Ett prototypsystem för detektion och lokalisering av fusk under tentamina genom signalspaning av WiFi-frekvenser

A Prototype-system for Detecting and Localising Cheating During Exams Through Signal Surveillance of WiFi-frequencies

Johannes Binde, Erik Frennborn, Lucas Glimfjord
Gustav Henriksson, Daniel Nguyen
Jacob Thorselius Pedersen

Handledare: Arne Linde

CHALMERS

Kandidatarbete vid Institutionen för data- och informationsteknik
Chalmers tekniska högskola, Göteborg 2020

Abstract

As learning institutions adopt more digital tools in their examination procedures new opportunities for cheating emerge. One example is exams where students are allowed to bring their own computers. To ensure the students are not communicating with outside parties during the exam new countermeasures are needed. The goal of this project was the development of a signal surveillance system which aims to detect and locate the source of unauthorized communication. The resulting prototype uses a software-defined radio system to listen in on and analyze 2.4 GHz WiFi traffic. With additional hardware and software modifications it could also be expanded to cover Bluetooth and cellular communication. Further work is needed to make the system ready for use, as the bridge between the receiver and the processing software remains incomplete.

Keywords: *surveillance, cheating, exam, SDR, GNU Radio, Monopulse, WiFi, Bluetooth, cellular.*

Sammandrag

I takt med att lärosäten digitaliserar sina examinationsmoment uppstår nya möjligheter att fuska. Ett specifikt exempel är tentamina där studenter tillåts ta med sina egna datorer. För att förhindra att studenterna tar kontakt med en utomstående part under examinationen krävs nya lösningar. Målet med detta projekt var att utveckla ett signalspaningssystem ämnat att detektera och lokalisera sådan otillåten kommunikation. Den resulterade prototypen använder mjukvarudefiniered radio för att detektera och analysera trafik över 2,4 GHz WiFi. Med ytterligare hårdvara och mjukvaruanpassningar kan det vara möjligt att utöka systemet så att det även täcker kommunikation över Bluetooth och mobilnät. Systemet är i dagsläget inte redo att användas då kopplingen mellan signalmottagarna och de mjukvarukomponenter som utför pejling- och detektionsberäkningarna inte är färdigställd.

Nyckelord: *signalspaning, fusk, tenta, prov, mjukvaruradio, GNU Radio, Monopulse, WiFi, Bluetooth, mobilnät.*

Förord

Vi vill tacka vår handledare Arne Linde för all konstruktiv kritik och vägledning under projektet. Vi vill också tacka Charles Keeling på avdelningen för fackspråk och kommunikation vid Chalmers för hans hjälp och råd under skrivarbetet. Vi tackar Alfred Patriksson vid Totalförsvarets forskningsintitut och Kevin Larsson på SAAB för deras hjälp i att samla information under arbetets början. Slutligen vill vi tacka Jan Johansson vid Research Institutes of Sweden för att ha hjälpt oss låna testutrustning.

Innehåll

1 Inledning

I takt med att lärosäten strävar efter att digitalisera examinationsmoment uppstår många nya möjligheter och metoder för tentander att fuska [1]. När elektroniska hjälpmedel tillåts under en tentamen måste det säkerställas att dessa enbart används i tillåtna syften. Tentanderna måste då hindras från att kunna kontakta en utomstående part eller använda sig av andra otillåtna hjälpmedel [2].

Ett sätt att motverka denna typen av fusk är att lärosätet tillhandahåller hårdvaran, exempelvis genom att låna ut datorer till studenter under tentamen. Med direkt kontroll över hårdvaran blir det betydligt enklare att designa ett system med begränsad funktionalitet samt att i efterhand kontrollera om något försök att kringgå dessa säkerhetsåtgärder har gjorts. Nackdelen med detta tillvägagångssätt är kostnaden av att köpa in och underhålla hårdvaran samt det administrativa arbetet kring utlåningen. Det utesluter heller inte att en tentand tar med en extern enhet som hålls dold, vilket innebär att problemet kvarstår.

Alternativt kan tentanderna tillåtas att ta med egna datorer till examinationen [2]. Detta kan minska kostnaderna avsevärt men öppnar upp för fler möjligheter till fusk. På egen hårdvara kan tentander köra vilken mjukvara de vill eller modifiera hårdvaran genom tillägg av till exempel ett extra nätverkskort eller annan typ av signalmottagare. Om medtagen hårdvara tillåts krävs alltså externa säkerhetsåtgärder för att kunna försäkra att ingen tentand bedriver otillåten kommunikation under examinationen. En sådan lösning skulle även kunna upptäcka dolda enheter som försöker kommunicera på ett otillåtet sätt.

1.1 Syfte

Syftet med detta projekt var att utveckla en prototyp till ett signalspaningsystem. Fokus har varit på att kunna detektera och lokalisera otillåten kommunikation under tentamina där studenter medtar egna datorer. Systemet var ämnat att användas av vakter i tentaminsalar för att underlätta deras arbete att motverka fusk.

1.2 Begränsningar

Projektet hade en tidsbegränsning på ungefär 4 månader. Projektet begränsades även av krav på att köpa in hårdvara eftersom den inte fanns tillgänglig vid projektets början. Projektet fick inte någon tydlig siffra för exakt hur mycket hårdvaran fick kosta men det fanns riktlinjer om att de flesta köpen under 3000 kr inte ifrågasattes. Detta tolkades som att denna summa helst inte borde överskridas men att det fanns visst svängrum om det skulle vara nödvändigt.

1.3 Arbetsmetodik

För att bygga en teoretisk bas och samla kunskap om existerande lösningar utfördes vid projektets början en litteraturstudie. Ett antal vetenskapliga artiklar samt kurslitteratur granskades och ett antal personer i organisationer med erfarenhet av kommunikationsteknologier och signalbehandling kontaktades, däribland *Totalförsvarets forskningsintitut* (FOI) och *Research Institutes of Sweden* (RISE). För att säkerställa att alla gruppmedlemmar hade en tillräcklig förståelse för samtliga koncept som projektet skulle involvera skedde ett utbyte av förkunskaper. Detta för att gruppmedlemmar med olika bakgrund bättre skulle förstå varandra och arbetet i helhet.

Utvecklingsarbetet har bedrivits i enlighet med arbetsmetoden Scrum [3]. Arbetet delades varje vecka in i deluppgifter som tilldelas en eller flera gruppmedlemmar beroende på omfattning och komplexitet. De huvudsakliga arbetsområdena var teoretisk signalbehandling, val av och arbete med hårdvara samt mjukvaruutveckling. Ursprungligen hölls två officiella möten varje vecka för att reflektera över, stämma av kring och planera arbetet. Under projektets andra halva hölls istället distansmöten dagligen.

För att testa systemet sattes det upp i en större tentaminasal. Med mobiltelefoner och bärbara datorer som sändare och mottagare mättes systemets kapacitet. Ett antal olika positioner och avstånd mellan sändare och systemet prövades. För att ha något att jämföra prototypens mätningar med användes ett system utlånat från RISE i samband med testningen.

Systemet jämfördes med andra lösningar på marknaden i termer av prestanda, precision och frekvensomfång. En teoretisk uppskattning av prototypsystemets funktionalitet användes eftersom systemet inte var färdigutvecklat när jämförelsen genomfördes. I och med att de andra systemens priser inte fanns tillgängliga gjordes heller ingen precis kostnadsjämförelse.

1.4 Rapportens disposition

I avsnitt 2 ges en överblick av projektets teoretiska och tekniska bakgrund med ett fokus på olika typer av pejlingalgoritmer. Därefter presenteras i avsnitt 3 en översikt av prototypsystemet, sedan redovisas i avsnitt 4 själva konstruktionsprocesen. I avsnitt 5 presenteras projektets resultat. Resultaten och arbetet i helhet diskuteras sedan i avsnitt 6, som avslutas i en jämförelse med andra signalspaningssystem på markanden. Slutligen sammanfattas projektet i avsnitt 7.

2 Teori och teknisk bakgrund

I detta avsnitt redovisas projektets teoretiska och tekniska bakgrund. Först ges en överblick av olika kommunikationsteknologier samt mjukvarudefinierad radio och programmet GNU Radio. Sedan följer en kort sammanfattning av några vanliga koncept inom signalbehandling följt av en mer detaljerad förklaring av tre olika typer av pejlingalgoritmer. Sist ges en förklaring av HTTP och REST API:er.

2.1 WiFi, mobilnät och Bluetooth

WiFi finns i ett antal olika versioner. Upp till och med WiFi 5 används antingen 2,4 GHz (2 401 MHz - 2 495 MHz) eller 5 GHz (5 030 MHz - 5 875 MHz) [4]. Vad gäller mobilnät används i Sverige i nuläget främst de så kallade 3G och 4G teknologierna [5]. Dessa är samlingsnamn för ett antal standarder inom samma generation. De underliggande teknikerna som används i Sverige täcker in frekvensband på 450 MHz, 700 MHz, 800 MHz, 900 MHz, 1 800 MHz, 1 900 MHz, 2 100 MHz samt 2 600 MHz [6]. Dock används inte alla frekvensband av alla tekniker eller operatörer. Bluetooth använder signaler med frekvenser runt 2,4 GHz (2 402 MHz - 2 480 MHz). Avståndet mellan de olika kanalera är 2,0 MHz vilket resulterar i 40 olika kanaler som signalerna varierar mellan [7].

2.2 Mjukvarudefinierad radio

Mjukvarudefinierad radio (SDR) bygger på principen att signalbehandling som traditionellt skulle utförts med specialdesignade kretsar i hårdvaran istället görs i mjukvara [8]. De analoga signalerna samlas in och konverteras till ett digitalt format som kan behandlas av en dator som sedan utför exempelvis filtrering eller pejlingberäkningar. På grund av att mjukvara oftast är betydligt lättare att modifiera eller byta ut än en hårdvarukrets erbjuder SDR-system i många fall ett flexiblare alternativ för signalbehandling. Samtidigt blir systemen i helhet ofta billigare att sammanställa då mindre funktionalitet krävs av hårdvaran och det finns en stor mängd både öppen och gratis mjukvara tillgänglig [9].

2.3 GNU Radio

GNU Radio är en kostnadsfri mjukvara som tillåter konstruktion av signalbehandlingssystem med hjälp av ett grafiskt gränssnitt bestående av flödesscheman [10]. Mjukvaran har även funktionalitet att kompilera flödesscheman till Python kod [10]. GNU Radio kommer med en variation av inbyggda funktioner för analys av radiosignaler, exempelvis filter och realtidsgrafer. Möjligheten att modifiera hur de olika filtren agerar med hjälp av ett grafiskt gränssnitt underlättar arbetet samtidigt som det tillåter förändring av olika parametrar i realtid [10].

2.4 Antennteori

För att välja en passande antenn krävs en del grundläggande kunskaper om antenner och deras egenskaper i kombination med kunskap om den typ av signaler som ska mottagas. Till en början måste den mottagna signalens polarisering vara känd. Polarisationen kan beskrivas som en egenskap hos en elektromagnetisk våg som beskriver tidsvarierande riktning hos dess elektriska fältvektor i relation till vågens riktning [11]. I praktiken innebär detta kortfattat att en antenn som är vertikalt polariserad kommer att ha svårigheter att ta emot signaler som är horisontellt polariserade medan en antenn som är horisontellt polariserad kommer att ha svårigheter att ta emot vertikalt polariserade vågor.

Hänsyn måste också tas till den givna antennens bandbredd. Med en antenns bandbredd menas det frekvensomfång för vilket antennen beter sig som tillverkaren specificerat att den gör. I detta beteende inkluderas bland annat parametrar så som förstärkning och strålningsmönster [11]. Strålningsmönstret för en given antenn är en ofta grafisk beskrivning av hur väl antennen kan ta emot olika signaler från olika infallsvinklar. För en rundstrålande antenn är denna förmåga idealt sett lika god i alla riktningar medan den för en riktad antenn skiljer sig beroende på vilken riktning antennen mottar vågen från. För att beskriva denna förmåga används ofta enheten isotropisk decibel (dBi), där 1 dBi är den förstärkning som en ideal rundstrålande antenn tar emot signaler med. Karakteristiskt för en riktad antenn är då att den för vissa infallsvinklar har en förstärkning som överstiger 1 dBi medan den för andra infallsvinklar är betydligt lägre än 1 dBi. Detta medför i praktiken att mottagningsförmågan endast är nog bra för att användas i vissa riktningar vilket är en önskvärd egenskap för vissa användningsområden.

2.5 Utrustning för testning

Utrustningen som användes för att testa systemets precision var en spektrumanalysator med ett riktat antennelement. Analysatorn var av modellen Spectrum Master MS2722C från Anritsu och antennen var en riktad Rohde & Schwarz RE300 med ett frekvensomfång på 0,5 GHz till 7,5 GHz. Utrustningen lånades via RISE.

2.6 Sampling av signaldata

I syfte att kunna avgöra huruvida det förekommer kommunikation måste data om frekvensspektrumet samlas in. Denna data måste vara så korrekt som möjligt, det vill säga att den skall vara så störningsfri som möjligt samtidigt som den inte innehåller någon vikning. Vikning innebär att det, på grund av bristande upplösning, inte går att urskilja vissa höga frekvenser från lägre frekvenser i en samplad signal. Eftersom datan behandlas digitalt måste den först samplas från en analog signal, vilket innebär att en del krav ställs på den använda hårdvaran.

Det allra viktigaste är att hårdvaran klarar av att sampla signaler med en nog hög frekvens. För att den skall klara av detta måste Nyquists samplingsteorem vara uppfyllt, eftersom den samplade signalen annars utsätts för vikning och blir felaktig [12]. Nyquists samplingsteorem innebär att den analoga signalen måste samplas med en frekvens $\omega_s > 2\omega$, där ω_s är samplingsfrekvensen. I händelse att det finns flera olika signaler, vilket ofta är fallet, är ω frekvensen för den signal vilken innehar högst frekvens av alla signalbidrag.

2.7 Algoritmer för pejling

Pejling innebär att med hjälp av signalbehandling lokalisera en signalkälla. Det är en teknologi med en rad olika användningsområden, inte minst inom militära applikationer. Det finns en mängd olika tillvägagånssätt för att pejla som varierar i precision, komplexitet och hårdvarukrav. I detta stycke beskrivs tre olika typer av pejlingsystem: Monopulse, Time Difference of Arrvial (TDOA) och Nearest Node.

2.7.1 Monopulse

Monopulse är en teknik som baseras på att åtminstone två antenner med olika gain-funktioner G_1 och G_2 som mäter samma signal och jämför signalstyrkan [13]. I praktiken fås de två olika gain-funktionerna enklast genom användning av två identiska antenner. Det fungerar förutsatt att de två antennerna inte är rundstrålande samt att de inte är riktade åt samma håll. Detta innebär att antennerna har en lika stor vinkel θ_s från det som anses vara antennkombinationens riktning, se Figur 1. Denna vinkel varieras enklast genom att fysiskt manipulera antennelementen på ett sådant sätt att önskat θ_s uppnås. Detta medför att gain-funktionerna, som kan modelleras med avseende på den infallande signalens vinkel jämfört med antennen i praktiken blir förskjutna relativt antennkonstellationens riktning.

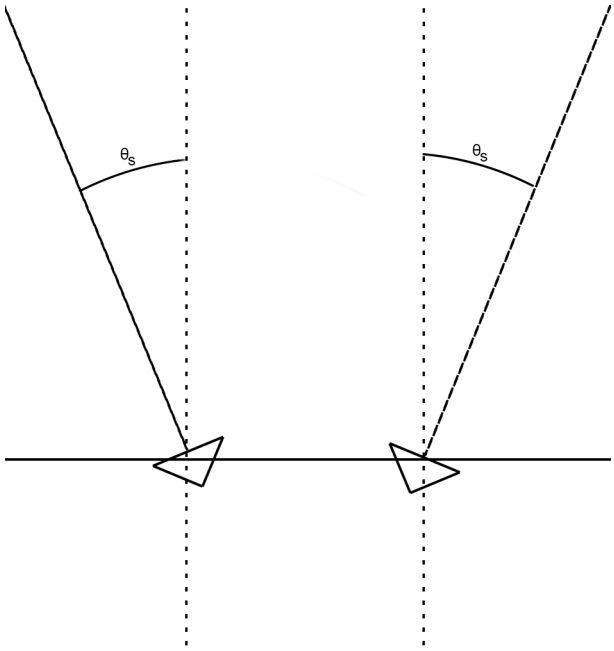
I det ideala fallet antas en specifik förstärkning endast en gång i det studerade området, eftersom det annars blir svårt att bestämma vilken infallsvinkel den inkommande signalen har, vilket i sin tur leder till svårigheter för att positionsbestämma den mottagna signalens källa. Om varje förstärkningsvärde däremot kan knytas till en unik vinkel så går det alltid att hitta signalens källa jämfört med den mottagande antennen med hjälp av en sum-diff-kvot enligt

$$\begin{aligned} A_1 &= A_x G(\theta - \theta_s) \\ A_2 &= A_x G(\theta + \theta_s) \\ D &= A_1 - A_2 \\ S &= A_1 + A_2 \\ R(\theta) = \frac{D}{S} &= \frac{A_1 - A_2}{A_1 + A_2} = \frac{A_x}{A_x}\frac{G(\theta - \theta_s) - G(\theta + \theta_s)}{G(\theta - \theta_s) + G(\theta + \theta_s)}. \end{aligned} \tag{1}$$

A_x är signalens amplitud vid källan, denna är okänd. A_1 & A_2 är de amplituder som uppmäts av de riktade antennerna och θ_s är en vinkel från antennkonstel-

lationens mittlinje enligt Figur 1. Den okända A_x elimineras, vilket tillåter en precis positionsberäkning [14].

I det ideala fallet fås två identiska grafer förskjutna med θ_s åt varsitt håll för y-axel. För att eliminera den okända A_x används då en sumdiff-kvot enligt ekvation (1). Då fås en udda funktion som är inverterbar inom ett interval runt vinkel 0.



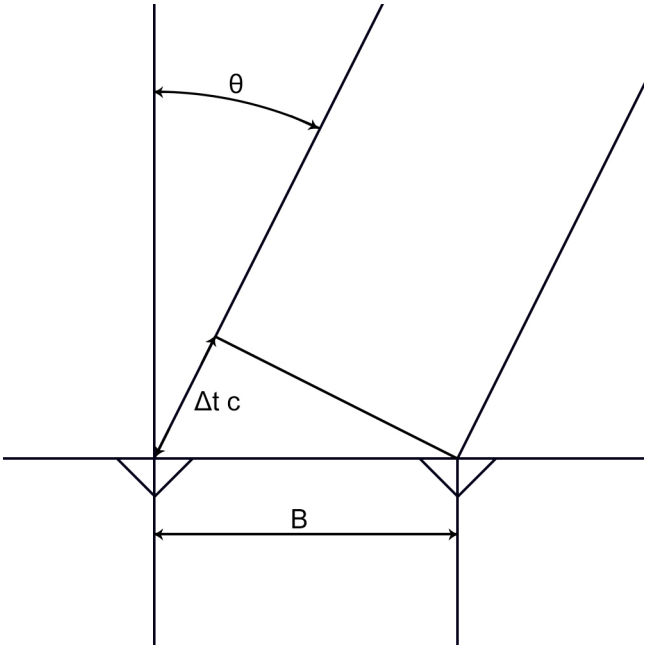
Figur 1: Monopulse-system

2.7.2 Time Difference of Arrival (TDOA)

Time difference of arrival (TDOA) bygger på ett system med två eller flera mottagare för att bestämma en sändares position. Detta görs genom att mottagarna placeras på kända avstånd B från varandra. Med hjälp av tidsskillnaden mellan när de olika mottagarna fångar upp signalen kan sedan en approximativ position beräknas. Då signalerna propagerar i nära ljusets hastighet ställs höga krav på hårdvaran att kunna mäta små tidskillnader med hög precision. Vidare måste mottagarna vara noga tidssynkroniserade, vilket utgör ytterligare en utmaning. Det är väsentligt att den inkommande signalen i princip infaller parallellt med den mottagande antennen för att algoritmen ska fungera. För att uppnå detta krävs att avståndet B mellan mottagarna är litet i relation till avståndet mellan mottagarna och sändaren. Då dessa krav uppfylls beräknas vinkeln vilken sändaren sänder ifrån enligt

$$\theta = \arcsin\left(\Delta t \frac{c}{B}\right) \tag{2}$$

där B är avståndet mellan mottagarna, Δt är tidsdifferensen som uppmäts mellan mottagarna och c är ljusets hastighet [15]. Se Figur 2 för en överblick av systemet.



Figur 2: TDOA-system

2.7.3 Nearest Node

Ett flertal mottagare placeras ut i rummet och sammankopplas till en central server. Mottagarna kör mjukvara som lyssnar efter WiFi signaler, exempelvis airodump-ng. När en mottagare upptäcker kommunikation på en förbjuden kanal så meddelar den kanal, tid och amplitud till servern. Servern använder då data från fler mottagare för att göra en uppskattning av var i rummet sändaren befinner sig. Sedan kan, utifrån vilken mottagare som registrerat högst amplitud, en uppskattning av var i rummet sändaren befinner sig göras.

2.8 HTTP och REST API:er

Nätverkskommunikation via protokollet HTTP sker via ett antal request-metoder [16]. När en nätverks-API designas bör dessa metoder has i åtanke eftersom de kommer vara centrala för de anrop som kan göras och genom att använda varje metod till det den är ämnad för kan en mer strukturerad API uppnås [17]. För att konstruera en allmänt mer väldefinierad API kan REST användas [18]. REST inför ett antal regler som måste uppfyllas av varje API-anrop. Dessa i kombination med korrekt användning av HTTP-metoderna kan användas då en API ska designas för att uppnå en högre nivå av separation och abstraktion vid kommunikation över ett nätverk.

3 Systemöversikt och specifikation

Systemet kan huvudsakligen delas upp i tre olika delar. Så kallade ankare utför själva signalupptagningen, servern tar emot den insamlade datan och utför detektions- och pejlingberäkningar och slutligen presenterar klienten datan för en användare.

3.1 Ankare

Ett ankare är den del av systemet som samlar in signaldata från omgivningen med hjälp av antenner och SDR-enheter. Ankaret bearbetar kontinuerligt den inkommande rådatan och skapar vid detektion av datatrafik ett objekt med nödvändiga datapunkter som krävs för att servern ska kunna utföra positionsberäkningar. Insamlingen av rådatan behöver utföras med en hög samplingsfrekvens vilket medför en begränsning av hur många ankare en server simultant kan hantera. För att motverka detta utförs en stor del av bearbetningen och all analys av rådatan hos ankaret medan objektet med nödvändig information sedan skickas till servern för vidare bearbetning.

3.2 Server

Servern är den del av systemet där all data sparas och den slutgiltiga bearbetningen sker. Servern är länken mellan ankarna, som samlar in rådatan, och klienterna där datan sedan representeras. Från de ankare som är anslutna till servern mottas nödvändig information för att kunna göra en positioneringsberäkning av var i rummet datatrafik har detekterats. Fördelen med att inte göra all bearbetning hos servern, förutom att antalet ankare kan vara mycket högre, är att implementationen av ankare enkelt kan varieras med kravet att samma typ av dataobjekt fortfarande skickas vidare utan att behöva skriva om någon kod hos servern. De anslutna klienterna kan hämta information rörande både tidigare och nuvarande detektioner från servern.

3.3 Klienter

Klientdelen av systemet Representationen av resultaten sker på klientsidan genom att detekterade signalkällor visas upp så att deras position i relation till ankarna och salen blir tydlig, men även så att användaren kan avgöra hur sannolikt det är att fusk pågår. För att göra den bedömningen kan användaren överskåda alla detektioner vid en viss tidpunkt eller under ett längre tidsintervall. Utöver att avgöra signalkällors position är det även möjligt att se alla detektioner och deras data i mer detalj, vilket kan vara användbart för en mer avancerad användare eller för felsökning av systemet. Mjukvaruimplementationen av klientdelen är uppdelad och bortkopplad från servern för att göra systemet mer flexibelt när det sätts upp i en eller flera salar.

4 Systemkonstruktion

I detta avsnitt redovisas konstruktionsprocessen av prototypsystemet och en mer detaljerad förklaring av systemet ges. Designval på mjukvarusidan och valet av hårdvara presenteras och motiveras.

4.1 Hård- och mjukvara

För att underlätta kommunikationen mellan systemets olika delar definierades ett antal datastrukturer vars syfte var att se till att datarepresentationen var konsekvent i hela systemet. Eftersom både ankaret och servern implementerades i Python kunde de smidigt dela exakt samma datastrukturer, medan klienten fick implementera en egen version av dem i JavaScript. De tre viktigaste datastrukturerna i systemet var två som representerade respektive ankare och sändare samt en tredje vilken representerade en detektion. Ett flertal övriga strukturer användes för att representera mer abstrakta datamängder internt, exempelvis i samband med pejlingberäkningarna.

Detektionsstrukturen representerar datan som skickas från ett ankare till servern, det vill säga en amplitud, en frekvens och en tidsstämpel. För att kunna koppla en detektion till ett ankare innehåller datastrukturen, utöver den faktiska datan, också ett unikt id kopplat till ett specifikt ankare. Ankarstrukturen innehåller i sin tur ett unikt id som används för att särskilja olika ankare, ankarets fysiska position och en vinkel som beskriver hur stor vinkeln mellan ankarets antenner är. Slutligen representerar sändarstrukturen den datan som skickas från servern till klienten. Den består av en position, en amplitud, en frekvens och en tidsstämpel.

4.1.1 Ankare

Till ankaret användes två stycken ADALM PLUTO enheter, en SDR-enhet från Analog Devices Incorporated främst avsedd för undervisnings- och lärandeändamål. Enheten är officiellt specificerad för att ta emot samt skicka signaler mellan 325 MHz och 3,8 GHz [19]. Det finns även möjlighet att genom en mjukvarumodifikation utöka frekvensomfånget till mellan 70 MHz och 6 GHz [20]. Enheten kan kopplas till en dator för dataöverföring och strömförsörjning via USB.

Antennerna som användes är från företaget Delock. De är riktade och är specificerade främst för frekvensbanden 2,4 till 2,4835 GHz samt 5,1500 till 5,8750 GHz [21]. Antennerna har en gain som sträcker sig mellan 7,5 och 10 dBi beroende på signalens infallsvinkel. För att kunna samla in data och omvandla informationen så att den blir användbar kompletterades SDR-enheterna och antennerna med mjukvaran GNU Radio. Att mjukvaran kan kompilera flödesscheman till Python-kod underlättade utvecklingsprocessen för medlemmar utan programmeringserfarenhet eftersom de kunde utforma ett signalbehandlingssystem med hjälp av ett grafiskt gränssnitt [10].

4.1.2 Server

Servern arbetar huvudsakligen genom ett REST-API som används både av ankaret och klienten. De detektioner som ankaret skickar vidare tas emot och sparas i databasen. Om en detektion nyligen kommit in från ett annat ankare använder servern utöver att spara undan detektionen i databasen en algoritm som med ankarnas position och vinklar tar fram en punkt i rummet där detektionen identifierats. Denna nya punkt sparas också i databasen så att en klienten sedan kan hämta ut och analysera informationen vidare.

Databasen implementerades i MongoDB, en NoSQL-databas där datan sparas i så kallade dokument som består av fält med tillhörande värden [22]. Databasen designades för att innehålla tre dokument för de tre huvudsakliga datastukturerna: detektioner, ankare och sändare. När ett nytt objekt ska föras in i ett dokument konverteras det till JSON-format och på samma sätt konverteras det från JSON tillbaka till respektive datastruktur vid uthämtning. Då data hämtas från ett dokument kan det även ske filtrering på datan, exempelvis genom att bara hämta alla sändare som finns inom ett givet frekvens- och tidsintervall. Det är även möjligt att ta bort data från dokument på samma vis som den kan hämtas, det vill säga genom att definiera ett antal krav och sedan ta bort alla fält som matchar de kraven.

Utöver stöd för att hantera de detektioner som ankarna skickar, håller servern också koll på vilka ankare som är anslutna och dess position. Vid anslutning av ett nytt ankare ska det ange sin position och får då tillbaka ett id som används för att identifiera vilket ankaret är vid framtida dataöverföring. Ett flertal ankare kan kopplas till samma server.

För klienterna finns det stöd att hämta ut detektionerna och deras positioner inom givna tidsintervall samt stöd för att hämta frekvensdata från servern. Däremot finns det ingen implementation hos ankaret som skickar vidare frekvensdata till servern.

För att underlätta uppsättningen av systemet och enkelt möjliggöra byte av hårdvaran som servern kör på har servern byggts i Docker, ett containerisationsystem. Detta möjliggör att systemet alltid fungerar likadant oavsett vilken hårdvara och operativsystem det körs på förutsatt att Docker-engine går att installera.

4.1.3 Klienter

Klienterna möjliggör ett sätt att visualisera de detektioner som ankarna och servern identifierat och bearbetat. Antalet klienter som är anslutna till servern är tekniskt sett obegränsat men kan i praktiken bli begränsat av nätverkets kapacitet.

Användargränssnittet delades upp i tre olika vyer för att representera de olika typerna av data i systemet. De tre vyerna var kart-vy över detektioner vid en viss tid, överblicks-vy över alla detektioner och detalj-vy över en detektion.

För att representera detektionernas position vid en viss tidpunkt ritades detektionerna ut som semi-transparenta punkter i ett koordinatsystem. Att de var semi-transparenta gjorde att det blev tydligt ifall flera detektioner var på samma position, eftersom opaciteten då blev högre i de områden som täcktes av flera punkter. Varje punkt var klickbar och ledde till detalj-vyn för den detektionen för att göra det lätt för användaren att se detaljer om specifika detektioner. Under koordinatsystemet finns en slider där förändringar av vilken tid som skulle visas kunde göras. En slider agerar som ett positionsreglage, i det här fallet var positionen som observerades mätt i sekunder. Att det var en slider gjorde att det var lätt att se hur detektionerna förändrades över tid och på så vis avgöra vad som kunde vara brus och vad som kunde vara ett försök till fusk. I koordinatsystemet fanns även ankarna representerade för att göra det enklare för användaren att få en överblick över hur hela systemet var uppsatt. Det gjorde det även lättare att se vart en detektion var i relation till ankarna. Ovanför koordinatsystemet presenterades generellt beskrivande text om frekvens och antal detektioner som representerades i koordinatsystemet vid den tidpunkten.

En överblick av alla detektioner i systemet implementerades i form av en tabell, sorterade efter tidsstämpeln från tillfället de detekterades. Varje detektion representerades i tabellen med fem datapunkter. Ett detektions-id som främst används internt för att skilja olika detektioner åt och kan användas för exempelvis felsökning. Ett ankar-id som noterar vilket ankare som registrerade detektionen. Det är likt detektionens-id inte relevant för en normal användare, men ansågs givande att ha med som felsökningsinformation. Ett amplitudvärde som beskriver vilken amplitud signalen hade när den detekterades. Detta kan användas för att avgöra hur stark signalen var och kan även användas för att kontrollera ifall algoritmen har räknat ut positionen korrekt. Frekvensen beskriver vilken frekvens signalen hade när den detekterades, vilket kan vara intressant för att se vilka frekvenser som är vanligast bland detektionerna. Slutligen sparas en tidsstämpel som helt enkelt beskriver vid vilken tidpunkt detektionen skedde. Det används för att se hur detektionerna förändras och hur fördelningen ser ut över tid.

Slutligen finns det en vy vars syfte var att ge mer detaljerad information om enskilda detektioner. Syftet med en sådan vy är att en användare ska kunna analysera enskilda detektioner för att kunna verifera att datan är korrekt. Det är således en vy mer ämnad för felsökning än för normal användning.

4.2 Pejling

Eftersom de inköpta antennernas datablad inte innehöll någon gain-funktion beräknades en approximativ sådan. Detta utfördes med hjälp av visuell inspektion av det strålningsmönster som fanns angivet i antennernas datablad. I figuren för strålningsmönster drogs räta linjer som approximativt stämmer överens med det faktiska strålningsmönstret. Med hjälp av MATLAB-funktionen Polyfit hittades sedan en lösning till en polynomekvation som beskriver antennens mottagningsförmåga beroende av den infallande signalens vinkel jämfört med antennens framsida. Den approximativa signalen beräknades enbart för ett område om 180°, eftersom detta är det område vilket systemet var tänkt att verka i. Den resulterande approximationen beskrivs av följande polynom:

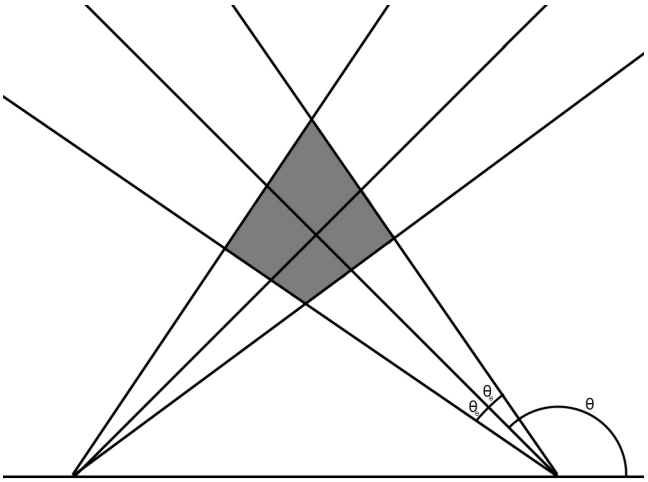
$$G(\theta) = 0.000000033\theta^4 - 0.00000000565\theta^3 - 0.000452\theta^2 + 0.0000334\theta + 2.176712. \tag{3}$$

Efter ett antal värden prövats, så identifierades att en offset på 30° gav det bästa resultatet. Detta resulterade i att den normaliserade funktionen blev inverterbar för 90° framför ankaret, vilket innebar att system fick en användbar vinkel på 90°. På grund av komplexiteten i att invertera (1) så genererades ett lookup table (LUT) med beräkningar av kvot till vinkel för att slippa göra dessa tidskrävande operationer i realtid. LUT:en generades med en noggrannhet på en datapunkt per var femtedels grad. När två amplituder uppmätts så beräknades den normaliserade kvoten enligt

$$r = \frac{A1 - A2}{A1 + A2}. \tag{4}$$

Sedan matchas den normaliserade kvoten med det förberäknade värdet från LUT:en som ligger närmast.

För att beräkna felmarginalen för den framtagna vinkeln så görs beräkningar utefter nio stycken olika scenarion. I de olika scenariorna varieras amplitud upp och ner samt mellan de två ankarna. Sedan används den största skillnaden mellan de olika beräknade värdena för att utgöra felmarginalen. När en vinkel beräknats från minst två ankare så används vinklarna i samband med ankarnas position för att beräkna en skärningpunkt. Denna process upprepas med olika felmarginaler för att genera ett område inom vilket sändaren bör finnas, se Figur 3.



Figur 3: Resultat av pejlingberäkningar med felmarginaler

4.3 Detektion

Till projektet har ett digitalt lågpassfilter av typen Finite Impulse Response (FIR) använts, vilket är ett filter som dämpar högre frekvenser, samtidigt som det släpper igenom, och i vissa fall förstärker, lägre frekvenser [23]. Filtret har implementerats med GNU Radio, vilket avsevärt förenklat filterdesginen jämfört med om denna gjorts utan hjälpmedel. Att filtret implementerades i GNU Radio har även inneburit att vissa av filtrets egenskaper, i detta fall dess bandbredd och avstängningsfrekvens, kunnat modifieras samtidigt som programmet körts [10]. En frequency-sink-modul i GNU Radio har använts för att ge en grafisk representation av vilka signaler som systemet fångar upp. Denna modul presenterar datan i form av en waterfall-graf, en Fouriertransform av signalen samt signalen i tidsdomänen.

Lågpass-filtret har använts för att filtrera all den indata som kommer från ADALM-PLUTO eftersom den använder sig av en lokal oscillator (LO) [24]. Den lokala oscillatorn utför mixning (heterodyning) på de mottagna signalerna, vilket innebär att de får ett frekvensskift lika stort som den lokala oscillatorns frekvens, en så kallad Intermediate Frequency (IF) [25]. I praktiken medför detta att de mottagna signalerna kan betraktas som signaler med lägre frekvens, vilket möjliggör sampling av signaler för vilka hårdvaran ursprungligen inte uppfyller Nyquists samplingskriterium. Med hjälp av ett lågpassfilter och justering av den lokala oscillatorns frekvens kan därför de signalfrekvenser som är relevanta att studera skiftas så att de har en frekvens nära 0 Hz och inom lågpassfiltrets bandbredd. Om ADALM-PLUTO istället filtrerat signalen utan nedmixning hade det inneburit att samplingsfrekvensen behövt vara mycket högre.

5 Resultat

Här redovisas projektets resultat. Då systemet i helhet inte är färdigställt läggs fokus på de olika delar som går att individuellt utvärdera.

5.1 Detektionstestning

För att utvärdera hur väl systemet fungerade i praktiken utfördes ett test i en större tentamenssal. Salen har en golvyta på $25,4 \times 38,6$ meter och en uppskattad takhöjd på 9,0 meter.

Bärbara datorer och mobiltelefoner anslutna till ett 2,4 GHz WiFi-nätverk placerades på olika platser i salen. Genom att strömma video samt att utnyttja ping-kommandon skickades data kontinuerligt från enheterna. Antennerna riktades i olika vinkar och på olika avstånd mot enheterna och access punkten. Sändarnas position varierades även för att mäta vilken effekt dessa förändringar fick på den insamlade signaldatan.

Den insamlade datan jämfördes kontinuerligt med datan som samlades in av spektrumanalysatorn som lånades av RISE. Resultatet av jämförelsen var att systemets hårdvara producerade mycket mer brus än förväntat vilket ledde till att det var svårt att urskilja de faktiska signalerna från brus. Det verkade även, baserat på antennernas mottagna signaler vid olika vinklingar och positioner, som att antennerna inte var vinklade på det sätt som var angivet i databladet. Då antennerna vändes 180° från signalkällan mottogs nämligen en nära identisk signalstyrka, vilket indikerar att antennerna snarare verkar mer rundstrålande än riktade. Detta ledde till att det inte utfördes fler tester eftersom datan inte hade tillräcklig kvalitet.

5.2 Pejlingalgoritmen

Pejlingalgoritmen är Monopulse-baserad vilket medför krav på att två SDR-enheter är anslutna och att deras antenner är placerade med en vinkel om 30° från en gemensam skiljelinje. Resultatet från algoritmen är ett detektionsobjekt som innehåller en vinkel och relevant metadata för att kunna avgöra tiden den detekterades och vilket ankare den kommit ifrån. Den informationen skickas sedan vidare till servern via dess REST-API. För att servern ska kunna arbeta vidare med detektionerna behövs två olika ankare och en jämförelse mellan detektionerna från de båda ankarna.

6 Diskussion av resultat och designval

I detta avsnitt diskuteras projektets resultat. Svårigheter och lärdomar som uppstått under arbetes gång reflekteras över, valet av pejlingalgoritm motiveras och hinder som uppkommit under projektets gång summeras. En överblick av några av de etiska aspekter som uppstår i samband med övervakning och datainsamling ges. Slutligen jämförs den utvecklade prototypen med ett antal andra system på marknaden i termer av prestanda och funktionalitet.

6.1 Hård- och mjukvara

Den större delen av mjukvaruarbetet med systemet gjordes i Python 3.8 med vissa undantag som var tvungna att skrivas i det äldre Python 2.7. Valet att använda just Python för att utveckla mjukvaran har varit både till för- och nackdel under projektet. Å ena sidan är Python ett bra språk att skriva prototypkod i då prestanda inte är i fokus och det istället är viktigare att snabbt kunna skriva fungerande kod och enkelt modifera den under arbetets gång. Å andra sidan har det uppstått problem på grund av att Python är ett interpreterat språk. Det finns ingen kompilator som ser till att all kod är fungerande innan man börjar köra. Detta orsakade många timmar av debuggande för att hitta små syntaxfel i delar av kod som inte rörts under längre tid. För att försöka åtgärda detta skrevs enhetstest och integrationstest varpå ytterligare problem hittades och åtgärdades. Dessa problem kunde ha undvikits eller minskats till stor del om ett språk som exempelvis Rust använts i vilket det finns en kraftfull typsystem och en kompilator som underlättar identifieringen av fel i koden tidigt i utvecklingsprocessen. Men valet att använda ett språk likt Rust hade också kunnat innebära negativa konsekvenser. GNU Radio är huvudsakligen byggt i Python, så att integrera extern kod skriven i Rust hade troligvis visat sig mer komplicerat.

Valet att dela upp systemet i tre olika delar gjordes tidigt och underlättade att strukturera arbetet så att flera personer kunde arbeta oberoende av varandra utan konflikter. Detta krävde även tydlig kommunikation mellan de som arbetade på de olika delarna eftersom delarna i slutändan skulle kommunicera med varandra. Att se till att de kommunikationsprotokoll som användes mellan enheterna var definierade tidigt i processen samt att förändringar i dessa kommunicerades ordentligt var viktigt för att minimiera konflikter och problem.

Motiveringen till ett uppdelat system ligger dels på att det möjliggör en mer flexibel uppsättning av systemet och möjliggör en skräddarsydd implementation för varierande ändamål, lokaler och upplägg för olika slags examinationsmoment. Det uppdelade systemet ger därför möjligheten att byta ut exempelvis ankaret mot en med andra antenner och hårdvara som passar sig bättre till de miljöerna.

Det visade sig betydligt svårare än förväntat att i realtid exportera data från GNU Radio till ankarets mjukvara, vilket var ett betydande hinder för projektet. Det skulle för vidare arbete vara givande att utveckla i ett ramverk med bättre utvecklingsstöd, däribland mer utförlig dokumentation då utvecklingen saktades ned av oklara typdefinitioner och dataflöden.

6.1.1 Ankare

Ankaret i sig är den delen av arbetet som tagit mest tid och energi under arbetet. Med facit i hand står det klart att hårdvaran varit den mest problematiska faktorn under projektet. De krav som ställdes på hårdvaran från början var för höga. Att den skulle ha låg kostnad och behandla signaler på ett frekvensband från 450 MHz upp till 6 GHz var inte ett hållbart mål. Det ägnades mycket tid åt att hitta en kombination av SDR-enheter och antenner som uppfyllde kraven. Det visade sig svårt att sammanställa ett system som kunde täcka hela frekvensomfånget och som landade inom projektets budget. Detta medförde att kraven på vilka frekvenser arbetet skulle innefatta fick arbetas om först till WiFi på 2,4 GHz och 5 GHz och efter det till enbart WiFi på 2,4 GHz. Hade dessa avgränsningar gjorts tidigare under projektet hade det funnits större möjligheter att välja alternativ hårdvara som varit smidigare att arbeta med. ADALM-PLUTO krävde mycket tid att ansluta till datorerna och till GNU Radio. Bland annat uppstod det problem med att ansluta två stycken till samma dator vilket krävdes för pejlingalgoritmen.

Begränsningen till enbart 2,4 GHz har inte haft en större inverkan på slutsatsen av arbetet eftersom detekteringen på olika frekvenser i teorin fungerar på samma sätt. Ett byte av hårdvaran från de antenner och SDR:er som använts till andra enheter inriktad på exempelvis de mobila näten kräver inga större omarbetningar av systemet. Det som skulle behöva anpassas är insamlingen av rådata, men det borde vara relativt enkelt att med GNU Radio utöka behandlingen av data för de nya frekvenser. Det viktiga är att strukturen på resultatet som skickas vidare till pejlingalgoritmen från behandlingen förblir konsekvent.

GNU Radio valdes tidigt som det primära mjukvaruverktyget för signalbehandling i projektet. Här uppstod ett antal problem med kompatibilitet mellan olika versionerna av mjukvaran som varit inblandad. För att kunna använda de hårdvaruspecifika mjukvarumodulerna till ADALM-PLUTO krävdes en relativt specifik version av GNU Radio som i sin tur begränsade arbetet på serversidan. Mjukvarumodulerna behövde kompileras och det visade sig därav vara mer tidskrävande att sätta upp utvecklingsmiljön. Allt detta resulterade i att det blev svårare och tog längre tid att få igång och konfigurera den nödvändiga mjukvaran på alla olika datorer, vilket saktade ner arbetets utveckling. Det är möjligt att MATLAB hade visat sig vara ett mer smidigt alternativ till GNU Radio.

6.1.2 Server

Målet med att utveckla en fungerande prototyp av servern som kunde behandla, bearbeta, lagra och skicka vidare datan från ankarna hindrades till stor del av att datan från GNU Radio aldrig kunde skickas in i systemet. Detta gjorde att många antaganden fick tas om hur datan skulle se ut och vilka krav som därav skulle sättas på servern. Resultatet blev att konstgjord testdata fick användas på serversidan för att kunna utvärdera systemets funktionalitet. Detta innebär att det inte är säkerställt att servern hade fungerat tillsammans med ankarna när de istället får sin data direkt från GNU Radio.

En möjlig förbättringsmöjlighet på serversidan är att implementera websockets i servern för att göra det lättare att skicka stora mängder data mellan ankaret och servern samt mellan servern och klienten utan mycket overhead. Det hade exempelvis varit till stor nytta när detektioner vid en viss tidpunkt hämtas från servern till klienten för att ritas ut i kart-vyn.

En vidare förbättringsmöjlighet hade varit att lägga ett större fokus på säkerhet inom systemet. I det nuvarande systemet är det till exempel möjligt för en attackerare, på ett nätverk anslutet till systemet, att låtsas vara ett ankare och således skicka felaktig data till servern. Detta hade lätt kunnat resultera i att servern lokaliserade en potentiell fuskare helt baserad på falsk och felaktig data. Ett sätt att öka säkerheten hade kunnat vara att skriva om hur ankare identifierar sig så att bara giltiga ankare kan registreras i systemet och då även bara tillåta dessa ankare att skicka in detektioner.

Valet av databas stod huvudsakligen mellan två alternativ, MongoDB och InfluxDB. Från början skulle den rådata som samlats in alternativt de FFTer som skapats sparas i databasen. Detta skulle innebära sparande av mycket tidsbaserad data, eftersom InfluxDB är en *time series database*, en databas optimerad för att hantera tidsbaserad data var det ett bra alternativ. Dock hade den mängd data som behövt sparas och skickas från ankare till servern krävt väldigt högpresterande hårdvara och mycket datatrafik sinsemellan och därav sparades varken FFTer eller rådatan, vilket innebar att InfluxDB blev mindre lämpligt att använda. Det fanns även viss data som inte var tidsbaserad, däribland vilka ankare som är registrerade i systemet. Valet blev därför MongoDB som inte är optimerad för tidsbaserad data. Det fanns dessutom enklare bibliotek för att kommunicera med databasen och det blev mindre komplicerat att implementera i vår redan existerande kodbas. InfluxDB kan dock vara ett bra alternativ för framtida utveckling. Ökar man mängden tidsbaserad data blir det ett bättre val, till exempel om man väljer att spara mer av den data som samlas in eller väljer att expandera systemet till fler ankare utspridda över flera rum.

6.1.3 Klienter

Klientdelen av systemet har ett flertal brister och förbättringsmöjligheter. Det finns flera rent estetiska delar som nedprioriterades i detta arbetet eftersom målet var en prototyp, men även vissa funktionella. Ett funktionellt problem på klientsidan är att den i nuläget förlitar sig på lokal testdata eftersom att servern inte implementerar allt som det finns vyer för. Detta är inte ett alltför stort problem då det är relativt lättlöst.

För att göra klienten mer användarvänlig finns det ett antal områden som kan förbättras. I allmänhet så kan alla vyer kunnat förbättras genom att modifiera designen till att vara mer än det absolut minsta som behövs för att representera datan. Detta har inte varit i fokus i det här arbetet eftersom målet har varit en prototyp och inte en färdig produkt. Det är dock viktigt att arbeta med om produkten skulle färdigställas. Utöver de estetiska ändringarna finns det rum för förbättringar när det kommer till interaktionen med klienten. Exempelvis hade det varit önskvärt för användaren att kunna definiera ett tidsintervall istället för en tidspunkt på kart-vyn, eftersom det hade gjort det lättare att få en överblick över hur detektionerna såg ut över tid.

6.2 Val av hårdvara

Valet av hårdvara var komplicerat då ett flertal faktorer påverkade beslutet. Dessutom skedde sökandet efter hårdvara parallellt med sökande av vilken metod som skulle användas, därav förändrades kraven på hårdvara under sökningsarbetet. Utöver det var projektets budget en kraftigt begränsade faktor, detta då hårdvara inom fältet primärt säljs till företag.

Valet av ADALM-PLUTO gjordes primärt av budgetskäl, då den var en av de få SDR-enheter som klarade kravet att täcka 2,4 GHz till 2,5 GHz men samtidigt inte överskred budgeten. Enheten är officiellt specificerad för att ta emot samt skicka signaler mellan 325 MHz och 3,8 GHz [19]. Det finns även möjlighet att genom en mjukvarumodifikation utöka frekvensomfånget till mellan 70 MHz och 6 GHz [20]. Detta hade tillåtit projektet att övervaka WiFi 2,4 GHz, WiFi 5 GHz, Bluetooth, 3G och 4G. De tre faktorerna som avgjorde valet var i första hand att den kunde garanterat levereras snabbt då den fanns i lager hos Chalmers leverantör, vilket var viktigt då utvecklingsarbetet till stor del var låst i väntan på hårdvara. Vidare är ADALM-PLUTO kompatibelt med en rad mjukvarupaket så som GNU Radio och MATLAB. Detta ansågs vara relativt viktigt eftersom det skulle påskynda utvecklingsarbetet, då dessa program ansågs lätta att använda och projektmedlemmar hade tidigare erfarenhet med dem. Slutligen kan enheten kopplas till en dator för dataöverföring och strömtillförsel via USB, vilket innebär att det inte behöver någon extern strömkälla [19].

ADALM-PLUTO använder sig av en lokal oscillator där heterodyning utförs. Om den istället filtrerat signalen utan mixas ned hade det inneburit att samplingsfrekvensen behövt vara mycket hög. Det hade då även varit nödvändigt att använda ett bandpassfilter eftersom signalerna då skulle inneha sin ursprungliga högre frekvens.

Den huvudsakliga motiveringen till valet av antenn var att den skulle vara riktad, vilket är ett krav för att den valda algoritmen skulle kunna fungera. Samtidigt skulle antennen kunna uppfylla de krav som ställdes för projektets mål medan den även var mycket prisvärd.

6.3 Pejlingalgortimen

I valet av pejlingalgoritm behövde projektets begränsningar tas i åtanke. Det var viktigt att den valda algoritmen inte krävde utrustning som överskred projektets budget samt inte heller ställde för höga krav på tillgängliga datorresurser. Den behövde även vara effektiv nog att fungera i samband med att signaldatan behandlas i realtid. Algoritmen behövde även kunna användas när avståndet till sändaren var kort. Många traditionella pejlingtekniker visade sig vara dåligt lämpade för att lokalisera sändare inom ett mindre område. Då en stor del av dessa härstammar från militär användning är de snarare tänkta att lokalisera mål på avstånd av hundratals meter.

TDOA visade sig olämplig för projekt eftersom signalen måste infalla nära parallellt för att TDOA ska ge korrekta värden. Därav måste antingen utrustningen placeras långt ifrån sändarna eller så måste mottagarna placeras nära varandra. Detta eftersom när avståndet B minskar så minskar även sträckan Δtc, se figur 2 alternativt ekvation 2. Eftersom ljusets hastighet (c) inte påverkas, så måste upplösningen på Δt öka för att bibehålla samma vinkelupplösning. Att öka upplösningen på Δt innebär i det här fallet att samplingsfrekvensen behöver öka. Detta ställer höga krav på hårdvaran och visade sig därav inte vara lämpligt inom projektets budget.

Nearst Node är en ganska simpel metod, dock ökar kostnaden snabbt i takt med storleken på ytan som ska täckas eller om precisionen behöver förbättras då detta skulle innebära att extra enheter skulle behöva införskaffas. Eftersom projektets syfte var att utveckla ett system för att verka under examination så var både relativt stora ytor och hög precision viktigt. Därav beslutades att denna metod var olämplig för projektet.

Monopulse-algoritmer är bäst lämpade för projektet då ett kort avstånd till sändare inte är ett problem samt att kraven på hårdvaran var mer konservativa. De viktiga hårdvarukraven för Monopulse är fyra riktade antenner och mottagare med tillräckligt hög precision i amplitudmätning, vilket det finns hårdvara som uppfyller samtidigt som den inte överskred budgeten. Utöver det är algoritmen inte så beräkningsintensiv vilket innebär att kravet på datorresurser inte blir alltför högt och att mjukvaran inte ställs för särskilt tunga prestandakrav vilket skyndar på utvecklingsarbetet. Implementationen av den besluta-

de pejlingsalgoritmen visade sig vara enkel eftersom den inte krävde alltför hög precision eller alltför avancerad hårdvara.

En konsekvens av att hårdvaran inte färdigställdes i sin helhet var att det inte blev möjligt att testa implementation med riktigt indata. Därav var det svårt att definitivt bevisa att implementationen fungerade korrekt. Algoritmen klarade dock det mjukvarutest som utvecklades, vilket ger en indikation till att metoden bör fungera i praktiken.

6.4 Hinder under arbetsprocessen

Under projektets andra halva bedrevs arbetet huvudsakligen på distans på grund av Corona-pandemin. Detta försvårade utvecklingsarbetet då många arbetsuppgifter krävde tillgång till hårdvaran. Som lösning tog två gruppmedlemmar anspråk över varsin hårdvaruenhet. Sedan användes skärmdelning med fjärrstyrning över Zoom för att ge de andra medlemmarna indirekt tillgång till hårdvaran. Detta gjorde fortsatt utveckling med hårdvaran möjlig men långsammare än om arbetet hade kunnat fortgå som innan.

Även samtliga av projektets möten under andra halvan gjordes på distans via Zoom. Här hade de ändrade omständigheterna en mer positiv inverkan. Att schemalägga och bedriva möten utan kravet på fysisk närvaro visade sig mycket smidigt och ledde till regelbundna, dagliga möten. Detta ökade sammanhållningen i gruppen och möjliggjorde en mer kontinuerlig utvärdering av arbetet. Det hade troligtvis varit givande att ha distansmöten tidigare i projektet, redan innan det blev en nödvändighet.

6.5 Etiska aspekter

All typ av övervakning väcker en mängd etiska och rättsliga frågor att ta ställning till. Vem utsätts för övervakningen? Har de gett sitt medgivande? Vilken data samlas in? Vem har tillgång till uppgifterna? Kommer de att sparas under längre tid? Går det att knyta datan till en person i efterhand?

I projektets fall är målet för övervakning studenter under ett examinationsmoment. Med antagandet att institutionen som använder sig av spaningsystemet följer lagen ska studenterna informeras och ge sitt medgivande, troligtvis som ett krav för att få delta i examinationen [26]. Det finns så klart en risk att kommunikation från personer i närheten som inte deltar i examinationsmomentet fångas upp av misstag. Detta är olyckligt men då projektet inte bearbetar eller analyserar denna informationen vidare, samt att denna typ av spaning är nära trivial att bedriva för vem som helst med en telefon eller bärbar dator, har tid eller resurser inte lagts på att motarbeta detta.

Om spaningsdatan ska kunna användas som underlag för en undersökning då det finns misstanke om fusk kommer den att behöva sparas. Frågan blir då hur den ska sparas, hur länge och vem som ska ha tillgång till den. God praxis här är att kryptera datan, inte spara den längre än nödvändigt för undersökningen

och att begränsa antalet personer som har tillgång till den till endast de som verkligen behöver det. Lättare sagt än gjort, inte minst i valet av vem eller vilka som ska ha möjlighet att avkryptera datan. Sedan finns det också externa bestämmelser och lagar som behöver följas, exempelvis GDPR [27].

Den tekniska aspekt som kan vara mest etiskt problematisk är pejling, att från spaningen försöka identifiera källan, personen, till misstänkt kommunikation. Här kan man välja att begränsa sig till att ge mindre specifik information, exempelvis att peka till en del av salen istället. Oavsett hur detaljerad pejlinginformation man ger finns ett ansvar att inte producera felaktig information och att vara tydlig med den potentiella osäkerheten i resultatet.

Det kan tänkas att man kan förhålla sig mer liberalt till en del etiska aspekter i projektet om syftet är forskning, att ta reda på vad som är möjligt, snarare än att utveckla en produkt redo att användas. I så fall måste man vara tydlig med denna målsättning och uppmärksamma vilka delar av det resulterade systemet som skulle behöva justeras för att göra det lämpligt för vidare användning.

6.6 Jämförelse med andra system

Prototypsystemet utvecklat i projektet jämfördes med fyra andra alternativ på marknaden som med varierande grad uppfyllde de krav och begränsningar som sattes. De valda systemen var ett kommersiellt system från Sagax Communications, ett Monopulse-baserat system utvecklat vid Linköping universitet i samarbete med Totalförsvarets forskningsintitut (FOI), en handhållen enhet från Clever Intelligence Unity (CIU) och slutligen Cellbusters Zone Protector. Systemen jämfördes först med kraven och begränsningarna på projektet för att sedan ställas i kontrast mot prototypsystemet.

6.6.1 Sagax Communications system

Sagax Communications hävdar att deras system täcker in ett frekvensband från 40 MHz till 6 GHz, vilket omfattar alla frekvenser de som används av WiFi (både runt 2,4 GHz och 5 GHz), Bluetooth och relevanta mobilnät. Den marknadsförda enheten har även ett inbyggt system för riktningsberäkning [28]. Sagax Communications har dessutom släppt en rapport om denna riktningsberäkning. I rapporten så placerades systemet vid tre positioner och beräknar utifrån dessa riktningen till tre målpunkter. Det maximala felet som uppmättes var 4° och medianfelet var 2° [29]. Det går dock inte att utifrån dessa begränsade datapunkter avgöra om systemets prestanda kvarstår vid kortare avstånd till målen. Givet att vinkelfelet inte försämras när sändaren närmar sig så skulle deras system uppfylla kraven för detta projekt. Sagax Communications uppger inte priset på sina system.

Om systemet exempelvis skulle placeras på kortsidan i en sal vore detta en suboptimal placering då den inte utnyttjar sin totala upptagningsförmåga på 360° samt att det ökar längden på sökområdet. I fallet när sändaren placerades mittemot på motstående kortsida så skulle sökområdet som bildas anta formen

av en likbent triangel. Detta skulle innebära ett sökområde på $\frac{36.82 \tan(4) * 36.8}{2} = 94\ m^2$ utifrån måtten i salen där systemet testades. Alternativt så skulle två sådana system kunna kopplas in som ankare och placeras i hörnen på vardera ände av en kortsida av en sal. Sedan placeras sändare mitt på motstående kortsida. Detta skulle resultera i ett sökområde som definieras av noderna (9,6, 35,8); (12,4, 46,3); (12,4, 29,4) och (15,2, 35.8). Dessa fyra noder resulterar i en fyrkant, där arean kan beräknas genom att dela upp den i fyra rätvinkliga trianglar och sedan summera arean av dem. I innevarande fall så resulterar det i en area på 47 m^2.

6.6.2 Monopulse-system utvecklat vid Linköpings universitet

Systemet utvecklat vid Linköpings universitet är vad i detta projekts prototypsystem motsvarar ett ankare. En enhet från systemet kan avgöra från vilken riktning signalen kommer men flera enheter behövs för att kunna göra en lokalisering. Systemet bygger på specialdesignad hårdvara som inte finns kommersiellt tillgängligt. Det AD9361 IC chip som används i hårdvaran är specificerad för ett frekvensband mellan 80 MHz och 6 GHz. Vid korta avstånd till sändaren antas Signal to Noise Ratio (SNR) vara minst 18 dB [30]. Vilket ger systemet en felmarginal på cirka 3° inom ett spann av 60° framför antennen, se avsnitt 6.12 i [14]. Detta innebär ett sökområde på 27 m^2, i samma situation som för systemet från Sagax ovan.

6.6.3 RFD-10EU från Clever Intelligence Unity

Clever Intelligence Unity (CIU) har vad de kallar en handhållen LTE, 4G, 3G, 2G, GSM och WiFi signaldetektor som heter RFD-10EU [31]. Enligt specifikationerna ska den klara av att hantera nästan alla de frekvenser som WiFi och mobilnät använder sig av i Sverige. Däremot klarar den inte av att hantera 450 MHz bandet som används av NET1 på vissa ställen i Sverige. Det större problemet är att den inte har stöd för att detektera WiFi kring 5 GHz vilket moderna enheter brukar använda. Enheten kan detektera signaler som skickas upp till 15 m från enheten men den har ingen inbyggd pejling av signalernaskällans position. Baserat på hur stark signal som detekteras så avger den ljud som varierar i volym i relation till signalstyrkan av den detekterade signalen. I kombination med den skärm som finns på enheten tillåter användaren att genom att förflytta sig i rummet avgöra positionen av den detekterade signalen.

Priset på enheten visas inte på CIUs hemsida men med tanke på att hårdvaran och den inbyggda mjukvaran inte verkar vara särskilt sofistikerad är troligtvis kostnaden inte alltför hög. Däremot, för att effektivt kunna använda enheterna behövs minst en för varje sal och troligtvis flera i de större salarna vilket kan driva upp kostnaden om det ska användas på stor skala.

Eftersom systemet inte kan detektera signaler på 5 GHz för WiFi och man dessutom manuellt måste avgöra positionen genom manuell testning anses CIUs system inte möta kraven för detta projekt.

6.6.4 Cellbusters Zone Protector

Cellbusters har utvecklat ett system kallat Zone Protector [32]. Systemet är ett signalspanningssystem, specificerat för att detektera signaler på frekvensband mellan 20 MHz och 6 GHz som skickats på ett avstånd från 1,5 m till 30 m beroende på inställning. Därmed så har Zone Protector möjlighet att övervaka de frekvensband som initialt specificerades, det vill säga WiFi på både 2,4 GHz och 5 GHz, Bluetooth samt mobilnät som 3G och 4G. Cellbusters publicerade inte kostnaden för detta system på sin hemsida.

Systemet saknar dessvärre lokaliseringssystem. Därmed kan den enbart användas för att upptäcka sändningar samt ange vilken frekvens och signalstyrka som mottagits. Till skillnad från RFD-10EU så är Zone Protector inte trivial att flytta. Det är inte praktiskt möjligt att lokalisera sändare, vilket som konsekvens gör att Zone Protector enbart kan upptäcka om något sänds, men inte ange var i rummet sändaren befinner sig.

6.6.5 Jämförelse mellan prototypsystemet och andra system

I jämförelse med de teoretiska specifikationerna på prototypsystemet är prototypen det som är bäst lämpat och uppfyller kraven bäst. Varken CIUs RFD-10EU eller Cellbusters Zone Protector har möjlighet att lokalisera varifrån en signal kommer ifrån, vilket prototypsystemet klarar. Därav anses prototypsystemet mer lämpligt för ändamålet.

De andra två systemen, det som utvecklats vid Linköpings universitet och av Sagax Communications är huvudsakligen motsvarigheten till ankaret i prototypsystemet. Eftersom prototypsystemet inte är operativt skulle dess ankare kunna ersättas med någon av dessa lösningar. Däremot skulle dessa system troligvis innebära en högre kostnad som skulle överskrida projektets budget. Båda är dessutom utvecklade för militära ändamål vilket kan vara problematiskt vid användning i civila situationer, eftersom de då kan vara ämnade för vissa frekvenser som främst är avsedda för militär användning [33]. För att kunna koppla samman dessa system med prototypsystemet skulle troligtvis vissa omskrivningar behöva göras på båda sidor, vilket sannolikt ökar kostnaden ytterligare.

Sammanfattningsvis är varken Cellbuster Zone Protector eller CIUs RFD-10EU acceptabla för att detektera fusk, men systemet från Linköpings universitet och det från Sagax Communications skulle kunna ge bra resultat, speciellt om de används som ankare i det prototypsystem som utvecklats. Däremot skulle kostnaden bli relativt hög.

7 Slutsats och sammanfattning

Projektets ursprungliga mål var att konstruera ett system som kan övervaka sändningar och lokalisera källan till kommunikation över WiFi, mobilnät och Bluetooth. Detta avgränsades under arbetets gång till att enbart fokusera på WiFi runt 2,4 GHz. Av budgetanledningar begränsades systemet till enbart en ankarenheten, vilket innebar att arbetet med pejling främst blev teoretiskt. Detta eftersom pejling av en position i praktiken med den valda Monopulse-algoritmen kräver minst två ankarenheter.

Systemet är inte redo att användas. Hårdvarumässigt består det av två SDR-enheter med varsin antenn kopplade till en bärbar dator. Dessa kan fånga upp signaler kring 2,4 GHz, vilket har testats i en större tentamenssal. Signaldatan matades in i GNU Radio där den filtreras och bearbetas för att underlätta fortsatt bearbetning. Tanken är att denna behandlade data sedan skickas vidare till en mjukvaru back-end där beräkningar för pejling och avgörandet om kommunikationen sker över en otillåten kanal görs. Denna brygga mellan hårdvaru- och mjukvarusidan är inte färdig.

Mjukvarusidan består huvudsakligen av två delar: en serverdel för pejling- och detektionsberäkningar samt en klientdel som presenterar resultatet för en användare. Målet var, likt resten av systemet, att framställa en fungerande prototyp, men både serverdelen och klientdelen har i dagsläget ett flertal brister som behöver åtgärdas för att systemet ska kunna sättas i bruk. En stor anledning till att utvecklingen inte har kommit så långt som planerat beror på att hårdvarudelen av projektet har stött på ett flertal hinder. Detta har gjort att mjukvaran till stora delar har designats utifrån antaganden på vilken typ av data som kan förväntas från ankaret. Trots bristerna är mjukvaran en grund som utan större svårigheter kan vidareutvecklas till en fullt fungerande produkt.

Det finns flera olika sätt att fortsätta utvecklingen med systemet. För att kunna detektera en större bredd av fusk kan kapacitet att hantera andra typer av kommunikation, exempelvis mobilnät, Bluetooth och 5 GHz WiFi, med hjälp av ytterligare hårdvara samt vissa ändringar i mjukvaran vara aktuellt. Nuvarande prototyp har en ankarenhet bestående av två SDR-enheter med en antenn vardera. För att bedriva pejling i praktiken krävs två ankare och dessa behöver då sammankopplas för att synkronisera mätningarna.

I syfte att förbättra pejlingalgoritmens noggrannhet skulle felmarginalen kunna förbättras med flera beräkningar av positionen. Detta skulle kunna göras exempelvis med hjälp av en heatmap, vilket borde resultera i ett relativt litet område efter ett antal detektioner. Detta utefter antagandet att om mätfelen sker slumpmässigt inom det specificerade intervallet så bör positionsestimaten med felmarginal alltid innehålla sändare men sällan hamna på samma plats. Därav bör antalet positionsestimat som innehåller sändare vara högre än omgivingen och då bör positionsestimatet bli bättre desto fler som används. Användning av likande algoritmer skulle kunna möjliggöra användning av billigare system med större vinkelfelmarginal och ändå leverera en liten sökyta.

Sammanfattningsvis är det utvecklade systemet inte färdigt att användas. Dock finns mycket av den nödvändiga funktionaliteten på plats. För att få ett färdigt system krävs det att bryggan mellan hårdvaran och mjukvaran hos ankaret färdigställs. Förutom det finns det en rad andra vidareutvecklingsområden för att förbättra systemets noggrannhet och funktionalitet. Dessutom krävs mer utförlig testning för att verifiera systemets precision och prestanda.

Referenser

[1] Martin Appel, "Så fuskar dina elever – och så stoppar du det," April 2018, [Hämtad 2020-02-04]. [Online]. Tillgänglig: https://tidningengrundskolan.se/sa-fuskar-dina-elever-och-sa-stoppar-du-det/

[2] Chalmers tekniska högskola, "Digital tentamen," Januari 2020, [Hämtad 2020-02-04]. [Online]. Tillgänglig: https://student.portal.chalmers.se/sv/chalmersstudier/tentamen/Sidor/Digital-tentamen.aspx

[3] Scrum.org, "What is scrum?" April 2020, [Hämtad 2020-04-12]. [Online]. Tillgänglig: /https://www.scrum.org/resources/what-is-scrum

[4] IEEE Computer Society, "Ieee standard for information technologytelecommunications and information exchange between systemslocal and metropolitan area networks specific requirements," December 2016. [Online]. Tillgänglig: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7786995

[5] K. Fransén, "The swedish, telecommunications market first half year 2019," [Hämtad 2020-04-28]. [Online]. Tillgänglig: https://statistik.pts.se/media/1484/swedish-telecoms-market-en-1h-2019_t.pdf

[6] Induo AB, "Frekvenser för 5g, 4g, gsm och 3g," [Hämtad 2020-02-08]. [Online]. Tillgänglig: https://www.induo.com/s/g/gsm-3g-4g-frekvensband/

[7] D. S. Johannes K Becker, David Li, "Tracking anonymized bluetooth devices," Mars 2019, [Hämtad 2020-02-08]. [Online]. Tillgänglig: https://content.sciendo.com/view/journals/popets/2019/3/article-p50.xml?tab_body=pdf

[8] D. H. Smith, "Software defined radios - architectures, systems and functions." [Online]. Tillgänglig: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170005302.pdf

[9] J. Verhaevert and P. Van Torre, "Design and realization of a 2.45 ghz transmitter and receiver as a modular unit for a mimo sdr," in *2015 Loughborough Antennas Propagation Conference (LAPC)*, 2015, pp. 1–4, DOI: 10.1109/LAPC.2015.7366044.

[10] gnuradio.org, "Gnuradio, the free and open software radio ecosystem," April 2020, [Hämtad 2020-04-23]. [Online]. Tillgänglig: https://www.gnuradio.org/

[11] C. A.Balanis, *Antenna Theory*, 4th ed. John Wiley & Sons, Incorporated, 2016, ISBN 978-1-118-64206-1.

[12] S. N. Alan V.Oppenheim, Alan S.Willsky, *Signals and Systems*, 2nd ed. Pearson, 2014, ISBN 978-1-292-02590-2.

[13] D. K. B. Samuel M. Sherman, *Monopulse Principles and Technices*, 2nd ed. Artech House, 2011, ISBN: 978-1-60807-1-753.

[14] A. Patriksson, "Radio signal doa estimation," Master's thesis, Linköping University, 2019.

[15] A. D. Martino, *Introduction to modern EW systems, Second Edition.* 685 Canton Street Norwood, MA: Artech House, 2018, ch. 4, pp. 237–285.

[16] Mozilla Contributors, "An overview of http," Maj 2020. [Online]. Tillgänglig: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

[17] ——, "Http request methods," Maj 2020. [Online]. Tillgänglig: https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods

[18] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[19] amiclaus, *ADALM-PLUTO Detailed Specifications*, ANALOG DEVICES, 2018. [Online]. Tillgänglig: https://wiki.analog.com/university/tools/pluto/devs/specs

[20] RTL-SDR.com, "Adalm-pluto sdr hack: Tune 70 mhz to 6 ghz and gqrx install," 2017. [Online]. Tillgänglig: https://www.rtl-sdr.com/adalm-pluto-sdr-hack-tune-70-mhz-to-6-ghz-and-gqrx-install/

[21] Delock, "Delock wlan antenna - datasheet." [Online]. Tillgänglig: https://www.kjell.com/globalassets/mediaassets/513253_30107_datasheet_en.pdf

[22] MongoDB, Inc., "Mongodb," Maj 2020. [Online]. Tillgänglig: https://www.mongodb.com/faq

[23] B. Karlström, *Kretsanalys*, 1st ed. Studentlitteratur, 2017, ISBN: 978-91-44-11769-0.

[24] A. Devices. Why Pluto. [Online]. Tillgänglig: https://wiki.analog.com/university/tools/pluto/users/name

[25] P. Ghelfi, F. Scotti, D. Onori, and A. Bogoni, "Photonics for ultrawideband rf spectral analysis in electronic warfare applications," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 25, no. 4, pp. 1–9, 2019, DOI: 10.1109/JSTQE.2019.2902917.

[26] Infrastrukturdepartementet, "SFS 2003:389 Lag om elektronisk kommunikation." [Online]. Tillgänglig: http://rkrattsbaser.gov.se/sfst?bet=2003:389

[27] European Parliament and Council of the European Union, "General data protection regulation," Maj 2016. [Online]. Tillgänglig: https://gdpr-info.eu/

[28] "Integrated sigint and comint systems," Sagax Communications. [Online]. Tillgänglig: https://sagaxcommunications.com/products-sigint-comint-systems/

[29] "Multimission direction finding receiver sdf-3000 field trial report," Sagax Communications. [Online]. Tillgänglig: https://sagaxcommunications.com/signals-intelligence-resources/sigint-direction-finding-field-trial/

[30] R. DIONICIO, "Snr – signal to noise ratio – what is it?" 2015. [Online]. Tillgänglig: https://www.packet6.com/what-is-snr-signal-to-noise-ratio/

[31] "Lte 4g 3g 2g gsm wi-fi bug mobile phone detector anti-cheating in exam for europe." [Online]. Tillgänglig: https://www.cleverintelligenceunity.tw/en/product-444725/LTE-4G-3G-2G-GSM-Wi-Fi-Bug-Mobile-Phone-Detector-Anti-\/-Cheating-in-Exam-for-Europe-RFD-10EU.html

[32] "Zone protector™ – detect cell phones & hidden transmission devices." [Online]. Tillgänglig: http://www.cellbusters.com/cell_phone_detector_zone_protector

[33] M. Wikberg, "Rapport:inventering frekvensanvändning över 6 ghz," 05 2017, ISSN 1650-9862.