DE HAAGSE HOGESCHOOL

Quintor

Welke communicatietechniek is inzetbaar voor PIN-betalingen wanneer het mobiele netwerk overbelast is?

Afstudeerverslag

Jordy Ubink

Afstudeerverslag

Welke communicatietechniek is inzetbaar voor PIN-betalingen wanneer het mobiele netwerk overbelast is?

Afstudeerperiode: 2020-2021
Opdrachtgever: Quintor

Jordy Ubink

De Haagse Hogeschool, Delft

Eerste druk
24 maart 2021

Administratief

Afstudeerblok: 2020-2.1
Startdatum uitvoering afstudeeropdracht:.... 31 augustus 2020
Inleverdatum afstudeerdossier:..... 26 maart 2021

Achternaam: ..... Ubink
Voorletters:..... J.M.
Roepnaam:..... Jordy

Opleiding:..... HBO-ICT
Differentiatie:..... Network & System Engineering
Afstudeerprogramma: ..... Internet of Things
Domein: ..... Internet of Things
Locatie school: ..... Delft

Naam begeleidend examinator:..... Gert den Neijsel
Naam expert examinator: ..... Karel van der Lelij

Bedrijf

Naam: ..... Quintor
Bezoekadres:..... Lange Vijverberg 4-5, Den Haag
Postcode: ..... 2513 AC
Plaats: ..... Den Haag

URL: ..... https://quintor.nl

Opdrachtgever

Achternaam: ..... Tillema
Voorletters:..... J.
Titel:..... Ingenieur
Functie:..... CEO

Bedrijfsmentoren

Achternaam: ..... ten Hoor
Voorletters:..... S.
Titel:..... Ingenieur
Functie:..... Java Developer

Voorwoord

Van 31 augustus 2020 tot 26 maart 2021 heb ik gewerkt aan de laatste fase van mijn studie: de afstudeeropdracht. Ik studeer HBO-ICT: Network en System Engineering aan de Haagse Hogeschool.

Tijdens deze opdracht heb ik verschillende communicatieprotocollen onderzocht. Volgens heb ik het meest kansrijke protocol toegepast in een project. Deze opdracht past goed bij mijn afstudeerrichting en interesse: Internet of Things.

Dit project heb ik uitgevoerd voor Quintor. Ik wil mijn stagebegeleider, Sander ten Hoor, bedanken voor zijn begeleiding tijdens mijn afstudeeropdracht.

Jordy Ubink
Leiden, 26 maart 2021

Referaat

Bij festivals is het mobiele internetnetwerk erg instabiel. Dat komt doordat er veel mensen met een mobiele telefoon zich op dezelfde plek bevinden. Deze instabiliteit zorgt ervoor dat PIN-transacties zullen mislukken. Tijdens deze afstudeeropdracht wordt een oplossing voor dit probleem gezocht. De onderzoeksvraag luidt: "*Welke communicatietechniek is inzetbaar voor PIN-betalingen wanneer het mobiele netwerk overbelast is?*"

Na een uitgebreid vooronderzoek is het LTE-M protocol gekozen om verder te onderzoeken. Er is een mock-bankapplicatie gerealiseerd om dit protocol te testen. Hiermee zijn PIN-betalingen in verschillende omgevingen gesimuleerd.

Het LTE-M protocol bleek zeer stabiel te zijn. Alle transacties zijn succesvol verwerkt, zelfs in omstandigheden met weinig signaalsterkte. Dit protocol kan dus gebruikt worden voor PIN-betalingen bij festivals.

Inhoudsopgave

Bijlagen

1. Begrippenlijst

In deze lijst staan de complexe begrippen kort uitgelegd. De begrippen staan ook in de voetnoten van dit verslag, wanneer de termen geïntroduceerd worden.

Begrip	Definitie
De casus	"*Welke communicatietechniek is inzetbaar voor PIN-betalingen wanneer het mobiele netwerk overbelast is?*" Deze opdracht wordt aangehouden als leidraad.
De opdracht / het project	Het vinden van een geschikt communicatieprotocol.
Mobiel netwerk	Technische term: Cellular network. Een netwerk waarin apparaten (zoals telefoons) verbonden zijn met een zendmast van een ISP.
Plan van Aanpak	Het concrete plan voor het uitvoeren van de opdracht, gemaakt in fase 2 (definitiefase).
Proof of Concept	En aantal tests om aan te tonen dat een standaard voldoet aan de gestelde eisen.
Protocol	In deze context: Een gestandaardiseerde techniek die gebruikt kan worden voor communicatie.
ANVP-X	Elke user story heeft een ID, bijvoorbeeld ANVP-3. De afkorting staat voor "Alternatief netwerk voor PIN-verkeer".
AT-commando	De commando's waarmee een modulatiechip wordt geconfigureerd. Deze commando's zijn te vinden in de handleiding van de fabrikant.
Bericht	Bij communicatie wordt data in 1 of meerdere (internet-)pakketten verstuurd. Dit zijn berichten.
Cat	LTE-Categorie/versie. Elke categorie heeft andere eigenschappen, zoals snelheid en energiezuinigheid.
CI/CD	Continuous Integration (CI): Het automatisch testen en analyseren van code bij elke commit. Continuous Delivery (CD): Het automatisch 'builden' van de software, eventueel met automatische deployment.
Co-existence	De mate waarin een protocol bestand is tegen interferentie van andere protocollen.
dBm	Het zend- of ontvangvermogen in mW op een logaritmische schaal. Voorbeeld: 30dBm = 1000mW.
DLE	Data Length Extensions. Een alternatief frame-type voor Bluetooth waardoor berichten een payload kunnen hebben van 251 bytes in plaats van 27 bytes.
Duty Cycle	De verhouding van de actieve zendtijd ten opzichte van een periode. De maximale duty cycle van LoRa is gebruikelijk 1%.
FEC	Forward error correction: Het redundant sturen van data, zodat eventuele errors bij het ontvangen van data gecorrigeerd kunnen worden.
Hop	Een verbinding tussen 2 eindbestemmingen is opgebouwd uit meerdere hops. Elke tussenstap naar de volgende node, switch of router wordt een 'hop' genoemd.
IEEE	Institute of Electrical and Electronics Engineers. Een organisatie die technische standaarden ontwikkelt.
IoT	Internet of Things. Een netwerk dat bestaat uit apparaten en sensors die (op grote afstand) gegevens kunnen versturen en ontvangen.
ISP	Internet Service Provider. Een bedrijf dat (mobiel) internet verschaft. Bijvoorbeeld: KPN, T-Mobile, Vodafone.
Laag	Een laag uit het OSI-model. Communicatietechnieken zijn opgebouwd uit de lagen: Fysieke- (1), data-link- (2), netwerk- (3), transport- (4) en applicatielaag (5-7).
Minimum viable product	Een deels-werkend product. De opdrachtgever geeft aan de hand hiervan feedback voor het verloop van de rest van het project.
Modem	De hardware die het mogelijk maakt om via het mobiele netwerk te communiceren.

Begrip	Definitie
MTU	Maximum Transmission Unit: Het maximale aantal bytes dat past in een packet of frame van het gespecificeerde protocol.
mW	milliWatt of 0,001 Watt. Het vermogen voor bijvoorbeeld het zenden of+C61 ontvangen van RF-data.
Netwerk-sniffer	Een programma dat alle verzonden en ontvangen internetpackets van een computer kan lezen.
Node	Een mesh-netwerk is opgebouwd uit apparaten (nodes) die berichten kunnen doorsturen naar andere nodes in de buurt.
Open standaard	Een protocol waarvan de specificatie openbaar is, waardoor de ontwikkelaar hoopt dat er veel vraag naar ontstaat en dat chipfabrikanten dit protocol zullen implementeren in hun producten.
Payload	De data uit de applicatielaag die verstuurd dient te worden.
Postman	Een programma om HTTP-requests te sturen naar een server.
Private network	Een mobiel netwerk dat alleen beschikbaar is voor de afnemer. Hiervoor worden (tijdelijke) zendmasten of base stations opgezet.
Raspberry Pi	Een eenvoudige, kleine computer met veel interfaces. Dit board wordt gebruikt in het Proof of Concept.
Spatial scope	Een categorie voor de geografische afstand tussen communicerende apparaten waarin een protocol valt.
Superseded	Letterlijk: opgevolgd. De status van een verouderd protocol als het wordt vervangen door een nieuwer protocol. Alleen de nieuwste protocollen worden meegenomen in het onderzoek.
Symbol rate	De snelheid waarmee symbolen verzonden worden. Een symbool bestaat uit 1 of meer bits, afhankelijk van de modulatietechniek.
Throughput	De hoeveelheid nuttige data die verzonden kan worden per tijdseenheid.
Unit test	Een testmethode voor de kleinst testbare softwareonderdelen. Deze onderdelen worden getest in isolatie van de rest van de applicatie.
Uplink/downlink	Uplink: Een bericht dat van end-device naar een base station verstuurd wordt. Downlink: Een bericht dat van een base station naar een end-device verstuurd wordt.

2. Inleiding

De wereld wordt steeds moderner: Mensen worden luier, apparaten worden steeds meer aan het internet gekoppeld en steden worden ‘smart cities’. Het Internet of Things (IoT) wordt steeds groter. Het softwarebedrijf Quintor wil inspelen op dit concept, maar heeft weinig ervaring op dit vakgebied. Daarom wordt een onderzoek uitgevoerd naar mogelijke communicatietechnieken.

Een scenario waarbij communicatie een kritisch proces is, zijn de PIN-betalingen op festivals. Bij drukke evenementen is het mobiele netwerk overbelast, waardoor alle apparaten een slechte internetverbinding hebben. Het gevolg hiervan is dat PIN-terminals niet draadloos kunnen werken. Om hier een oplossing voor te bieden, wordt met deze afstudeeropdracht naar het antwoord gezocht op de vraag: “Welke communicatietechniek is inzetbaar voor PIN-betalingen wanneer het mobiele netwerk overbelast is?”

In dit document staat de zoektocht beschreven naar de verschillende protocollen. Deze afstudeeropdracht is gericht op stabiele verbindingen bij drukbezochte festivals. Eerst wordt op internet onderzoek gedaan naar verschillende draadloze communicatieprotocollen. Het LTE-M protocol voldoet aan de gestelde eisen en wordt onderworpen aan een Proof of Concept.

Voor het Proof of Concept wordt een mock-bankapplicatie gerealiseerd. Hiermee worden PIN-transacties gesimuleerd en het geselecteerde protocol getest. Bij deze test wordt de reactietijd, slaagkans en signaalsterkte bijgehouden.

Na het uitvoeren van dit project, is de onderzoeksvraag beantwoord: LTE-M is een zeer betrouwbaar protocol en het kan worden ingezet voor PIN-betalingen bij festivals. Met de geschreven software en documentatie kan Quintor het protocol toepassen in bestaande en nieuwe projecten.

3. Opdrachtomschrijving

In dit hoofdstuk staat de aanleiding van dit onderzoek beschreven en wat de afstudeeropdracht inhoudt. Dit hoofdstuk is een samenvatting van de opdrachtomschrijving. Het volledige, originele afstudeerplan is te vinden in Bijlage D.

Aanleiding

Op festivals en andere drukbezochte evenementen is het mobiele netwerk vaak overbelast. Dit komt door de vele aanwezige smartphones die via dezelfde zendmasten verbonden willen blijven met het internet. Als gevolg hiervan wordt de internetverbinding instabiel voor alle telefoons en andere apparaten. Ook zullen PIN-betalingen regelmatig mislukken als de betaalterminals verbonden zijn met het mobiele netwerk.

Probleemstelling

Wanneer pinnen niet meer mogelijk is, zal dit leiden tot frustratie bij klanten en organisatoren. Dit komt doordat:

- Klanten nu contant moeten betalen, als ze al contant geld bij zich hebben;
- Mensen in de wachtrij ongeduldig worden, omdat contant betalen langer duurt dan pinnen;
- Eigenaren van festivals en kraampjes minder omzet zullen maken.

Bij festivals zal deze onrust soms leiden tot onvoorspelbaar en ongewenst gedrag van de bezoekers. De probleemstelling wordt beschreven als volgt:

Probleemstelling: Veel PIN-betalingen mislukken bij grote evenementen, waarvoor een oplossing voor bedacht moet worden.

Doelstelling

Het doel van deze afstudeeropdracht is het oplossen van het eerdergenoemde probleem. Ondanks dat het mobiele netwerk overbelast is, kan er wel draadloos gecommuniceerd worden op andere manieren. Zoals de titel van dit document suggereert, worden alternatieve communicatietechnieken hiervoor onderzocht. Via dit alternatieve netwerk zullen PIN-betalingen betrouwbaar uitgevoerd moeten worden.

De hoofdvraag wordt beantwoord aan de hand van de volgende deelvragen:

deelvraag 1.	Met welke kenmerken onderscheiden protocollen zich?
deelvraag 2.	Hoe wordt de opdracht aangepakt?
deelvraag 3.	Welk protocol voldoet aan die eisen?
deelvraag 4.	Voldoet dat protocol ook aan de eisen volgens het Proof of Concept?

Fasering

Dit onderzoek is te verdelen in 5 fases, te zien in Bijlage A - Planning. Bij het lezen van dit document, is het handig om deze planning bij de hand te houden. De fases zijn als volgt:

fase 1:	Er wordt oriënterend onderzoek gedaan rond de opdracht.
fase 2:	De projectsituatie en eisen worden vastgesteld in een Plan van Aanpak.
fase 3:	Er wordt breed en diep onderzoek gedaan naar de specificaties van communicatieprotocollen, waarna een protocol wordt geselecteerd voor verder onderzoek.
fase 4:	Het meest-geschikte protocol wordt getest door middel van een Proof of Concept. Hiermee wordt getoetst op de eisen uit fase 2.
fase 5:	Documentatie wordt geschreven over de implementatie van het gekozen protocol.

Verwacht resultaat

Na het uitvoeren van het Proof of Concept en het documenteren van de implementatie van het te kiezen protocol, is het project afgerond. Er zijn 2 uitkomsten mogelijk:

Uitkomst 1: Het onderzochte protocol voldoet aan de gestelde eisen:
Het protocol is geschikt en kan worden ingezet bij festivals en voor andere use cases.

Uitkomst 2: Het onderzochte protocol voldoet *niet* aan de gestelde eisen:
Het protocol voldoet niet aan 1 of meerdere eisen uit het pakket van eisen. Hierdoor is het ongeschikt voor het gebruik op festivals. Daarentegen zou Quintor dit protocol wel kunnen toepassen voor andere use cases. Daarom wordt het Proof of Concept alsnog volledig uitgewerkt en wordt er niet gewisseld naar een ander protocol. Er wordt geprobeerd om het project zo compleet mogelijk te maken, door om de tekortkomingen van het protocol heen te werken.

In beide uitkomsten is antwoord gegeven op de vraag: “Is het gekozen protocol inzetbaar voor PIN-betalingen als het mobiele netwerk overbelast is?”.

4. Werkzaamheden

In dit hoofdstuk staan de werkzaamheden van de afstudeerder beschreven. Deze zijn verdeeld over vijf fases, zie Bijlage A - Planning. Op de keuzes in deze fase wordt gereflecteerd in hoofdstuk 6.

4.1. Fase 1: Oriënterend onderzoek rondom afstudeeropdracht

Beroepstaken: A-1, G-c

In deze fase is het probleemdomein onderzocht. Tijdens de oriëntatiefase is er oppervlakkig en breed onderzoek gedaan binnen het probleemdomein. De kennis uit dit korte vooronderzoek is noodzakelijk voor het realiseren van het Plan van Aanpak in de volgende fase. De deelvraag die in deze fase beantwoord wordt luidt:

Deelvraag 1: Met welke kenmerken onderscheiden protocollen zich?

Om antwoord te krijgen op deze deelvraag, zijn de volgende vragen opgesteld:

4.1.1. Welke protocollen bestaan er en wat zijn de verschillen tussen die protocollen?
4.1.2. Wat zijn de reeds bestaande oplossingen voor internetverkeer bij festivals?
4.1.3. Op welke manier hebben draadloze verbindingen invloed op elkaar?
4.1.4. Welke overige informatie is nu over het hoofd gezien, maar is relevant voor deze opdracht?

4.1.1. Verschil tussen draadloze protocollen

In de oriëntatiefase zijn 40 protocollen gevonden en oppervlakkig onderzocht. In deze paragraaf worden alleen de kenmerken ervan aangekaart. In fase 2 worden er eisen gesteld aan deze eigenschappen. De uiteindelijke lijst met gevonden protocollen na afloop van het project is te zien in Bijlage B. Een volledig overzicht van de eigenschappen van alle protocollen is te zien in Bijlage C.

Type protocollen

Draadloze lange-afstandsprotocollen kunnen verdeeld worden in 3 hoofdcategorieën:

Categorie 1: LPWAN

LPWAN staat voor low-power wide-area network. De energiezuinigheid is niet een relevante eigenschap voor dit onderzoek. De grote afstand waarmee deze protocollen kunnen communiceren daarentegen is wel relevant. Bij het onderzoeken van deze categorie moet er rekening gehouden worden met de volgende limitaties:

- Throughput
- Grootte van de berichten[1]
- Aantal berichten per uur of per dag.

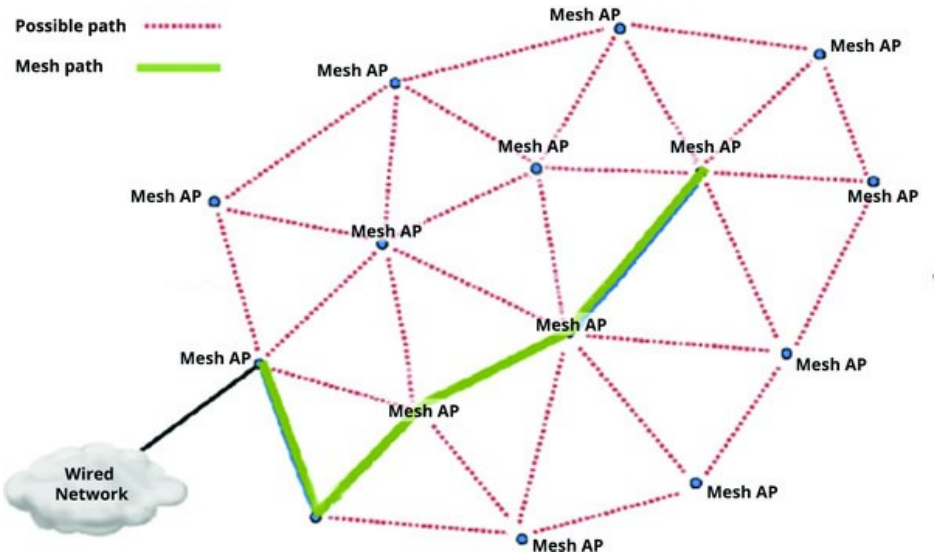
De betekenis van deze eigenschappen is hieronder beschreven bij ‘Technische kenmerken’.

Categorie 2: Mesh-netwerken

Een Mesh-netwerk is een netwerk-infrastructuur van nodes[2] die verbonden zijn met nodes in de buurt. Door deze ‘bruggen’ kan er een pad gevormd worden vanaf elke node naar de internet-gateway, zie figuur 1. Elke extra node kan het communicatiebereik vergroten.

---

[1] **Bericht:** Bij communicatie wordt data in 1 of meerdere (internet-)pakketten verstuurd. Dit zijn berichten.
[2] **Node:** Een mesh-netwerk is opgebouwd uit apparaten (nodes) die berichten kunnen doorsturen naar andere nodes in de buurt.



figuur 1: Concept van een mesh-netwerk. Nodes verbinden via andere nodes die zich dichter bij de gateway bevinden. [1]

Categorie 3: Mobiele netwerken (Cellular networks)

In een mobiel netwerk verbinden apparaten zich met de zendmast van een ISP[3]. Een groot voordeel hiervan is dat er al nationale en internationale dekking is. Een nadeel is dat de gebruiker een abonnement moet afsluiten en niet zelf controle heeft over de netwerkinfrastructuur.

Doordat het gebruikelijke mobiele netwerk overbelast is, is het niet een geschikte oplossing voor deze opdracht, maar er zijn wel andere mogelijkheden om op deze manier een werkend systeem te maken:

- Private network: Een eigen netwerkinfrastructuur door het opbouwen van eigen zendmasten.
- Application Priority: Voorrang krijgen boven minder kritieke datastromen, zie hoofdstuk 4.3.3 – LTE-M.
- Andere generatie: Het gebruik van modulatietechnieken uit een andere telefoongeneratie, zie hoofdstuk 4.3.3 – LTE-M.

(Realistische) communicatie-afstand

Een van de belangrijkste kenmerken voor het uit te zoeken protocol is de haalbare communicatie-afstand. De maximale afstand tussen twee apparaten hangt af van de volgende factoren:

Zendvermogen en zendtijd

Bij een hoger zendvermogen zal het te versturen signaal verder komen. Andersom kan het zendvermogen ook beperkt worden, zodat het energieverbruik wordt teruggedrongen. Daarnaast is de signaalsterkte afhankelijk van de antenne van de zender en van de ontvanger.

Nationale en internationale wetten kunnen restricties opleggen aan het maximale zendvermogen en de duty cycle[4]. Zo zijn LoRa-apparaten in de meeste gevallen gelimiteerd tot een duty cycle van 1% [2].

Fout-oplossend vermogen

Wanneer apparaten te ver uit elkaar staan, zal het signaal wegvallen of gestoord worden door andere radiogolven. Hier kan op 2 manieren mee omgegaan worden:

---

[3] **ISP:** Internet Service Provider. Een bedrijf dat (mobiel) internet verschaft. Bijvoorbeeld: KPN, T-Mobile, Vodafone.

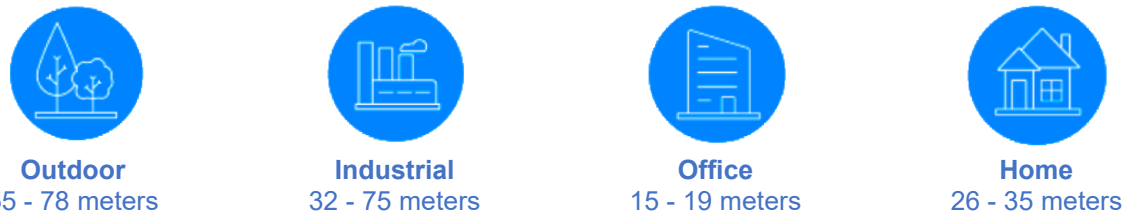
[4] **Duty Cycle**: De verhouding van de actieve zendtijd ten opzichte van een periode [51]. De maximale duty cycle van LoRa is gebruikelijk 1%.

Error-detection: CRC (Cyclic Redundancy Check) is een checksum waarmee gecontroleerd wordt of een bericht intact is.

Error correction: FEC[5] (Forward Error Correction) is een techniek waarbij de bits uit een bericht redundant verstuurd worden. Als een deel van het bericht of van de FEC-bits corrupt is geraakt, kunnen deze elkaar corrigeren en zo weer een correct bericht vormen. Dit werkt echter niet als een groot deel van een bericht verstoord is.

Omgeving

De omgeving is ideaal wanneer er zich geen objecten tussen de zender en de ontvanger bevinden. Ook heeft het materiaal van deze obstakels invloed op de signaalsterkte. Daarom geven specificaties van protocollen vaak de communicatieafstand voor binnen- en buitengebruik aan, zie figuur 2. Signalen van een lagere frequentie (langere golflengte) zullen gemakkelijker door compacte elementen, zoals beton, penetreren.



figuur 2: Invloed van de omgeving op de communicatieafstand. De gemelde afstanden zijn voor Bluetooth LE 1M (standard Bluetooth) [3].

Technische kenmerken

Deze paragraaf beschrijft technische eigenschappen van protocollen die niet met de communicatieafstand te maken hebben.

Payload-grootte

Bij het ontwerpen van een systeem moet er rekening gehouden worden met de MTU[6]. MTU staat voor de hoeveelheid data dat een protocol kan inkapselen (encapsulate). Een voorbeeld hiervan is te zien in figuur 3. Hier is te zien dat bij een TCP/IP segment de grootte gelimiteerd wordt door de MTU van een ethernet frame (1500 bytes). (Hoewel TCP/IP wordt benoemd in figuur 3, is dit slechts een voorbeeld. De keuze voor de te implementeren protocollen komt later in het project aan bod.)

De payload[7] in de applicatielaag[8] wordt gezien als 'nuttige' bytes. De rest van het bericht is overhead om het bericht succesvol te versturen. Veel LPWAN-protocollen hebben een erg lage MTU, zie tabel 1. Als de payload groter is dan toegestaan, zal het gesplitst moeten worden over meerdere packets. Daarom moet bij het uitzoeken van een geschikt protocol rekening gehouden worden met de te verzenden data. Daarnaast kunnen gewenste protocollen uit de netwerk- en transportlaag niet gebruikt worden, als de MTU in een data-linkprotocol te beperkt is.

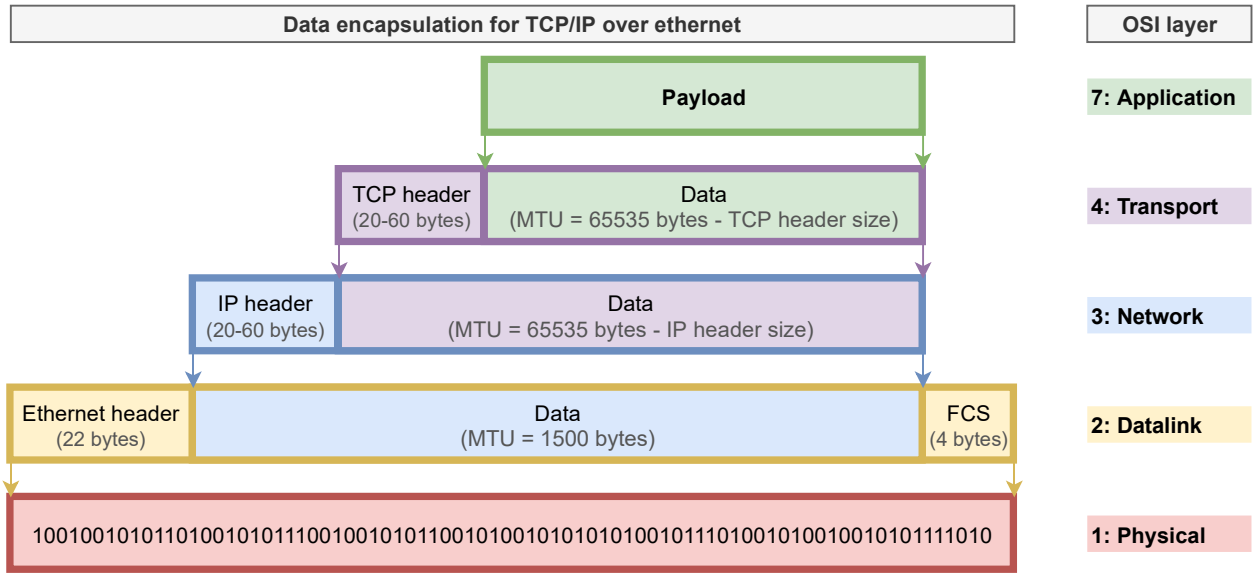
---

[5] **FEC:** Forward error correction: Het redundant sturen van data, zodat eventuele errors bij het ontvangen van data gecorrigeerd kunnen worden.

[6] **MTU:** Maximum Transmission Unit: Het maximale aantal bytes dat past in een packet of frame van het gespecificeerde protocol.

[7] **Payload:** De data uit de applicatielaag die verstuurd dient te worden.

[8] **Laag:** Een laag uit het OSI-model. Communicatietechnieken zijn opgebouwd uit de lagen: Fysieke- (1), data-link- (2), netwerk- (3), transport- (4) en applicatielaag (5-7).



figuur 3: Een encapsulation voorbeeld voor het uitleggen van de MTU. MTU is de maximale hoeveel data dat een protocol kan inkapselen. De grootte van het bericht in dit voorbeeld wordt beperkt door de grootte van een ethernet-frame (MTU = 1500). Ingekapselde lagen worden ook beperkt door de MTU van de laag erboven.

Protocol	Payload grootte (bytes)
LoRaWAN	51-247
Bluetooth	27 (standaard) / 251 (DLE[9])
Sigfox	100/600
Ethernet-frame	1500

tabel 1: Payload van verschillende protocollen. Deze gegevens komen uit verder onderzoek in fase 3.

Aantal berichten per dag

Wanneer een (LPWAN-)protocol gekoppeld wordt aan de infrastructuur van een ander bedrijf, wordt er vaak een daglimiet gesteld op het aantal berichten dat verzonden (uplinks) en ontvangen (downlinks) kan worden, zie tabel 2. Bij een PIN-transactie is zowel het aantal uplinks als downlinks van belang.

Protocol / Infrastructuur	Uplinks	Downlinks
LoRaWAN van KPN [4]	Onbeperkt (max duty cycle: 1%)	3 / uur
LoRaWAN van The Things Network [2]	30 seconden / dag	10 / dag
Sigfox [5]	6 / uur	4 / dag

tabel 2: Restricties op het aantal uplinks en downlinks voor LoRaWAN en Sigfox per device.

Niet-technische kenmerken

Deze paragraaf beschrijft de niet-technische eigenschappen die van belang zijn voor dit project.

Hardware

Er bestaan verrassend veel communicatieprotocollen waarvoor geen hardware beschikbaar is. Hier kunnen verschillende redenen voor zijn:

- Het protocol is erg nieuw.
- Het protocol is een open standaard[10] die niet geïmplementeerd wordt door chipfabrikanten.

---

[9] **DLE:** Data Length Extensions. Een alternatief frame-type voor Bluetooth waardoor berichten een payload kunnen hebben van 251 bytes in plaats van 27 bytes.

[10] **Open standaard**: Een protocol waarvan de specificatie openbaar is, waardoor de ontwikkelaar hoopt dat er veel vraag naar ontstaat en dat chipfabrikanten dit protocol zullen implementeren in hun producten.

Daarnaast zijn sommige protocollen gemakkelijker te implementeren dan anderen. Dat hangt af van de complexiteit van het protocol en of het protocol gestandaardiseerd is door operating systems of applicaties.

Bekendheid van het protocol

Bekende protocollen worden praktisch gezien beter ondersteund dan onbekende protocollen. Ook is het makkelijker om daar documentatie over te vinden. Wanneer deze informatie niet openbaar toegankelijk is, is het lastig om het protocol te implementeren.

Status

De protocollen die ontwikkeld zijn als IEEE-standaard[11], hebben een status. Deze kan onder andere "superseded" en "withdrawn" zijn. Protocollen die niet meer worden ondersteund, zijn geen goede optie voor dit project en vallen af. De IEEE definieert deze statussen als [6]:

Superseded standard The standard has been replaced by a new standard.
Withdrawn standard The standard is no longer market relevant or active.

4.1.2. Reeds bestaande oplossingen

Er is onderzoek verricht naar de reeds bestaande oplossingen voor de slechte verbinding bij festivals. Een oplossing hiervoor is het opzetten van een private network[12]. Veel informatie hierover is te vinden via de bedrijven MCS en GIGTECH. MCS heeft IoT-oplossingen[13] voor veel verschillende zaken en heeft bijvoorbeeld veel apparaten aan het LoRa-netwerk gekoppeld op Schiphol. GIGTECH is gespecialiseerd in het opbouwen van een redundant netwerk bij festivals.

Een private network houdt in dat zendmasten op eigen terrein worden opgezet op een gelicentieerde bandbreedte, om zo een betrouwbare verbinding te creëren. Hierdoor kan met standaard 4G-PIN-terminals en andere apparatuur een betrouwbare verbinding opgelegd worden, zonder modificatie van de hardware.

Hieronder staan de verschillen benoemd tussen een private network en een mobiel netwerk van een ISP:

Voordelen private network

- Alleen bevoegde werknemers en apparaten kunnen er gebruik van maken, wat zorgt voor meer veiligheid en snelheid.
- De prioriteit van apparaten kan worden ingesteld, zodat kritieke processen altijd succesvol verlopen.
- De netwerkinfrastructuur, zoals zendmasten, kan men na afloop van een evenement weer gebruiken op andere locaties.
- Eventueel kan het netwerk gebruikt worden voor het verschaffen van internet aan de bezoekers.

Nadelen private network

- Het (laten) opzetten van de netwerkinfrastructuur kost veel tijd en geld.
- Het opzetten van een private network is veel complexer dan het gebruikmaken van een bestaande infrastructuur.

Het afnemen van een private network bij bijvoorbeeld MCS of GIGTECH lijkt een geschikte oplossing voor het probleem van dit project. Ik heb echter niet voor een private network gekozen, omdat Quintor een communicatieprotocol wil dat zij ook voor andere use cases kan inzetten.

---

[11] **IEEE:** Institute of Electrical and Electronics Engineers. Een organisatie die technische standaarden ontwikkelt.
[12] **Private network**: Een mobiel netwerk dat alleen beschikbaar is voor de afnemer. Hiervoor worden (tijdelijke) zendmasten of base stations opgezet.
[13] **IoT**: Internet of Things. Een netwerk dat bestaat uit apparaten en sensors die (op grote afstand) gegevens kunnen versturen en ontvangen.

4.1.3. Co-existence

In deze paragraaf is de invloed van radiogolven van andere zenders onderzocht. De onderzoeksvragen zijn:

- Wat zorgt ervoor dat er geen internet is bij een festival?
- Zullen andere protocollen hier ook hinder van ondervinden?

Bij het uitzoeken of verschillende protocollen kunnen samenwerken of elkaar storen, is de term "co-existence"[14] van belang. Het zoeken naar experimenten van vergelijkbare aard bleek lastig, maar een zoektocht op internet leverde één duidelijk artikel op: "The myth of non-overlapping channels: interference measurements in IEEE 802.11" [7]. Hierin werd een proef uitgevoerd waarbij 2 Wi-Fi-stations op korte afstand van elkaar continu data zenden. Deze stations zijn getest op verschillende afstanden. Ook is nagegaan of het uitmaakt of de stations op hetzelfde Wi-Fi-kanaal zenden of een ander kanaal gebruiken.

Conclusie: Wanneer 2 of meerdere stations dicht bij elkaar zenden, verstoren ze elkaars signaal. Dit is ook het geval wanneer ze op verschillende kanalen (frequenties) zenden. Echter, dit probleem bleek zich alleen voor te doen wanneer de stations erg dicht bij elkaar stonden en continu aan het zenden waren.

Relevantie voor dit onderzoek: Op een festival bevinden zich veel smartphones die kunnen communiceren via Wi-Fi, Bluetooth en het mobiele netwerk. Er is geconcludeerd dat dit geen probleem is vanwege de onderstaande redenen. In de evaluatie van hoofdstuk 6.1 wordt deze aangenomen conclusie genuanceerd.

- Er bevinden zich weinig telefoons in de buurt (binnen een meter) van de PIN-terminals.
- Smartphones zijn normaliter niet verbonden met een Wi-Fi- of Bluetooth-netwerk bij festivals, waardoor ze niet zullen storen op het mobiele netwerk.
- De telefoons van de festivalgangers zijn verbonden met het mobiele netwerk. Echter, de telefoons zullen door de beperkte verbinding weinig zenden en daardoor nauwelijks interfereren op het netwerk van de PIN-terminals.
- Het genoemde onderzoek [7] is 13 jaar oud. Naar verwachting is de zendapparatuur op de markt sinds dien beter geworden.
- Het genoemde onderzoek [7] is uitgevoerd met consumentenhardware. Waarschijnlijk zal professionele zendapparatuur, zoals antennes, beter bestand zijn tegen interferenties.

4.1.4. Overige relevante ontdekkingen

In deze paragraaf staan de uitkomsten van kleinere onderzoeken beschreven die relevant zijn voor het project, maar niet bij de bovenstaande onderwerpen passen.

Signaalsterkte

Een belangrijke factor bij het vergelijken en testen van protocollen is de signaalsterkte. Er zijn testen uitgevoerd met een mobiele telefoon om een indicatie te krijgen van gebruikelijke waardes, zie tabel 3.

De signaalsterkte is gemeten is dBm[15]. Het zendvermogen kan omgezet worden tussen dBm en mW[16] met de volgende formules [8]:

$$zendvermogen\ in\ dBm\ (A) = 10\ \times \log_{10}(P)$$
$$zendvermogen\ in\ mW\ (P) = 10^{A/10}$$

[14] **Co-existence:** De mate waarin een protocol bestand is tegen interferentie van andere protocollen.
[15] **dBm**: Decibel milliwatt. Het zend- of ontvangvermogen in mW op een logaritmische schaal. Voorbeeld: 30dBm = 1000mW.
[16] **mW:** milliWatt of 0,001 Watt. Het vermogen voor bijvoorbeeld het zenden of+C61 ontvangen van RF-data.

De ASU (Arbitrary Strength Unit) wordt gebruikt voor het berekenen van de signaalsterkte. Elk protocol heeft hier zijn eigen formule voor. De formule is [9]:

$$Signaalsterkte\ (RSRP)\ in\ dBm = -140 + ASU \qquad bij\ 0 < ASU < 97$$

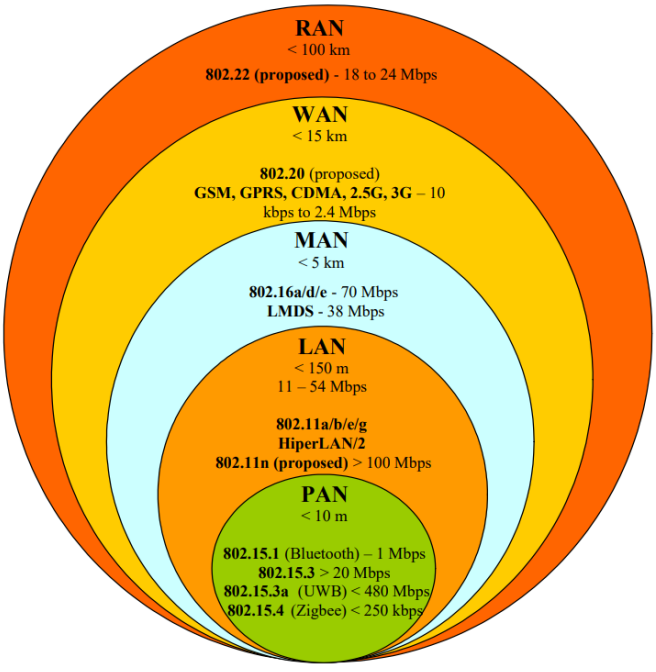
tabel 3: Test met mobiele telefoon ter indicatie van signaalsterkte.

Locatie	Techniek	Signaalsterkte (RSRP)		ASU	Bars
Op kantoor, dakterras	"4G" (LTE)	-91 dBm	7,943 × 10^{-10} W	49	5 (max)
Thuis, in de tuin	"4G+" (LTE-A)	-99 dBm	1,259 × 10^{-10} W	41	4
Op kantoor, bovenste étage	"4G" (LTE)	-105 dBm	0,316 × 10^{-10} W	35	3-4
Thuis, begane grond	"4G+" (LTE-A)	-110 dBm	0,100 × 10^{-10} W	29	2
In het ziekenhuis	"4G" (LTE)	-116 dBm	0,025 × 10^{-10} W	24	1

Spatial scopes

Protocollen kunnen gecategoriseerd worden op hun communicatieafstand. Deze categorieën worden "spatial scopes"[17] genoemd en zijn afgebeeld in figuur 4.

PAN-netwerken hebben typisch een bereik van ongeveer 10 meter. Deze zouden tijdens het onderzoeken van protocollen gelijk af kunnen vallen. Echter, Uit fase 3 blijkt dat dat niet altijd het geval is. Veel mesh-netwerken (zoals Bluetooth-Mesh en ZigBee) kunnen veel verder communiceren dan 10 meter, maar zitten toch in het PAN-domein. Ook heeft Bluetooth een 'long range'-variant, die weer in het MAN-domein valt.



figuur 4: Spatial scopes voor communicatieprotocollen en hun typische bereik.

[17] **Spatial scope:** Een categorie voor de geografische afstand tussen communicerende apparaten waarin een protocol valt.

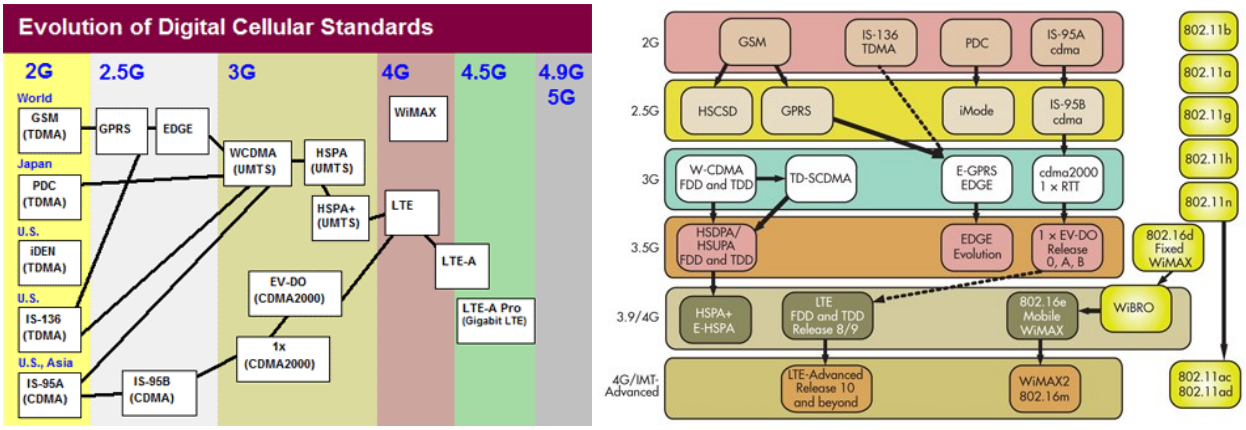
Aanbod mobiele netwerken

Binnen de categorie 'mobiele netwerken' vallen veel protocollen. Deze protocollen horen bij een bepaalde generatie. Aan een generatie zitten meerdere eigenschappen verbonden.

Verschillende bronnen hebben andere meningen over welk protocol in welke generatie past. Dit is bijvoorbeeld te zien in figuur 5. LTE voldoet volgens veel bronnen niet aan de eisen van een 4G netwerk. Daarom wordt LTE vaak als 3,9G beschouwd, terwijl ISP's dit als 4G bestempelen. LTE-Advanced is de opvolger van LTE en "4G+" genoemd door ISP's. Hierdoor ontstaat vaak een spraakverwarring over wat een "G" betekend [10]. De belangrijkste verschillen tussen de generaties zijn, volgens whatsag.com [11]:

- **1G:** De eerste generatie van draadloze telefonie technologie. Alleen bedoeld voor bellen.
- **2G:** Eerste generatie met internet, SMS, binnenlandse roaming en telefonisch vergaderen.
- **3G:** Internetsnelheid rond de 7,2 Mbit/s en bellen met betere geluidskwaliteit.
- **4G:** Internetsnelheid rond 100 Mbit/s. IPv6 mogelijkheden en clients kunnen gemakkelijk overschakelen tussen heterogeneous networks, zoals een sub-netwerk in een winkelcentrum.
- **5G:** Internetsnelheid rond 1-20 Gbit/s, latency van 1ms en 10-100x zoveel apparaten per netwerk.

Binnen het aanbod van mobiele netwerken, richten een aantal protocollen zich op IoT-devices. De belangrijkste zijn Narrowband-IoT en LTE-M. Deze zijn verder onderzocht in fase 3.



figuur 5: Enkele protocollen voor mobiele netwerken. De definitie van een "generatie" (G) verschilt per bron. (Links [12]; Rechts [13])

4.1.5. Resultaat

Uit deze fase is het volgende naar voren gekomen:

Een lijst met protocollen die onderzocht kunnen worden in fase 3. Deze lijst is te vinden in Bijlage B.

Er is al één oplossing gevonden voor het internetprobleem bij festivals: het opzetten van een private network. Deze optie is echter niet gekozen. Quintor implementeert liever zelf een protocol zonder dat er zendmasten opgebouwd hoeven te worden. Daarom zal deze optie afvallen. Daarnaast is het verbindingsprobleem bij festivals bedoeld als casus voor de afstudeeropdracht, niet als het daadwerkelijke doel.

Met de informatie die opgedaan is in deze fase kon begonnen worden aan fase 2: Het schrijven van het Plan van Aanpak.

4.2. Fase 2: Definitiefase

In deze fase zijn de requirements onderzocht en is het Plan van Aanpak – Bijlage E gerealiseerd. In dit hoofdstuk komen het pakket van eisen, de projectbeheersing en het testplan aan bod. De deelvraag van deze fase luidt:

Deelvraag 2: Aan welke eisen moet het te kiezen protocol voldoen?

Om antwoord te krijgen op deze deelvraag, zijn de volgende vragen opgesteld:

4.2.1. Hoe ziet de situatie rond de opdracht eruit?
4.2.2. Aan welke eisen moet het te kiezen protocol voldoen?
4.2.3. Wat is Quintor's doel van dit project?
4.2.4. Hoe worden de requirements getest?
4.2.5. Wat zijn de risico's en hoe wordt hiermee omgegaan?
4.2.6. Welke methodiek wordt gebruikt in dit project?

Backlog

Er is geen specifieke backlog opgenomen in het Plan van Aanpak, aangezien dit in grote mate afhangt van de uitkomst van fase 3, de onderzoeksfase. Daarom zal de backlog pas gerealiseerd worden in het begin van fase 4, de realisatiefase. In het Plan van Aanpak is echter een testplan opgenomen, waardoor er wel een plan is voor de uit te voeren werkzaamheden.

4.2.1. Situatie analyseren

In deze paragraaf wordt ingegaan op de omstandigheden van de opdracht en van de test.

Eisen opstellen

Middels deze afstudeeropdracht wil Quintor informatie opdoen over draadloze communicatieprotocollen. Deze opdracht maakt geen deel uit van een bestaand project, maar is informatie-inwinning voor toekomstige projecten. Daarom waren bij aanvang geen randvoorwaarden en eisen bekend.

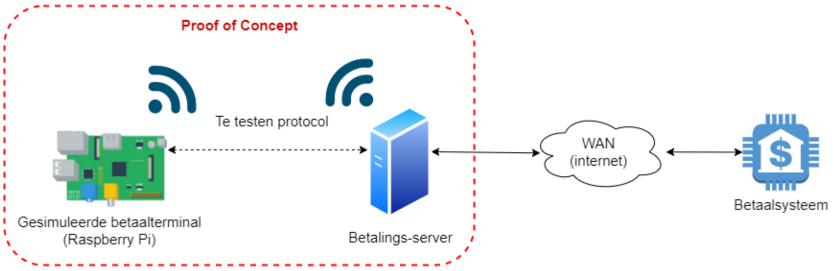
Door eigen onderzoek zijn eisen beredeneerd. Om de eisen te verantwoorden, is de herkomst ervan beschreven in dit hoofdstuk. Naast de uit te voeren opdracht is ook rekening gehouden met de bruikbaarheid van het protocol voor andere toepassingen, zodat de uitkomst van deze opdracht waardevoller is voor Quintor.

Proof of Concept situatie

Zoals beschreven staat in het afstudeerplan - Bijlage D, zullen er geen echte PIN-betalingen plaatsvinden, omdat het verwerken van betalingen niet het doel is van dit project. Er wordt met deze opdracht gefocust op de implementatie van het protocol en het testen ervan. In plaats van betaalterminals worden Raspberry Pi's[18] gebruikt om PIN-terminals te simuleren. De reden hiervan is dat een Raspberry Pi een simpele en goedkope computer is; geschikt om het Proof of Concept mee uit te voeren.

Het realiseren van een echt betaalsysteem zit niet in de scope van de opdracht. Alleen de systemen binnen het netwerk van het te testen protocol worden opgezet, zie figuur 6. In werkelijkheid zou de server de betalingen via een betrouwbare, bekabelde verbinding doorsturen naar een betaalplatform, maar in plaats daarvan zal de server zelf de betalingen afhandelen.

[18] **Raspberry Pi:** Een eenvoudige, kleine computer met veel interfaces. Dit board wordt gebruikt in het Proof of Concept.



figuur 6: Simplistische weergave van het Proof of Concept. Alleen de verbinding tussen de gesimuleerde betaalterminals en de server worden getest.

4.2.2. Eisen opstellen

Er zijn 16 eisen opgesteld voor de protocollen die onderzocht worden. Deze eisen zijn geprioriteerd volgens de MoSCoW-notatie. De letter van de eis geeft de prioriteit aan: Must have, Should have, Could have of Won't have. Deze eisenlijst is besproken met de bedrijfsmentor en is goedgekeurd.

Met de kennis uit het vooronderzoek is de eisenlijst zo compleet mogelijk ingevuld. Aan het eind van de opdracht bleek dat deze eisenlijst niet verder aangevuld hoefde te worden.

Technische eisen

In de eisen hieronder worden voorwaarden gesteld aan de specificatie van de te onderzoeken protocollen.

M1: Grootte van berichten

Voorwaarde M1: Applicatie-payload van ≥ 240 bytes per bericht.

Bij veel LPWAN-protocollen is de MTU erg laag, waardoor deze voor veel toepassingen niet inzetbaar is. Om erachter te komen hoe groot de applicatielaag-payload moet zijn, is gemonitord welke data er verstuurd wordt bij het doen van een betaling. Vervolgens is onderzocht welke data essentieel is en welke overbodig.

In een webbrowser is gelogd welke data er verstuurd wordt bij het doen van een betaling. De betaling is gedaan via de website van ING. De data is met de ingebouwde sniffer van Google Chrome verzameld. De betaal-request is te zien in figuur 7.

De betaal-requests zijn in JSON-formaat. De beschrijving van elk figuur geeft de 'geminimaliseerde grootte' aan. Dat houdt in: De grootte van de payload, na het verwijderen van de spaties en enters. Deze karakters zijn bedoeld om de leesbaarheid te verbeteren, maar ze hebben geen impact op de functionaliteit.


```
{
    "sourceAccount": {
        "type": "c-24",
        "value": "NL[redacted]",
        "encryptedValue": "[redacted]",
        "currency": "EUR"
    },
    "sourceAccountName": "Hr J M Ubink",
    "description": "Test voor afstudeeropdracht",
    "country": "NL",
    "destinationAccountName": "[redacted]",
    "destinationAccount": {
        "value": "[redacted]"
    },
    "amount": {
        "currency": "EUR",
        "value": "0,01"
    },
    "schedule": {
        "startDate": "18-11-2020"
    }
}
```


figuur 7: Payload van een online overboeking. Minimalistische payload = 465 bytes. Gevoelige informatie is gecensureerd.

Doordat de betaling is uitgevoerd met een webbrowser, is er veel HTTP-overhead meegestuurd in de vorm van headers. Deze zijn niet relevant voor het op te zetten systeem. Headers die relevant zijn, zoals authenticatie informatie, kunnen eenmalig verstuurd worden in het begin van een communicatiesessie.

Verkleinen van de payload

De payload kan verkleind worden. De eerste stap is het vervangen van de key-names door 1-letter key names, zie figuur 8.

```
{
    "a": {
        "b": "c-24",
        "c": "NL",
        "d": "",
        "e": "EUR"
    },
    "f": "Hr J M Ubink",
    "g": "Test voor afstudeeropdracht",
    "h": "NL",
    "i": "",
    "j": {
        "k": ""
    },
    "l": {
        "m": "EUR",
        "n": "0,01"
    },
    "o": {
        "p": "15-09-2020"
    }
}
```

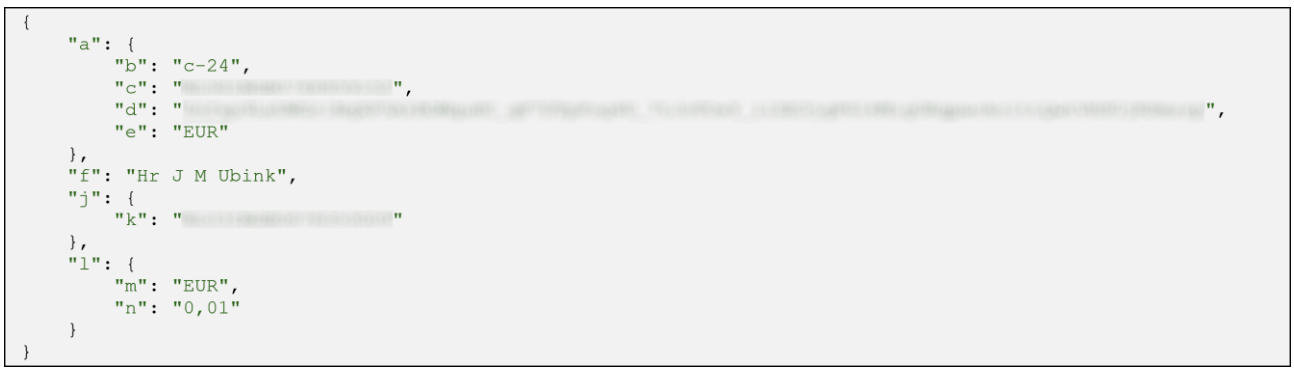

figuur 8: Payload van online overboeking waarbij de key-namen 1 letter zijn. Minimalistische payload = 321 bytes.

Als tweede optimalisatiestap kunnen een aantal velden vervallen. Waardes die bij elke betaling hetzelfde zijn, hoeven niet elke keer tussen de terminal en de server verstuurd te worden. Deze zijn niet relevant voor PIN-transacties bij een festival. Deze velden staan beschreven in tabel 4.

tabel 4: Waardes die weggelaten kunnen worden in de payload van een betaalterminal. Waardes die niet veranderen kunnen van tevoren handmatig of bij eerste verbinding met de server ingesteld worden.

Veld	Reden om weg te laten
Description	Hiervoor is vaak een generieke omschrijving.
Country	Het land zal niet veranderen tijdens een evenement.
destinationAccountName	De betaalgegevens van de ontvanger zullen niet aangepast worden tijdens een evenement.
schedule / startDate	Wordt gebruikt voor een geplande betaling. PIN-betalingen zijn directe betalingen.

De payload die overblijft is 225 bytes lang, zie figuur 9. De sourceAccountName-waarde in dit voorbeeld is 12 karakters lang. Deze lengte kan per betaalkaart verschillen. Voor extra speling, is de eis gezet op een MTU van minstens 240 bytes. Om het bericht nog kleiner te maken, zou er een ander payload-format gebruikt kunnen worden, zoals CSV. De payload van 225 bytes geeft echter een indicatie van de benodigde hoeveelheid data.



```
{
    "a": {
        "b": "c-24",
        "c": "",
        "d": "",
        "e": "EUR"
    },
    "f": "Hr J M Ubink",
    "j": {
        "k": ""
    },
    "l": {
        "m": "EUR",
        "n": "0,01"
    }
}
```


figuur 9: Uitkomst. Minimalistische payload van 225 bytes.

M2 & M3: Hardware beschikbaarheid

Voorwaarde M2: Minstens 3 componenten (bijvoorbeeld modulatiechips, ontwikkelboards) die dit protocol ondersteunen moeten leverbaar zijn. Anders is er geen vertrouwen dat het protocol goed ondersteund is.

In de oriëntatiefase is gebleken dat er protocollen zijn die niet, of niet goed ondersteund worden. Sommige protocollen zijn een open standaard, zonder implementatie. Door deze eis moeten er minstens 3 hardwarecomponenten op de markt zijn. Anders wordt geconcludeerd dat het protocol niet goed ondersteund is.

Voorwaarde M3: Er moet een Raspberry Pi module voor het gebruik van dit protocol beschikbaar zijn.

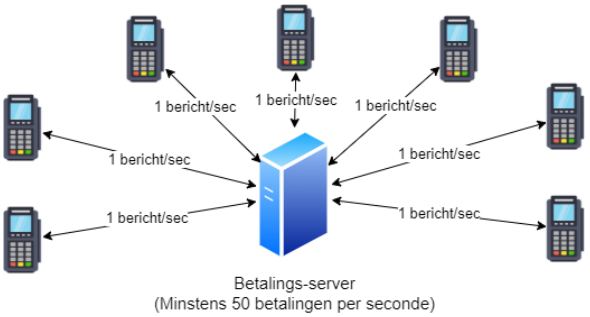
Het Proof of Concept wordt uitgevoerd met een Raspberry Pi. Voor veel protocollen is een Raspberry Pi-module beschikbaar. Als hardware hiervoor compatible is, dan heeft de opdracht meer kans van slagen.

M4 & M5: Aantal berichten per tijdseenheid

Eisen M4 en M5 gaan over het aantal berichten dat ontvangen en verzonden mogen worden. De limiet kan verschillen:

- M4: Het aantal uplinks en downlinks van een betaalterminal.
- M5: Het aantal uplinks en downlinks voor de gateway of server.

Afhankelijk van het protocol is het mogelijk dat een gateway meer berichten kan verzenden en ontvangen dan een end-device. Een server moet de berichten van alle terminals kunnen beantwoorden, figuur 10. Dit probleem speelt vooral bij LPWAN- en mesh-netwerken een rol.



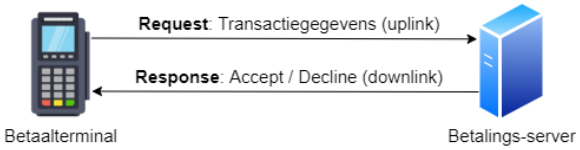
figuur 10: De Een server moet berichten van meerdere betaalterminals kunnen verwerken. Het vereiste aantal uplinks en downlinks voor de server (eis M5) ligt hoger dan dat van de betaalterminals (eis M4).

Voorwaarde M4: Nodes/terminal: 3600 berichten/uur of meer, zowel uplinks als downlinks.

Test situatie

Er is onderzoek gedaan naar de message flow van een PIN-betaling. Helaas is hier geen bron voor gevonden. Er wordt aangenomen dat er meer berichten heen en weer worden gestuurd bij een echte PIN-betaling. Denk hierbij aan het oplossen van een cryptografische challange met een private key die op de PIN-pas staat.

Het doel van het Proof of Concept is om het protocol zo goed mogelijk te testen en niet het volledig nabouwen van een betaalsysteem. Daarom wordt er een simpel request-response model opgezet. Het testsysteem wordt geïmplementeerd zoals te zien in figuur 11. Hierbij zal een betaalterminal 1 uplink en 1 downlink nodig hebben per transactie.



figuur 11: Minimale data in een gesimuleerd betaalsysteem.

Limitaties
Zoals uitgelegd in hoofdstuk 5.1.1 kan er een limiet zijn op het aantal berichten dat verzonden of ontvangen mag worden. Dit kan liggen aan het protocol of de cloud service die is afgenomen. LoRaWAN via KPN heeft bijvoorbeeld een limiet van 3 downlinks per uur, wat lang niet genoeg is.

Mesh-netwerk
Indien er gebruikt gemaakt wordt van een mesh-netwerk, moeten de forward-nodes de capaciteit hebben om alle betalingen te forwarden. Nodes moeten het verkeer van meerdere terminals door kunnen voeren, daarom zou het aantal berichten van de nodes hoger moeten liggen dan het aantal berichten van de betaalterminals.

Minimum van 3600 berichten/uur
Er wordt een eis gesteld dat 3600 berichten/uur moeten kunnen worden verstuurd en ontvangen. Dat komt in de testsituatie neer op 1 betaling per seconde. Bij het op te zetten systeem zal een betaling maar 1 request (uplink) en 1 response (downlink) bevatten, zie figuur 11. Daarnaast wordt aangenomen dat er realistisch gezien niet meer dan 1 betaling per seconde, per betaalterminal, wordt verricht.

Voorwaarde M5: Gateway/server: Moet 50 berichten (betalingen) per seconde kunnen zenden en ontvangen.

Als eis wordt gesteld dat de tussenserver 50 betalingen per seconde moet kunnen verwerken. Hierdoor zullen 500 betaalterminals gemiddeld elke 10 seconden een betaling kunnen uitvoeren. Deze eis gaat niet over de snelheid van de computer, maar over de limiet op het aantal berichten dat kan worden ontvangen en verzonden.

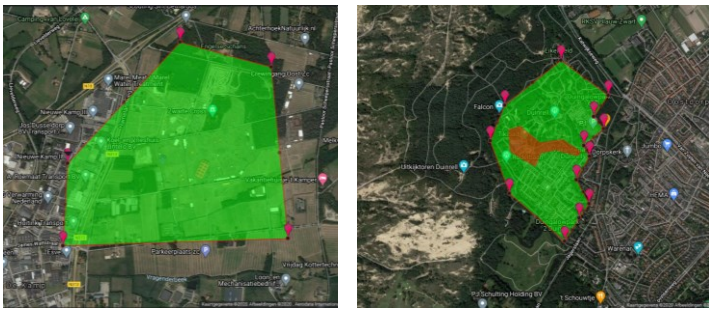
M6 & S1: Bereik
Voorwaarde M6: Bereik ≥ 30 hectaren (387m).
Voorwaarde S1: Bereik ≥ 60 hectaren (548m).

Een van de belangrijkste kenmerken is de communicatieafstand. Er zijn twee locaties vergeleken, zie figuur 12. Het doel hiervan was om te beredeneren welke communicatieafstand wenselijk is, om hier vervolgens een eis aan te verbinden.

Bereik bepalen

De eerste locatie was de Zwarte Cross, het grootste festival van Nederland [16]. Het volledige terrein is ongeveer 160 hectaren groot, inclusief kampeerterrein, parkeerplaatsen en het omheen gelegen recreatieterrein. Ook op deze plekken is het vaak wenselijk om te kunnen pinnen.

Het tweede terrein was vakantiepark Duinrell. Het volledige vakantiepark is 48 hectaren en het attractiepark is 9 hectaren. Ik heb voor dit park gekozen, omdat het een stuk kleiner is dan de Zwarte Cross, maar alsnog van nuttige grootte. Ik ben bekend met dit park, waardoor het makkelijker was om een beeld te krijgen bij de genoemde oppervlaktes.

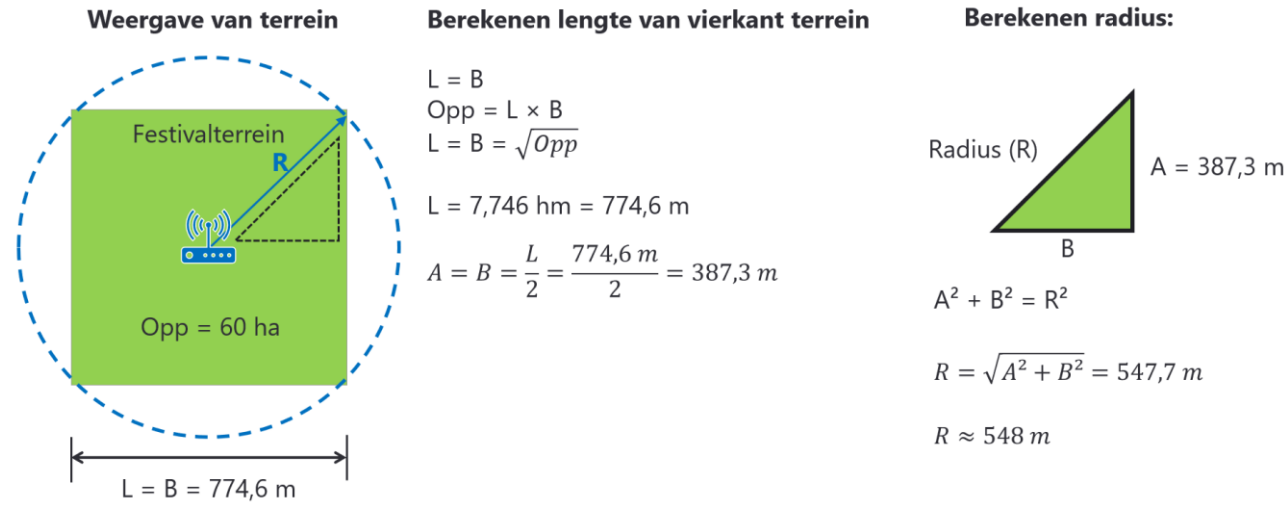


figuur 12: Oppervlakte is gemeten met oppervlakte-tool van draftlogic.com. [14]
Links - groen: Oppervlakte Zwarte Cross: (~160 ha)
Rechts - Rood: Oppervlakte Duinrell attractiepark. (~9 ha) - Groen: Oppervlakte Duinrell, hele vakantiepark. (~48 ha)

Naar verwachting zal het moeilijk worden om protocollen te vinden die een gebied van 160 hectaren kunnen dekken. Na overleg met de bedrijfsmentor wordt gesteld dat het te kiezen protocol geschikt is voor veel use cases als het ten minste 30 hectaren kan bedekken. Echter, de wens is dat het protocol 60 hectaren kan dekken. Wanneer grotere terreinen van internet moeten worden voorzien, moet er een andere oplossing gevonden worden, zoals het gebruiken van meerdere internet aansluitpunten.

Oppervlakte omzetten naar zendafstand

Hectaren zijn niet een geschikte eenheid om de communicatieafstand mee te bepalen. Daarom zijn de hierboven genoemde oppervlaktes omgezet naar afstanden. Dat is gedaan met de methode van figuur 13, waarbij uitgegaan wordt van een vierkant festivalterrein. De afstanden die hieruit kwamen zijn 387 m (vereist) en 548 m (gewenst).



figuur 13: Methode om het benodigde zendbereik te berekenen.

S2: Succesvolle datatransmissie

Voorwaarde S2: Falen in de communicatie worden afgehandeld door het protocol.

Om erachter te komen of een bericht aankomt, zijn acknowledgements nodig, maar niet elk protocol maakt daar gebruik van. Het is noodzakelijk om falen in de communicatie te detecteren. Dit probleem kan op twee manieren opgelost worden:

1. Foutdetectie of oplossend vermogen in de data-linklaag.
2. Het toepassen van controle in de transportlaag, zoals gebeurt met TCP.

Bij de eerste optie wordt elk bericht beantwoord met een acknowledge door de eerstvolgende hop[19]. Veel protocollen hebben oplossend vermogen, waardoor de oorspronkelijke data vaak kan worden herleid uit een corrupt bericht.

Bij de tweede optie wordt een acknowlegde gestuurd door de eindbestemming. Bij het gebruik van TCP wordt elk bericht geacknowledged. In het geval dat een bericht niet aankomt, kan het opnieuw verzonden worden. Het gebruik van zo'n transportlaagprotocol is alleen mogelijk als de MTU van het protocol daar groot genoeg voor is.

S3 & S4: Snelheid van berichten

Voorwaarde S3: 1 kbyte/s voor eind-nodes.

De eind-nodes moeten een minimale snelheid hebben van 1 kilobyte per seconde voor zowel uplinks als downlinks. In eis M1 is gesteld dat de payload van een bericht ongeveer 240 bytes is. Bij een snelheid van 1 kbyte/s zal zo'n bericht in ongeveer 1/4de seconde verstuurd worden. Er wordt aangenomen dat als er meer latency zal plaatsvinden, het systeem ongewenst traag wordt.

Voorwaarde S4: Een bericht heen-en-weer sturen van end-node naar gateway duurt ≤ 1000ms.

Een bericht uitwisselen tussen de betaalterminal en tussenserver zal maximaal 1000ms duren. Dit is wederom om irritaties wegens lang wachten te voorkomen.

Om een idee te krijgen van verschillende round-tripsnelheden, zijn er testen uitgevoerd met behulp van ICMP-pings. Verschillende servers zijn ongeveer 100 keer gepingd. De gemiddelde response-tijd is te zien in tabel 5. Deze testen zijn uitgevoerd via een bekabelde ethernetverbinding, een Wi-Fi-verbinding en via het mobiele netwerk. Hieruit blijkt dat de responsetijd bijna altijd lager is dan 100ms.

Het beantwoorden van een bericht kan in de praktijk langer duren, omdat het door het systeem verwerkt moet worden. De vertraging is afhankelijk van de complexiteit van de applicatie en de snelheid van de server en niet van het protocol. De 1000ms eis gaat over de round-triptijd van een ping. Indien ICMP-pingen niet mogelijk is, wordt een gebruikelijk ping-alternatief van het protocol gebruikt om de snelheid te testen.

[19] **Hop:** Een verbinding tussen 2 eindbestemmingen is opgebouwd uit meerdere hops. Elke tussenstap naar de volgende node, switch of router wordt een 'hop' genoemd.

tabel 5: De round-tripsnelheid van ICMP-pings naar verschillende servers. De response time is een gemiddelde van 100 pings.

Van	Naar	Ping response time (gemiddeld)
Laptop (bekabeld	8.8.8.8 (DNS Google)	11 ms
Laptop (Wi-Fi)		12 ms
Mobiel (Wi-Fi)		26 ms
Mobiel (4G+)		49 ms
Laptop (bekabeld	9.9.9.9 (Quad9 DNS)	79 ms
Laptop (Wi-Fi)		60 ms
Mobiel (Wi-Fi)		95 ms
Mobiel (4G+)		109 ms
Laptop (bekabeld	Nu.nl (Webserver)	28 ms
Laptop (Wi-Fi)		30 ms
Mobiel (Wi-Fi)		34 ms
Mobiel (4G+)		83 ms
Laptop (bekabeld	quintor.nl (Webserver)	9 ms
Laptop (Wi-Fi)		9 ms
Mobiel (Wi-Fi)		27 ms
Mobiel (4G+)		49 ms
Laptop (bekabeld	**Gemiddelde van alle servers**	20 ms
Laptop (Wi-Fi)		21 ms
Mobiel (Wi-Fi)		31 ms
Mobiel (4G+)		66 ms

S5: Beveiliging

Voorwaarde S5: Het protocol moet encryptie ondersteunen.

Bij PIN-betalingen speelt de beveiliging een grote rol. Vanzelfsprekend moet de payload versleuteld worden, zodat alleen de eindbestemming het bericht kan lezen. Met deze eis wordt echter de encryptie van de link-layer (laag 2) bedoelt.

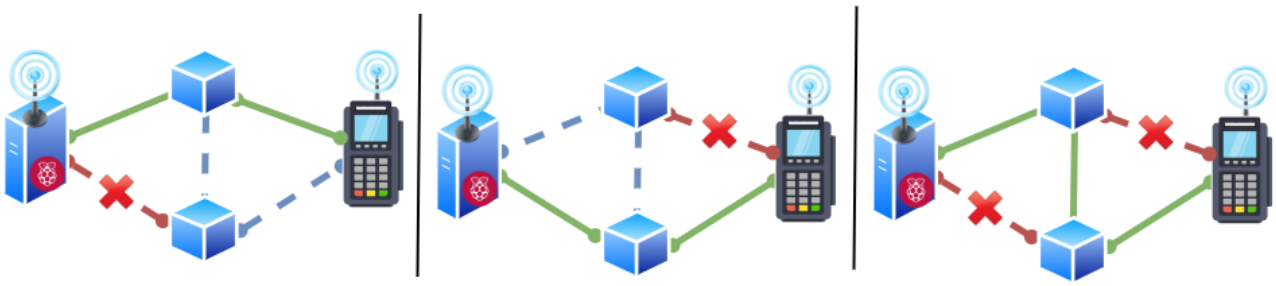
Draadloze communicatieprotocollen hebben vaak encryptie op laag 2, denk hierbij aan Wi-Fi (WPA) en Bluetooth. Als alleen de applicatielaag-payload (laag 7) versleuteld is, zijn de source- en destination IP-adres (laag 3) en port (laag 4) zichtbaar, die beide gevoelige informatie bevatten.

C1: Redundant mesh-netwerk

Voorwaarde C1: Als de verbinding tussen 2 nodes uitvalt, moet een andere route van eind-node naar gateway genomen kunnen worden.

In hoofdstuk 4.1.1 is onderscheid gemaakt tussen 3 soorten protocollen: LPWAN-, mesh- en cellulaire netwerken. Als er wordt gekozen voor een mesh-netwerk, moet de redundantie gecontroleerd worden. Hiervoor zullen 2 forward-node gebruikt worden, zoals te zien in figuur 14. Hierdoor zijn er meerdere paden naar de server mogelijk, voor het geval dat 1 node uitvalt. Er wordt aangenomen dat als een verbinding via 1 node mogelijk is, het ook zal werken met meerdere nodes. Daarom zullen de verbindingspaden maar 1 node hebben.

In het Proof of Concept wordt getest of de nodes automatisch omschakelen naar een andere verbinding wanneer een node of verbinding wegvalt.



figuur 14: Redundant mesh-netwerk. Wanneer een link of node uitvalt, zal er verbinding gemaakt worden via een ander pad.

Niet-technische eisen

Een protocol kan aan alle technische eisen voldoen, maar alsnog een slechte keuze zijn. Hieronder staan eisen opgesteld over de implementatie van het protocol.

M7: Hoe gemakkelijk het is om te integreren

Voorwaarde M7: Er moet informatie openbaar beschikbaar zijn over hoe het te integreren is.

Als er geen goede documentatie vindbaar is, zal het implementeren te veel tijd kosten of onmogelijk worden. Deze eis is vooral bij onbekende protocollen van toepassing.

S6: Gebruik in de praktijk

Voorwaarde S6: Een protocol moet praktisch voldoen aan alle gestelde eisen.

In de specificaties van protocollen worden vaak theoretische uitspraken gedaan over kenmerken, die niet haalbaar zijn in de praktijk. Deze eis is van toepassing op alle hierboven beschreven technische eisen.

Voorbeeld: Een mesh-protocol heeft een theoretisch bereik van 100m tussen 2 nodes. Dit betekent niet dat met 100 hops een netwerk van 10km opgezet kan worden.

Het toepassen van protocollen voor ongebruikelijke doeleinden leidt in sommige gevallen tot waardevolle uitkomsten. Echter, ik heb ervoor gekozen om niet een protocol te kiezen dat in de praktijk nooit gebruikt wordt voor dit soort systemen. Dat zou de kans van slagen voor dit project verkleinen.

C2: Bekendheid protocol

Voorwaarde C2: Het protocol moet redelijk bekend zijn.

Protocollen met de status “superseded” of “withdrawn” moeten vermeden worden. Deze protocollen worden in de toekomst steeds minder gebruikt. Na verloop van tijd zal er geen hardware daarvoor verkrijgbaar zijn.

C3: Kosten

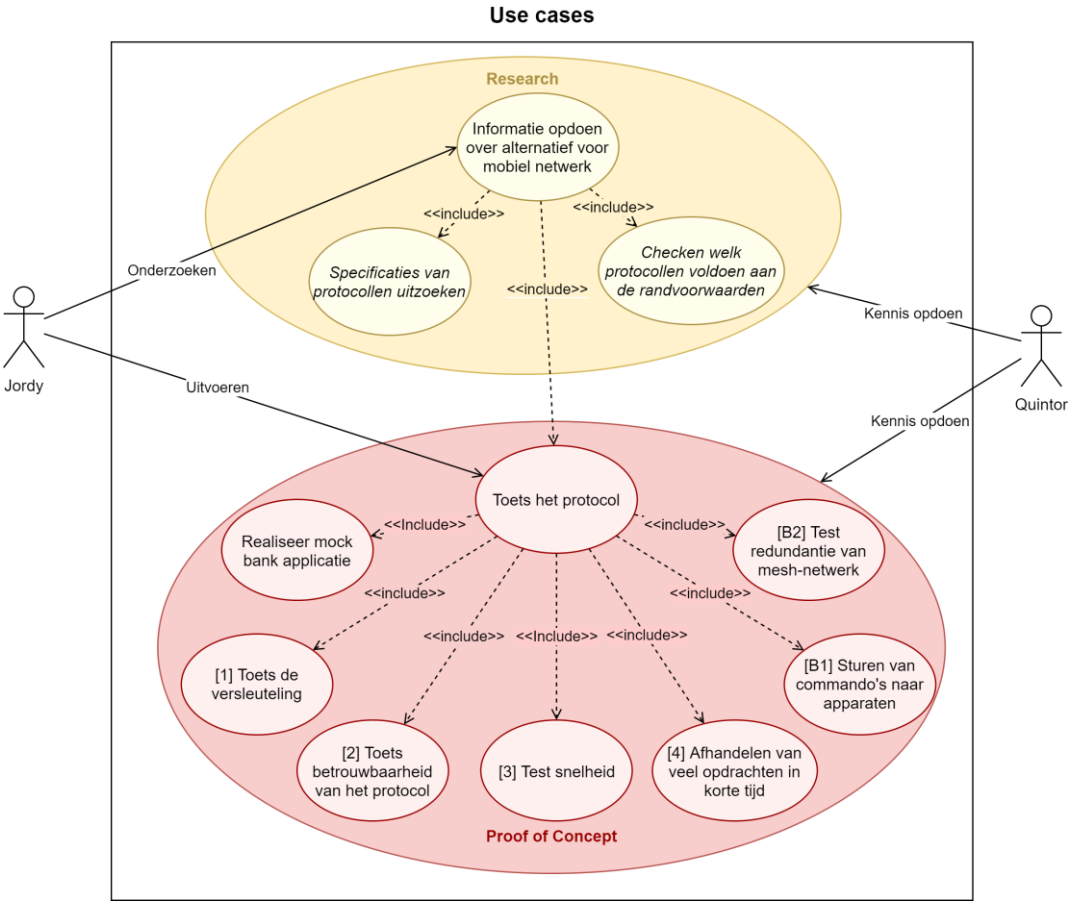
Voorwaarde C3: De kosten voor dit project zullen meegenomen worden bij het presenteren van de mogelijkheden.

Er moet rekening gehouden worden met de totale kosten van het Proof of Concept. Als een protocol of de evaluation kit veel geld kost, moet het ook grotere voordelen hebben ten opzichte van andere protocollen. Naast hardware kunnen ook andere zaken geld kosten, zoals het abonnement van een provider.

4.2.3. Doel

Het use case-diagram in figuur 15 beschrijft welke taken binnen de scope van het project vallen. Hier is duidelijk de verdeling te zien tussen de fase 3 (Uitgebreid onderzoek) en fase 4 (Proof of Concept).

Na dit project zal Quintor genoeg basiskennis hebben om het gekozen protocol te implementeren in nieuwe systemen. Dat wordt de focus voor dit project.



figuur 15: Taken en belangen voor dit project. Alle uit te voeren taken zijn verdeeld over 2 categorieën, welke overeenkomen met fase 3 en 4. [15]

4.2.4. Testplan

In het Plan van Aanpak staat een concept testplan beschreven. Hoewel de daadwerkelijke testsituatie afhangt van het te kiezen protocol, zijn de testen zo generiek beschreven, dat ze voor elk protocol gebruikt kunnen worden. Een uitzondering is use case [B2], welke bedoeld is voor het testen van een mesh-netwerk.

Bij de test van use case [3] (succesvolle datatransmissie) is het aantal betalingen (**x**) niet vastgesteld. Deze waarde hangt af van het te onderzoeken protocol. Dit aantal hangt bijvoorbeeld af van:

- De limiet op het aantal uplinks en downlinks van het protocol per tijd.
- De besteedbare databundel.

Zo kunnen 1000 berichten voldoende zijn om de betrouwbaarheid te testen. Indien het protocol het toestaat, kan dit opgeschaald worden naar veel meer berichten.

4.2.5. Risico's

Er is vooruit gekeken naar de komende fases om de risico's van deze opdracht in te schatten. Hierbij zijn de kans en het gevolg beredeneerd. Wanneer zo'n situatie zich voordoet, moet hierop gereageerd worden. Er is bij elk risico een oplossing beschreven, om de opdracht weer in goede banen te leiden.

4.2.6. Methodiek

In de onderzoeksfase en reflectiefase wordt gebruikgemaakt van verschillende methodes.

Onderzoeksfase (fase 3)

Tijdens deze fase wordt gewerkt volgens een stappenplan.

1. Onderzoek doen naar alle draadloze communicatietechnieken. In deze stap worden de technieken niet diep, maar oppervlakkig onderzocht.
2. Een selectie maken van communicatietechnieken die de oplossing zouden kunnen bieden voor dit probleem.
3. Verdiepen in de specificaties van de geselecteerde communicatietechnieken. De meest geschikte techniek wordt geselecteerd voor stap 4. Indien er 2 technieken even goed lijken te werken, kunnen beide technieken getest worden.
4. Een Proof of Concept maken die aantoont of het mogelijk is om deze techniek te gebruiken voor PIN-betalingen.

Realisatiefase (fase 4)

Bij de realisatie van het Proof of Concept wordt gebruikgemaakt van Scrum. Er wordt gewerkt met sprints van 2 weken. Quintor maakt zelf veel gebruik van Scrum en verlangt van haar afstudeerders dat dit gebruikt wordt voor hun afstudeeropdracht. Hoe Scrum uiteindelijk geïmplementeerd wordt, staat beschreven in hoofdstuk 4.4.

4.2.7. Resultaat

Na deze fase zijn de volgende zaken gerealiseerd:

Een Plan van Aanpak, waarin de opdracht in meer detail wordt toegelicht.

Een pakket van eisen. De voorwaarden hierin kunnen in fase 3 gebruikt worden om protocollen te beoordelen op geschiktheid.

Een concept testplan, waarmee de eisen getest kunnen worden in de praktijk.

4.3. Fase 3: Uitgebreid onderzoek

Beroepstaken: A-2

In deze fase wordt het een protocol gekozen dat gebruikt wordt voor het Proof of Concept in de volgende fase. De deelvraag van deze fase luidt als volgt:

Deelvraag 3: Welk protocol voldoet aan die eisen?

De taken die hierbij horen zijn als volgt:

4.3.1. Het zoeken van zo veel mogelijk protocollen.
4.3.2. Checken welke protocollen aan de requirements voldoen.
4.3.3. Een selectie van protocollen volledig onderzoeken.
4.3.4. Keuzes voor de protocollen en de gewenste hardware voorleggen bij de opdrachtgever.

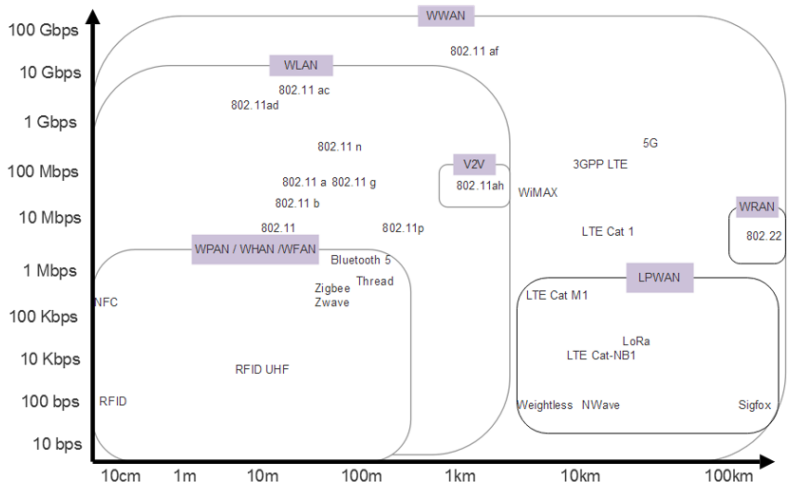
Volgens de planning zou deze fase 6 weken duren. Om een vollediger beeld te krijgen van de bestaande protocollen, zijn er veel meer protocollen onderzocht dan gepland. Doordat de scope groter werd, is deze fase met 2 weken verlengd. Hierdoor is echter wel een betere protocolkeuze gemaakt voor het Proof of Concept.

4.3.1. Zoveel mogelijk protocollen zoeken

Tijdens het oppervlakkige onderzoek in de oriëntatiefase zijn 24 draadloze communicatieprotocollen gevonden. Bij aanvang van deze fase is naar zo veel mogelijk protocollen gezocht, waarbij er in totaal 36 protocollen zijn gevonden. Een overzicht van alle gevonden protocollen is te vinden in Bijlage C - Onderzochte communicatieprotocollen. Deze paragraaf beschrijft het zoekproces.

Zoekmethode

Op internet staan veel overzichten van communicatieprotocollen, zoals in figuur 16. Hoewel dit een effectieve manier om aan een groot aantal protocollen te komen, is het noodzakelijk om ze te fact-checken. Een andere manier is door protocollen te zoeken op Wikipedia. Onder aan de pagina staat vaak een “See also”-deel waarin andere protocollen te vinden zijn. Daarnaast kunnen verschillende protocollen gevonden worden door met Google te zoeken naar IoT-, mesh-, long range- en WWAN-netwerken.



figuur 16: Draadloze protocollen met de daarbij behorende snelheid en afstand. [16]

Standaarden

In hoofdstuk 4.1.4 is vermeld dat protocollen op spatial scope gecategoriseerd kunnen worden, zie figuur 4. De daadwerkelijke afstanden die protocollen kunnen bereiken is anders dan gebruikelijk is voor hun spatial scope. De reden hiervoor is dat deze protocollen een nieuwe fysieke laag toevoegen aan de reeds bestaande basis. Bijvoorbeeld IEEE 802.11ah is voor een groot deel gebaseerd op de bestaande Wi-Fi protocollen en blijft, ondanks het zendbereik van 1 km, in de IEEE 802.11 range (WLAN). Deze standaardnummering bracht daarom verwarring tijdens het onderzoek.

Hieronder staan enkele voorbeelden.

- IEEE 802.11 **af** en **ah** behoren tot de WLAN-scope (< 150 m), maar hebben een bereik van 1km.
- IEEE 802.15.4**g** is een uitbreiding voor WPAN-netwerken (< 10 m), waardoor op 4 km afstand gecommuniceerd kan worden. [17]

Het is belangrijk om te weten hoe de versienummering van de relevante IEEE-protocollen werkt. In tabel 6 staat een lijst van protocollen die aan bod kwamen tijdens dit onderzoek. Wanneer een standaard superseded is, is het lastig om te achterhalen door welke actieve standaard het is opgevolgd, omdat dat vaak niet wordt vermeld op de site van IEEE: standards.ieee.org.

tabel 6 IEEE standaarden die relevant zijn voor dit project. Deze worden toegelicht in Bijlage I.

IEEE Familie	Working group	Uitbreiding	Betekenis
IEEE 802	IEEE 802.11 (WLAN)	IEEE 802.11a IEEE 802.11b IEEE 802.11g IEEE 802.11n (Wi-Fi 4) IEEE 802.11ac (Wi-Fi 5) IEEE 802.11ax (Wi-Fi 6) IEEE 802.11be (Wi-Fi 7)	Ter informatie. Wi-Fi voor consumentengebruik.
		IEEE 802.11af: (White-Fi / Super Wi-Fi)	Lange afstandsprotocol dat gebruikmaakt van TVWS (TV White Spaces).
		IEEE 802.11ah (Wi-Fi HaLow)	Een sub-GHz protocol dat op lange afstand veel data kan versturen.
	IEEE 802.15 (WPAN)	IEEE 802.15.4 (Low rate-WPAN)	Deze standaard beschrijft de fysieke- en data-linklaag waarop de volgende protocollen op zijn gebaseerd: ZigBee, ISA100.11a, WirelessHART, MiWi, 6LoWPAN, Thread, MiWi, en SNAP.
		IEEE 802.15.4g (SUN) (superseded)	Een uitbreiding op de fysieke laag van IEEE 802.15.4, waardoor op lange afstanden gecommuniceerd kan worden. Wi-SUN is een IPv6-implementatie hiervan. Deze uitbreiding heeft de status 'superseded'. Dat betekent dat de standaard vervangen is, maar nog wel voorkomt
		IEEE 802.15.4z-2020	Laatste actieve uitbreiding van IEEE 802.15.4.
	IEEE 802.16 (WirelessMAN / Wi-MAX)		Een langeafstandsprotocol met hoge throughput.
	IEEE 802.22 (Wi-FAR) (WRAN)		Cognitive Radio network in TVWS.

4.3.2. Protocollen testen op requirements

Om te bepalen welke protocollen in detail onderzocht moeten worden, is onderzocht welke protocollen aan de gestelde requirements voldoen. Dit onderzoek staat beschreven in Bijlage I. Hierdoor zijn 25 protocollen afgestreept.

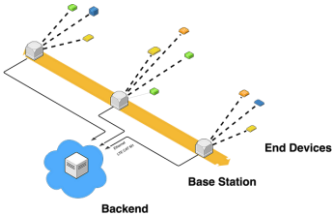
4.3.3. Selectie protocol volledig onderzoeken

De protocollen die in dit hoofdstuk beschreven zijn, hadden het meeste kans om te voldoen aan de gestelde eisen. Daarom zijn deze protocollen geselecteerd om in detail te onderzoeken.

Weightless-P

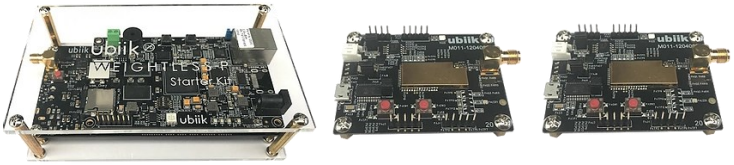
Wat is Weightless?

Weightless-P is een LPWAN-technologie dat bedoeld is voor het Internet of Things. End-devices maken verbinding met een base station, zie figuur 17. De base stations zijn verbonden met een private network of het internet. Met dit protocol kunnen apparaten op 2 km afstand communiceren. Daarnaast is dit protocol voor zowel uplinks als downlinks geschikt, terwijl andere LPWAN-protocollen alleen bedoeld zijn voor uplinks (bijvoorbeeld LoRa en Sigfox).



figuur 17: Systeem architectuur van Weightless-P. [18]

Deze techniek leek erg mooi, maar het blijkt moeilijk te implementeren. Ubiik is de enige leverancier van Weightless-hardware en hun starter kit is 1500 USD (figuur 18).



figuur 18: Weightless starter kit. Links: Base station, rechts: 2 end-devices. [19]

Contact met Weightless

Weightless-P leek op papier erg mooi, maar niet alle informatie was op internet te vinden. Er is een mailtje gestuurd naar de Weightless Alliance om achter de specificaties te komen. Helaas hebben ze niet gereageerd. De gestelde vragen waren:

- Hoeveel berichten kunnen er per tijd verstuurd worden?
- Wat is de payload van een bericht?
- Hoeveel latency bevat het protocol?
- Hoe kan het protocol geïmplementeerd worden?
- Zijn er andere leveranciers naast Ubiik?

Conclusie

Weightless-P is geschrapt. De redenen zijn:

- De start kit is te duur (1500 USD). Er was geen goedkopere hardware leverbaar via andere leveranciers.
- Veel informatie was onvindbaar en vragen werden niet beantwoord.
- De requirements konden niet gekeurd worden.

DASH7

Wat is DASH7?

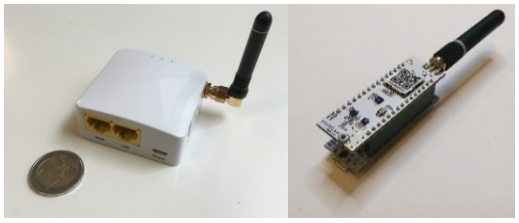
DASH7 is een full stack-protocol gebaseerd op ISO/IEC 18000-7. Een 'full stack-protocol' is een protocol dat een implementatie bevat voor alle lagen van het OSI-model. Hoewel de meeste RFID-protocollen bedoeld zijn voor het lezen van tags op korte afstand, is dit protocol bedoeld voor communicatie op 500 m in bebouwde omgevingen tot 10 km erbuiten.

Hoe werkt het

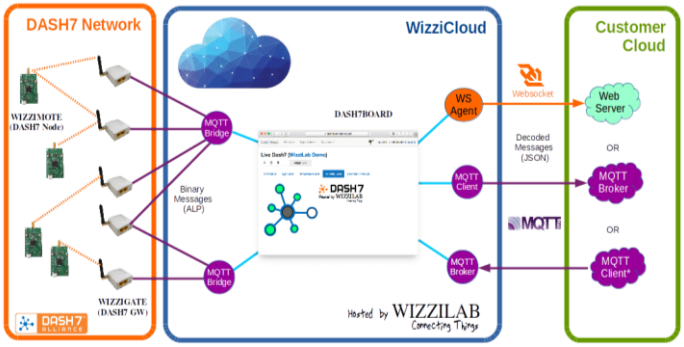
DASH7 heeft 2 modes: Pull en Push. Wanneer het pull model wordt gebruikt, komt er een verzoek vanuit het base station naar het end-device voor data. Wanneer het push-model wordt gebruikt, kunnen end-devices op eigen initiatief zenden naar de base stations [20]. Dit protocol werkt in de ongelicenseerde banden 433 MHz met een snelheid van 28-200 kbit/s [21].

Beschikbare hardware

Wizzilab is de enige vindbare leverancier van DASH7-hardware (figuur 19). Deze end-devices kunnen gekoppeld worden aan de cloud-omgeving van Wizzilab, zie figuur 20. De backend van de klant kan aansluiten op de cloud van Wizzilab. Helaas meldt Wizzilab dat deze apparatuur alleen geschikt is voor het testen van het protocol en niet voor end-use.



figuur 19: WizziKit prototyping hardware voor het testen van het DASH7-protocol. Links: Gateway (€149), rechts: end-device (€89). [22]



figuur 20: Netwerk architectuur van DASH7, welke aan de cloud van Wiizilab gekoppeld is. [23]

Zelf softwarestack bouwen

Hoewel er geen op DASH7 gerichte hardware is, bestaat er een andere mogelijkheid om dit protocol te implementeren. OSS-7 (Open Source Stack voor DASH7) is een GitHub-repository waar een software-stack staat om DASH7 te implementeren. [24] Deze software kan als referentiemateriaal gebruikt worden of op een wireless MCU gezet worden. Een blog van Van Ginneken [25] laat zien hoe OSS-7 geïmplementeerd kan worden op een Raspberry Pi in combinatie met een wireless MCU. Omdat OSS-7 open source is, zou het op elke MCU met een radio geïmplementeerd kunnen worden. In de DASH7 documentatie [26] staan 6 verschillende componenten waarop de software stack getest is.

OSS-7 heeft DASH7 nog niet volledig geïmplementeerd en het is nog in ontwikkeling. Daarnaast is de documentatie voor deze software stack incompleet. Daarom is de verwachting dat het implementeren van OSS-7 erg complex zal worden.

De DASH7 Alliance hoopt dat door OSS-7 meer fabrikanten dit DASH7 zullen implementeren, waardoor er meer DASH7-hardware op de markt komt.

Conclusie
Volgens de specificaties van DASH7 lijkt het een bruikbaar protocol, maar er wordt gesteld dat het niet geschikt is voor dit project. De redenen hiervoor zijn als volgt:

- Er is geen hardware speciaal voor DASH7 beschikbaar op de markt. Er is geen off-the-shelf-oplossing.
- Zowel de software als de documentatie van de open source stack OSS-7 is nog niet compleet.
- Het implementeren van DASH7 met eigen hardware zal veel tijd kosten, wat ten koste gaat van de rest van het project.

Multefire
Wat is Multefire?
Multefire is een protocol voor het gebruik van LTE-technologieën, specifiek 3GPP release 13 en 14, in het ongelicenseerde spectrum. Het normale mobiele netwerk maakt gebruikt van gelicenseerde frequenties. Multefire gebruikt het ongelicenseerde spectrum, zoals de 5 GHz-band die ook door Wi-Fi gebruikt wordt. Bedrijven kunnen hierdoor hun eigen ‘mobiele netwerk’ opbouwen. [27]

Mogelijkheden
Multefire kan ingezet worden door een bedrijf, zoals een vliegveld of mijnplaats. Daarnaast kan een Multefire ingezet worden als ‘neutral host’. In dat geval kunnen meerdere organisaties gebruikmaken van dezelfde ‘hot spots’. Met een gebruikersnaam en een wachtwoord kunnen gebruikers zich aanmelden bij het netwerk, zonder SIM-kaart.

Een ander voordeel ten opzichte van gelicenseerde mobiele netwerken is Quality of Service (QoS). Met QoS kunnen apparaten voorrang krijgen boven andere netwerken in het netwerk, zodat kritieke processen een hogere beschikbaarheid hebben.

Daarnaast ondersteunt Multefire ook LTE-M en NB-IoT. Hierdoor kunnen slimme apparaten toch aan het netwerk gekoppeld worden.

Nadelen
Er zijn ook wat nadelen aan Multefire verbonden. Gelicenseerde netwerken beschikken over een betrouwbaar, storingsvrij netwerk. Multefire maakt gebruik van gedeelde- en ongelicenseerde spectrums, waardoor ze te maken hebben met interference van andere apparaten. Er mag alleen gezonden worden, wanneer andere apparaten ‘stil’ zijn. Dit principe heet ‘listen before talk’. Ook is het toegestane zendvermogen veel lager, waardoor het bereik minder is. In de 2,4 GHz-band zal het een base station een bereik hebben van 50 tot 250 meter voor indoor apparaten [28]. Dit bereik kan worden opgeschaald met meerdere base stations.

Een ander nadeel is dat Multefire niet compatible is met LTE-chipsets [29]. De Multefire specificatie 1.0 was rond in 2017. Doordat het een erg nieuw protocol is, is er nog geen hardware beschikbaar.

Conclusie
Ik heb niet voor Multefire gekozen vanwege de volgende redenen:

- Er is geen end-device hardware te vinden om deze optie mee te implementeren.
- Er zijn geen base stations in Nederland om de verbinding mee te testen. Er zou in dat geval zelf een zendmast opgezet moeten worden.
- De range is maximaal 250 meter, wat niet genoeg is volgens eisen M5 en S1.

Wi-Fi HaLow (IEEE 802.11ah)

Wat is Wi-Fi HaLow?

Wi-Fi HaLow is een long-range protocol dat gecreëerd is door de Wi-Fi Alliance. Het protocol maakt gebruik van het 863-868 MHz spectrum in plaats van 2,4 of 5 GHz, waardoor langere afstanden worden behaald en minder energie nodig is. Dit protocol is de eerste techniek die voldoet aan alle eisen. Een overzicht hiervan is te zien in tabel 7.

Overeenkomsten met Wi-Fi

Met Wi-Fi HaLow bestaat een netwerk uit 1 base station (access point) en meerdere verbonden end-devices. De end-devices verbinden met het base station via zijn SSID, net zoals bij normale Wi-Fi. Ook wordt er gebruikgemaakt van WPA3 en IP. [30]

Verschillen met andere LPWAN-protocollen

In tegenstelling tot veel andere LPWAN-protocollen, kan met Wi-Fi HaLow gemakkelijk een eigen netwerk opgezet worden. Bij andere LPWAN-protocollen, zoals LoRa, Sigfox, DASH7 en LTE-M, moet er verbonden worden met een gateway, die met de achterliggende infrastructuur praat.

tabel 7: Wi-Fi HaLow specificities. [31]

Modulation	OFDM	**Maximum payload length**	1500 bytes
Frequency	Unlicensed ISM bands Europe: 863-868 MHz, North America: 902-928 MHz	**Range**	1.5 km (urban)
Bandwidth	2/4/8 MHz	**Interference immunity**	Very high
Maximum data rate	150 kbps to 78 Mbps	**Authentication & encryption**	Yes (WPA3)
Bidirectional	Yes / Half-duplex	**Adaptive data rate**	Yes
Maximum messages/day	Unlimited	**Standardization**	IEEE

Hardware

Wi-Fi HaLow is helaas niet compatible met standaard Wi-Fi chipsets. Voor dit protocol is de SX-NEWAH-EVK gevonden, zie figuur 21. Dit is een Wi-Fi HaLow evaluation board dat aan een Raspberry Pi gekoppeld kan worden. Silex heeft een video gepubliceerd, waarin uitgelegd wordt hoe het board aangesloten en gebruikt kan worden. Hieruit blijkt dat het een zeer gemakkelijk te implementeren protocol is [32]. Wanneer de driver van de chipset is geïnstalleerd op de Raspberry Pi, wordt de evaluation kit gezien als Wi-Fi adapter. Hierdoor zal de computer de hardware zien als een generieke internetverbinding, waardoor er bij het realiseren van het Proof of Concept geen rekening gehouden hoeft te worden met de adapter. Meer informatie is te vinden in Bijlage F - Hardware voorstel.

figuur 21: SX-NEWAH-EVK. Een Raspberry Pi-compatible Wi-Fi HaLow evaluation board (€211,85 per stuk). [50]

Conclusie

Wi-Fi HaLow leek de uitkomst voor het probleem en werd meegenomen met het hardwarevoorstel in hoofdstuk 4.3.4.

- Het voldoet aan alle eisen.
- Het is gemakkelijk te implementeren. De computer ziet geen verschil tussen Wi-Fi HaLow en andere internet interfaces.
- Het is ontwikkeld door een bekende alliantie.
- Het protocol dekt een grote range.

Bluetooth (coded PHY)

Wat is Bluetooth Coded PHY?

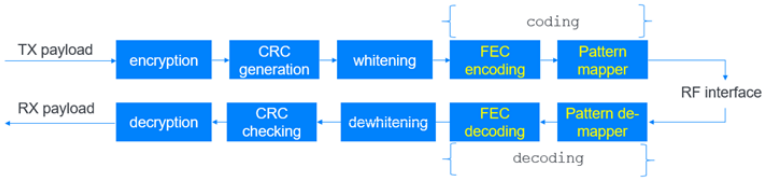
Bluetooth versie 4 en hoger worden 'Bluetooth Low Energy' (BLE) genoemd en hebben 4 implementaties voor de fysieke laag, zie tabel 8. (B)LE 1M heeft een data rate van 1 Mbit/s en is de standaard variant. Later zijn Coded PHY (long range) en LE 2M (2 Mbit/s) erbij gekomen. Met Coded PHY kunnen afstanden tot 400 m (binnen) en 1000 m (buiten) worden behaald, wat ruim genoeg is voor eis M5.

tabel 8: De specificaties van de verschillende fysieke lagen voor Bluetooth. [33]

Physical layer:	LE 1M (standard)	LE Coded S=2	LE Coded S=8	LE 2M
Symbol rate	1 Ms/s	1 Ms/s	1 Ms/s	2 Ms/s
Data rate	1 Mbit/s	500 kbit/s	125 kbit/s	2 Mbit/s
Error detection	CRC	CRC	CRC	CRC
Error correction	None	FEC	FEC	None
Range multiplier	1	2	4	0,8
Bluetooth 5 requirement	Mandatory	Optional	Optional	Optional

Verschil tussen coded PHY en 1M (standaard) Bluetooth

Coded PHY maakt gebruikt van Forward Error Correction (FEC) om langere communicatieafstanden te bereiken. Hierbij wordt data redundant verzonden, zodat errors gecorrigeerd worden, zie figuur 22. De symbol rate[20] blijft hetzelfde als bij LE 1M, maar de throughput wordt lager.



Input-bit voor FEC Encoder	Output met S=2	Output met S=8
0	0	0011
1	1	1100

figuur 22: Links: FEC in Bluetooth 5 bit stream processing [33]. Rechts: Beschrijving van hoe FEC-bits worden gegenereerd uit input-data.

Payload

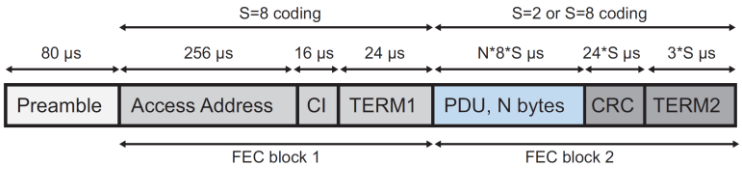
Eis M1 van deze opdracht stelt dat de payload van het protocol minstens 240 bytes moet zijn. Bluetooth heeft standaard een vaste payload van 27 bytes. Bluetooth beschikt echter over DLE (Data packet Length Extension), waarmee payloads van variabele lengtes tussen 27 en 251 bytes mogelijk zijn. De package format is te zien in figuur 23. In figuur 23 en figuur 24 is te zien dat het mogelijk is om DLE te gebruiken in combinatie met coded PHY.

Het was lastig om dit uit te zoeken, omdat meerdere bronnen online claimen dat DLE niet te gebruiken is in combinatie met coded PHY. Om zeker te zijn over de MTU, is navraag gedaan bij een medewerker van Nordic, een bedrijf dat Bluetooth hardware fabriceert. Het antwoord was: De Bluetooth SDK van Nordic heeft een limitatie op 2700 µs, wat neerkomt op een payload van 27 bytes. Hoewel het volgens de Bluetooth specificatie mogelijk is om met Bluetooth coded PHY-berichten van 251 bytes te zenden, is het verstandiger om de data op te splitsen in kleinere delen en als losse pakketjes te versturen. Hierdoor is de kans groter dat de data aankomt. [34]

Eis M1 is niet gehaald, maar het is wel mogelijk om meerdere berichten samen te voegen om de oorspronkelijke data te krijgen, waardoor transacties toch door kunnen gaan.

[20] **Symbol rate:** De snelheid waarmee symbolen verzonden worden. Een symbool bestaat uit 1 of meer bits, afhankelijk van de modulatietechniek. [52]

	Fields						
	Preamble	Access Address	CI	TERM1	PDU	CRC	TERM2
Number of Bits	Uncoded	32	2	3	16 – 2056	24	3
Duration when using S=8 coding (µs)	80	256	16	24	128 – 16448	192	24
Duration when using S=2 coding (µs)	80	256	16	24	32 – 4112	48	6



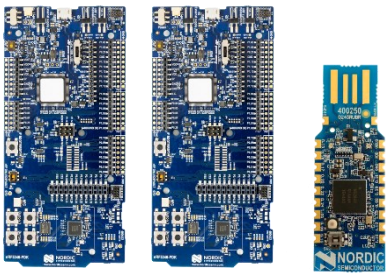
figuur 23: Bluetooth coded PHY - Package format [35]. Waar µs staat kan ook 'symbool' gelezen worden, want 1 Ms/s = 1s/µs.

LE Data Packet Length Extension feature supported	LE Coded PHY feature supported	CTEs supported on Data Physical Channel PDUs	Parameters with names containing "Octets"		Parameters with names containing "Time"	
			Minimum	Maximum	Minimum	Maximum
No	No	No	27	27	328	328
Yes	No	No	27	251	328	2120
No	No	Yes	27	27	328	336
Yes	No	Yes	27	251	328	2128
No	Yes	Don't care	27	27	328	2704
Yes	Yes	Don't care	27	251	328	17040

figuur 24: Overzicht van de MTU van Bluetooth met meerdere mogelijkheden [35]. De gele regel laat zien dat coded PHY en DLE samen gebruikt kunnen worden om een payload van 251 bytes te behalen.

Hardware

Bluetooth coded PHY is een redelijk nieuwe standaard, waardoor er weinig keuze voor hardware was. De nRF5340 PDK leek het beste, zie figuur 25. Na het doornemen van de SDK en de documentatie van Nordic is geconcludeerd dat het realiseren van een applicatie erg complex is en veel tijd zal kosten.



figuur 25: Bluetooth evaluation kit (€90,29). 4 board die met elkaar kunnen communiceren en 1 Bluetooth sniffer om mee te debuggen.

Conclusie

De eigenschappen van dit protocol zijn:

- Met coded PHY kan tot 400 meter indoor gezonden worden, waardoor eis M5 goed bevonden wordt.
- De MTU van 1 Bluetooth packet is niet groot genoeg, maar door data te splitsen over meerdere berichten, wordt de verbinding betrouwbaar. Daarom wordt eis M1 goedgekeurd.
- De implementatie lijkt lastig
- De hardware heeft lage aanschafkosten. Daarentegen zal dit niet opwegen tegen de werkuren die gebruikt moeten worden om dit protocol te implementeren.

Bluetooth coded PHY leek een geschikt protocol voor deze opdracht, maar het is niet de beste keuze voor dit project. In Bijlage F – 'Hardware voorstel' wordt deze optie meegenomen als 3de keuze.

LTE-M

Wat is LTE-M

LTE-M staat voor LTE categorie-M. LTE (Long Term Evolution) wordt doorgaans “4G” genoemd en zorgt voor mobiel internet. Cat-M[21] is de IoT-variant van LTE en is gericht op energiezuinige apparaten die weinig internet nodig hebben. Omdat niet alle informatie op internet te vinden was, is er contact opgenomen met KPN. Hier is veel nuttige informatie uit gekomen.

Overbelasting van het mobiele netwerk

KPN heeft 2G, 3G, 4G, LTE-M en 5G netwerken. Deze generaties hebben allemaal een bepaalde capaciteit. Wanneer 4G ‘vol’ zit, kunnen 3G en 2G nog capaciteit over hebben. Mobiele telefoons zullen altijd verbinding maken met de generatie waarmee ze het beste signaal hebben. Wanneer deze generatie ‘vol’ zit, zullen ze niet automatisch overschakelen naar een andere generatie. Hierdoor is het gebruik van 2G een mogelijkheid voor het verrichten van PIN-transacties. Echter, 2G wordt niet opgenomen als optie, omdat deze generatie in de toekomst zal verdwijnen. Daarnaast is de kans dat 3G ‘vol’ zit bij festivals zeer aanwezig, waardoor óók niet voor deze optie gekozen wordt.

Application Priority

Het internet in Nederland is ‘net-neutraal’. Dat wil zeggen dat er geen verschil is in de ‘Quality of service’ voor apparaten en websites, zodat er geen concurrentievoordeel ontstaat. Een uitzonding op deze wet zijn kritieke processen. Wanneer congestie op een netwerk plaatsvindt, zullen kritieke processen operationeel moeten blijven. Denk hierbij aan internet voor noodhulpverlening. Wanneer de noodzaak voor een proces aangetoond kan worden, is het mogelijk om een gegarandeerde, ongestoorde beschikbaarheid te hebben. PIN-betalingen bij festivals vallen hier ook onder, om zo onrust te voorkomen. Deze voorrang heet ‘Application Priority’ en kan toegepast worden bij LTE-M, 4G en 5G.

Verschillen en overeenkomsten met 4G

LTE-M wordt aangeboden door KPN. LTE-M is een 4G verbinding voor IoT-apparaten. De verschillen en overeenkomsten zijn hieronder benoemd. In tabel 9 staan de specificaties.

Overeenkomsten tussen LTE (4G) en LTE-M

- Een SIM-kaart is vereist.
- Er is een databundel nodig.
- Er kan via IP gecommuniceerd worden.
- VoLTE is in de toekomst mogelijk voor LTE-M.

Verschillen tussen LTE (4G) en LTE-M

- Het is een andere SIM-kaart dan een 4G-SIM-kaart.
- Meerdere apparaten kunnen dezelfde databundel delen.
- Via een online dashboard voor het inzien en beheren van LTE-M apparaten.
- Het beheersysteem kan via een VPN gekoppeld worden met de back-end van de klant.
- Het heeft een latency van 50-100 ms. [36]

tabel 9: Specificaties van LTE en LTE-M. [37]

Specificaties	Cat-1 (LTE)	Cat-M1 (LTE-M)
Totale benodigde bandbreedte	20 MHz	1,4 MHz
Maximale downloadsnelheid	10 Mbit/s	1 Mbit/s (full-duplex) of 375 kbit/s (half-duplex)
Maximale uploadsnelheid	5 Mbit/s	
Energieverbruik	Laag	Erg laag
Duplex mode	Full-duplex	Full-duplex of half duplex

[21] **Cat**: LTE-Categorie/versie. Elke categorie heeft andere eigenschappen, zoals snelheid en energiezuinigheid. [53]

Frequentieband

ISP's bieden mobiel internet aan in meerdere LTE-banden, zie tabel 10. Telefoons die willen verbinden met het netwerk van de IPS, moeten een of meerdere van deze banden ondersteunen. In figuur 26 staat een voorbeeld van alle LTE-banden die een telefoon ondersteunt. Hieraan is tevens te zien dat deze telefoon alle LTE-banden ondersteunt die in Nederland gebruikt worden.

tabel 10: Beschikbare LTE-M banden in Nederland. [38]

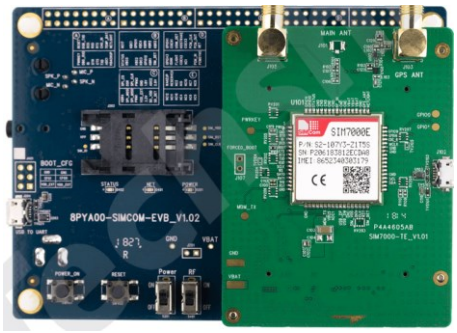
Provider	Band	Frequentie
T-Mobile	1	2100 MHz
KPN, T-Mobile, Vodafone	3	1800 MHz
KPN, Tele2, Vodafone	7	2600 MHz
T-Mobile	8	900 MHz
KPN, Tele2, Vodafone, Ziggo	20 (LTE-M)	800 MHz
T-Mobile	38	2600 MHz

COL-L29	GSM 900:35dBm, GSM 1800:32dBm, WCDMA 900/2100:25.7dBm, LTE Band 1/3/7/8/20/38/40:25.7dBm, Wi-Fi 2.4G:20dBm, Bluetooth:20dBm, Wi-Fi 5G:5150-5350MHz: 23dBm, 5470-5725MHz:30dBm, NFC: 60 dBuA/ min. op 10 m

figuur 26: Voorbeeld van de ondersteunde LTE-banden van een mobiele telefoon. De ondersteunde banden zijn te vinden in de handleiding. [54]

Hardware

Voor het Proof of Concept is de SIM7000E development kit uitgezocht. Deze kit bestaat uit 2 boards, zie figuur 27. Op de groene printplaat zit een SIM7000E modulatiechip bevestigd. Deze chip kan RF-signalen voor LTE-M en NB-IoT zenden en ontvangen. De blauwe printplaat is aan de groene bevestigd. Op deze printplaat zitten 2 SIM-kaartlezers en een USB-poort. Wanneer de juiste drivers op een computer of Raspberry Pi geïnstalleerd zijn, wordt de modem herkent wanneer het wordt aangesloten met een USB-kabel. Vervolgens kan er verbinding gemaakt worden met het internet. De SIM7000E kan via banden B3, B8, B20 en B28 met een LTE-M netwerk verbinden [39]. KPN levert alleen LTE-M op 800 MHz (band 20) [40], dus de kit is compatible.



figuur 27: De SIMCom SIM7000E Development Kit (€86,10). Een evaluation board waarmee een Raspberry Pi aan een LTE-M netwerk verbonden kan worden.

De maximale snelheid die behaald kan worden is 375 kbit/s uplink en 300 kbit/s downlink [39]. De snelheid van het LTE-M netwerk van KPN is 300 kbit/s up en 200 Kbit/s down op een half-duplex apparaat.

Installatie

De SIM7000E lijkt eenvoudig te implementeren. De volgende referenties kunnen gebruikt worden tijdens het Proof of Concept.

- De handleidingen en datasheet van de SIM7000E, waaronder een lijst met mogelijke AT-commando's[22].
- Een GitHub-repository met uitleg over hoe de SIM7070E development kit toegepast kan worden met een Raspberry Pi [41]. Deze kit komt erg overeen met de aangeschafte kit.
- Een contactpersoon bij TechShip, waar de evaluation kit is aangeschaft, die zijn hulp heeft aangeboden.
- Het KPN-forum voor LTE-M gerichte vragen.

[22] **AT-commando**: De commando's waarmee een modulatiechip wordt geconfigureerd. Deze commando's zijn te vinden in de handleiding van de fabrikant.

Conclusie

LTE-M is iets ingewikkelder dan Wi-Fi HaLow om te implementeren; toch lijkt het mij een geschikte keuze voor deze opdracht:

- Overal in Nederland is er dekking.
- De netwerkinfrastructuur is geregeld door een ISP.
- De snelheid en latency zijn volgens de eisen.
- Er is een groot vertrouwen dat dit protocol zal werken.

4.3.4. Keuze

Hardware voorstel

Van alle onderzochte protocollen waren er 3 waarvan het vertrouwen was dat ze aan alle specificaties voldoen en waardevol zijn voor Quintor. Daarom is hardware uitgezocht voor deze protocollen. De hardware bestaat uit evaluation kits en ander benodigd materiaal, zoals antennes en voedingen. Er is een hardwarevoorstel voorgelegd bij Quintor, zie Bijlage F. Aan de hand hiervan kon Quintor een keuze maken voor het te gebruiken protocol.

1. Wi-Fi HaLow (aanbevolen) €402,90
2. LTE-M (3 opties) €81,50, €86,10 en €136,54
3. Bluetooth coded PHY €90,29

Wi-Fi HaLow werd te duur bevonden, waardoor de keuze viel op LTE-M. De SIMCom SIM7070E is aangeschaft, omdat deze een grotere kans tot slagen had in tegenstelling tot de andere opties. Doordat deze optie uitverkocht was, is de SIMCom SIM7000E besteld, welke even goed was. Het verschil tussen de SIM7000E en SIM7070E zijn de ondersteunde frequenties en snelheden. Echter, beide modems[23] ondersteunen de LTE-M band van Nederland (band 20) en zijn beide sneller dan het bestaande LTE-M netwerk (300 kbit/s).

Keuze

Zowel Wi-Fi HaLow als LTE-M is een waardevolle protocol voor Quintor. Ze hebben beide hun eigen use cases. Het LTE-M protocol zal gebruikt worden in het Proof of Concept van de volgende fase.

4.3.5. Resultaat

Het grote aanbod aan protocollen was erg overweldigend. Er zijn meer protocollen onderzocht dan gepland.

Door grondig onderzoek naar de implementatie en benodigde hardware van protocollen is een goede, weloverwogen keuze gemaakt voor een protocol.

Naar aanleiding van het hardwarevoorstel heeft Quintor gekozen voor een LTE-M systeem. Met de opgedane kennis uit deze fase is er een goed beginpunt om te starten met het Proof of Concept in de volgende fase.

[23] **Modem**: De hardware die het mogelijk maakt om via het mobiele netwerk te communiceren.

4.4. Fase 4: Realisatiefase

In deze fase wordt het proof of concept iteratief gerealiseerd. Door een mock-bankapplicatie op te zetten met een LTE-M verbinding, wordt antwoord gegeven op de volgende deelvraag.

Deelvraag 4: Voldoet dat protocol ook aan de eisen volgens het Proof of Concept?

Wijziging ten opzichte van het afstudeerplan

Bij aanvang van deze fase zouden er nog maar 5 weken over zijn. Na het tussentijds assessment is besloten om de afstudeerperiode te verlengen. Er zijn 10 weken besteed aan de realisatiefase.

Ik heb besloten om te werken met sprints van 1 week in plaats van 2 weken. Hierdoor zijn meerdere momenten gecreëerd, waarop vooruitgekeken werd naar welke user stories de meeste prioriteit hadden. Hierdoor kon het project sneller bijgestuurd worden als dat nodig was. De retrospective van elke sprint staat beschreven in hoofdstuk 6.4 - Fase 4: Realisatiefase.

User stories en taken

In het begin van deze fase is een backlog opgesteld, te vinden in Bijlage G. Aan de user stories uit de backlog zijn 'story points' verbonden, die de verwachte realisatietijd aangeven. De geschatte tijd is beredeneerd door de user stories op te delen in kleinere taken.

Aan het begin van elke week zijn een aantal taken geselecteerd om uit te voeren tijdens de sprint met gezamenlijk ongeveer 40 story points. De bedrijfsmentor vervulde de rol van de opdrachtgever. Aan het eind van elke sprint is een minimum viable product[24] gepresenteerd en werd met hem besproken welke user stories er uitgevoerd zouden worden tijdens de volgende sprint. Elke sprint bestond uit:

- De user stories voor het realiseren van het Proof of Concept.
- Het bijhouden van het afstudeerverslag.
- Het voorbereiden van de demo-sessie (eens per 2 sprints).

In de vorige fase bleek dat het bijhouden van het afstudeerverslag vaak werd uitgesteld. Daarom is deze taak opgenomen in de sprints.

4.4.1. Sprint 1

Beroepstaken: C-10

User stories

Voor deze sprint heb ik de onderstaande user stories geselecteerd. Taken die tot het project behoren, zijn geprioriteerd met de MoSCoW-notatie. Taken over presentaties en verslagleggingen hebben een asterisk (*) als prioriteit.

Prior.	User story	Omschrijving	Points
*	ANVP-64	**Verslag bijwerken met feedback van TTA**	8
M	ANVP-21	**Als developer wil ik weten hoe Spring en Spring Boot werken, om het Proof of Concept te kunnen maken.** – Spring tutorials volgen – Leren hoe een unit test gemaakt wordt – Software Design maken	26
*	ANVP-28	**Verslag schrijven sprint 1**	8

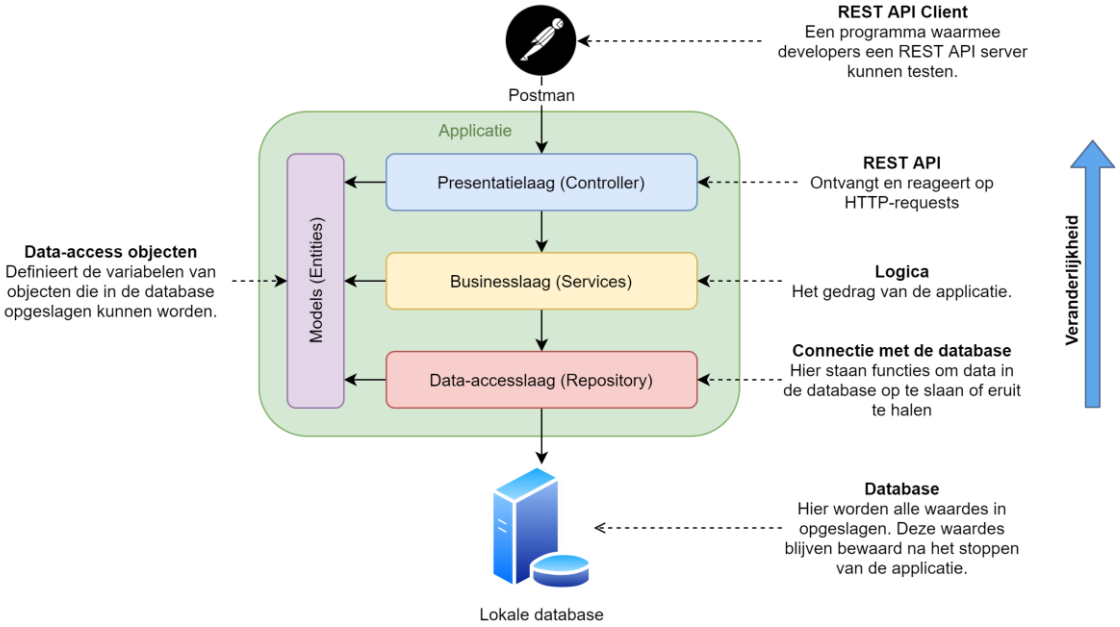
[24] **Minimum viable product**: Een deels-werkend product. De opdrachtgever geeft aan de hand hiervan feedback voor het verloop van de rest van het project.

Als eerste werd de feedback van het tussentijds assessment verwerkt in het verslag. Daarna was de belangrijkste taak om de Raspberry Pi te verbinden met het LTE-M netwerk. Helaas was er nog geen SIM-kaart beschikbaar, waardoor deze taak moest wachten tot een latere sprint. In plaats daarvan ben ik begonnen met het maken van het software-ontwerp en het bijleren over Spring, het Java-framework dat gebruikt moest worden. Er is besloten om te leren over unit tests, om vervolgens 'test-first' te programmeren. Dat houdt in: Eerst het schrijven van unit tests en vervolgens de software implementeren, om zo bugs vroegtijdig te detecteren en voorkomen. Daarnaast had ik geen ervaring met unittesten en dit was een vaardigheid die ik wilde leren.

Uitvoering

Software architectuur

In Spring zijn alle applicaties opgebouwd uit 3 lagen: de presentatielaag, de businesslaag en de data-accesslaag. Er is onderzoek gedaan naar wat zo'n softwareontwerp inhoudt. Deze informatie staat beschreven in figuur 28.

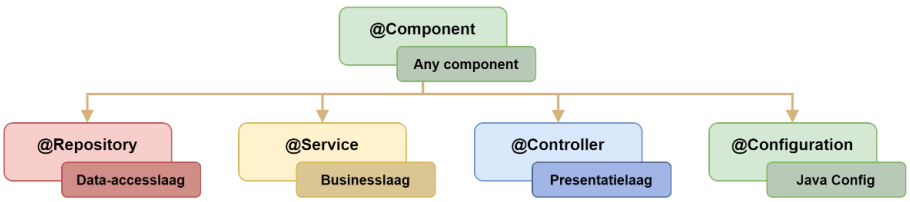


figuur 28: High level overview van een gebruikelijke Spring applicatie.

Leren over Spring

Een Spring applicatie bestaat uit 1 of meer 'components' van elke laag. Door het gebruik van annotaties, wordt gedefinieerd om wat voor een component het gaat, zie figuur 29. Het Spring-framework instantieert de benodigde componenten (dependencies), welke vervolgens worden meegegeven aan de dependant-componenten. Dit heet "dependency-injection". Unittesten wordt hierdoor eenvoudig gemaakt, doordat mock-componenten geïnjecteerd kunnen worden.

Spring Boot is een uitbreiding van Spring en biedt vele 'starters'. Dat zijn production-ready applicatiedelen die in elk project gebruikt kunnen worden, om sneller software te kunnen ontwikkelen.



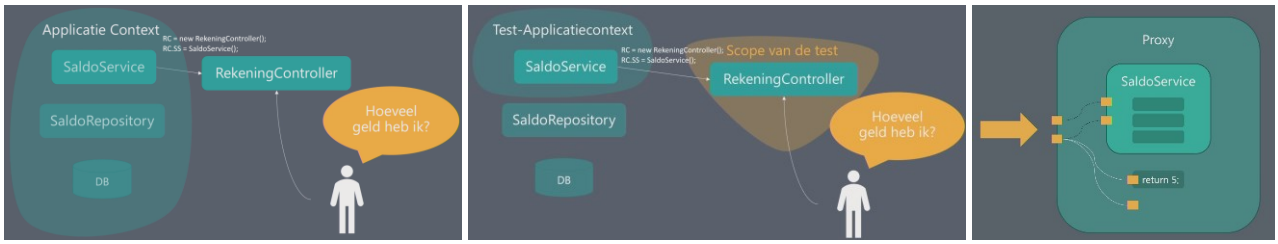
figuur 29: Spring componenten en bijhorende annotaties. [42]

Unit testen

Er is gekozen om unit tests te implementeren in het project. Hoewel het ontwikkelen van unit tests tijd kost, kan efficiënter geprogrammeerd worden, doordat bugs eerder worden gevonden. Daarnaast bieden deze tests een structuur voor het opzetten van de applicatie.

Unit tests controleren de werking van de kleinste testbare onderdelen. Deze onderdelen worden onafhankelijk van de rest van de applicatie getest. In figuur 30 is te zien dat een controller-component getest wordt, die afhankelijk is van een bepaalde service. Een mock-service wordt geïnjecteerd in plaats van de echte service.

Wanneer de controller functies aanroept op deze mock-service, zal altijd hetzelfde, vaste antwoord teruggegeven worden. Hierdoor is het gedrag van het te testen component niet afhankelijk van de werking van andere componenten. Een voorbeeld van een unit test is te zien in codefragment 1.



figuur 30: Methode voor het testen van voorbeeld-class 'RekeningController'. Links: applicatie, Midden: testomgeving (onafhankelijk van andere componenten), Rechts: Mockfuncties met hardcoded returnvalues. [43]

```
@Test
public void shouldMakeListOfUsers() {
    //Maak een lijst van 3 users aan.
    List<User> usersToAdd = new ArrayList<>(Arrays.asList(
            new User("Martijn", "martijn@email.nl"),
            new User("Jordy", "jordy@email.nl"),
            new User("Robin", "robin@email.nl")
    ));
    List<User> otherUsers = new ArrayList<>(Arrays.asList(
            new User("Martijn", "martijn@email.nl"),
            new User("Jordy", "jordy@email.nl"),
            new User("Robin", "robin@email.com") //let op dit verschil: .com ipv .nl
    ));

    //Voeg de users to aan de database.
    usersToAdd.forEach(mainService::slaUserOpInDatabase);

    //Haal alle users uit de database en zet ze in een nieuwe lijst, zodat ze vergeleken kunnen worden.
    List<User> savedUsers = new ArrayList<>();
    mainService.getAllUsers().iterator().forEachRemaining(savedUsers::add);

    //containsAll() werkt, omdat equals() geïmplementeerd is in de User-class.
    //Voer controle van unit test uit.
    Assertions.assertTrue(savedUsers.containsAll(usersToAdd));  //savedUsers bevat userToAdd: correct
    Assertions.assertFalse(savedUsers.containsAll(otherUsers)); //savedUsers bevat niet otherUsers: correct
}
```

codefragment 1: Een unit test-oefening. Deze test controleert of gebruikers worden toegevoegd door een service.

Softwareontwerp

Voordat er begonnen is met programmeren, is een softwareontwerp gemaakt, zie Bijlage G. Aan de hand hiervan kan de software gerealiseerd worden. In de alinea's hieronder wordt ingegaan op de keuzes die in dit ontwerp gemaakt zijn.

Databaseontwerp

Zowel de terminal als de server heeft een lokale database, waarin alle transacties worden opgeslagen. De database van de server bevat de ontvangen transacties van alle terminals. Een terminal heeft een database die alle geslaagde en niet-geslaagde transacties bevat die hij heeft verstuurd.

De volgende zaken worden bijgehouden voor alle transacties:

- Client: Timestamp voor zenden van een request en ontvangen van een response.
- Server: Timestamp voor het ontvangen van een request.
- Client: Boolean: Is er een response van de server ontvangen?
- Client: De signaalsterkte van de LTE-M modem tijdens het zenden van de transactie.

Hiermee worden de onderstaande zaken geanalyseerd. Hoewel dit niet allemaal eisen betreffen, is dit zeer handige informatie.

- Latency
  - Wat is de round-triptijd?
  - Hoeveel vertraging treedt op bij het zenden van een bericht (gebruikelijk en maximaal)?
  - Hoeveel vertraging treedt op bij het ontvangen van een bericht (gebruikelijk en maximaal)?
- Betrouwbaarheid van LTE-M netwerk
  - Wat is de kans dat een bericht niet integer verzonden of ontvangen wordt?
  - Vanaf welke signaalsterkte is het netwerk stabiel?

Protocol applicatielaag (HTTP)

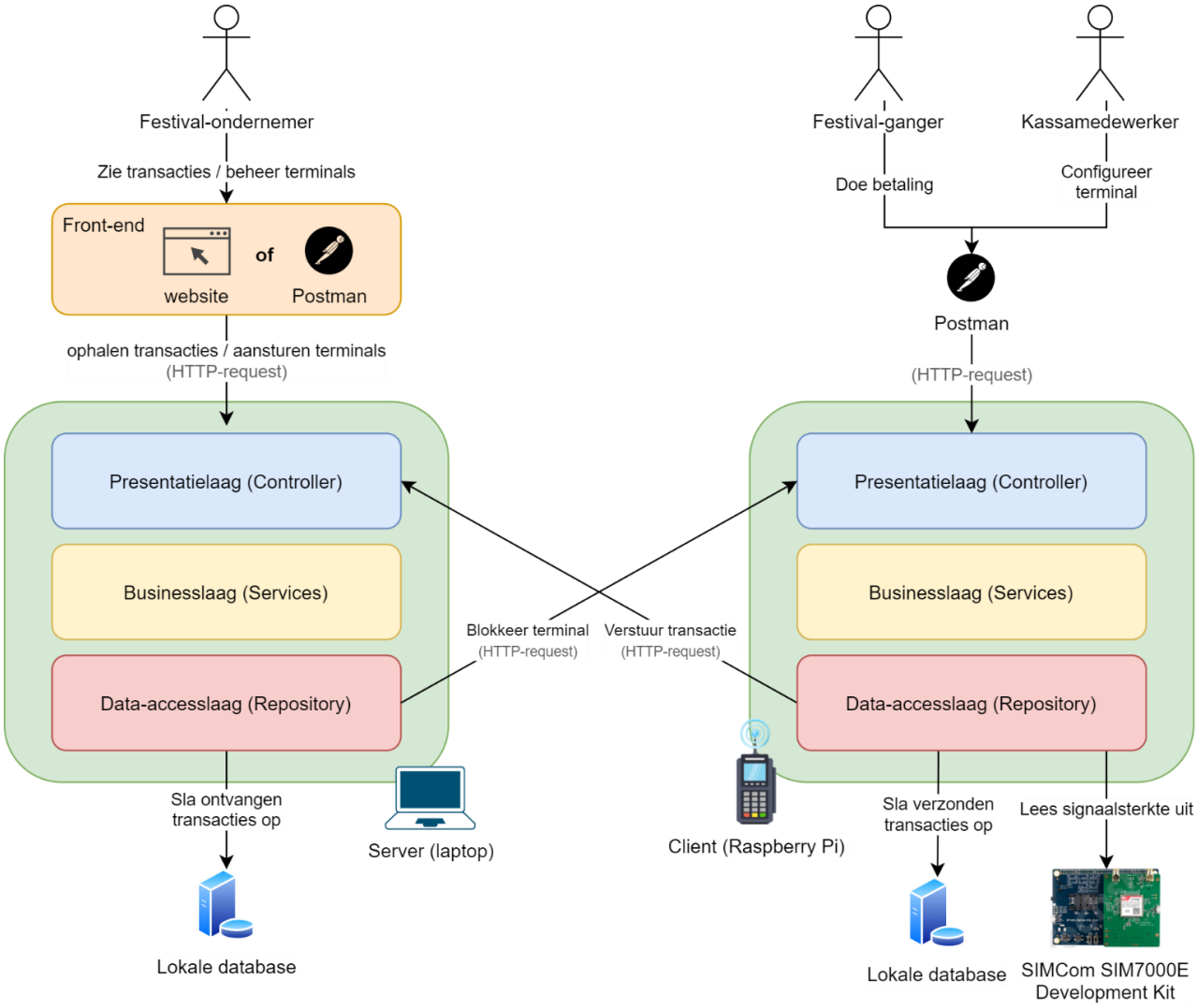
Zoals te zien is in figuur 31, communiceren de client en server met elkaar met HTTP-requests. Ik heb voor HTTP gekozen als applicatieprotocol, omdat het veel gebruikt wordt en daardoor goed wordt ondersteund. Het is daarnaast eenvoudig om met Spring een REST-API op te zetten.

Protocol transportlaag (TCP & UDP)

Voor HTTP-verbindingen wordt meestal TCP gebruikt. De belangrijkste verschillen tussen TCP en UDP staan beschreven in tabel 11. De mock-bankapplicatie wordt opgezet met TCP, waardoor de slaagkans van transacties wordt vergroot. Echter, doordat het opnieuw sturen van berichten automatisch wordt afgehandeld, is er geen zicht op hoeveel berichten er niet verzonden worden. Bij het testen van de verbinding wordt daarom UDP gebruikt. Wanneer betalingen niet succesvol verstuurd worden, is dat te zien door de database van de terminal en de server te vergelijken.

tabel 11: Verschillen tussen TCP en UDP.

Eigenschap	TCP	UDP
Overhead (Header)	20-60 bytes	8 bytes
Maximale hoeveelheid data per request.	Onbeperkt. Door TCP-sessies kunnen berichten gesplitst worden.	Grootte van 1 package: maximaal 1476 bytes. Berichten kunnen niet gesplitst worden door dit protocol. *Ethernet-MTU: 1500* *IP-header: minstens 20* *UDP-header: 8 –* *Data: 1472 bytes*
Betrouwbaarheid	Hoog. Wanneer een bericht niet aankomt, wordt het automatisch opnieuw verstuurd.	Laag. Het is onbekend of een bericht aankomt bij de server.
Snelheid	Langzaam. Er moet eerst een sessie opgezet worden met een handshake.	Snel. Een bericht kan gelijk verstuurd worden, zonder handshake.



figuur 31: High level overview van de opdracht. De PIN-terminals (rechts) sturen betaal-requests naar de PIN-server (links). De PIN-server kan een PIN-terminal blokkeren.

Ontworpen unit tests

Voor deze opdracht zijn er enkele unit tests geschreven, zie Bijlage G. Er is gekozen om alleen classes uit de businesslaag te testen. Bij het programmeren in Spring worden de connecties in de data-accesslaag en presentatielaag uitgevoerd door Spring Boot-starters. Dit zijn veelgebruikte libraries die al uitvoerig getest zijn. Hoewel deze zelf geconfigureerd moeten worden, hebben deze lagen geen tests nodig. In de businesslaag zitten classes die zelf geïmplementeerd moeten worden. Dat is de software die getest dient te worden.

4.4.2. Sprint 2

Beroepstaken: D-14, D-15

User stories voor deze sprint & bijwerken backlog

Het doel van deze sprint is om de basis op te zetten voor de mock-bankapplicatie. Daarvoor worden onderstaande user stories uitgevoerd.

Prior.	User story	Omschrijving	Points
M	ANVP-22	**Als een klant wil ik kunnen pinnen bij festivals, zodat ik niet contant hoef te betalen** – REST API maken met Spring – Betaling verwerken in een database	6
S	ANVP-34	**Als developer wil ik dat de applicatie te testen is zonder verbinding, zodat ik vroegtijdig fouten kan voorkomen** – Unit tests schrijven voor API requests voor server en client – Unit tests schrijven voor verwerken van goede en foute betalingen (servicelaag) – Unit test uitvoeren op GitLab na elke push	11
M	ANVP-40	**Performance testen: De impact van de signaalsterkte op de verbinding onderzoeken, zodat daar rekening mee gehouden kan worden in volgende projecten** – Test de round-trip time op verschillende locaties (met pings en betaalverkeer) – Testuitkomst analyseren – Continu zenden en meten hoeveel berichten er per seconde verwerkt kunnen worden met 1 LTE-M apparaat.	19
*	ANVP-58	**Verslag schrijven sprint 2**	8
*	ANVP-66	**Demo sprint 2**	5

User story ANVP-40 (Performance testen) is erg groot en wordt opgesplitst. Hierdoor kan een deel al uitgevoerd worden in sprint 2. De onderstaande taken worden afgescheiden:

- ANVP-41: Timer programmeren om HTTP-requests te timen.
  - Deze taak is ook gewijzigd naar must-have. De timestamps van transacties zijn belangrijk om op te nemen, omdat daarmee de latency berekend kan worden (eis S4).
- ANVP-42: Signaalsterkte loggen bij elk request.
  - Deze user story wordt niet gewijzigd naar must-have, omdat er verwacht wordt dat deze taak meer tijd kost dan andere, belangrijkere user stories.

Uitvoering

Basis betaalfunctionaliteit

Tijdens deze sprint is het eerste deel opgezet voor de betaalterminal (de client) en de betaalserver. Deze applicaties kunnen bediend worden via HTTP-requests, bijvoorbeeld via Postman[25]. De client kan de terminal registreren en betalingen naar de betaalserver versturen. Deze transacties worden bij zowel de client als de server opgeslagen in een database.

Testen

In Bijlage G zijn de unit tests gedocumenteerd. In codefragment 2 is de implementatie te zien van unit test 6.1.1. Elke keer dat de applicatie gerund wordt op de ontwikkelcomputer, worden de unit tests automatisch uitgevoerd, zie figuur 32.

Het oorspronkelijke plan was om *test-first* te programmeren. Dat houdt in dat de tests worden geprogrammeerd voordat de daadwerkelijke software geschreven wordt. Echter, het bleek dat het programmeren hierdoor erg lastig wordt. Hoewel er al een software-ontwerp met gedocumenteerde unit tests beschikbaar is, kunnen deze plannen veranderen bij het programmeren. Een developer met minder ervaring loopt hier het risico snel vast te lopen. Na het schrijven van enkele testen, heb ik besloten om te wachten met het implementeren van de overige unit tests, totdat de software hiervoor geschreven is. Hierdoor is bij het schrijven van de testen bekend hoe de tests uitgevoerd moeten worden.

[25] **Postman**: Een programma om HTTP-requests te sturen naar een server.

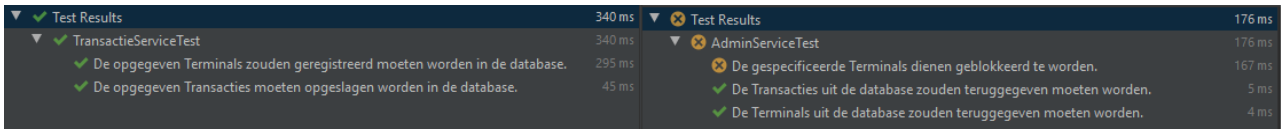
```java
@Test
 @DisplayName("De opgegeven Transacties moeten opgeslagen worden in de database.")
 void verwerkTransactieTest() {
     //Stap 1: Sla een paar willekeurige Transacties op in database via verwerkTransatie(..).
     //Genereer willekeurige Transacties.
     List<Transactie> nieuweTransacties = new ArrayList<>(Arrays.asList(
             TransactieGenerator.getRandomTransactie(),
             TransactieGenerator.getRandomTransactie(),
             TransactieGenerator.getRandomTransactie()
     ));

     for (Transactie transactie : nieuweTransacties) {
         //Genereer random terminal
         Terminal randomTerminal = TerminalGenerator.getRandomTerminal();
         //Voeg Terminal toe aan database.
         terminalRepository.save(randomTerminal);
         //Koppel Terminal aan transactie
         transactie.setTerminal(randomTerminal);
         //Transactie toevoegen aan database.
         transactieService.verwerkTransactie(transactie.getTerminal().getId(), transactie);
     }

     //Stap 2. Haal alle opgeslagen Transacties uit TransactieRepository.
     List<Transactie> transactiesInDatabase = transactieRepository.findAll();

     //Vergelijk
     Assertions.assertEquals(nieuweTransacties, transactiesInDatabase);
 }
```

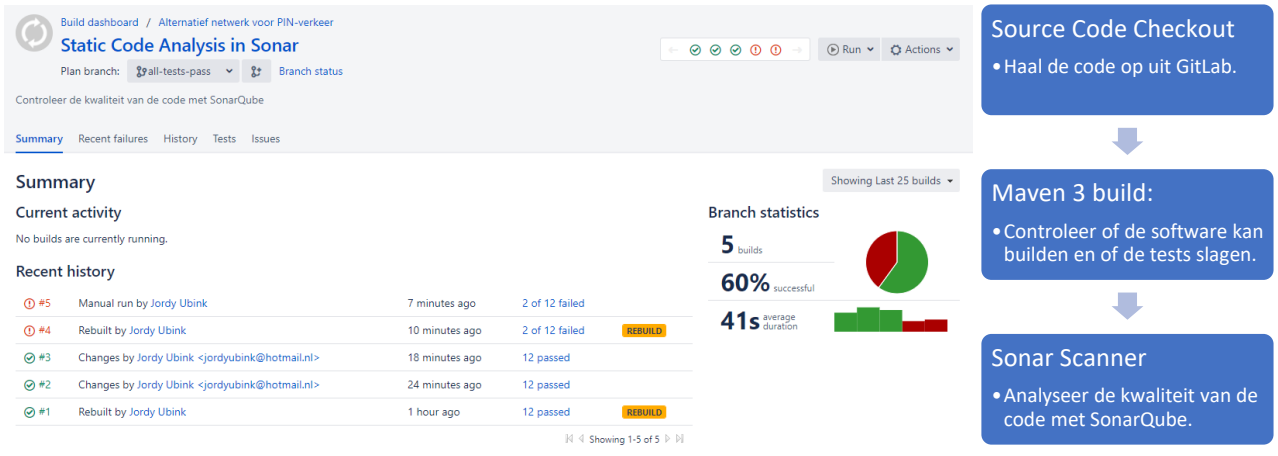
codefragment 2: Implementatie van unit test 6.1.1: TransactieService : verwerkTransactie.



figuur 32: De uitkomst van de unit tests van 2 verschillende classes. Rechts is te zien dat een test gefaald is.

Continuous integration

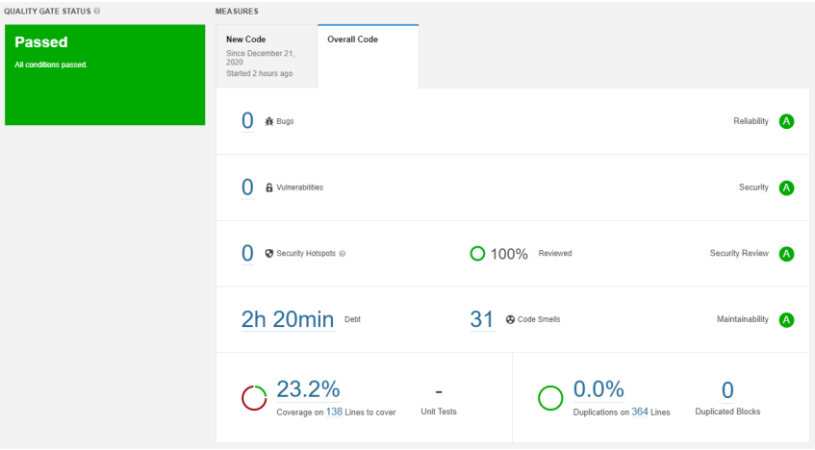
De softwareprojecten zijn gekoppeld aan 'Bamboo', een CI/CD-omgeving[26] die Quintor gebruikt voor haar projecten. In Bamboo is de CI ingericht om automatisch de geschreven code te controleren, zie figuur 33. Wanneer er een unit test niet slaagt, zal dit in deze omgeving te zien zijn.



figuur 33: Links: CI/CD-omgeving voor de backend van de betaalserver. In dit voorbeeld zijn de laatste 2 testen gefaald. Rechts: De CI-stappen die doorlopen worden door Bamboo.

[26] **CI/CD:** Continuous Integration (CI): Het automatisch testen en analyseren van code bij elke commit. Continuous Delivery (CD): Het automatisch 'builden' van de software, eventueel met automatische deployment.

Daarnaast wordt de code automatisch geanalyseerd met SonarQube, zie figuur 34. Deze tool voert geen unit tests uit, maar analyseert de code op bugs, vulnerabilities, security hotspots, test coverage en ‘code smells’. Wanneer SonarQube fouten vindt, wordt aangeven wat er fout is en waarom het fout is. Door het detecteren en oplossen van deze fouten wordt de kwaliteit van de code verbeterd.



figuur 34: SonarQube geeft inzicht in de bugs, beveiligingsproblemen en testcoverage van het project.

4.4.3. Sprint 3

Beroepstaken: C-9

User stories

Voor deze sprint zijn de onderstaande user stories gepland. Als deze user stories volbracht worden voordat de LTE-M SIM-kaart geleverd is, zullen extra user stories toegevoegd worden aan de sprint.

Dinsdag is de SIM-kaart bezorgd, waarna user story ANVP-20 werd toegevoegd aan de sprint: het opzetten van een LTE-M verbinding via een Raspberry Pi. Omdat dit de belangrijkste user story is, heeft deze taak voorrang gekregen boven de andere user stories.

Het LTE-M abonnement is afgenomen bij SIMPoint, omdat dit bedrijf een bundel van 500MB levert. Hierdoor is het mogelijk om veel tests uit te kunnen voeren.

Prior.	User story	Omschrijving	Points
S	ANVP-34	**Als developer wil ik dat de applicatie te testen is zonder verbinding, zodat ik vroegtijdig fouten kan voorkomen** – Unit test schrijven voor API requests voor server en client – Unit tests schrijven voor verwerken van goede en foute betalingen (servicelaag) – Unit test uitvoeren op GitLab na elke push	11
M	ANVP-41	**Timer programmeren om HTTP-requests te timen**	2
*	ANVP-59	**Verslag schrijven sprint 3**	8
M	ANVP-20	(Toegevoegd dinsdag 22 december) **Als developer wil ik een LTE-M systeem dat berichten kan ontvangen en versturen, zodat ik een betaalsysteem kan bouwen** – Flashen Raspberry Pi – Firewall instellen waardoor de databundel alleen gebruikt wordt voor een specifiek IP-adres – Zorgen dat het LTE-M apparaat kan pingen – TCP-verbinding opzetten tussen client en server – Stappenplan documenteren voor Quintor	14

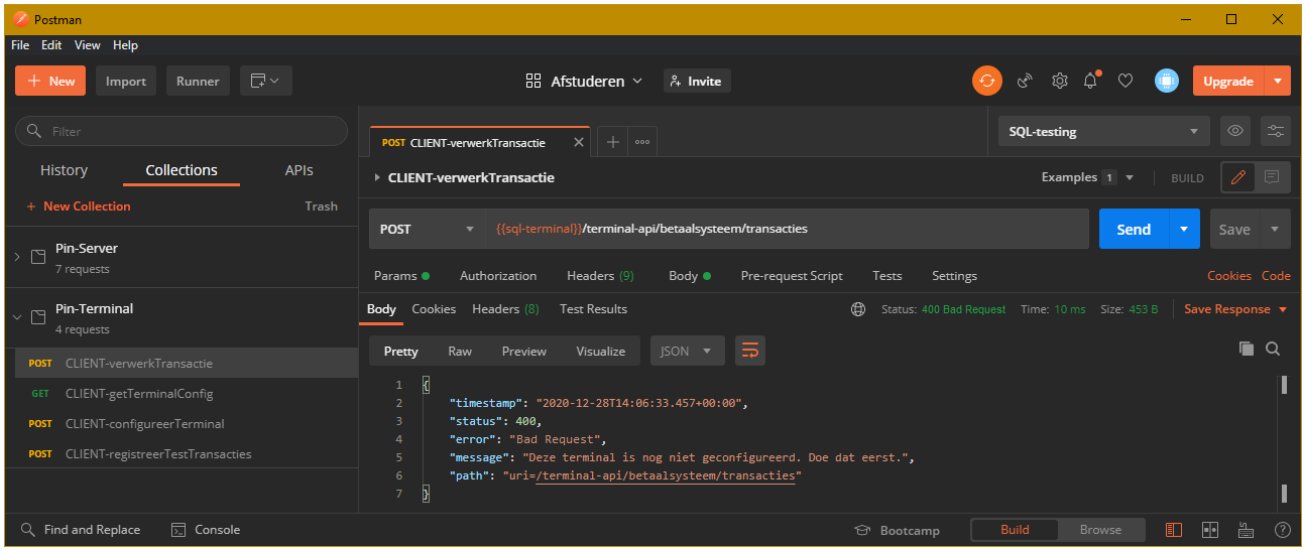
Uitvoering

Verder werken aan de software

Op de eerste dag van deze sprint is er verder gewerkt aan de software van het PIN-systeem. Wanneer er fouten optreden, zal de user een duidelijke response ontvangen waarin de fout en de oplossing beschreven staat, zie figuur 35.

Daarnaast wordt het CI-proces van de vorige sprint opgezet voor zowel de server- als de terminalsoftware en wordt dit proces automatisch getriggerd bij elke commit.

Tot slot is het gelukt om de terminalsoftware op de Raspberry Pi te runnen. Vanaf de ontwikkelcomputer is ‘remote debugging’ mogelijk.

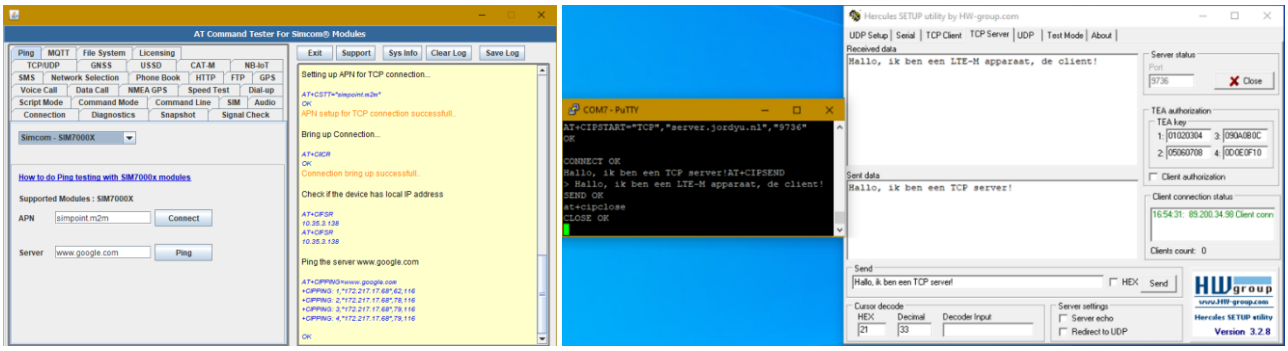


figuur 35: Wanneer gepoogd wordt een transactie te versturen, zonder eerst de terminal te registreren, zal de user een duidelijke melding te zien krijgen. De “Status: 200 OK” melding wordt op een later moment opgelost.

AT commando's

Om te testen of de LTE-M modem werkt, wordt het geconfigureerd en getest door AT-commando's te sturen. Dit is gedaan met behulp van de handleidingen van Techship. Dit was geen eenvoudig proces, omdat elke chip een aantal unieke commando's heeft, die niet overeenkomen met de handleiding. Na enig uitzoekwerk was het mogelijk om een verbinding op te zetten met een server over TCP. De modem kon alleen als client gebruikt worden, niet als server. Dat komt doordat SIMPoint een NAPT-firewall (Network Address Port Translation) heeft ingesteld. Dit is een beveiligingsfeature waardoor alleen verbindingen opgezet kunnen worden als ze door het LTE-M apparaat worden geïnitieerd. Hierdoor is het nog niet mogelijk om de apparaten op afstand te blokkeren, wat een van de eisen is uit het Plan van Aanpak. Bij KPN kan dit uitgezet worden door een andere APN te selecteren. Het moest nog uitgezocht worden of dit ook mogelijk was bij SIMPoint.

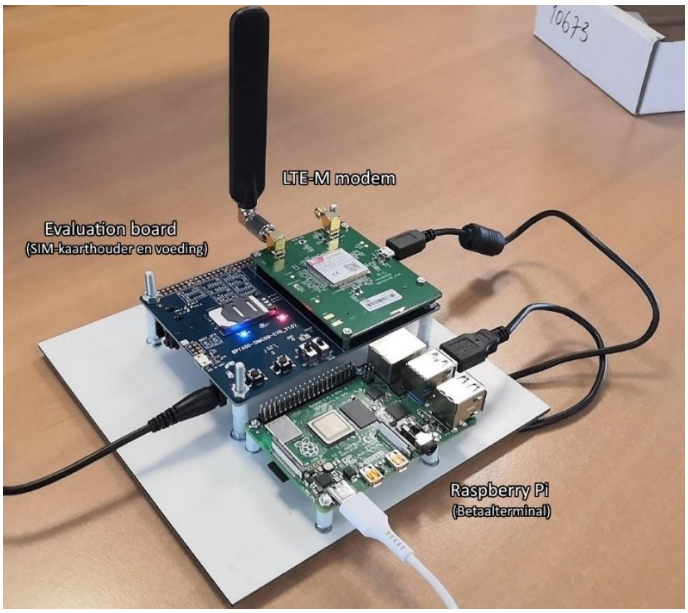
Later is de tool "AT Command Tester" gevonden. Dit is een grafische interface waarmee modems geconfigureerd kunnen worden, zie figuur 36. Deze tool geeft tevens inzicht in de benodigde AT-commando's voor vele specifieke opdrachten, welke ook gebruikt kunnen worden zonder dit programma. Hierdoor was het mogelijk om te pingen met andere hosts.



figuur 36: Links: AT Command Tester. Een grafische interface om modems mee te configureren. Rechts: Een TCP-verbinding die is opgezet met AT-commando's. De LTE-M modem is de client.

LTE-M met Raspberry Pi

Er is gepoogd om de aangeschafte LTE-M module te gebruiken als netwerk-interface voor een Raspberry Pi. Hierbij is de documentatie van TechShip gebruikt als naslagwerk. Via tools als 'netwerk-manager' en 'modem-manager' zou een verbinding in te stellen moeten zijn, maar dit is niet gelukt tijdens deze sprint.



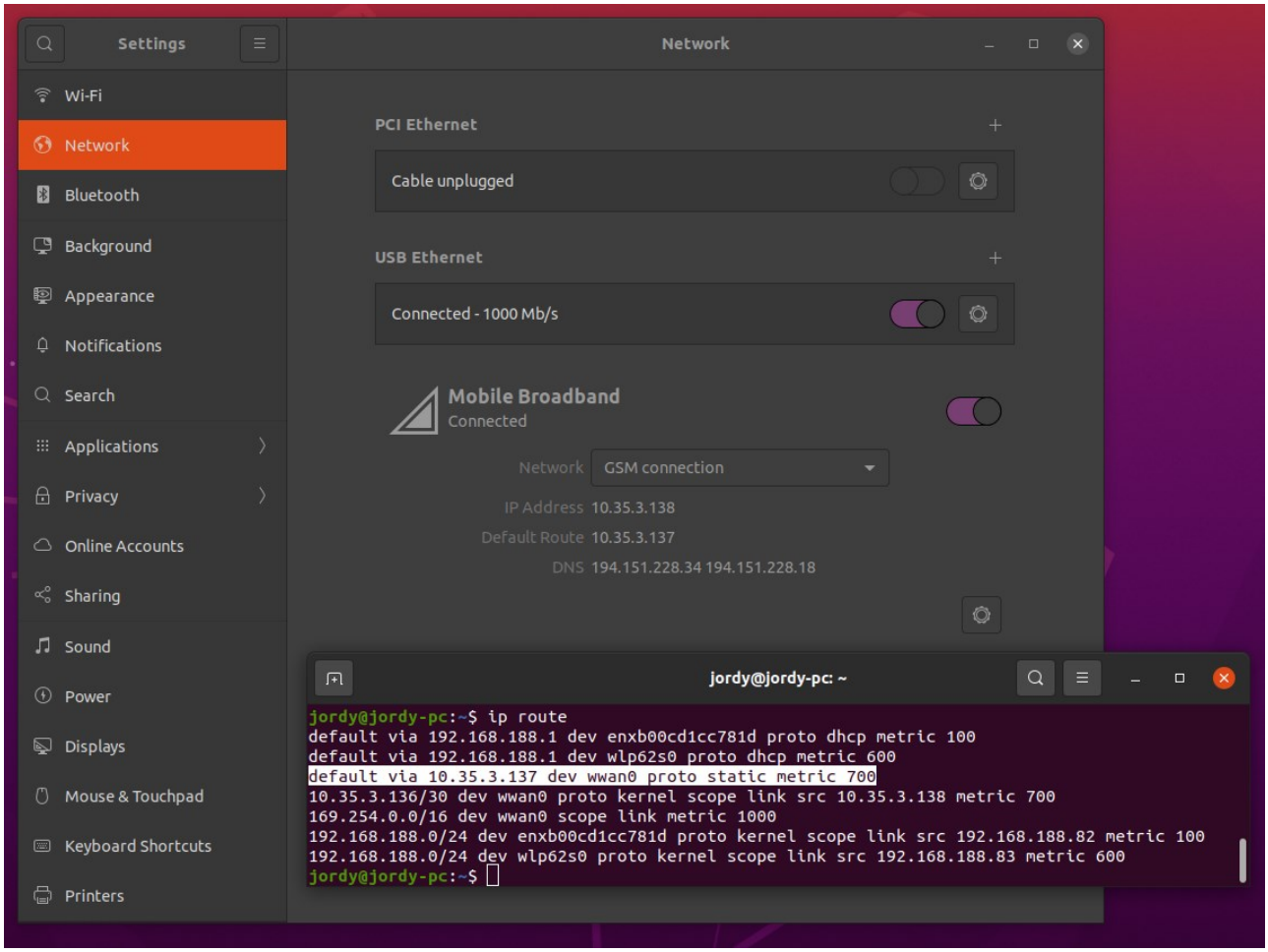
figuur 37: Een setup is gebouwd om de LTE-M verbinding te testen met een Raspberry Pi.

LTE-M met Ubuntu

Het instellen van een LTE-M verbinding op de Raspberry Pi wilde niet lukken. Daarom is geprobeerd om deze interface te gebruiken op een eigen laptop. Ik heb besloten om Ubuntu te gebruiken als operating system, omdat het gebaseerd is op Debian, net als Raspbian (het OS van de Raspberry Pi). De configuratie zou daarom ongeveer hetzelfde moeten verlopen. Daarnaast is Ubuntu een veel gebruikte distributie waar veel informatie voor te vinden is.

Dat bleek erg eenvoudig te gaan. Na het aansluiten van de LTE-M module, was de 'mobile broadband' optie beschikbaar in de netwerkinstellingen, zie figuur 38. Na het invoeren van de APN, kan er verbinding gemaakt worden met het internet. Dit bevestigt dat het mogelijk is om gebruik te maken van de LTE-M module. In de volgende sprint wordt onderzocht hoe deze verbinding opgezet kan worden via de terminal, om te achterhalen wat de stappen zijn voor de installatie bij een Raspberry Pi.

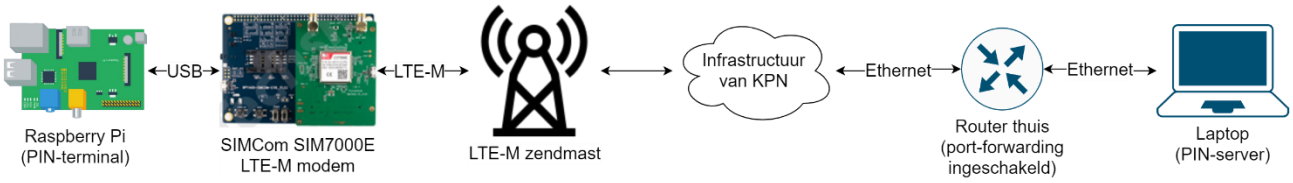
De module kan niet gebruikt worden met Windows 8-10, omdat deze besturingssystemen alleen werken met modems die de 'Mobile Broadband Interface Model'-interface gebruiken. Deze interface wordt niet ondersteund door de SIM7000E [44] en valt daardoor buiten de scope van dit project.



figuur 38: Netwerkinstellingen van Ubuntu. LTE-M module werkt out-of-de-box wanneer het via USB wordt aangesloten op een computer. In de terminal is te zien dat een default gateway voor deze interface is ingesteld. Wanneer de bekabelde- en Wi-Fi-verbinding uitgeschakeld worden, wordt de LTE-M verbinding gebruikt (gemarkeerde regel).

Technische infrastructuur

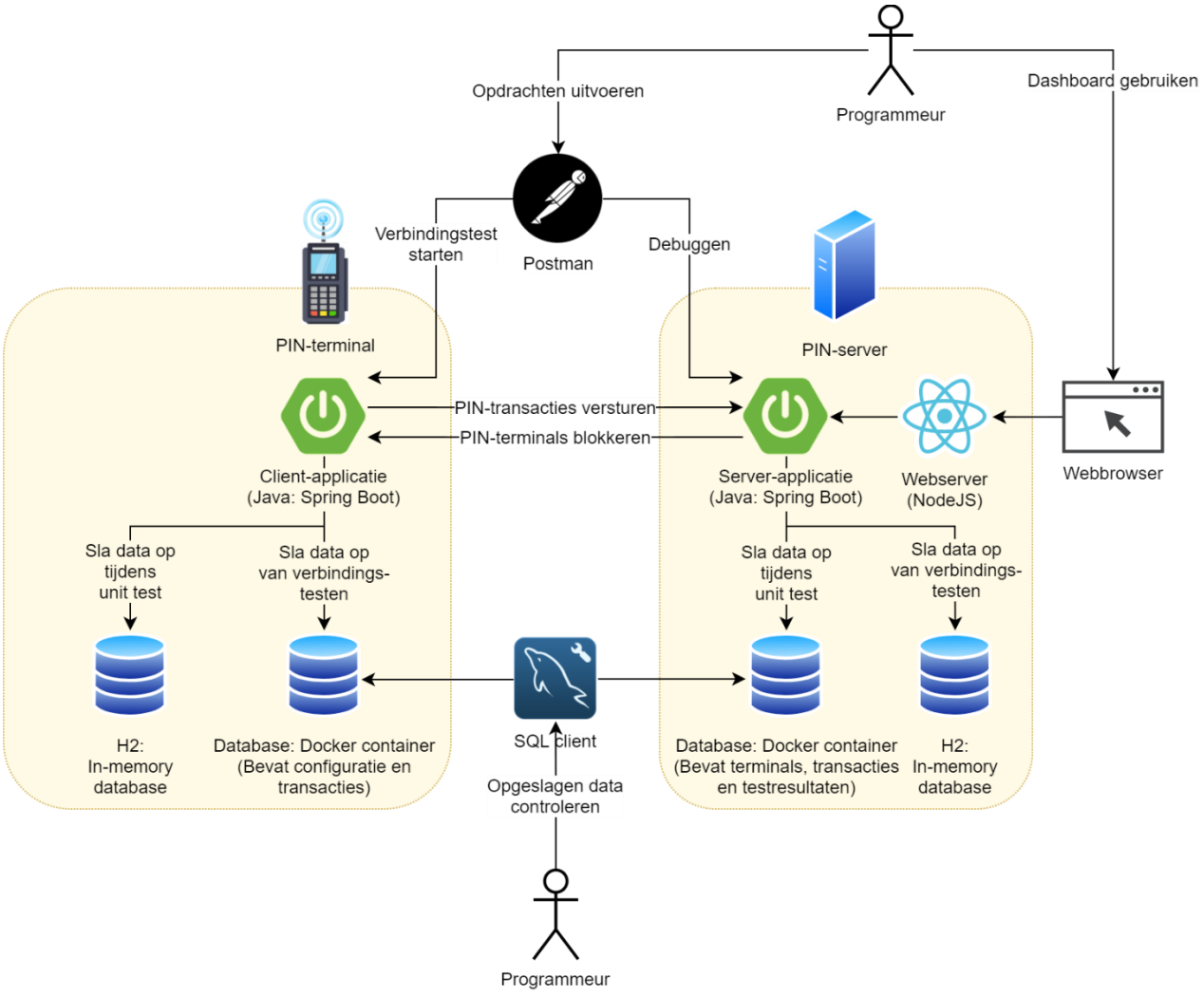
Beroepstaak C-9 kan in mindere mate aangetoond worden met dit project. Dit komt doordat van tevoren niet duidelijk was welk protocol en hoeveel apparaten er gebruikt zouden gaan worden. De uiteindelijke technische infrastructuur is te zien in figuur 39.



figuur 39: Infrastructuur van de verbonden apparaten.

De services die draaien op de Raspberry Pi en de laptop zijn te zien in figuur 40. Commando's kunnen naar de server en client verstuurd worden door middel van HTTP-requests. De databases worden als Docker container uitgevoerd. Hierdoor kunnen ze gemakkelijk aangemaakt, gestart en gestopt worden. Ook is het installatieproces voor de PC bijna hetzelfde als voor de Pi. Daarnaast wilde ik leren omgaan met Docker, omdat dit een zeer nuttige vaardigheid is om te beheersen.

Naast de database-container, is ook een H2-database geïmplementeerd. Dit is een in-memory database waarmee unit tests uitgevoerd kunnen worden. Deze database wordt altijd geleegd voordat de unit tests worden uitgevoerd, zodat de unit test altijd met dezelfde precondities starten.



figuur 40: Services die gebruikt worden in dit project.

4.4.4. Sprint 4

User stories

Prior.	User story	Omschrijving	Points
M	ANVP-20	**Als een developer wil ik een LTE-M systeem dat berichten kan ontvangen en versturen, zodat ik een betaalsysteem kan bouwen.** – Flashen Raspberry Pi – Firewall instellen waardoor de databundel alleen gebruikt wordt voor een specifiek IP-adres. – Zorgen dat het LTE-M apparaat kan pingen. – TCP-verbinding opzetten tussen client en server. – Stappenplan documenteren voor Quintor	14
M	ANVP-41	**Timer programmeren om HTTP-requests te timen**	2
*	ANVP-60	**Verslag schrijven sprint 4**	8
*	ANVP-65	**Verslag updaten voor nieuwe TTA.**	8

Uitvoering

LTE-M verbinding met Raspberry Pi

In deze sprint is opnieuw geprobeerd om de Raspberry Pi met het internet te verbinden via de LTE-M modem. De SD-kaart is opnieuw geflasht. De oude, incorrecte configuratie werd gewist, zodat met een schone lei begonnen kan worden. Het globale configuratieproces staat hieronder in grote lijnen beschreven. Hierna is de modem klaar voor gebruik.

1. Update het systeem.
2. Zorg ervoor dat het juiste protocol wordt gebruikt voor GSM-verbindingen.
3. Installeer een network-manager.
4. Configureer de GSM-verbinding.
5. Test de verbinding.

Indien gewenst, kan de Raspberry Pi geconfigureerd worden om de LTE-M modem alleen te gebruiken voor PIN-transacties. Hiermee wordt voorkomen dat de databundel snel leeg raakt door updates en andere verbindingen. Dit kan geconfigureerd worden met “ip route”-commando’s.

Schrijven installatiehandleiding

Om het installatieproces goed over te dragen aan Quintor, is er een handleiding geschreven, zie Bijlage H. Met deze handleiding kunnen de volgende systemen gebruikmaken van de LTE-M modem:

- LTE-M modem stand-alone
- Raspberry Pi (OS: Raspbian of Ubuntu voor Raspberry Pi)
- Ubuntu Desktop
- Windows 7 (hardware is niet geschikt voor Windows 8 & 10)

Het was de bedoeling om alleen de eerste 2 opties uit te zoeken. Echter, de installatieprocessen van Ubuntu en Windows 7 zijn ook toegevoegd aan de handleiding, zodat de LTE-M modem voor meerdere toepassingen gebruikt kan worden.

In de handleiding is het configuratieproces stap-voor-stap beschreven, zodat het eenvoudig doorlopen kan worden door iemand met weinig ervaring met de Linux-terminal. Een nieuwe Raspberry Pi is geïnstalleerd en geconfigureerd om alle stappen van de handleiding te controleren.

4.4.5. Sprint 5

User stories

Verandering user stories prioriteit naar 'must have'

Hieronder staan de user stories die gekozen zijn voor deze sprint. User stories ANVP-41 en ANVP-42 waren eerst bestempeld als 'should have' en 'could have'. Deze heb ik naar 'must have' veranderd wegens de onderstaande redenen.

ANVP-41: Zoals genoemd in het pakket van eisen, is het wenselijk dat de round-trip time korter is dan 1000 ms. Met deze user story wordt de latency getimed, zodat gecontroleerd kan worden of het protocol aan deze eis voldoet.

ANVP-42: Een PIN-systeem moet een stabiele verbinding hebben. Door het loggen van de signaalsterkte kan geanalyseerd worden hoe bepaalde factoren de verbinding beïnvloeden.

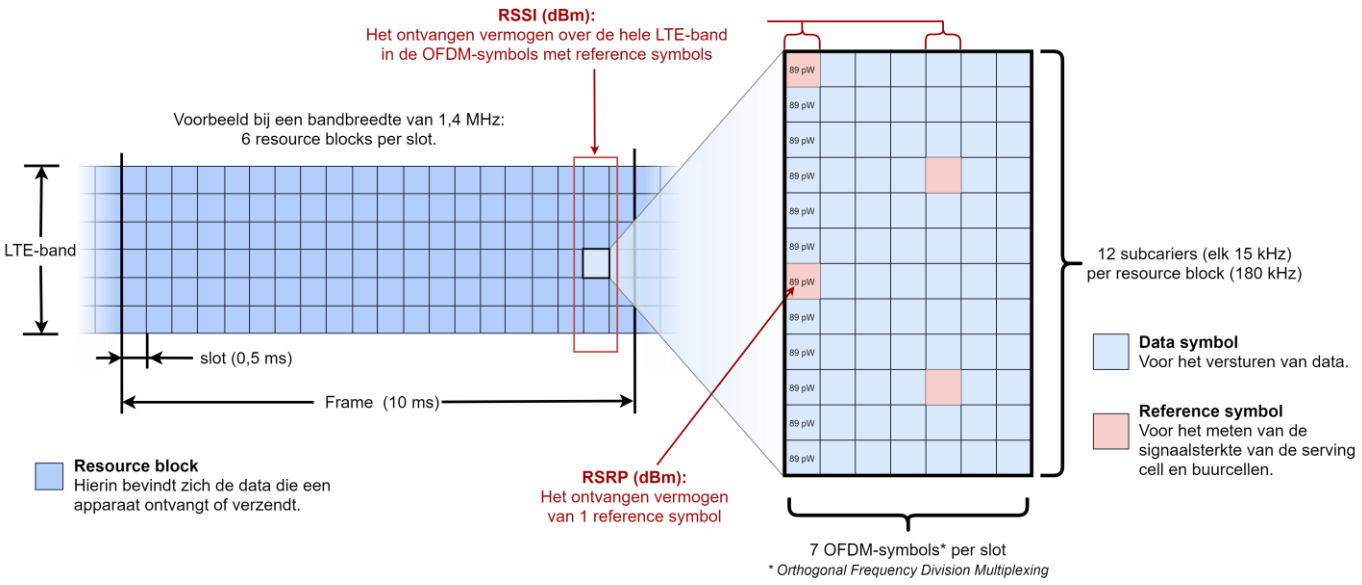
Prior.	User story	Omschrijving	Points
M	ANVP-41	**Timer programmeren om HTTP-requests te timen**	2
M	ANVP-42	**Signaalsterkte loggen bij elke request**	30
*	ANVP-61	**Verslag schrijven sprint 5**	8
*	ANVP-67	**Demo sprint 4** – Is wegens de kerstdagen uitgesteld naar sprint 5.	5

Uitvoering

In deze sprint is software ontwikkeld om signaalwaardes voor ANVP-42 te loggen. In sprint 10 worden deze gebruikt om conclusies mee te trekken.

Signaalsterkte

Op de volgende pagina staan 4 soorten signaalsterktes beschreven. Deze worden gemeten en gelogd bij elke transactie. Deze waardes geven inzicht in de kwaliteit van de verbinding. De waardes worden gemeten met een referentiesignaal dat de zendmast periodiek verstuurt, zie figuur 41. De kwaliteit die bij deze gegevens hoort, is te vinden in tabel 12.



figuur 41: Illustratie van hoe de LTE-bandbreedte verdeeld is in resource blokken. Hierin is de afkomst van de RSRP- en RSSI-waardes te vinden. Illustratie gebaseerd op [45] & [46].

RSRP **Reference Signal Received Power (De signaalsterkte van eigen verbinding)**
Het vermogen van het ontvangen referentiesignaal. Hierbij is de noise van andere signalen niet opgenomen. De RSRP-waarde is het hoogst als het apparaat dicht bij een zendmast staat, zonder obstakels in de buurt.

RSSI **Received Signal Strength Indicator (Totale signaalsterkte)**
Het totale vermogen dat de antenne detecteert binnen de bandbreedte. Hierbij worden ook ruis (noise) en storingen door andere channels gemeten.
$$RSSI = wideband\ power = noise + serving\ cell\ power + interference\ power$$

RSRQ **Reference Signal Received Quality (De kwaliteit van de verbinding)**
RSRQ is de verhouding tussen RSRP en RSSI. Deze waarde geeft inzicht in de kwaliteit van de verbinding. In dunbevolkte gebieden is de signaalsterkte laag (lage RSRP), maar er kan alsnog een betrouwbare verbinding worden opgezet (hoge RSRQ), doordat er weinig verstoringen zijn.
Als een andere LTE-cel een hogere RSRQ-waarde heeft, dan wordt er overgeschakeld naar deze cel.
De formule voor RSRQ luidt [46]:
$$RSRQ\ (factor) = \frac{RSRP\ (mW)}{RSSI\ (mW)} \times aantal\ resouce\ blocks$$

SINR **Signal to Interference & Noise Ratio**
De SINR (ook wel S/R) is afhankelijk van de RSRQ en de cel load. Deze waarde geeft inzicht in de throughput bij verschillende omstandigheden. De SINR-waarde is niet in de LTE-standaard gedefinieerd en wordt door fabrikanten op verschillende manieren berekend. [47]

tabel 12: Kwaliteit van het ontvangen signaal voor RSRP-, RSRQ- en S/N-waardes. [48] [49] Voor deze waardes geldt: Hoger is beter.

Signaalkwaliteit	RSRP (dBm) (vermogen: verbinding)	RSRQ (dB) (kwaliteit)	RSSI (dBm) (vermogen: totaal)	SINR (dB) (kwaliteit)
Range	Max: -44 dBm Min: -140dBm	Max: -3dB Min: -19,5dB	n.v.t.	n.v.t.
Excellent	> -80	> -10	> -65	> 20
Goed	-80 tot -90	-10 tot -15	-65 tot -75	13 tot 20
Redelijk	-90 tot -100	-15 tot -20	-75 tot -85	0 tot 13
Slecht	≤ -100	≤ -20	≤ -85	≤ 0

4.4.6. Sprint 6 & 7

Beroepstaken: D-14

User Stories

Na uitgebreid onderzoek bleek dat UDP te complex is om te gebruiken voor de mock-bankapplicatie, zoals beschreven is in sprint 1. Dit komt door de 'Spring Boot starter' die de HTTP-verbindingen regelt. Daarom zal TCP gebruikt worden, in plaats van UDP. Hierdoor wijzigen er twee user stories in de backlog:

Ten eerste vervalt user story ANVP-38, want er hoeft geen retry-policy geprogrammeerd te worden. TCP zorgt voor een stabiele verbinding.

Ten tweede moest een andere manier bedacht worden om de kwaliteit van de verbinding te analyseren. Daarom wordt een nieuwe user story toegevoegd: ANVP-69. Deze user story is ook een stuk complexer, waardoor er 2 sprints voor nodig zijn. Dit kost veel tijd, maar het is essentieel om de kwaliteit van het netwerk te onderzoeken.

Sprint	Prior.	User story	Omschrijving	Points
6	M	ANVP-49	**Als ondernemer wil ik de betaalterminals op afstand kunnen blokkeren, zodat ze niet door dieven gebruikt kunnen worden.** Variabel lezen/wijzigen met GET/POST request.	1
Geschrapt		~~ANVP-38~~	**Als ondernemer wil ik dat 99,9% van de transacties slagen, omdat klanten anders lang moeten wachten of contant moeten betalen.** – Maximaal 3 betaalpogingen doen. (1 poging = 2 seconden)	
6 & 7	M	ANVP-69	**Als developer wil ik dat TCP-verkeer geanalyseerd kan worden, zodat ik hier conclusies uit kan trekken.** – Loggen van TCP-verkeer-analyse voor de client en server. – Analyse-gegevens verwerken en opslaan in de database van de PIN-server.	60
6	*	ANVP-61	**Verslag schrijven sprint 6**	8
6	*	ANVP-67	**Demo sprint 6**	5
7	*	ANVP-63	**Verslag schrijven sprint 7**	8

Uitvoering

Op afstand blokkeren

Tot nu toe was elke verbinding van de client naar de server geïnitieerd. Voor dit project was het ook van belang dat server-naar-clientverbindingen opgezet kunnen worden, zodat de terminals geblokkeerd kunnen worden door de server. De server-naar-clientverbinding wordt echter standaard geblokkeerd door de ISP.

Wanneer een LTE-M abonnement geactiveerd wordt, staat standaard een NAT+NAPT-firewall geactiveerd. Dit houdt in dat de ISP het IP-adres en port-nummer van het apparaat maskeert met een ander adres en port. Hierdoor worden ingaande-berichten geblokkeerd, als het LTE-M apparaat niet zelf de verbinding initieert.

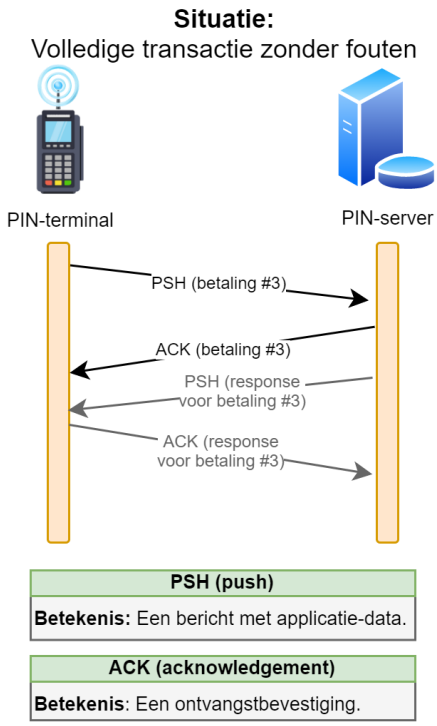
Om deze blokkade uit te schakelen, moest een aanvraag ingediend worden bij de ISP. Vervolgens moest een andere APN geconfigureerd worden op het LTE-M apparaat. Hierna was het mogelijk om op initiatief van de server verbinding te leggen naar de client en commando's te sturen.

Detecteer verbindingskwaliteit met TCP

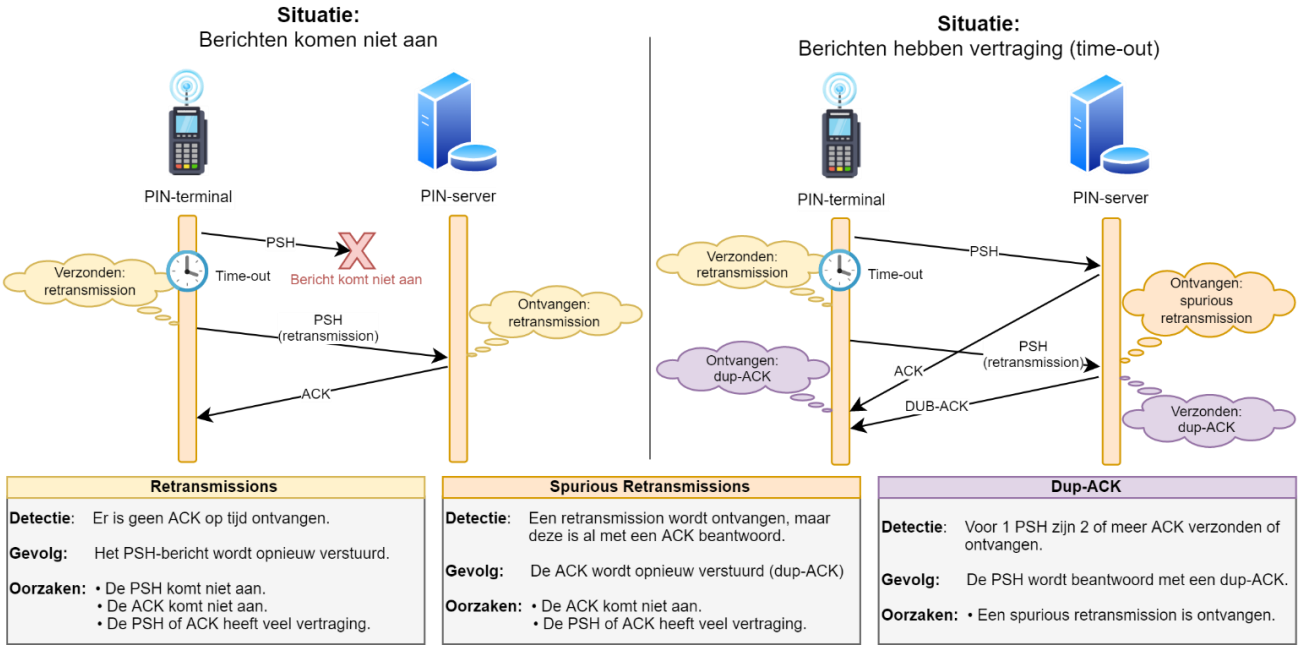
In tegenstelling tot UDP, handelt TCP de stabiele verbinding automatisch af. Hierdoor is het lastig om de kwaliteit van de verbinding te detecteren. Als oplossing wordt het netwerk-sniffprogramma[27] TShark gebruikt. TShark is een terminalvariant van Wireshark. Hiermee worden de volgende berichttypes geteld:

- Transmissions van een stabiele verbinding
  - PSH (push)
  - ACK (acknowledgement)
- Transmissions van een instabiele verbinding
  - Retransmissions
  - Spurious retransmissions
  - Dup-ACK's (Duplicate acknowledgements)

Met deze informatie kan de kwaliteit van het netwerk beoordeeld worden. In figuur 42 is te zien hoe een foutloze transactie verloopt. In figuur 43 wordt het verloop van een instabiele verbinding weergegeven.



figuur 42: Verbinding waarbij alle berichten (op tijd) aankomen. De TCP-handshake en termination zijn weggelaten.



figuur 43: Situaties waarbij transmissiefouten voorkomen.

[27] **Netwerk-sniffer:** Een programma dat alle verzonden en ontvangen internetpackets van een computer kan lezen.

4.4.7. Sprint 8 & 9

In deze twee sprints is het operator-dashboard gerealiseerd. Een uitgebreide uitleg over dit dashboard is te zien in Bijlage J. Met dit dashboard heeft de operator inzicht in:

- de geregistreerde betaalterminals
- de uitgevoerde transacties
- de resultaten van de verbindingstesten

User stories

Het doel van de mock-bankapplicatie is om te beoordelen of LTE-M een geschikt protocol is. Hoewel de testresultaten niet op het dashboard getoond hoeven te worden, heb ik besloten om deze functionaliteit toe te voegen. Dit werd user story ANVP-70. Het dashboard genereert nu automatisch statistieken na het uitvoeren van een verbindingstest, waardoor er inzicht komt in de verbindingskwaliteit en er op een betrouwbare manier conclusies getrokken kunnen worden.

Door dit dashboard is Quintor in staat om zelf automatische verbindingstesten uit te voeren. Het gerealiseerde systeem kan ook gebruikt worden voor het analyseren van andere protocollen. Dit project, waar inmiddels weken aan besteed is, kan nu dus in de toekomst hergebruikt worden.

Doordat het maken van dit dashboard veel tijd kost, zal het twee sprints in beslag nemen.

Sprint	Prior.	User story	Omschrijving	Points
8	M	ANVP-51	**Als evenementondernemer wil ik een betalingen-dashboard, zodat ik kan zien dat PIN-terminals werken.** – Functioneel dashboard maken – Mooi dashboard maken met React	10
8 & 9	M	ANVP-70	**Als gebruiker wil ik dat verbindingsstatistieken in het dashboard getoond worden, zodat ik eenvoudig de verbindingskwaliteit kan beoordelen.**	50
8	*	ANVP-71	**Verslag schrijven sprint 8**	8
8	*	ANVP-73	**Demo sprint 8**	5
9	*	ANVP-72	**Verslag schrijven sprint 9**	8

Uitvoering

Front-end keuze

Volgens de omschrijving van de afstudeeropdracht, moest het dashboard gemaakt worden met een JavaScript front-end. Welk framework hiervoor gebruikt moest worden, mocht ik zelf bepalen. Omdat ik hier nog geen ervaring mee had, heb ik eerst onderzoek gedaan naar de bekendste JavaScript frameworks:

- Angular (sinds 2010)
- React (sinds 2013)
- Vue.js (sinds 2014)

Van deze 3 frameworks wordt Vue.js veel minder gebruikt, waardoor het nuttiger is om een van de andere twee frameworks te leren gebruiken. Hoewel Angular een paar jaar ouder is dan React, worden beide frameworks ongeveer even vaak gebruikt. Meerdere sites stellen dat de learning-curve van React in het begin een stuk makkelijker is. Daarom heb ik voor React gekozen voor het realiseren van het dashboard.

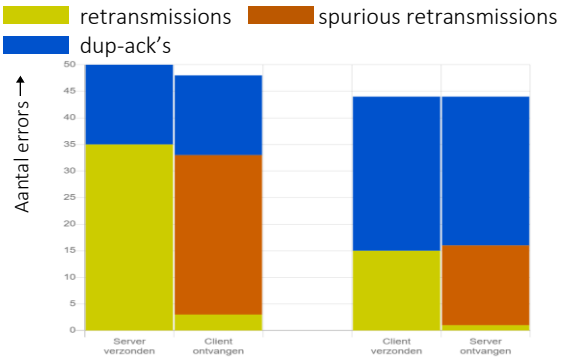
Afwegingen voor dashboard

Voor het dashboard zijn een aantal afwegingen gemaakt. Hieronder worden alleen de belangrijkste keuzes benoemd. Een volledig overzicht van het dashboard is te zien in Bijlage J.

Een project kan altijd groter en beter worden. Echter, ik heb besloten dat de gerealiseerde statistieken voldoende zijn om goede conclusies te trekken over het LTE-M protocol. Omdat dit project afgerond moest worden, heb ik besloten dat dit de laatste uitbreiding van het project was.

Verhouding retransmissions, spurious retransmissions, duplicate ack's

Deze te loggen TCP-errors zijn beschreven in de vorige sprint. Door de verzonden en ontvangen errors direct naast elkaar te zetten, valt snel op wat er fout is gegaan. Deze weergave laat vooral zien in welke richting de verbinding instabiel is. In figuur 44 is te zien dat bij ingaande LTE-M data (downlinks) 2 keer zoveel time-outs voorkomen als bij uitgaande data.

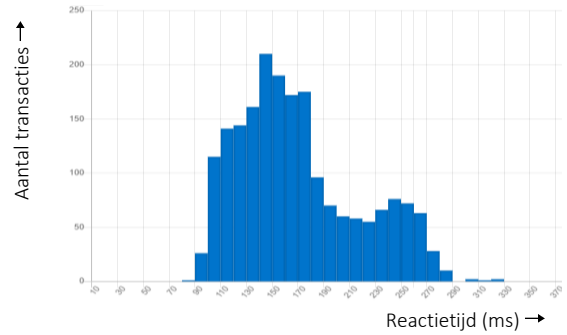


figuur 44: Gestapelde grafiek van 3 soorten TCP-errors. Links: Server → Client. Rechts: Client → Server.

Reactietijd

Mijn oorspronkelijke idee was om per test de gemiddelde reactietijd en de mediaan ervan weer te geven. Dit zal echter een verkeerd beeld geven, omdat de reactietijd van LTE-M verbindingen erg verschilt. Daarom heb ik hiervoor een histogram gemaakt, zie figuur 45. Deze grafiek geeft inzicht in de volgende zaken:

- de snelste reactietijd
- de langste reactietijd
- de meest voorkomende reactietijd

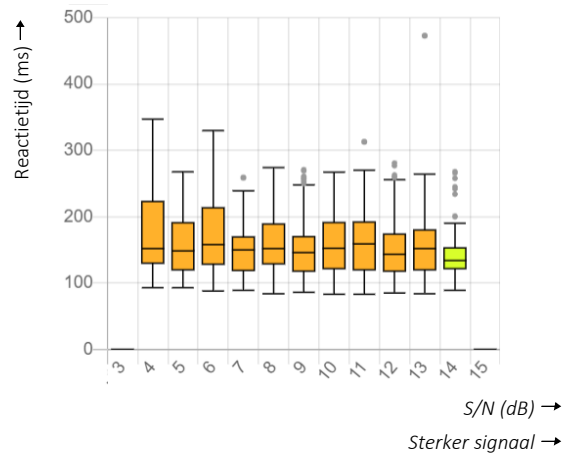


figuur 45: Reactietijd van een LTE-M test.

Invloed van signaalsterkte op de reactietijd

Met boxplotgrafieken wordt de invloed van de signaalsterktes (RSRP, RSSI, RSRQ en SINR) op de reactietijd inzichtelijk gemaakt, zie S/N (dB) → Sterker signaal → figuur 46. Deze grafieken zijn belangrijk, om de invloed van de omgeving te analyseren. Met deze grafiek kunnen de grenzen van het protocol onderzocht worden.

Tijdens het uitvoeren van een verbindingstest veranderen de signaalsterkte en de reactietijd continu. Staafdiagrammen kunnen deze informatie niet goed weergeven. Daarom heb ik besloten om boxplotdiagrammen te gebruiken. Hiermee kan per signaalsterke-stap zowel de mediaan, als de maximale en minimale reactietijd afgelezen worden.



figuur 46: Invloed van de SINR op de reactietijd. De puntjes zijn representeren de 'uitschieters'.

Testscore

Het is belangrijk dat de testuitvoerder snel en gemakkelijk de kwaliteit van het systeem kan aflezen. Daarom krijgt elke test een score tussen A en E. Het uiterlijk van de testscore en andere onderdelen is gebaseerd op het uiterlijk van SonarQube. Veel Quintor-medewerkers werken met SonarQube, waardoor dit dashboard vertrouwd aanvoelt.

Zoals te zien in tabel 13, hangt de score af van het aantal transacties zonder retransmissions. Dit bleek niet een goede maatstaf te zijn, omdat het aantal retransmissions niet bepalend is voor de verbindingskwaliteit. Het zou beter zijn om de onderstaande maatstaf te gebruiken:

- ✅ Alle transacties zijn succesvol afgehandeld.
- ❌ 1 of meerdere transacties waren onsuccesvol.

tabel 13: Score voor verbindingskwaliteit.

Score	Betekenis	Aantal transacties zonder retransmissions
A	Excellent	> 99,9 %
B	Goed	> 99,5 %
C	OK	> 99 %
D	Slecht	> 98 %
E	Ongeschikt	≤ 98 %

4.4.8. Sprint 10 - Resultaten

Beroepstaken: D-15

Dit is de laatste sprint van het project. De verbindingstest wordt onder verschillende omstandigheden uitgevoerd. De conclusies die in dit hoofdstuk getrokken worden, zijn gebaseerd op de testresultaten die te zien zijn in Bijlage K. In dit hoofdstuk wordt het testplan uit het Plan van Aanpak (Bijlage E) behandeld.

User stories

Prior.	User story	Omschrijving	Points
M	ANVP-40	**Performance testen: De impact van de signaalsterkte op de verbinding onderzoeken, zodat daar in volgende projecten rekening mee kan worden gehouden** – Test de round-trip time op verschillende locaties (met pings en betaalverkeer) – Testuitkomst analyseren – Continu zenden en meten hoeveel berichten er per seconde verwerkt kunnen worden met 1 LTE-M apparaat.	32
*	ANVP-74	**Verslag schrijven sprint 10**	8

[1] Encryptie: Gezien het gaat om betaalverkeer, moet de verbinding goed beveiligd zijn (vertrouwelijkheid)
Deze use case gaat over encryptie op de data-linklaag (laag 2) en op de applicatielaag (laag 7).

Laag 2: LTE-M is een variant van LTE (4G). Er is onderzoek gedaan naar de beveiliging van 4G-verbindingen. Deze verbindingen zijn versleuteld. Op het moment van schrijven is er nog geen manier bekend waarmee deze data onderschept en ontsleuteld kan worden. Daarom concludeer ik dat dit protocol veilig is.

Laag 7: Voor echte bankapplicaties is het noodzakelijk om data uit de applicatielaag te versleutelen (user story ANVP-46). Echter, deze user story helpt niet bij het onderzoeken van de beveiliging van LTE-M. Daarom heb ik deze taak geschrapt, zodat ik me meer kan richten op de andere user stories.

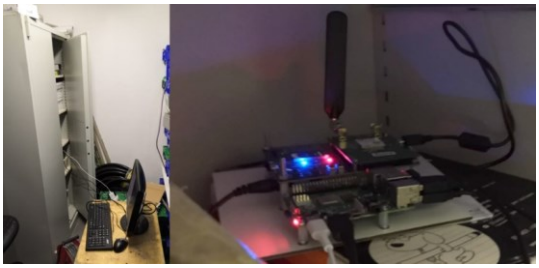
Conclusie: LTE-M is een vertrouwelijk protocol.

[2] Succesvolle datatransmissie: PIN-betalingen moeten gegarandeerd verstuurd en ontvangen worden
Met deze use case wordt onderzocht hoe betrouwbaar het LTE-M netwerk is. In andere woorden: Onderzoek of alle berichten die met dit protocol verstuurd worden, aankomen bij de ontvanger.

Testmethode

Deze use case is onderzocht met de gerealiseerde mock-bankapplicatie. Er worden 2000 transacties van de PIN-terminal naar de PIN-server verstuurd, terwijl in de achtergrond het internetverkeer ‘gesnift’ wordt. Deze verbindingstest is uitgevoerd op locaties met verschillende omstandigheden:

- Op kantoor (weinig obstakels)
- In de opslagruimte (veel obstakels)
- In een kluis (zeer beperkte verbinding), zie figuur 47



figuur 47: Verbindingstest waarbij de PIN-terminal zich in een kluis bevindt. Dit had invloed op de verbinding.

Uit deze verbindingstesten kwamen de volgende resultaten:

- Alle berichten kwamen aan.
- Het ontvangen signaal was duidelijk sterker op kantoor dan in de kluis.
- De testomgeving en de signaalsterkte hadden geen invloed op de reactietijden.
- De test in de kantooromgeving had de meeste retransmissions.

Conclusie

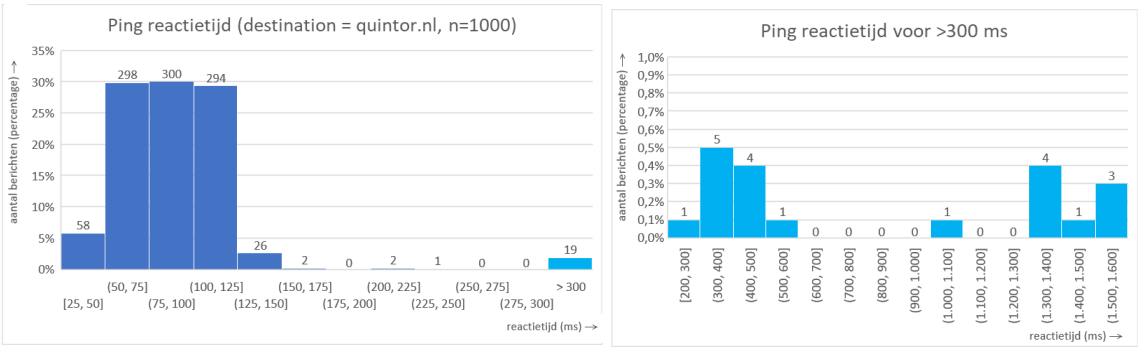
Ik had verwacht dat de verbinding in de opslagruimte en in de kluis instabiel zou worden. Echter, dat bleek geen effect te hebben op de reactietijden en het aantal retransmissions. Hieruit concludeer ik dat LTE-M een zeer stabiel protocol is. Zelfs bij de zwaarste test was het ontvangen signaal sterk genoeg om alle transacties foutloos uit te voeren.

[3] Snelheid round-trip message: Er moeten (veel simultane) PIN-betalingen plaats kunnen vinden zonder storende vertraging

Het doel van deze use case is onderzoeken hoe snel transacties uitgevoerd kunnen worden. Deze use case is onderzocht met twee methoden:

Methode 1: Pingtest

Met deze test zijn 1000 ICMP-pings verstuurd met een LTE-M modem. In figuur 48 zijn de round-trip tijden van deze test te zien. 90% van de pings had een reactietijd van 50-125 ms.

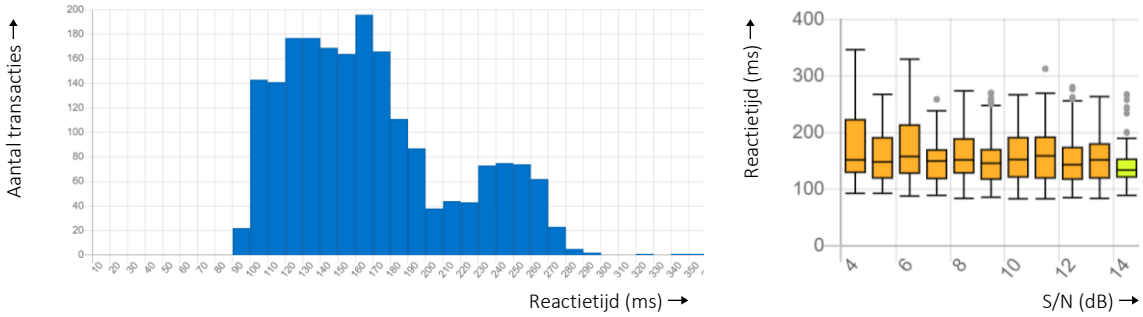


figuur 48: De resultaten van een pingtest met de server van quintor.nl. Meer informatie: Bijlage K – Hoofdstuk 1.

Methode 2: Mock-bankapplicatie

De mock-bankapplicatie houdt bij hoelang elke transactie duurt. De reactietijden zijn te zien in figuur 49. Hieruit volgen de volgende bevindingen:

- De reactietijd is hoger dan bij methode 1. Dat komt doordat een ICMP-ping sneller te verwerken is dan een transactie.
- Het histogram heeft 2 bergen. Dit komt doordat er meermaals een nieuwe TCP-verbinding moet worden opgezet, wat extra tijd kost.
- De signaalsterktes (RSRP, RSRQ, RSSI en S/N) hebben geen invloed op de reactietijd.
- Bijna alle 6000 transacties verliepen snel (3 testen van 2000 transacties). Hoewel er bij ICMP (methode 1) veel uitschieters waren, is dat niet het geval bij TCP (deze methode). Dit komt door de retransmissions.
    - 5995 transacties zijn binnen 500 ms verwerkt.
    - 4 transacties lagen tussen de 500 en 1000 ms.
    - 1 transactie duurde langer dan 1 seconde, namelijk 5239 ms.



figuur 49: Reactietijden van 2000 mock-transacties. [Test 3.1 uit Bijlage K]
Links: Histogram die het aantal transacties met een bepaalde reactietijd weergeeft.
Rechts: Boxplotdiagram waarin de signaalwaarde voor verschillende signaalkwaliteiten wordt weergegeven.

Conclusie

Volgens eis S4 moesten transacties in minder dan 1000 ms verwerkt kunnen worden. Slechts bij 1 transactie (0,017%) duurde het langer. Hieruit concludeer ik dat het LTE-M protocol snel genoeg is. Daarnaast is ondervonden dat de omgeving en signaalsterkte geen invloed hebben op de reactietijd.

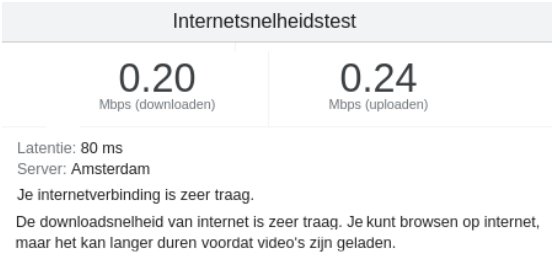
[4] Capaciteit: Omgaan met een high load (beschikbaarheid)

Het doel van deze use case is om te achterhalen hoeveel PIN-transacties tegelijk uitgevoerd kunnen worden bij gebruik van LTE-M. Omdat er maar 1 LTE-M apparaat ter beschikking is, wordt onderzocht hoeveel transacties 1 apparaat per seconde kan uitvoeren. Daarnaast is onderzocht wat er gebeurt als er meer transacties worden uitgevoerd dan mogelijk is volgens de LTE-M specificatie.

Limitaties

De snelheidslimitatie van LTE-M is hieronder te zien. Dit komt overeen met de snelheidstest in figuur 50.

	Up	Down
LTE-M netwerk van KPN	300 kbit/s	200 kbit/s
SIM7000E (gebruikte modem)	375 kbit/s	300 kbit/s



figuur 50: LTE-M internetsnelheid met een tool van M-Lab.

Testmethode: pingflooding

Pingflooding betekent dat de Raspberry Pi continu ICMP-pingrequests verstuurt naar een server, zonder eerst te wachten op een reply. De gerealiseerde mock-bankapplicatie verstuurt ongeveer 850 bytes per transactie. Voor extra speling zijn pings van 1000 bytes verstuurd bij de floodingtest. Daarnaast is op zowel de Pi als de server gesnift, zodat uplinks en downlinks los van elkaar onderzocht kunnen worden.

Resultaat

De resultaten van deze test waren erg informatief: 75% van de pings die verstuurd worden (uplinks), komen aan bij de server. Daarentegen komen *alle* ping-replies (downlinks) succesvol aan bij de Pi. Dit had ik niet verwacht, omdat volgens de specificaties de uploadsnelheid hoger is dan de downloadsnelheid. Gemiddeld kunnen er per seconde 17 pings van 1000 bytes succesvol verstuurd worden, wat neerkomt op een uploadsnelheid van 136kbit/s. Uitgebreide resultaten zijn te zien in Bijlage K – hoofdstuk 2.

Conclusie

Volgens eis M5 zouden bij een festival de aanwezige PIN-terminals gezamenlijk 50 transacties per seconde moeten kunnen uitvoeren. Deze test is uitgevoerd met 1 LTE-M apparaat. Om deze eis met 100% zekerheid te toetsen, zou deze test met 3 LTE-M apparaten uitgevoerd moeten worden. Echter, er kan veilig worden aangenomen dat meerdere PIN-terminals gezamenlijk aan de gestelde eis van 50 transacties per seconde voldoen. LTE-M apparaten zijn namelijk met hetzelfde snelle netwerk verbonden als alle andere 4G-apparaten, zoals beschreven in hoofdstuk 4.3.3 – LTE-M.

[B1] Commando's sturen: Een server moet berichten kunnen sturen naar de client

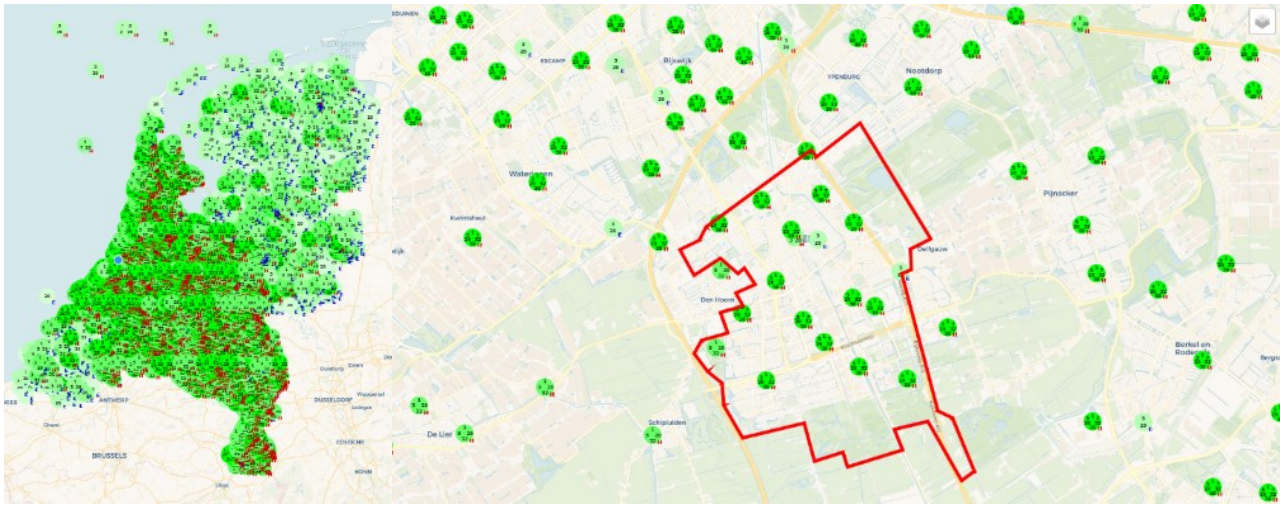
Deze use case is beantwoord in sprint 6. Voor consistentie met het testplan, wordt B1 hier nog kort toegelicht.

Het is gelukt om een verbinding te initiëren van de PIN-server naar een PIN-terminal, als het IP-adres bekend is. Dit is standaard niet mogelijk, omdat LTE-apparaten beveiligd zijn met een NAT-firewall van de ISP. Deze firewall kan op aanvraag van de klant opgeheven worden.

[B2] Redundantie: Als de verbinding tussen 2 nodes uitvalt, moet een andere route van eind-node naar gateway genomen kunnen worden

Bij het LTE-M protocol vallen de basestations (zendmasten) buiten het beheer van de gebruiker. Hierdoor is het niet mogelijk om te testen wat er gebeurt als een zendmast uitvalt. Ik verwacht dat er geen rekening gehouden hoeft te worden met het uitvallen van een zendmast om de volgende redenen:

- Zoals te zien is in figuur 51 zijn er meerdere zendmasten waarmee verbonden kan worden. Het signaal van een 'buurcel' is minder sterk. Echter, een LTE-M apparaat kan goed omgaan met zwakke signalen.
- De kans dat een zendmast (tijdelijk) uitvalt acht ik zeer klein, omdat deze door een ISP beheerd wordt.



figuur 51: Alle LTE-M zendmasten van één ISP in Nederland (links) en Delft (rechts). Elke groene bol representeert een 4G-zendmast van het KPN-netwerk met band 20 (800 MHz) als carrier. Dit zijn de zendmasten waarmee de gebruikte modem kan verbinden.

Eindoordeel

Alle testen uit het testplan zijn uitgevoerd. Uit bovenstaande resultaten concludeer ik dat LTE-M voldoet aan alle gestelde eisen en dat het ingezet kan worden voor PIN-betalingen bij festivals.

4.5. Fase 5: Verslaglegging

Om het gerealiseerde project goed over te dragen aan Quintor, is in deze fase de nog ontbrekende documentatie gerealiseerd.

Bijlage H is een handleiding die beschrijft hoe een LTE-M modem geconfigureerd kan worden. Hierdoor kan het LTE-M apparaat ingezet worden voor een bankapplicatie of voor andere toepassingen. Bijlage J is een handleiding waarin uitgelegd wordt hoe het dashboard gebruikt kan worden bij het analyseren van de verbindingstesten. In Bijlage K staan de resultaten van verschillende tests. De data die hierin staat is nuttig voor het beoordelen van het protocol.

5. Conclusie

De hoofdvraag van de afstudeeropdracht is: “Welke communicatietechniek is inzetbaar voor PIN-betalingen wanneer het mobiele netwerk overbelast is?” Om deze vraag te beantwoorden, is na uitgebreid onderzoek het beste protocol geselecteerd en aan een Proof of Concept onderworpen.

De protocollen die uit het onderzoek als beste naar voren kwamen, zijn Wi-Fi HaLow en LTE-M. Om financiële redenen is voor LTE-M gekozen. Dit protocol biedt veel voordelen: Allereerst biedt het in de meeste gevallen een plug-and-play-ervaring, waardoor het gemakkelijk te integreren is. Daarnaast wordt de netwerkinfrastructuur in binnen- en buitenland beheerd door internetproviders, waardoor het erg betrouwbaar is. Ook voldoet het aan alle gestelde criteria uit het Plan van Aanpak. Tot slot maakt ‘application priority’ het mogelijk om een betrouwbare verbinding te behouden. Wanneer het mobiele netwerk bij drukke festivals overbelast is, blijven PIN-betalingen en andere kritieke processen foutloos doorgaan.

Een Proof of Concept in de vorm van een mock-bankapplicatie is gerealiseerd om LTE-M te testen. Zelfs onder zware omstandigheden opereert dit protocol foutloos. De huidige maximale snelheid in Nederland is 300 kbit/s, waardoor het snel genoeg is voor IoT-toepassingen.

Met de geschreven software en handleidingen beschikt Quintor over de kennis om LTE-M toe te passen bij PIN-systemen en andere uitdagingen. Ook kan zij de mock-bankapplicatie gebruiken om andere protocollen mee te testen. Mocht later een alternatief protocol vereist zijn, dan zal de lijst met gevonden informatie over de andere draadloze communicatieprotocollen goed van pas komen.

Uit dit onderzoek is gebleken dat LTE-M een zeer geschikt protocol is. Naast PIN-systemen kan het ook voor andere toepassingen ingezet worden. ‘Internet of Things’ wordt een steeds belangrijker begrip. Dankzij deze afstudeeropdracht is Quintor er klaar voor om uitdagingen uit dit nieuwe veld op te pakken.

6. Evaluatie en reflectie

In dit hoofdstuk reflecteer ik op mijn keuzes van hoofdstuk 4 – Werkzaamheden. Ik beoordeel wat er goed en fout ging en de keuzes die ik gemaakt heb tijdens mijn afstudeerstage.

6.1. Fase 1: Oriënterend onderzoek rondom afstudeeropdracht

Beroepstaken: A-1, G-c

Toen ik aan dit project begon, waren er veel onduidelijkheden. Hoewel ik 2 weken had gepland voor deze fase, had ik niet verwacht dat ik ruim 2 weken nodig zou hebben. Ik heb erg onderschat hoeveel protocollen er zijn en hoe moeilijk het kan zijn om specifieke specificaties te onderzoeken.

Verschil tussen draadloze protocollen

Na afloop van dit project heb ik 42 communicatieprotocollen gevonden die potentieel geschikt zijn. Tijdens het onderzoeken van deze protocollen in fase 3, kwamen er nieuwe protocollen bij. Hierdoor is die fase veel langer geworden dan verwacht. Ik zie nu in dat ik had moeten stoppen met het opnemen van nieuwe protocollen in mijn onderzoek, zodat ik meer tijd over zou houden voor fase 4: de realisatiefase.

In deze fase kwam ik veel protocollen tegen die superseded of withdrawn zijn. Hoewel dit niet hoeft te betekenen dat ze ongeschikt zijn, heb ik ze toch afgestreept. Dit deed ik om te voorkomen dat de lijst steeds langer zou worden. Ook ben ik in de veronderstelling dat ingetrokken protocollen geen goede keuze zijn om toekomstige projecten mee te realiseren.

Co-existence

Het is bekend dat de internetverbinding bij festivals instabiel is. De reden daarvoor was echter onbekend. Ik vroeg me af of het maximale aantal gebruikers voor het mobiele netwerk dan bereikt is, óf dat er zoveel apparaten aanwezig zijn, dat de communicatie van elk protocol verstoord wordt. Wanneer de tweede stelling het geval zou zijn, had dit voor veel problemen kunnen zorgen. Het was lastig om hier een direct antwoord op te vinden.

Ik heb een white paper gevonden [7] die inging op een soortgelijke kwestie. De onderzoekers van deze white paper hebben onderzoek verricht naar het effect van Wi-Fi-stations die op korte afstand van elkaar aan het zenden zijn. Door de informatie uit deze white paper verwachtte ik dat mobiele netwerken niet zullen storen op andere netwerken. Aangezien het onderzoek ging over verstoringen bij Wi-Fi, en *niet* over mobiele netwerken, gaf dit geen compleet antwoord op de vraag. De white paper was wel erg leerzaam voor de rest van dit project.

Later vond ik het antwoord op de site van MSC. Een private zendmast kan opereren wanneer het normale netwerk overbelast is. Daarom zal de limitatie bij de zendmast of het protocol zitten; en komt de storing niet doordat meerdere telefoons elkaars verbinding verstoren. Dit betekent dat verschillende communicatieprotocollen naast elkaar gebruikt kunnen worden, zolang ze niet dezelfde frequentie gebruiken. In fase 3 heb ik voor de zekerheid deze vraag voorgelegd bij KPN, die mijn conclusie bevestigde.

Overige relevante ontdekkingen

Ik heb een begin gemaakt met het onderzoeken van welke technologie bij elke telefoniegeneratie hoorde; voor de volledigheid van mijn onderzoek. Hier ben ik later mee gestopt, omdat bronnen elkaar hierover tegenspraken. Daarnaast leverde het niet veel relevante kennis op.

Tijdens deze fase heb ik een gesprek gehad met mijn afstudeercoach. De onderzoeksvraag is: “Welke communicatietechniek is inzetbaar voor PIN-betalingen wanneer het mobiele netwerk overbelast is?”. Later heb ik met mijn coach afgesproken dat het te onderzoeken protocol niet het beste protocol voor deze casus hoeft te zijn. Als er een protocol gevonden wordt waarmee meer use cases van Quintor uitgevoerd kunnen worden, dan zal dát protocol geselecteerd worden. Hierdoor wordt het eindresultaat van mijn opdracht meer waardevol voor Quintor.

6.2. Fase 2: Definitiefase

Situatie analyseren

Deze afstudeeropdracht sluit niet aan op een bestaand project, waardoor er van tevoren geen randvoorwaarden waren. Hierdoor heb ik zelf moeten uitzoeken welk resultaat de opdrachtgever aan het eind van de opdracht wilde hebben. Deze eisen zijn natuurlijk wel overlegd met de bedrijfsmentor. Hoewel ik voor dit project alle eisen zelf moest bedenken, zou ik bij een niet-verzonnen project meer met de opdrachtgever overleggen om zijn wensen te onderzoeken.

Eisen opstellen

Om de payload van eis M1 te bepalen, heb ik de communicatie van een online geldoverboeking gemonitord. Daarna heb ik bepaald welke eigenschappen van deze data nodig zijn bij het realiseren van een betaalsysteem op een festival. Het is mogelijk dat bij een echte PIN-transactie meer variabelen worden meegestuurd dan ik geconcludeerd heb, waardoor de payload nog groter wordt dan beschreven in eis M1. Om dit probleem te verhelpen, zouden berichten in meerdere delen gesplitst kunnen worden. Het zou echter handiger zijn om voortaan meer rekening mee te houden met extra data.

Bij eis M3 heb ik beschreven dat de hardware voor het te kiezen protocol compatible moest zijn met een Raspberry Pi, omdat het Proof of Concept met Pi's uitgevoerd wordt. Echter, het is niet vereist dat de modules specifiek voor een Raspberry Pi gemaakt moeten zijn (denk hierbij aan Raspberry Pi-'hats'). De hardware hoeft alleen compatible te zijn, maar dat zou ook met standaard (USB-)drivers kunnen. Deze eis zou anders geïnterpreteerd kunnen worden, door de manier van schrijven in het Plan van Aanpak.

Eisen M6 en S1 beschrijven het vereiste zendbereik van de protocollen. Voor deze eisen heb ik 2 terreinen vergeleken: de Zwarte Cross (160 ha), wat ongeveer de grootte is van mijn woonwijk en Duinrell (48 ha), een vakantiepark. Achteraf had ik beter een kleiner festivalpark kunnen gebruiken als tweede voorbeeld, aangezien de opdracht over festivals en evenementen gaat. De reden dat ik Duinrell als referentie heb gekozen, is omdat ik erg bekend ben met dit park. De oppervlakte van dit terrein was voor mij meer verhelderend dan dat van elk ander festivalterrein. Hiervoor waren hectaren een abstracte eenheid voor mij.

In het pakket van eisen komen geen won't do's voor. Als ik die wel gespecificeerd had, was duidelijk welke taken er niet uitgevoerd zouden worden tijdens dit project. Dat geeft meer duidelijkheid over de scope van het project. Bij toekomstige projecten zal ik hierop moeten letten. De volgende won't do's hadden opgenomen kunnen worden in het Plan van Aanpak:

- Er zal geen PIN-systeem geïmplementeerd of nagemaakt worden. De mock-bankapplicatie zal alleen de verbinding testen. De logica achter PIN-transacties is niet belangrijk.
- Er zal geen (private) network worden getest waarvoor zendmasten opgezet moeten worden.

Beheersing

De feedback uit deze paragraaf hangt samen met fase 3. In het afstudeerplan (Bijlage D, hoofdstuk 3.1) heb ik de volgende stappen beschreven:

1. Onderzoek doen naar alle draadloze communicatietechnieken. In deze stap worden de technieken niet diep, maar oppervlakkig onderzocht. (fase 3)
2. Een selectie maken van communicatietechnieken die de oplossing zouden kunnen bieden voor dit probleem. (fase 3)
3. Verdiepen in de specificaties van de geselecteerde communicatietechnieken. De meest geschikte techniek wordt verder geselecteerd voor stap 4. Indien er 2 technieken even goed lijken te werken, kunnen beide technieken vergeleken worden. (fase 3)
4. Een Proof of Concept maken die aantoont of het mogelijk is om deze techniek te gebruiken voor PIN-betalingen. (fase 4)

Ik wilde ervoor zorgen dat ik niet te veel tijd kwijt zou zijn met fase 3. Door stappen 1 en 2 zou ik protocollen afstrepen, voordat ik onderzocht had of ze aan de eisen voldeden of niet. Hierdoor zou er een kleinere selectie overblijven om uitgebreid te onderzoeken. Toen ik eenmaal begonnen was met het uitgebreide onderzoek, was ik deze afspraak vergeten. Ik had ook geen deadlines in de planning gezet. Ik heb de eisen van bijna elk protocol onderzocht, omdat ik bang was dat ik het meest-geschikte protocol over het hoofd zou zien. Als ik mezelf in fase 3 aan het stappenplan had gehouden, dan was ik weken eerder klaar geweest en had ik meer tijd over voor fase 4. Dat zou tot meer informatie hebben geleid voor Quintor.

Om deze uitloop te voorkomen, had ik meerdere dingen moeten doen. Ten eerste had ik harde deadlines in mijn planning moeten zetten. Genoeg tijd hebben voor het Proof of Concept is belangrijker dan een onderzoek dat 100% compleet is. Ten tweede moet ik niet alle mogelijke opties willen onderzoeken. In de toekomst zal ik vaker voor keuzes staan met veel opties. Ik moet me niet verdiepen in alle mogelijkheden. Dit is uiteindelijk mijn valkuil geworden. Ten derde kan ik in de risicoanalyse al beschrijven hoe ik verder te werk moet gaan als taken te lang duren.

Doordat ik te veel tijd heb besteed aan fase 3, heb ik mijn afstudeerperiode verlengd. Dit heb ik gedaan, om meer tijd te kunnen besteden aan fase 4 en om alsnog een mooi eindproduct op te leveren. Ik ga hard aan dit verbeterpunt werken, zodat ik in de toekomst niet dezelfde fout maak.

6.3. Fase 3: Uitgebreid onderzoek

Beroepstaken: A-2

Zoveel mogelijk protocollen zoeken - IEEE-standaarden

Een van de valkuilen van deze fase was de verwarrende versienummering van de IEEE-standaarden. Er waren voor mij twee dingen erg onduidelijk:

Ten eerste kunnen protocollen behoren tot de 'basis-standaard' van een andere scope. IEEE 802.11ah hoort bijvoorbeeld eerder bij de WRAN spatial scope dan WLAN, in tegenstelling tot de andere IEEE 802.11-protocollen. Achteraf besef ik dat deze protocollen een uitbreiding zijn van de basis-standaard en nauw samenhangen met andere uitbreidingen daarvan.

Ten tweede was het erg verwarrend dat de IEEE-website bij veel 'superseded'-protocollen niet aangeeft door welke standaard het vervangen is. Een voorbeeld hiervan was IEEE 802.15.4g-2012 (Wi-SUN). Deze standaard heeft de status 'superseded' en is opgevolgd door IEEE 802.15.4z-2020. Doordat er geen IEEE 802.15.4g-standaard was met een hoger jaartal, nam ik aan dat deze uitbreiding niet meer gebruikt werd. Daardoor heb ik het Wi-SUN-protocol niet verder onderzocht. Na het indienen van het hardwarevoorstel kwam ik erachter dat ik de verkeerde aanname gemaakt had.

Protocollen testen op requirements

Protocollen onderzoeken

Het onderzoeken van de protocollen duurde erg lang. Dat had meerdere oorzaken.

Ten eerste waren veel specifieke eisen erg lastig te vinden. Een voorbeeld daarvan was het uitzoeken van de praktische communicatieafstanden van ZigBee en Z-Wave mesh-netwerken. Uiteindelijk heb ik een onderzoek gevonden waarin deze afstanden getoetst werden en aanbevelingen werden gedaan. Het leek erop dat de slechtere specificaties van een protocol opzettelijk niet, of vaag gedeeld worden door de bedrijven die ze ontwikkelen.

Ten tweede had ik sneller contact moeten leggen met deskundigen. Wanneer ik iets uit wilde zoeken, bleef ik er vaak te lang mee bezig. Tijdens dit project heb contact opgenomen met de Weightless Alliance, KPN en Nordic. Hoewel Weightless nooit heeft geantwoord, hebben KPN en Nordic me snel van nuttige informatie voorzien.

Selectie maken vóór het uitgebreide onderzoek

In de evaluatie van hoofdstuk 6.2 was ik in de veronderstelling dat ik van tevoren een lijst met protocollen had moeten selecteren om uitgebreid te onderzoeken. Nu ik fase 3 heb afgerond, ben ik blij dat ik dat niet gedaan heb. In dat geval zou ik, zonder de kennis van nu, een selectie maken van verkeerde protocollen:

- Wi-Fi HaLow en Bluetooth zou ik afgestreept hebben, met de aanname dat deze protocollen een kleine communicatieafstand hebben.
- LTE-M zou ik hebben afgestreept, omdat ik niet bekend was met 'application priority'.

Als ik de selectie met verkeerde protocollen had onderzocht, was ik later alsnog tot de conclusie gekomen dat deze ongeschikt waren. Waarna alsnog een nieuwe selectie protocollen gemaakt had moeten worden.

Rond het eind van deze fase heb ik besloten om WiMAX, White-Fi en Ingenu RPMA niet te onderzoeken, hoewel deze drie ook goede opties hadden kunnen zijn.

Selectie protocol volledig onderzoeken

SDK uitzoeken

Voor het hardwarevoorstel (Bijlage F) heb ik een aantal evaluation kits uitgezocht. Quintor zou bepalen of ze tevreden zijn met het voorgestelde protocol en of de voorgestelde hardware aangeschaft kon worden. Zij konden nog niets zeggen over het budget voor de afstudeeropdracht tot de opties duidelijk waren.

Ik heb volledig uitgezocht hoe de Wi-Fi HaLow kit geïmplementeerd moest worden. Quintor vond deze echter te duur was. Hierdoor moest ik een van de andere opties kiezen. Achteraf had ik het anders kunnen doen: Ik zou eerst het hardware voorstel doen en wachten op het oordeel van Quintor. Na goedkeuring van het budget zou ik pas verder gaan met het grondig onderzoeken van de evaluation kits.

Bluetooth specificatie

Bluetooth is een protocol met enorm veel opties. De ontwikkelaar mag zelf bepalen welke opties hij ondersteunt. Daardoor is het een zeer complex protocol en was het een uitdaging om te onderzoeken. Een ander probleem was dat ik nergens op internet informatie kon vinden over het combineren van coded PHY en DLE. Gelukkig heeft Nordic mijn vragen kunnen beantwoorden. Van alle protocollen was Bluetooth het meest complexe protocol om te onderzoeken.

Beheersing

In fase 4 is Scrum gebruikt voor de projectbeheersing. Dit had achteraf ook een goede manier geweest om deze fase mee uit te voeren. Met deze manier van werken wordt de planning nauwlettend in de gaten gehouden en is duidelijk welke taken belangrijk zijn.

6.4. Fase 4: Realisatiefase

Sprint 1 Retrospective

Beroepstaken: C-10

Wat ging goed	Wat had beter gedaan kunnen worden?
• De sprint verliep volgens plan. • Ik heb veel geleerd over Spring.	• Thuis werken is lastiger dan op locatie, maar helaas was dat niet mogelijk.

Acties

- In dit verslag worden 5 pagina's besteed aan deze sprint. Dit moet bij de volgende sprints minder worden. Voor deze sprint is het niet erg, omdat er veel belangrijke informatie beschreven moest worden.
- De beroepstaken moeten bij de benodigde user stories beschreven worden, zodat ik weet op welke taken ik me moet focussen.

Sprint 2

Beroepstaken: D-14, D-15

Wat ging goed	Wat had beter gedaan kunnen worden?
• Deze week was erg productief. Dit kwam onder anderen door de goede voorbereiding uit sprint 1. • De statische code-analyse met SonarQube stond voor deze fase niet in de planning. Daarom heb ik besloten dit alleen te implementeren, als het niet te veel tijd zou vergen. Dit is gelukt en het heeft de kwaliteit van mijn code verbeterd.	• Deze week ging goed en ik kon geen verbeterpuntjes bedenken.

Acties

- Doorgaan zoals het nu gaat.

Sprint 3

Beroepstaken: C-9

Wat ging goed	Wat had beter gedaan kunnen worden?
• Ondanks de kerstvakantie heb ik ongeveer een volledige werkweek kunnen besteden aan dit project. • Effectief kunnen werken.	• Als ik eerder had overgestapt naar het testen met Ubuntu, dan was ik er eerder achter gekomen hoe de modem geconfigureerd moest worden.

Acties

- Doorgaan zoals het nu gaat.

Sprint 4

Wat ging goed	Wat had beter gedaan kunnen worden?
• Effectief gewerkt. • Duidelijke handleiding geschreven.	• Alles ging goed

Acties

- Doorgaan zoals het nu gaat.

Sprint 5

Wat ging goed	Wat had beter gedaan kunnen worden?
• Er waren weinig problemen. • Ik heb veel kennis opgedaan over het analyseren van de betrouwbaarheid van LTE-protocollen. Hierdoor is het later mogelijk om conclusies te trekken over de kwaliteit van het protocol.	• Ik had de complexiteit van LTE-signaalsterktes onderschat.

Acties

- Doorgaan zoals het nu gaat. Iets minder letten op de details. Ik hoef me niet in alles te verdiepen.

Sprint 6

Beroepstaken: D-14

Wat ging goed	Wat had beter gedaan kunnen worden?
• Oplossingsgericht de problemen opgepakt. • De backlog-wijziging heeft voor veel extra werk gezorgd. Echter, dit was de beste keuze.	• Eerder het plan aanpassen, als blijkt dat het oude plan niet werkt.

Acties

- Een extra sprint inzetten voor het afronden van user story ANVP-69. Maak hier 60 storypoints van.

Sprint 7

Wat ging goed	Wat had beter gedaan kunnen worden?
• Gestructureerd met Wireshark onderzoeken waar fouten zitten en deze oplossen.	• Deze sprint is goed gegaan.

Acties

- Doorgaan zoals het gaat.

Sprint 8

Wat ging goed	Wat had beter gedaan kunnen worden?
• In plaats van het schrijven van eigen componenten, is er eerst onderzoek gedaan naar bruikbare componenten uit bibliotheken. Hierdoor is de kwaliteit van de site een stuk beter geworden.	• Er zijn wat bugs gevonden in de software van de vorige sprint. Die moeten in deze sprint nog opgelost worden. Alle geschreven software moet goed gecontroleerd worden.

Acties

- Kritischer nalopen of de Wireshark-log en de waardes van de mock-bankapplicatie overeenkomen.

Sprint 9

Wat ging goed	Wat had beter gedaan kunnen worden?
• Een project is nooit af en kan altijd nog groter en beter worden. Daarom is het goed dat ik een duidelijk eindpunt gedefinieerd heb. Ik heb een streep achter het project gezet en alleen de belangrijkste dingen geïmplementeerd. • Het programmeren van een front-end is nieuw voor mij. Hiervan heb ik veel geleerd. Ik verwacht dat ik deze kennis voor toekomstige projecten goed kan gebruiken. • Het dashboard is mooi geworden en ik ben trots op het resultaat.	• Deze sprint verliep goed. Niks op aan te merken.

Acties

- Blijf in de gaten houden wat de belangrijkste taken zijn en rond het project af.

Sprint 10

Beroepstaken: D-15

Wat ging goed	Wat had beter gedaan kunnen worden?
• Alle tests uit het testplan zijn uitgevoerd. Hier zijn zeer nuttige resultaten uit gekomen. • Alles is goed afgrond.	• Ik had onderschat hoelang het duurt om de testen uit te voeren. Hier moet meer rekening mee gehouden worden.

Acties

- Doorgaan zoals het gaat. Alleen de documentatie moet nog geschreven worden.

6.5. Evaluatie van gebruikte aanpak

Zoals genoemd is in het afstudeerplan (Bijlage D) en het Plan van Aanpak (Bijlage E) heb ik Scrum gekozen als projectmethode. Omdat dit een éénpersoonsproject is, week de ontwikkelmethode een beetje af van Scrum. Zo was er bijvoorbeeld geen formele daily stand-up.

Wat goed ging, was dat ik per sprint keek naar welke user stories het belangrijkst waren. Wanneer het nodig was, waren user stories van MoSCoW-prioriteit veranderd. Ook waren er nieuwe, belangrijkere stories toegevoegd aan de backlog, als deze niet voorzien waren. Doordat ik werkte met sprints van elk 1 week, kon er snel geanticipeerd worden.

Wat meestal goed ging, was het bijhouden van de grootte van de user stories. Zo is ANVP-40 opgedeeld en ontstonden user stories ANVP-41 en ANVP-42. Daarentegen hadden ANVP-69 en ANVP-70 ook opgesplitst moeten worden, want deze user stories waren respectievelijk 60 en 70 Scrum-punten waard.

Aan elke sprint was een user story toegevoegd voor het bijhouden van dit afstudeerverslag. Hierdoor heb ik elke sprint tijd besteed aan dit verslag, zodat het niet allemaal achteraf geschreven hoefde te worden. Ik ben van plan om documentatie in de toekomst ook op te nemen in de sprint planning.

7. Beroepstaken

Beroepstaken: G-f

In dit hoofdstuk staan de beroepstaken beschreven die ik heb bewezen tijdens deze afstudeeropdracht. De fases hebben betrekking op de planning in Bijlage A - Planning.

ID	Beroepstaak	Hoe aangetoond	Fase/ sprint	Pagina
A-1	Analyseren probleemdomein & opstellen probleemstelling	Het onderzoeken van de kenmerken en andere belangrijke factoren voor deze opdracht.	1	13/68
A-2	Informatie vergaren, analyseren, beoordelen & verwerken	Het onderzoeken van de specificaties van veel protocollen.	3	32/72
C-9	Ontwerpen Technische Infrastructuur	Een testomgeving is opgezet waarin een Raspberry Pi kan communiceren met een server via LTE-M.	4.3	51/74
C-10	Ontwerpen embedded (& Realtime) systemen	Klassendiagrammen, sequentiediagrammen en database-structuren zijn gemodelleerd voor zowel de betaalterminals als de betaalserver.	4.1	43/74
D-14	Realiseren van software	Het programmeren van de betaalterminal en -server in het Spring-framework en het programmeren van het web-dashboard met het React-framework.	4.2, 4.6	48/74 58/75
D-15	Testen	Het maken en uitvoeren van een testplan waarmee de prestaties van het LTE-M netwerk worden geanalyseerd en het schrijven van unit tests.	4.2, 4.10	48/74 62/76
G-c	Kritisch, onderzoekend & methodisch werken	Onderzoek doen naar de eisen waarop het protocol getest wordt.	2	13/68
G-f	Leren leren: Voorbereiden op volgende studiefase en beroep	Het uitvoeren van werkzaamheden uit de beschreven beroepstaken. Quintor heeft haar eigen vrijwillige cursussen en workshops, waar ik zo veel mogelijk gebruik van maak.	-	78

8. Bronnen

[1] F. Everaardt, „Mesh networking: draadloze verbindingsofficieren,” 3 september 2017. [Online]. Available: https://nl.hardware.info/artikel/7534/2/mesh-networking-draadloze-verbindingsofficieren-trucs. [Geopend 10 november 2020].

[2] A. v. Bentem, „Limitations: data rate, packet size, 30 seconds uplink and 10 messages downlink per day Fair Access Policy [guidelines],” The Things Network, 26 juli 2020. [Online]. Available: https://www.thethingsnetwork.org/forum/t/limitations-data-rate-packet-size-30-seconds-uplink-and-10-messages-downlink-per-day-fair-access-policy-guidelines/1300. [Geopend 11 november 2020].

[3] Bluetooth SIG, „Understanding Bluetooth Range (standaard configuratie aangehouden),” Bluetooth, g.d.. [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/range/#estimator. [Geopend 11 november 2020].

[4] Rick S, „Downlink and Uplink limitation - Free Dev Portal,” KPN, 20 juli 2017. [Online]. Available: https://zakelijkforum.kpn.com/lora-forum-16/downlink-and-uplink-limitation-free-dev-portal-9479. [Geopend 11 november 2020].

[5] Sigfox, „Technical Quickstart,” Sigfox, g.d.. [Online]. Available: https://build.sigfox.com/technical-quickstart. [Geopend 11 november 2020].

[6] IEEE Standards Association, „HOW ARE STANDARDS MADE?,” IEEE, g.d.. [Online]. Available: https://standards.ieee.org/develop/develop-standards/process.html. [Geopend 11 november 2020].

[7] F. Paul, V. Danilo en R. Fabio, „The myth of non-overlapping channels: interference measurements in IEEE 802.11,” januari 2007. [Online]. Available: https://www.researchgate.net/publication/202880366_The_myth_of_non-overlapping_channels_interference_measurements_in_IEEE_80211. [Geopend 11 november 2020].

[8] mobilefish.com, „LoRaWAN - LoRa/LoRaWAN tutorial 5: Decibel, dBm, dBi, dBd,” 22 september 2018. [Online]. Available: https://www.mobilefish.com/developer/lorawan/lorawan_quickguide_tutorial.html. [Geopend 11 november 2020].

[9] 3GPP, „Requirements for support of radio resource management - ETSI TS 136 133: sub-clause 9.1.4. RSRP Measurement Report Mapping,” januari 2011. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/136100_136199/136133/10.01.00_60/ts_136133v100100p.pdf. [Geopend 11 november 2020].

[10] whatsag.com/, „What are the 4G Technology Standards,” What's a G?, g.d.. [Online]. Available: https://whatsag.com/4g-and-lte-standards/understanding_4g.php. [Geopend 14 november 2020].

[11] whatsag.com, „A History of the G Mobile Generations,” What's a G, g.d.. [Online]. Available: https://whatsag.com/mobile-technology/generation_history.php. [Geopend 14 november 2020].

[12] Computer Desktop Encyclopedia, „IS-136,” g.d.. [Online]. Available: https://encyclopedia2.thefreedictionary.com/IS-136. [Geopend 11 november 2020].

[13] L. Frenzel, „The Evolution Of LTE,” 8 januari 2013. [Online]. Available: https://www.electronicdesign.com/content/article/21795506/the-evolution-of-lte. [Geopend 11 november 2020].

[14] daftlogic.com, „Google Maps Area Calculator Tool,” daftlogic.com, 11 08 2018. [Online]. Available: https://www.daftlogic.com/projects-google-maps-area-calculator-tool.htm. [Geopend 24 11 2020].

[15] J. M. Ubink, „Plan van Aanpak - Welke communicatietechniek is inzetbaar voor PIN-betalingen wanneer,” Haagse Hogeschool, Leiden, 2020.

[16] supernova, „Architecture à base d'IoT et de Edge Computing,” 9 maart 2020. [Online]. Available: http://www.wikiwai.com/2020/03/09/architecture-a-base-diot-et-de-edge-computing/.

[17] P. Mannion, „Wi-SUN Alliance Adds Long-Range Mesh to Wireless IoT Fray,” Electronics 360, 13 November 2015. [Online]. Available: https://electronics360.globalspec.com/article/5943/wi-sun-alliance-adds-long-range-mesh-to-wireless-iot-fray. [Geopend 11 november 2020].

[18] Ubiik, „What is Weightless?,” Ubiik, g.d.. [Online]. Available: https://www.ubiik.com/lpwan-technology. [Geopend 27 november 2020].

[19] Ubiik, „Ubiik Weightless Starter Kit,” Ubiik, g.d.. [Online]. Available: https://www.ubiik.com/starterkit. [Geopend 27 november 2020].

[20] DASH7 Academy, „DASH7 Academy,” 2 april 2015. [Online]. Available: https://www.youtube.com/watch?v=ead-fFj4fyc&list=PL6jN_KMUDdhG2-RTyzPkCslqji7iLVzvk&index=1. [Geopend 28 november 2020].

[21] TU Delft, „DASH7,” 28 maart 2011. [Online]. Available: http://wikid.io.tudelft.nl/WikID/index.php/DASH7. [Geopend 28 november 2020].

[22] Wizzilab, „Shop,” Wizzilab, g.d.. [Online]. Available: https://wizzilab.com/shop. [Geopend 28 november 2020].

[23] Wizzilab, „Wizzikit - Service Prototyping,” g.d.. [Online]. Available: https://wizzilab.com/wizzikit. [Geopend 28 novembder 2020].

[24] University of Antwerp, „OSS-7: Open Source Stack for Dash7 Alliance Protocol,” 17 november 2020. [Online]. Available: https://github.com/MOSAIC-LoPoW/dash7-ap-open-source-stack. [Geopend 28 november 2020].

[25] C. v. Ginneken, „EZRPi,” g.d.. [Online]. Available: https://christophe.vg/technology/EZRPi. [Geopend 28 november 2020].

[26] University of Antwerp, „Supported hardware platforms,” 22 november 2018. [Online]. Available: https://mosaic-lopow.github.io/dash7-ap-open-source-stack/docs/supported-hardware/. [Geopend 28 november 2020].

[27] EverythingRF.com, „What is MulteFire Technology?,” 27 september 2017. [Online]. Available: https://www.everythingrf.com/community/what-is-multefire-technology. [Geopend 29 november 2020].

[28] Multefire, „MulteFire Release 1.1 Technical Overview White Paper,” g.d.. [Online]. Available: https://www.multefire.org/wp-content/uploads/MulteFire_Release-1.1_WhitePaper_03JAN.pdf. [Geopend 21 oktober 2020].

[29] Multefire, „Frequently Asked Questions (FAQ),” g.d.. [Online]. Available: https://www.multefire.org/faq/. [Geopend 29 november 2020].

[30] Wi-Fi Alliance, „Wi-Fi HaLow™: Wi-Fi® for IoT applications,” mei 2020. [Online]. Available: https://www.wi-fi.org/downloads-registered-guest/Wi-Fi_HaLow_White_Paper_20200518_0.pdf/36881. [Geopend 29 november 2020].

[31] Silex technology, „How does 802.11ah compare with other LPWAN technologies?,” 30 april 2020. [Online]. Available: https://www.silextechnology.com/hubfs/White_Papers/SX-NEWAH%20White%20Paper_FINAL2020_v8.pdf. [Geopend oktober oktober 2020].

[32] Silex Technology, „How to get your SX-NEWAH-EVK up and running for your 802.11ah evaluation in 12 minutes,” 30 juli 2020. [Online]. Available: https://www.youtube.com/watch?v=Bf5jxr2XnPo. [Geopend 29 november 2020].

[33] Bluetooth SIG, „Exploring Bluetooth 5 – Going the Distance,” 13 februari 2017. [Online]. Available: https://www.bluetooth.com/blog/exploring-bluetooth-5-going-the-distance/. [Geopend 29 november 2020].

[34] Vidar Berg, „Nordic Q&A: General questions about notifications, low level BLE packets and SoftDevice (PHY, connection interval, connection event length, ATT MTU and DLE),” 3 november 2020. [Online]. Available: https://devzone.nordicsemi.com/f/nordic-q-a/47073/general-questions-about-notifications-low-level-ble-packets-and-softdevice-phy-connection-interval-connection-event-length-att-mtu-and-dle/278195#278195. [Geopend 6 november 2020].

[35] Bluetooth SIG, „BLUETOOTH CORE SPECIFICATION Version 5.2 | Vol 6, Part B,” 31 december 2019. [Online]. Available: https://www.bluetooth.com/specifications/bluetooth-core-specification/. [Geopend 14 oktober 2020].

[36] Digi International Inc., „LTE-M vs. NB-IoT: Determine the Differences Between Low Bandwidth Protocols,” 13 maart 2017. [Online]. Available: https://www.youtube.com/watch?v=RlZ9rXbrJqM&feature=emb_title. [Geopend 29 november 2020].

[37] NimbeLink, „LTE Cat 1 vs. Cat M1: Choose the Right IoT Modem for the Job,” 21 juli 2017. [Online]. Available: https://nimbelink.com/blog/lte-cat-1-vs-cat-m1-choose-right-iot-modem-job/. [Geopend 29 november 2020].

[38] Luke, „Alles over 4G: frequenties, banden en de techniek,” 20 april 2016. [Online]. Available: https://www.droidapp.nl/tips-en-tricks/alles-over-4g-frequenties-banden-techniek/. [Geopend 30 november 2020].

[39] SIMCom, „SIM7000E Datasheet,” 2017. [Online]. Available: https://simcom.ee/documents/SIM7000E/SIM7000E_SPEC_2017-9-21.pdf. [Geopend 30 november 2020].

[40] M. L., „LTE-M | Frequently Asked Questions (FAQ),” g.d.. [Online]. Available: https://zakelijkforum.kpn.com/lte-m-forum-60/lte-m-frequently-asked-questions-faq-10867. [Geopend 30 november 2020].

[41] R. J. Heerekop, „Piece-of-Cake LTE-CatM1 SIM7070E RaspberryPI Modem debug tooling v1.0," 1 juni 2020. [Online]. Available: https://github.com/rrrRbert360/LTECatM1_SIM7070E. [Geopend 30 november 2020].

[42] Quintor Academy, „Architectuur & transacties," g.d.. [Online]. Available: https://academy.quintor.nl/moodle/mod/lesson/view.php?id=408&pageid=218. [Geopend 7 december 2020].

[43] Quintor Academy, „Inversion of Control & Dependency Injection," g.d.. [Online]. Available: https://academy.quintor.nl/moodle/mod/lesson/view.php?id=435. [Geopend 7 december 2020].

[44] Techship, „How to configure Simcom SIM7100, SIM7500 and SIM7600 series modules for usage in Windows 8 and 10?," g.d.. [Online]. Available: https://techship.com/faq/how-to-configure-simcom-sim7100-sim7500-and-sim7600-series-modules-for-usage-in-windows-8-and-10/. [Geopend 28 12 2020].

[45] Simpletechpost.com, „RSRP, RSSI and RSRQ," mei 2015. [Online]. Available: http://www.simpletechpost.com/2015/05/rsrp-rssi-and-rsrq.html. [Geopend 17 maart 2021].

[46] Keysight Technologies, Inc., „LTE Physical Layer Overview," g.d.. [Online]. Available: http://rfmw.em.keysight.com/wireless/helpfiles/89600b/webhelp/subsystems/lte/content/lte_overview.htm. [Geopend 17 februari 2020].

[47] CableFree, „LTE Metrics including RSRP, RSRQ and SINR," 18 januari 2018. [Online]. Available: https://www.cablefree.net/wirelesstechnology/4glte/lte-rsrq-sinr. [Geopend 17 febuari 2020].

[48] CableFree Wireless Technology, „LTE Metrics including RSRP, RSRQ and SINR," g.d.. [Online]. Available: https://www.cablefree.net/wirelesstechnology/4glte/lte-rsrq-sinr/. [Geopend 15 januari 2021].

[49] Teltonika-Networks, „Mobile Signal Strength Recommendations," 27 juli 2020. [Online]. Available: https://wiki.teltonika-networks.com/view/Mobile_Signal_Strength_Recommendations. [Geopend 17 februari 2021].

[50] Silex Technology, „SX-NEWAH Evaluation," g.d.. [Online]. Available: https://www.silextechnology.com/connectivity-solutions/embedded-wireless/sx-newah-evaluation. [Geopend 29 november 2020].

[51] Institute for Telecommunication Sciences, „Telecommunications: Glossary of Telecommunication Terms," 23 augustus 1996. [Online]. Available: https://www.its.bldrdoc.gov/fs-1037/dir-013/_1849.htm. [Geopend 30 november 2020].

[52] Keysight Technologies, Inc., „Symbol Rate (Digital Demod)," g.d.. [Online]. Available: http://rfmw.em.keysight.com/wireless/helpfiles/89600B/WebHelp/Subsystems/digdemod/content/dlg_digdemod_fmt_symrate.htm. [Geopend 30 november 2020].

[53] CableFree, „LTE UE Category & Class Definitions," g.d.. [Online]. Available: https://www.cablefree.net/wirelesstechnology/4glte/lte-ue-category-class-definitions/. [Geopend 11 november 2020].

[54] Huawei Technologies Co., Ltd., „Frequentieband en vermogen," in *Honor 10 (COL-L29) Quick Start Guide*, 2018, p. 73.