.com/wiki/Linksys_WRT1900AC_v2

[43] Linksys WRT1900ACS Dual-Band Wi-Fi Router with Ultra-Fast 1.6 GHz CPU. *Linksys* [online]. c2019 [cit. 2019-12-07]. Dostupné z: https://www.linksys.com/cz/p/P-WRT1900ACS/

[44] Netgear R7800. In: *Info Depot Wiki* [online]. San Francisco (CA): Wikimedia Foundation, 2019, 3.10.2019 [cit. 2019-12-07]. Dostupné z: http://en.techinfodepot.shoutwiki.com/wiki/Netgear_R7800

[45] Nighthawk X4S Smart WiFi Gaming Router. *Netgear* [online]. 2019 [cit. 2019-12-07]. Dostupné z: https://www.netgear.com/home/products /networking/wifi-routers/R7800.aspx

[46] ZyXEL NBG6817 (Armor Z2). In: *Info Depot Wiki* [online]. San Francisco (CA): Wikimedia Foundation, 2019, 4.11.2019 [cit. 2019-12-07]. Dostupné z: http://en.techinfodepot.shoutwiki.com/wiki/ZyXEL_NBG6817_(Armor_Z2)

[47] Linksys WRT3200ACM. In: *Info Depot Wiki* [online]. San Francisco (CA): Wikimedia Foundation, 2019, 6.11.2019 [cit. 2019-12-07]. Dostupné z: http://en.techinfodepot.shoutwiki.com/wiki/Linksys_WRT3200ACM

[48] Linksys WRT3200ACM AC3200 MU-MIMO Gigabit Wi-Fi Router. *Linksys* [online]. c2019 [cit. 2019-12-07]. Dostupné z: https://www.linksys.com/cz/p/P-WRT3200ACM/

[49] Monitor mode support. *Turris Forum* [online]. 2019, 16.10.2019 [cit. 2019-12-08]. Dostupné z: https://forum.turris.cz/t/monitor-mode-support/11281

[50] Difference - Promiscuous vs. Monitor Mode (Wireless Context). *High on wires: Interesting concepts in the world of Computer Netowrking* [online]. 2008, 13.10.2008 [cit. 2019-12-10].
Dostupné z: http://lazysolutions.blogspot.com/2008/10/difference-promiscuous-vs-monitor-mode.html

[51] KOMAZEC, Bojan. Promiscuous vs monitor mode of a wireless network interface. *My Public Notepad* [online]. 2015, 6.10.2015 [cit. 2019-12-20]. Dostupné z: https://www.bojankomazec.com/2015/08/promiscuous-vs-monitor-mode-of-wireless.html

[52] MICHEL, Paul. Iw. *Linux Wireless* [online]. 2019, 18.6.2019 [cit. 2020-05-27]. Dostupné z: https://wireless.wiki.kernel.org/en/users/documentation/iw

[53] TOURRILHES, Jean. IWCONFIG: Linux - Manuál programátora. *Debian* [online]. 2004, 22.6.2004 [cit. 2020-05-27]. Dostupné z: https://manpages.debian.org/stretch/wireless-tools/iwconfig.8.cs.html

[54] VAN KEMPEN, Fred N., Alan COX, Phil BLUNDELL, Andi KLEEN a Bernd ECKENFELS. IFCONFIG: Linux System Administrator's Manual. *Debian* [online]. 2008, 3.10.2008 [cit. 2020-05-27].
Dostupné z: https://manpages.debian.org/stretch/net-tools/ifconfig.8.en.html

[55] The-tcpdump-group/libpcap. *GitHub* [online]. 2020 [cit. 2020-05-27]. Dostupné z: https://github.com/the-tcpdump-group/libpcap

[56] TCPDUMP/LIBPCAP. *TCPDUMP&LIBPCAP* [online]. c2010-2020 [cit. 2020-06-08]. Dostupné z: https://www.tcpdump.org/index.html

[57] Manpage of TCPDUMP. *Tcpdump* [online]. 2019, 2.4.2019 [cit. 2019-12-13]. Dostupné z: https://www.tcpdump.org/manpages/tcpdump.1.html

[58] Kismet. *Kismetwireless* [online]. c2019 [cit. 2019-12-16]. Dostupné z: https://www.kismetwireless.net/

[59] Compiling quickstart: Compiling: Quick Setup. *Kismetwireless* [online]. c2019, 29.11.2019 [cit. 2019-12-16]. Dostupné z: https://www.kismetwireless.net/docs/readme/quickstart/

[60] Kismetdb to PCAP. *Kismetwireless* [online]. c2019, 29.11.2019 [cit. 2019-12-16].Dostupné z: https://www.kismetwireless.net/docs/readme/kismetdb_to_pcap/

[61] BEAUPRÉ, Antoine. Horst manpage. *Ubuntu manuals* [online]. Canonical, c2019, 14.10.2012 [cit. 2019-12-17]. Dostupné z: http://manpages.ubuntu.com/manpages/trusty/man8/horst.8.html

[62] About. *Wireshark: Go Deep* [online]. 2020 [cit. 2020-06-08]. Dostupné z: https://www.wireshark.org/

[63] WU, Peter. CaptureFilters. *Wiki Wireshark* [online]. 2016, 19.10.2016 [cit. 2020-05-27]. Dostupné z: https://wiki.wireshark.org/CaptureFilters

[64] D.2. tshark: Terminal-based Wireshark: Appendix D. Related command line tools. *Wireshark* [online]. 2020 [cit. 2020-05-27]. Dostupné z: https://www.wireshark.org/docs/wsug_html_chunked/AppToolstshark.html

[65] AWUS1900 802.11ac AC1900Ultra-speed USB Adapter. *Euro DK* [online datasheet]. c2010-2020 [cit. 2020-05-28]. Dostupné z: https://www.eurodk.com/files/catalogue/alfa/datasheet-awus1900.pdf

[66] Turris OS versions. *Docs Turris: Turris Documentation* [online]. 2020, 7.2.2020 [cit. 2020-05-28]. Dostupné z: https://docs.turris.cz/basics/tos_versions/

[67] What is Kali Linux? *Kali Linux: by Offensive security* [online]. Amterdam, c2020, 25.11.2019 [cit. 2020-06-08]. Dostupné z: https://www.kali.org/docs/introduction/what-is-kali-linux/

[68] Aircrack-ng. *Aircrack-ng* [online]. c2009-2020 [cit. 2020-05-28]. Dostupné z: https://www.aircrack-ng.org/

[69] RTL8812AU/21AU and RTL8814AU driver with monitor mode and frame injection. *GitHub* [online]. GitHub, c2020 [cit. 2020-05-28]. Dostupné z: https://github.com/aircrack-ng/rtl8812au

[70] NetworkManager (Česky). *Archlinux* [online]. c2002-2020, 16.10.2019 [cit. 2020-05-29]. Dostupné z: https://wiki.archlinux.org/index.php/NetworkManager_(%C4%8Cesky)

[71] Airmon-ng. *Aircrack-ng* [online]. 2019, 18.8.2019 [cit. 2020-05-29]. Dostupné z: https://www.aircrack-ng.org/doku.php?id=airmon-ng

[72] LOMBARDO, Dario. Sshdump. *Wireshark* [online]. [cit. 2020-05-30]. Dostupné z: https://www.wireshark.org/docs/man-pages/sshdump.html

[73] Gcc install Busybox conflict. *Forum Turris* [online]. 2020, 7.3.2020 [cit. 2020-05-31]. Dostupné z: https://forum.turris.cz/t/gcc-install-busybox-conflict/12421

[74] 4-Way Handshake. *Wifi professionals* [online]. 2019, 24.1.2019 [cit. 2020-06-01]. Dostupné z: https://www.wifi-professionals.com/2019/01/4-way-handshake

[75] Airdecap-ng. *Aircrack-ng* [online]. 2020, 26.9.2009 [cit. 2020-06-01]. Dostupné z: https://www.aircrack-ng.org/doku.php?id=airdecap-ng

[76] What are RadioTap Headers? *WifiNigel* [online]. 2013, 30.11.2013 [cit. 2020-06-01]. Dostupné z: https://wifinigel.blogspot.com/2013/11/what-are-radiotap-headers.html

[77] 802.11ac (VHT) - Just the Facts. *Certified Wireless Network Professionals* [online]. c2020, 07.05.2012 [cit. 2020-06-02]. Dostupné z: https://www.cwnp.com/802-11ac-vht-just-the-facts/

[78] Wireless basic configuration: Short Guard Interval and multipath effect FAQ: 26.3.2020. *Sonicwall* [online]. c2020 [cit. 2020-06-02]. Dostupné z: https://www.sonicwall.com/support/knowledge-base/wireless-basic-configuration-short-guard-interval-and-multipath-effect-faq/170504672960493/

[79] Demystifying Modulation and Coding Scheme (MCS) Index Values. *Digital Air Wireless* [online]. c2018 [cit. 2020-06-02]. Dostupné z: https://www.digitalairwireless.com/articles/blog/demystifying-modulation-and-coding-scheme-mcs-index-values

[80] PyShark: Python packet parser using wireshark's tshark. *Https://pages.github.com/* [online]. c2020 [cit. 2020-06-03]. Dostupné z: https://kiminewt.github.io/pyshark/

[81] HANAWA, Yoshio. Chanhop.sh. *GitHub Gist* [online]. c2020 [cit. 2020-06-03]. Dostupné z: https://gist.github.com/hnw/6fbd3ac3bb59d0c93fc0bd2a823cf5cb

[82] Compal Broadband Networks CH7465LG-LC. *WikiDevi Wireless Cat* [online]. 2020, 16.4.2020 [cit. 2020-06-04]. Dostupné z: https://wikidevi.wi-cat.ru/Compal_Broadband_Networks_CH7465LG-LC

[83] Networking/Computing Tips/Tricks: The Purpose of the IP ID Field Demystified. *CellStream Inc.: Instruction. Knowlage, Skills.* [online]. c1998-2020, 28.3.2016 [cit. 2020-06-04]. Dostupné z: https://www.cellstream.com/reference-reading/tipsandtricks/314-the-purpose-of-the-ip-id-field-demystified

Seznam symbolů, veličin a zkratek

ACK	Acknowledgement
AES	Advanced Encryption Standard
AP	Access Point
BSSID	Basic Service Set Identifier
CCK	Complementary Code Keying
CCMP	Counter-mode Cipher Block Chaining Message Authentication Code Protocol
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
ČTU	Český telekomunikační úřad
DA	Destination address
dBm	Decibel vztažený na 1 miliwatt
DCF	Distributed Coordination Function
DECT	Digital Enhanced Cordless Telecomunications
DHCP	Dynamic Host Configuration Protocol
DIFS	Distributed Coordination Function InterFrame Space
DLNA	Digital Living Network Alliance
DNS	Domain Name Service
DS	Distribution system
DSSS	Direct Sequence Spread Spectrum Radio
DVB-T	Digital Video Broadcasting – Terrestrial
EIRP	Equivalent Isotropically Radiated Power, Effective Isotropic Radiated Power
ESSID	Extended Service Set Identifier
FHSS	Frequency Hopping Spread Spectrum Radio
FTP	File Transfer Protocol
GB	Gigabajt
GHz	Gigahertz
GPL	General Public License
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HE	High Efficiency
HT	High Throughput
HTTP/S	Hypertext Transfer Protocol/Secure
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IFS	InterFrame Space

IoT Internet of Things
ISM Industrial, Scientific, Medical
LAN Local Area Network
LLC Logical Link Control
LTE Long Term Evolution
MAC Media Access Control
Mbit/s Megabit za sekundu
MHz Megahertz
MIMO Multiple-Input Multiple-Outpu
MPDU MAC protocol data units
ms milisekunda
mSATA mini-Serial Advanced Technology Attachmen
MU-MIMO Multiple user Multiple-Input Multiple-Output
NAS Network Attached Storage
NAV Network Allocation Vector
OFDM Orthogonal Frequency Divison Multiplexing
OFDMA Orthogonal Frequency-Division Multiple Access
PCAP Packet Capture
PCF Point Coordination Function
PLCP Physical Layer Convergence Protocol
PMD Physical Medium Depandent
PPDU PLCP protocol data units
QAM Quadrature Amplitude Modulation
QoS Quality of Service
QPKS Quadrature phase shift keying
RA Reciever address
RAM Random Access Memory
RFID Radio Frequency Identification
RFMON Radio Frequency Monitor
RSSI Received Signal Strength Indication
RTS/CTS Request To Send/Clear To Send
SA Source address
SFP Small Form-Factor Pluggable
SFTP SSH File Transport Protocol
SIFS Short InterFrame Space
SISO Single-Input Single-Output
SSD Solid-State disk
SSH Secure Shell
SSID Service Set Identifier

SU-MIMO Single user Multiple-Input Multiple-Output
TA Transmitter address
TCP Transmission Control Protocol
TKIP Temporal Key Integrity Protocol
UDP User Datagram Protocol
UMTS Universal Mobile Telecommunications System
USB Universal Serial Bus
VHT Very High Throughput
VLAN Virtual Local Area Network
VPN Virtual Private Network
WAN Wide Area Network
WECA Wireless Ethernet Compatibility Alliance
WEP Wired Equivalent Privacy
Wi-Fi Wireless Fidelity
WLAN Wireless Local Network
WPA Wi-Fi Protected Access

Seznam příloh

A Skript ChanAnalyze.py

Výpis A.1: Obsah skriptu ChanAnalyze.py

```
#!/usr/bin/env python3
# import použitých balíčků.
import pyshark
import collections
import numpy as np
import argparse
from operator import itemgetter
import matplotlib.pyplot as plt
import termplotlib as tpl

#Vytvoření argumentů.
parse = argparse.ArgumentParser()
#Argument pro zadání cesty k souboru.
parse.add_argument("-f", help="Path to PCAP file", required=True)
#Argument pro vypsání výsledků do terminálu.
parse.add_argument("-t", help="Plot bar graph in terminal", action="store_true", required=False)
args = parse.parse_args()

#Spuštění čtení souboru předaného argumentem "-f" pomocí Pyshark
#Rámce se ukládají do proměnné "cap".
cap = pyshark.FileCapture('{}'.format(args.f), only_summaries=False)
print("Analyzing file {}..".format(args.f))

#Funkce pro výpis výsledků do terminálu.
def terplot():
    fig = tpl.figure()
    fig.barh(list(counter.values()), (list(counter.keys())), force_ascii=True)
    fig.show()

#Funkce pro vynesení výseldků do grafu.
def matplotgraph():
    plt.style.use('ggplot')
    y_pos = np.arange(len(list(counter.keys())))
    plt.bar(y_pos, list(counter.values()), align='center',alpha=0.5, color=['b'])
    plt.xticks(y_pos, list(counter.keys()))
    plt.ylabel("Počet zařízen")
    plt.xlabel("Číslo kánalu")
    plt.title("Počet zařízení na konkrétní Wi-Fi kanál")
    plt.savefig('channels_graph.png')
    plt.show()

channelList = []
#Procházení rámců ze souboru v proměnné "cap".
for packet in cap:
  try:
    #Přidá MAC adresu a číslo kanálu do listu jako pár [[macX,kanalX], [macY, kanalY]..].
    channelList.append([packet.wlan.ta, packet.wlan_radio.channel])
  except:
    continue

new_channeList = []
#Kontrola duplicity. Prochází channelList a přidává pár [mac,kanal] do nového listu.
#Pokud narazí na pár, který ještě není v novém listu, tak jej přidá. Stejné ignoruje.
for elem in channelList:
    if elem not in new_channeList:
        new_channeList.append(elem)

#Seřadí channelList podle cisel kanálu.
sorted_channelList = sorted(new_channeList, key = itemgetter(1))
#Počítá počet záznamů pro jednotlivé kanály.
counter = collections.Counter(sublist[1] for sublist in sorted_channelList)

#Pokud je přítomen argument "-t" tak se zavolá funkci terplot(), jinak matplotgraph().
if(args.t):
    print("Kanál | Počet zařízení")
    terplot()
else:
    matplotgraph()
```

B Skript LiveCap.py

Výpis B.1: Obsah skriptu CapAnalyze.py

```
#!/usr/bin/env python3
# import použitých balíčků.
import pyshark
import argparse
import collections
import matplotlib.pyplot as plt
import numpy as np
import termplotlib as tpl
from datetime import datetime

#Vytvoření argumentů.
parse = argparse.ArgumentParser()
#Argument pro předání názvu rozhraní.
parse.add_argument("-i", help="Capture interface", required=True)
#Argument pro vypsání výsledků do terminálu.
parse.add_argument("-t", help="Plot bar graph in terminal", action="store_true", required=False)
#Argument pro analýzu počtu zachycených rámů na MAC adresu.
parse.add_argument("-mac", help="Analyzes the number of frames per MAC", action="store_true",
     required=False)

args = parse.parse_args()
#Spuštění zachytávání pomocí Pyshark, zachycené rámce se ukládají do proměnné "cap".
cap = pyshark.LiveCapture(interface='{}'.format(args.i), only_summaries=False)
capList = []

# Inicializace grafu
plt.ion()
plt.style.use('ggplot')
fig = plt.figure(figsize=(16,8))
axes = fig.add_subplot(111)
data_plot=plt.plot(0,0)
line, = axes.plot([],[])

#Funkce pro analýzu počtu zachycených rámů na MAC adresu.
#Přidá do listu MAC adresu vysílače
def macanalyze():
            frameList.append(packet.wlan.ta)

#Funkce pro analýzu typu rámců. Porovnává hodnotu v poli wlan.fc.type a
# přidává do listu údaj o typu každého rámce.
def frametypeanalyze():
        if(packet.wlan.fc_type == '0'):
             frameList.append("Management")
        elif(packet.wlan.fc_type == '1'):
             frameList.append("Control")
        elif(packet.wlan.fc_type == '2'):
             frameList.append("Data")
        else:
             frameList.append("Other")

#Funkce pro výpis vysledků do terminálu
def terplot():
    print(datetime.now().strftime("%Y-%m-%d %H:%M:%S"))
    fig = tpl.figure()
    fig.barh(list(counter.values()), (list(counter.keys())), force_ascii=True)
    fig.show()
    print("-----------------------------------------------------------------\n")

#Funkce pro zobrazení výsledků do grafu
def matplotgraph():
        plt.bar(x,list(counter.values()), align='center',alpha=0.5, color=['b', 'g', 'r', 'c', '
             m'])
        plt.xticks(x, list(counter.keys()), rotation=45)
        plt.ylabel("Počet")
        plt.xlabel("Typ rámce")
        #plt.ylabel("Cetnost")
        #plt.xlabel("MAC")
        #plt.title('Pocet ramcu na MAC')
        plt.draw()
        plt.pause(0.1)

frameList = []
#Pokud je přítomen argument "-mac" zavolá se funkce macanalyze()
#Jinak se zavolá funkce frametypeanalyze()
```

```
if(args.mac):
      for packet in cap.sniff_continuously(packet_count=20):
          macanalyze()
          counter = collections.Counter(frameList)
          x = np.arange(len(list(counter.keys())))
          if(args.t):
            terplot()
          else:
            matplotgraph()
else:
      for packet in cap.sniff_continuously(packet_count=20):
          frametypeanalyze()
          counter = collections.Counter(frameList)
          x = np.arange(len(list(counter.keys())))
          if(args.t):
             terplot()
          else:
             matplotgraph()
             counter = collections.Counter(frameList)
             x = np.arange(len(list(counter.keys())))
#zobrazí graf
plt.show(block=True)
```

C Skript DataLenCounter.py

Výpis C.1: Obsah skriptu DataLenCounter.py

```
#!/usr/bin/env python3
# import použitých balíčků.
import pyshark
import collections
import numpy as np
import argparse

#Vytvoření argumentů.
parse = argparse.ArgumentParser()
#Argument pro zadání cesty k prvnímu souboru.
parse.add_argument("-f", help="Path to reference PCAP file", required=True)
#Argument pro zadání cesty k druhému souboru.
parse.add_argument("-f2", help="Path to second PCAP file", required=True)

args = parse.parse_args()

#Spuštění čtení 1. souboru předaného argumentem "-f" pomocí Pyshark
#Rámce se ukládají do proměnné "cap".
cap = pyshark.FileCapture('{}'.format(args.f), only_summaries=False)
print("Analyzuji první soubor {}..".format(args.f))


file1DataLenBytes = 0 #Proměnná pro součet velikosti dat v paketech
file1TotalFrames = 0 #Proměnná pro celkový součet paketů v 1. souboru

#Prochází všechny pakety v 1. souboru a přičítá velikost dat v nich
for packet in cap:
  file1DataLenBytes += int(packet.tcp.len)
  #Počítá celkový počet zachycených paketů
  file1TotalFrames += 1

file1DataLenMegaBytes = (file1DataLenBytes / 1048576) #Převod z B na MB
print("Počet rámců: ", file1TotalFrames)
print("Bajty: ", file1DataLenBytes)
print("Megabajty: ", file1DataLenMegaBytes)

#Spuštění čtení 2. souboru předaného argumentem "-f2" pomocí Pyshark
#Rámce se ukládají do proměnné "cap2".
cap2 = pyshark.FileCapture('{}'.format(args.f2), only_summaries=False)
print("Analyzuji druhý soubor {}..".format(args.f2))


file2DataLenBytes = 0 #Proměnná pro součet velikosti dat v rámcích
file2TotalFrames = 0 #Proměnná pro celkový součet rámců v 2. souboru
rssiSum = 0      #Proměnná pro součet hodnot RSSI

#Prochází všechny rámce v 2. souboru a přičítá velikost dat v nich.
for packet in cap2:
  file2DataLenBytes += int(packet.data.len)
  #Sčítá hodnoty RSSI
  rssiSum += int(packet.wlan_radio.signal_dbm)
  #Počítá celkový počet zachycených rámců
  file2TotalFrames += 1

file2DataLenMegaBytes = (file2DataLenBytes / 1048576) #Převod z B na MB
rssiPrumer = rssiSum / file2TotalFrames #Výpočet průměrného RSSI
print("Počet rámců: ", file2TotalFrames)
print("Bajty: ", file2DataLenBytes)
print("Megabajty: ", file2DataLenMegaBytes)
print("Průměrná síla signálu (RSSI): ", round(rssiPrumer))

lossInBytes = file1DataLenBytes - file2DataLenBytes # Výpočet ztráty v bajtech
lossInPct = (lossInBytes/file1DataLenBytes) * 100 # Výpočet ztráty v %

print("Ztráta v bajtech: ", lossInBytes)
print("Ztráta v procentech: ", lossInPct)
```