



# Drone mesh netwerk simulatie

Onderzoeksrapport

HAN Arnhem

561399

MWJ.Berentsen@student.han.nl

Versie 1

Alten Nederland B.V.

Docent: J. Visch, MSc

Assessor: ir. C.G.R. van Uffelen

M.W.J. Berentsen

15 mei 2019

## Samenvatting

Optioneel een samenvatting van het onderzoek. Hier kunnen anderen snel inzicht krijgen in wat jij hebt onderzocht en wat je conclusie is.

- Kunnen derden snel inzicht krijgen in jouw onderzoek?
  - Staat de conclusie erin vermeld?
-

# Inhoudsopgave

<b>1</b>	<b>Inleiding</b>	<b>4</b>
<b>2</b>	<b>Hoofd- en deelvragen</b>	<b>5</b>
2.1	Hoofdvraag . . . . .	5
2.2	Deelvragen . . . . .	5
2.3	Onderzoeksmethode . . . . .	6
2.3.1	De gekozen onderzoeksmethodes . . . . .	6
2.4	Invloed op het project . . . . .	7
<b>3</b>	<b>Criteria</b>	<b>8</b>
<b>4</b>	<b>Literatuur</b>	<b>9</b>
4.1	Welke simulatie software is geschikt om een drone in na te bootsen? . . . . .	9
4.1.1	Gazebo of Webots? . . . . .	10
4.1.2	Conclusie . . . . .	10
4.2	Wat is de grootte van de payload die een netwerkmodule aangesloten op een drone moet versturen om nuttig te zijn voor het aansturingssysteem? . . . . .	11
4.2.1	Binaire serialisatie voor het samenstellen van berichten . . . . .	11
4.2.2	Wat is nuttige informatie die in een bericht kan staan voor of van een aansturingssysteem? . . . . .	12
4.2.3	Wat is de grootte van een bericht voor de locatie bepaling van drones? . . . . .	12
4.2.4	Wat is de grootte van een bericht waarin een netwerkmodule zijn verbinding met anderen aangeeft? . . . . .	14
4.2.5	Conclusie . . . . .	15
4.3	Welke mesh netwerk hardware is geschikt voor het onderling verbinden van drones? . . . . .	16
4.3.1	Wat is een meshnetwerk? . . . . .	16
4.3.2	LPWAN LoRa . . . . .	18
4.3.3	nRF24l01+ . . . . .	19
4.3.4	Conclusie keuze hardware: LoRa vs nRF . . . . .	20
4.4	Welke routingstechniek is geschikt voor de te gebruiken hardware? . . . . .	21

4.4.1	Proactieve routingstechniek . . . . .	21
4.4.2	Reactieve routingstechniek . . . . .	22
4.4.3	Hybride routingstechniek . . . . .	23
4.4.4	Conclusie . . . . .	24
4.5	Wat is minimaal benodigd in een simulatie om abstracte drone te representeren? . . . . .	24
<b>5</b>	<b>Oplossingsrichtingen</b>	<b>25</b>
5.1	Hybride Lightweight Mobile Routing . . . . .	25
5.1.1	Hoe werkt Lightweight Mobile Routing? . . . . .	25
5.1.2	Eigen implementatie idee voor de techniek . . . . .	26
5.1.3	Netwerk ondersteunen met heartbeats . . . . .	27
5.1.4	Conclusie . . . . .	27
5.2	Drone aansturing: Een stap terug voor netwerkherstel . . . . .	28
5.2.1	Een enkele drone die verloren is . . . . .	28
5.2.2	Een groep drones die verloren is . . . . .	29
5.2.3	Conclusie . . . . .	31
<b>6</b>	<b>Experimenten</b>	<b>32</b>
6.1	Test hybride lightweight mobile routing protocol . . . . .	32
6.2	Is het mogelijk voor netwerkmodule om een netwerk te herstellen door het de mogelijkheid te geven zich vrij te bewegen? . . . . .	36
6.3	Hoe gedraagt een fysieke node uit het mesh netwerk zich bij slecht of geen bereik? . . . . .	36
<b>7</b>	<b>Conclusie</b>	<b>37</b>
7.1	Antwoord op de hoofdvraag . . . . .	37
7.2	Advies . . . . .	37
	<b>Literatuur</b>	<b>38</b>
<b>A</b>	<b>Schets voor casus drone netwerk</b>	<b>40</b>

# 1 — Inleiding

De inleiding beschrijft:

- Waarvoor het onderzoek gedaan wordt;
- Beschrijf waarom het onderzoek nu wordt uitgevoerd;
- Het doel van het onderzoek.

---

Het volgende onderzoek rapport wordt uitgevoerd ten behoeve van de het afstudeerproject van Maurice Berentsen, Hogeschool van Arnhem en Nijmegen.

Het doel van dit onderzoek is het zetten van de eerste stap in de ontwikkeling van het dronenetwerk. De eerste stap is ontwikkelen van een netwerkmodule voor het onderling verbinden van drones. Het is van belang dat het meshnetwerk van de drones snel kan reageren op uitval van netwerkpunten.

## 2 — Hoofd- en deelvragen

In dit hoofdstuk worden de hoofd- en deelvragen genoemd en onderbouwd. Er wordt een scope bepaalt met wat er onderzocht wordt en dus ook wat niet. Vervolgens wordt de onderzoeksmethode toegelicht en beargumenteerd. Tenslotte wordt de invloed van het onderzoek op het afstudeerproject beschreven.

### 2.1 Hoofdvraag

Het doel van dit onderzoeksrapport is het beantwoorden van de volgende opgesteld hoofdvraag:

*Hoe kunnen netwerkmodules een groep van ongeveer 100 drones voorzien van een onderling meshnetwerk die snel reageert op uitval van netwerkpunten?*

Doordat de uitvoerende student van dit onderzoek niet beschikt over een certificaat om met drones te mogen vliegen behoudt dit onderzoek zich tot een simulatie. Van de netwerkmodule wordt wel een fysiek prototype ontwikkeld.

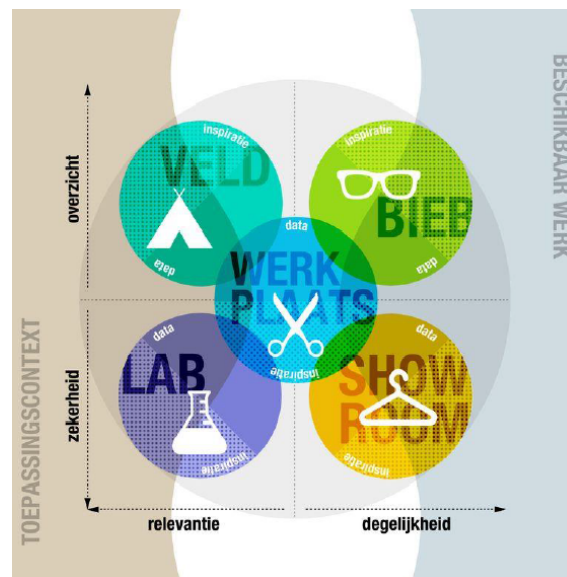
### 2.2 Deelvragen

Om de hoofdvraag te kunnen beantwoorden zijn er de volgende deelvragen opgesteld:

- Welke simulatie software is geschikt om een netwerk van ongeveer 100 drones in na te bootsen?
- Wat is de grootte van de payload die een netwerkmodule aangesloten op een drone moet versturen om nuttig te zijn voor het aansturingssysteem?
- Welke mesh netwerk hardware is geschikt voor het onderling verbinden van drones?
- Welke mesh netwerk techniek is geschikt voor de te gebruiken hardware?
- Hoe gedraagt een node uit het mesh netwerk zich bij slecht of geen bereik?
- Is het mogelijk voor netwerkmodule om een netwerk te herstellen door het de mogelijkheid te geven zich vrij te bewegen?
- Wat is minimaal benodigd in een simulatie om abstracte drone te representeren?

## 2.3 Onderzoeksmethode

De onderzoeksmethodes volgen de structuur van de *ICA methodenkaart* (Van Turnhout et al., 2014). De methodenkaart is een onderzoek framework voor professionals in ICT en media. Het ontwerp van het framework gaat uit van het idee van triangulatie. Triangulatie is het combineren van verschillende theorieën, methoden of databronnen om zo tot betere antwoorden te komen op je onderzoeksvragen (Oates, 2005). Door het definiëren van zo geheten werkplaatsen en hun samenhang geeft het de onderzoeker een set van methoden en technieken. De vijf verschillende werkplaatsen binnen het framework zijn: *lab*, *veld*, *showroom*, *bieb* en *werkplaats*.



Figuur 2.1: De vijf werkplaatsen binnen het framework en hun systematiek. Overgenomen uit *De informatieprofessional 3.0* van Van Turnhout et al. (2014)

### 2.3.1 De gekozen onderzoeksmethodes

De deelvragen gaan gedeeltelijk over keuzes van gebruik. Welke software en hardware is geschikt, is een vraag die veel naar voren komt. Hieruit valt te concluderen dat de onderzoeker zich nog aan het oriënteren is. "De onderzoeksruimte bieb bevat een verzameling methoden en technieken die dienen tot oriëntatie op beschikbaar werk." (Van Turnhout et al., 2014) Daarom wordt er gekozen om een bieb onderzoek uit te voeren naar de deelvragen die gaan over geschiktheid. Dit zodat de onderzoeker zich nog kan oriënteren naar het beschikbare aanbod en nog op nieuwe deelvragen kan komen.

Lab onderzoek wordt pas later uitgevoerd omdat niet alle hardware voor handen is, maar ook omdat er niet genoeg tijd voor beschikbaar is. Van Turnhout et al. zegt het volgende over lab onderzoek: De onderzoeksruimte lab bevat methoden die geschikt zijn om de oplossing te toetsen aan een aspect van de toepassingscontext. Daarom is het lab onderzoek geschikt voor het moment waarop de hardware beschikbaar is en de onderzoeker het gedrag hiervan exact in kaart wil brengen en wil bewijzen dat de gestelde eisen behaald worden.

Oates (2005) stelt dat een systeem bouwen 'bewijs' is dat een nieuw soort toepassing gebouwd kan worden. Als dit het doel is heeft het onderzoek waarschijnlijk hoofdzakelijk een werkplaats karakter, maar alle andere onderzoeksruimtes kunnen een rol spelen. Gedurende het huidige onderzoek zal er door de onderzoeker software geschreven worden om bewijs te leveren dat een netwerkmodule een groep drones kan voorzien van een onderling netwerk. Daarom past de werkplaats onderzoeksruimte goed bij dit onderzoek.

## **2.4 Invloed op het project**



### 3 — Criteria

In [Tabel 3.1](#) worden codes gebruikt als naam voor de criteria, wat deze betekenen wordt eerst toegelicht:

- ALG: Een algemene eis die in elke onderzoek meegenomen moet worden.
- SK: Een eis die gesteld worden in de keuze naar simulatie software.
- DR: Een eis voor de te gebruiken drone in de simulatie.
- PH: Eisen opgesteld aan de prototype hardware.
- PS: Eisen opgesteld aan de prototype software.

Naam	Beschrijving
ALG1	Is voorzien van API documentatie.
ALG2	Ondersteunt aansturing vanuit C++ .
ALG3	Heeft ondersteuning voor het Linux platform Ubuntu 18.04
ALG4	Software is gratis in gebruik voor studenten
SK1	Heeft ondersteuning voor UAV (unmanned air vehicle) ook wel drone genoemd
SK2	Ondersteunt de simulatie van locatie bepaling sensoren zoals GPS.
SK3	Heeft een kant en klare oplossing voor simulatie van externe krachten zoals wind.
SK4	Heeft een ingebouwde pathfinding oplossing.
SK5	Ondersteunt ROS als middleware.
SK6	Ondersteunt de detectie van botsingen.
SK7	Ondersteunt de simulatie 100 drones tegelijk.
DR1	De drone is een quadcopter.
DR2	De drone is voorzien van een API voor aansturing op basis van coördinaten.
DR3	De API sluit aan op de middleware ROS melodic.
DR4	De drone is bruikbaar binnen de gekozen simulatie software.
DR5	De drone moet een kleine goedkope drone representeren.
PH1	Ondersteunt een bereik van minimaal 500 meter.
PH2	Ondersteunt het gebruik van grid networking om tot een mesh te kunnen komen.
PH3	Maakt gebruik van een openbare bandbreedte.
PH4	Voorziet zichzelf van een draadloos netwerk en leunt dus niet op technieken zoals 4G
PH5	Is een product ontwikkeld voor low energy toepassingen.
PS1	Ondersteunt een mesh netwerk van minimaal 100 nodes.
PS2	Het netwerk is zelf Herstellend.
PS3	Ondersteunt het draaien op een raspberry pi of een arduino (UNO of NANO).
PS4	Ondersteunt het versturen van zelfgemaakte berichten.

Tabel 3.1: Opgestelde criteria

## 4 — Literatuur

### 4.1 Welke simulatie software is geschikt om een drone in na te bootsen?

Voor het onderzoek wordt gebruik gemaakt van een simulatie omgeving. Welke software wordt gebruik staat vastgesteld in deze paragraaf. De keuze van software wordt bepaald door het uitvoeren van de bieb onderzoeksmethode. In de [Tabel 3.1](#) staan de criteria toegelicht waar de simulatie software aan moet voldoen. Als hoofdcriteria wordt een lijst opgesteld met simulatoren die ondersteuning hebben voor ROS. Vervolgens worden deze aan de hand van een kruistabel onderworpen aan de andere criteria. Deze techniek is gebaseerd op de "Comparison Chart" van (Van Turnhout et al., z. j.) uit de "*CMD Methods Pack*"

De kandidaten zijn:

- **Actin** is een simulatie framework van het bedrijf Energid. Het is in staat om de beweging van verschillende robots gelijktijdig aan te sturen.
- **Gazebo** is een opensource robot simulatie framework bijzonder geschikt voor het simuleren van robotica in outdoor omgevingen door de uitgebreide Physics Engine Support.
- **V-REP (Virtual Robot Experimentation Platform)** is een platform geschikt voor het snel bouwen van robot prototypes.
- **Webots** is een open source-ontwikkelomgeving die wordt gebruikt voor het modelleren, programmeren en simuleren van mobiele robots.
- **OpenRAVE** biedt een ontwikkelomgeving voor het testen van motion planning algoritmes aan de hand van simulaties.

Simulator \ Eis	ALG1	ALG2	ALG3	ALG4	SK1	SK2	SK3	SK4	SK5	SK6
Actin	✓	✓	✓	x	✓	x	✓	x	✓	✓
Gazebo	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
V-REP	✓	x	✓	✓	✓	✓	✓	✓	✓	x
Webots	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
OpenRAVE	✓	✓	✓	✓	x	✓	✓	✓	✓	x

Tabel 4.1: Kruistabel simulatiekeuze

In de [Tabel 4.1](#) komt de simulatie software van Gazebo en Webots naar voren als enige simulatoren die voldoen aan alle gestelde eisen. Het is nu zaak om te uit te zoeken welke van de twee het meest geschikt is voor het uitvoeren van dit onderzoek.

#### 4.1.1 Gazebo of Webots?

In januari 2018 hebben Pitonakova, Giuliani, Pipe en Winfield een publicatie geschreven waarin zij drie simulatoren vergelijken in het aanbod van features en performance. Zij hebben een vergelijking gedaan tussen V-REP, Gazebo en ARGoS. ARGoS is niet meegenomen in de overweging omdat het geen ondersteuning heeft voor ROS.

In de conclusie van de publicatie komt Webots als het meest gebruiksvriendelijk naar voren in het gebruik en is in het bezit van de meeste features. Het grote nadeel van Webots is dat het veel kracht vraagt van de computer waardoor het niet geschikt is voor de simulatie van veel drones tegelijk.

Daarmee komt de keuze uit op Gazebo. De resultaten in de publicatie van Pitonakova et al. (2018) tonen aan dat de software in staat is om veel drones tegelijk aan te kunnen sturen. Pitonakova et al. (2018) geven als nadeel aan Gazebo dat de software niet in staat is om 3d meshes te manipuleren, de user interface niet innovatief is en Gazebo soms problemen met dependencies door de verschillende versies van ROS. Dit laatste brengt een risico met zich mee voor het onderzoek daarom is het ook zaak dat dit als criteria meegenomen word in de deelvraag van [Paragraaf 4.5 Wat is minimaal benodigd in een simulatie om abstracte drone te representeren?](#).

#### 4.1.2 Conclusie

Voor het onderzoek wordt Gazebo gebruikt omdat het voldoet aan de gestelde [Criteria](#). De meest recente versie van Gazebo, die niet meer beta is op 19 februari 2019, is Gazebo 9. Deze versie vereist het gebruik van ROS Melodic volgens haar eigen website ([zie link](#)).

## 4.2 Wat is de grootte van de payload die een netwerkmodule aangesloten op een drone moet versturen om nuttig te zijn voor het aansturingssysteem?

De payload is het gedeelte van een bericht die vrij is naar de gebruiker om invulling te geven. De maximale grootte van een payload verschilt per protocol. Om een keuze te kunnen maken waar de hardware en het bijhorende protocol aan moet voldoen is het dus zaak om eerst te weten hoe groot deze vrije payload ruimte moet zijn.

Het doel van de berichten is om ze zo kort als mogelijk samen te stellen. Daarom is het belangrijk om de structuur van de berichten zo klein mogelijk te houden. Om deze reden wordt niet gekeken naar een tekstuele serialisatie mogelijkheden zoals XML, YAML, JSON maar naar binaire mogelijkheden omdat deze veel efficiënter zijn (Tauro, Ganesan, Mishra & Bhagwat, 2012).

### 4.2.1 Binaire serialisatie voor het samenstellen van berichten

"Binary Serialization is converting the object in binary format and being able to store it in a storage medium" (Tauro et al., 2012). Het binair samenstellen van informatie is een efficiënte manier omdat de data met een minimale overhead wordt samengesteld (Castillo, Rosales & Blanco, 2018) Het nadeel die deze wijze met zich meebrengt is dat het resultaat van het bericht niet meer makkelijk leesbaar is voor mensen. Voor dit project ligt daar geen focus op dus is dit nadeel te verwaarlozen.

Castillo et al. (2018) hebben onderzoek gedaan naar een efficiënte manier naar de inzet van binaire berichten te vervanging van tekstuele serialisatie. Hoewel dat in het huidige onderzoek niet het geval is kan er wel geleerd worden van de manier van bericht gebruik. Castillo et al. (2018) tonen in hun onderzoek aan dat berichten vaak een herhalende structuur hebben en dat dit een onnodige overhead creëert in de berichten. Een voorbeeld maken Castillo et al. (2018) zichtbaar in [Figuur 4.1](#) waarover zij het volgende zeggen: "While JSON compared with XML, eliminated parameter name redundancy, it keeps repeating (red box) the definition of each field for each serialized element (green box)".

The figure shows two side-by-side code snippets. The left snippet is a JSON message from 'http://localhost:8080/Json/SyncReply/Contacts'. It contains a list of contacts with fields 'FirstName', 'LastName', and 'Email'. The right snippet is an XML message from 'http://localhost:8080/XML/SyncReply/Contacts'. It contains a similar list of contacts but with more verbose XML tags. In the JSON, the field definitions are repeated for each contact (highlighted in a red box). In the XML, the field definitions are repeated for each contact (highlighted in a green box). The XML message is significantly longer than the JSON message.

```
http://localhost:8080/Json/SyncReply/Contacts
{
  "Contacts": [
    {
      "FirstName": "Dennis",
      "LastName": "Bellot",
      "Email": "dennis.bellot@gmail.com"
    },
    {
      "FirstName": "Steve",
      "LastName": "Jobs",
      "Email": "steve@apple.com"
    },
    {
      "FirstName": "Steve",
      "LastName": "Ballmer",
      "Email": "steve@microsoft.com"
    },
    {
      "FirstName": "Eric",
      "LastName": "Schmidt",
      "Email": "eric@google.com"
    },
    {
      "FirstName": "Larry",
      "LastName": "Ellison",
      "Email": "larry@oracle.com"
    }
  ]
}

http://localhost:8080/XML/SyncReply/Contacts
<ContactsResponse xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Contacts>
    <Contact>
      <Email>dennis.bellot@gmail.com</Email>
      <FirstName>Dennis</FirstName>
      <LastName>Bellot</LastName>
    </Contact>
    <Contact>
      <Email>steve@apple.com</Email>
      <FirstName>Steve</FirstName>
      <LastName>Jobs</LastName>
    </Contact>
    <Contact>
      <Email>steve@microsoft.com</Email>
      <FirstName>Steve</FirstName>
      <LastName>Ballmer</LastName>
    </Contact>
    <Contact>
      <Email>eric@google.com</Email>
      <FirstName>Eric</FirstName>
      <LastName>Schmidt</LastName>
    </Contact>
    <Contact>
      <Email>larry@oracle.com</Email>
      <FirstName>Larry</FirstName>
      <LastName>Ellison</LastName>
    </Contact>
  </Contacts>
</ContactsResponse>
```

Figuur 4.1: JSON and XML messages. Overgenomen uit *International Journal of Computer Applications* van Castillo et al. (2018)

Wat zij hebben gedaan om dit op te lossen is het apart versturen van de parameternamen zodat deze in de opvolgende berichten van dat type niet meer nodig is. Het systeem die bedacht is geeft een enorme flexibiliteit omdat er van te voren nog geen bericht types bekend hoeven te zijn. Deze flexibiliteit is alleen niet van toepassing op het doel van dit project, daarom kan de student in zijn software al van te voren berichttypes samenstellen waardoor er de overhead nog kleiner gemaakt wordt. Voor nu wordt er vanuit gegaan dat 1 byte genoeg zal zijn omdat de communicatiesoftware van de netwerkmodule niet meer dan 256 ( $2^8$ ) berichtentypes hoeft te ondersteunen. Mocht het blijken dat dit toch niet genoeg is kan er een byte toegevoegd worden waardoor er 65536 ( $2^{16}$ ) types mogelijk zouden zijn. Daarnaast moet een bericht om nuttig te zijn voor het netwerk altijd aan een apparaat gekoppeld kunnen worden. De opdracht gaat over een netwerk van ongeveer 100 drones wat ruim past in een byte. Daarom wordt er gekozen om een byte te gebruiken wat 255 unieke ID's mogelijk zou maken. Hiermee komt het eerste deel van het bericht dus al tot stand het telt nu minimaal 2 bytes.

ID	Berichttype	overig
00 tot FF	00 tot FF	?

Tabel 4.2: Berichtsamenstelling met type aanduiding

#### 4.2.2 Wat is nuttige informatie die in een bericht kan staan voor of van een aansturingssysteem?

Hoewel er nog niet bekend is welk type netwerk er gebruikt gaat worden is het al wel bekend dat dit een low power long range netwerk zal zijn om aan de requirements te voldoen. Een gedeelde eigenschap van zulke netwerken is de lage bandbreedte waardoor ze niet geschikt zijn voor realtime applicaties. Deze kennis kan toegepast worden op de informatie die de drones met hun netwerkmodules bij zich dragen. Informatie zoals snelheid en richting heeft geen prioriteit om verstuurd te worden omdat deze niet relevant is voor het aansturingssysteem aangezien hier geen beslissingen op gebaseerd kunnen worden. Wat wel van belang is voor het aansturingssysteem is informatie als waar bevindt een netwerkmodule zich op welk moment in tijd. Daarnaast kan informatie zoals de status van de batterij en de sterkte van de verbinding van belang zijn voor het systeem omdat deze informatie gebruikt kan worden om preventieve beslissingen te nemen in het onderhoud van het netwerk.

Een bericht die het aansturingssysteem tenminste zal verzenden zal het commando zijn om een drone naar een coördinaat te sturen. Daarnaast zal er een bericht moeten komen waarmee het systeem de status van het netwerk kan controleren, wat voor bericht dit zal zijn is nog niet bekend.

#### 4.2.3 Wat is de grootte van een bericht voor de locatie bepaling van drones?

De huidige locatie is essentiële administratie voor de verdeling van de drones. Omdat de drones zich kunnen verplaatsen is naast de locatie ook het tijdstip waarop het bericht verstuurd wordt van belang.

## Coördinaat vanuit de informatie van de GPS module

Een positie van een locatie wordt aangeduid door coördinaten te gebruiken. Deze worden bepaald door het gebruik van een GPS module aan boord van het systeem. GPS modules gebruiken triangulatie van satelliet signalen om hun positie te bepalen.

Een coördinaat is opgebouwd door de aarde te verdelen in graden over de assen latitude (horizontaal) en longitude (verticaal). Een coördinaat loopt van -180 tot 180 graden. De aarde heeft een omtrek van 40.07.5161,2 meter gedeeld door 360 graden levert dit per graad een stap op van 111,32 kilometer. Het volgende programma laat zien dat een float gebruikt kan worden in dit geval voor een accuraatheid van 5 cijfers achter het decimaal.

```
#include <stdio.h>

float exampleFloatPositive = 179.1234567890;
float exampleFloatNegative = -179.1234567890;

int main( int argc, const char* argv[] )
{
    printf( "positive float = %f\n", exampleFloatPositive );
    printf( "negative float = %f\n", exampleFloatNegative );
}
```

Door een coördinaat te gebruiken van 5 cijfers achter het decimaal kan een plek op aarde met een precisie van ongeveer 1,10 meter nauwkeurig aangegeven worden. Het is gebruik hiervan is realistisch omdat met alleen realtime GPS data de accuraatheid ongeveer 4 meter is (U.S. Government, 2008). Er wordt per as een float gebruikt dit houdt in dat minimaal 8 bytes nodig zijn voor een coördinaat. Dit is 4 bytes per float en exclusief de separator.

## Hoogte van de drone

Kulhavy, Hung, Unger en Zhang (2017) hebben onderzoek gedaan naar de accuraatheid van de hoogtebepaling bij drones. Uit dit onderzoek is gebleken dat een continue vliegende drone een accuraatheid heeft van ongeveer twee meter. Hieruit is te halen dat het geen zin heeft om op centimeter niveau de hoogte van een drone aan te geven. Op het moment dat er een signed integer van 16 bits gebruikt zou worden kan er een afstand van 32.767 meter zowel boven als onder zeeniveau verstuurd worden. Het is een realistische waarde om te gebruiken aangezien de grootste hoogte die gebruik kan worden dan 32 kilometer is. Deze hoogte is al ver in de stratosfeer en zelfs al boven het niveau die een weerballon haalt. De keuze om een signed integer te gebruiken is om ondersteuning te bieden aan plekken onder het zeeniveau waarvan Nederland een groot voorbeeld is.

Voor de hoogte moeten 2 bytes gereserveerd worden.

## Tijd vanuit de GPS module

De drone maakt al gebruik van de GPS module voor de positiebepaling. In de berichten die de GPS module ontvangt van de satellieten wordt ook de tijd meegestuurd. Deze tijd is volgens (U.S. Government, 2008) in 95 procent van de gevallen accuraat tot 40 Nano seconden. In het geval van het drone netwerk is dit veilig te gebruiken als bron voor tijd. De tijd zal aangegeven worden in het unix time format zodat deze meteen de datum met zich draagt. Unix time maakt gebruik van 32 bits formaat, er is gekozen voor een

unsigned variant aangezien er geen interesse is naar data voor 1 januari 1970 ook wel epoch genoemd.

Voor de tijd en datum worden 4 bytes gereserveerd.

## Conclusie

De bovenstaande sub paragrafen kunnen samengesteld worden tot een bericht waarin alle informatie zit die nodig is om te bepalen waar een drone zich op een moment bevindt. Een systeem kan met dit bericht in ieder geval elke drone afzonderlijk in kaart brengen.

	ID	Bericht type	latitude	longitude	hoogte	tijd
Aantal Bytes	1	1	4	4	2	4

Tabel 4.3: Locatie bericht in bytes

In 4.3 is het aantal bytes te zien benodigd voor een locatie bericht. Concluderend wordt voor dit bericht een grootte van minimaal 16 bytes gebruikt.

### 4.2.4 Wat is de grootte van een bericht waarin een netwerkmodule zijn verbinding met anderen aangeeft?

Een van de wensen van de Alten is het in kaart brengen van de onderlinge verbindingen van de modules in het drone netwerk. Als voorbeeld hoe dit eruit kan zien is de schets afkomstig van de casus uit het plan van aanpak toegevoegd als bijlage te vinden in [Appendix A](#). De informatie van de locatie van de netwerkpunten zijn al bekend door het gebruik van het bericht in [Subparagraaf 4.2.3](#) Wat overblijft als onbekende informatie is de onderlinge verbinding.

In het netwerk kunnen twee rollen verdeeld worden. De rol van zender aan het device die informatie wil versturen naar het basis station. Of de rol van ontvanger aan het device die de informatie van zender ontvangt. Een device die beide rollen kan aannemen wordt een transceiver genoemd. Als werkbaar voorbeeld wordt het netwerk uit de [Schets voor casus drone netwerk](#) gebruikt. De ontvangers en zenders zijn verdeelt in [Tabel 4.4](#).

Ontvanger	BS	9	7	5	3	2	8	6	11	13	12	14
Zender	9,11	7,8	5	3	2	1	6	4	13,10	12	14	15,16,17

Tabel 4.4: De ontvangers en zenders uit de casus

## Topologie met PlantUML

Om snel aan te kunnen tonen dat er weinig informatie nodig is om een topologie op te stellen gebruikt de student PlantUML. Hiermee wordt aangetoond dat een bericht die de ontvanger aanmaakt zodra een zender zich bij hem aanmeldt waar in staat wie de zender is genoeg is voor een topologie. De keuze om de verantwoording bij de ontvanger neer te leggen is omdat een zender niet bij elke soort netwerk weet wie de ontvanger is. De uitwerking hiervan is zichtbaar in [Figuur 4.2](#)





netwerk is het onaannemelijk dat deze groter zullen zijn dan deze 17 bytes. Om toch ruimte te houden voor extra informatie in een bericht voor bijvoorbeeld data verificatie worden er 4 bytes toegevoegd aan het maximum. Dit zou voldoende ruimte moeten geven doordat er één 32 bits waarde of twee 16 bits waardes toegevoegd kunnen worden.

Daarmee is de conclusie dat de maximale grootte 21 byte is voor een bericht om nuttig te zijn voor een aansturingssysteem.

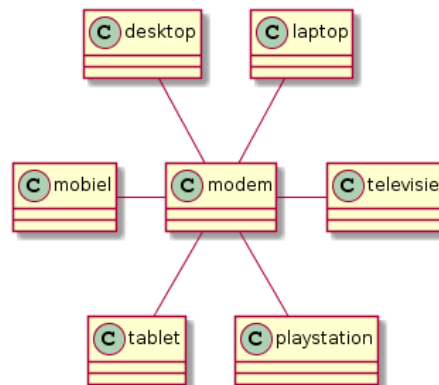
### 4.3 Welke mesh netwerk hardware is geschikt voor het onderling verbinden van drones?

Om een onderling netwerk op te bouwen is hardware nodig. Deze hardware moet voldoen aan de gestelde [Criteria](#) zodat het geschikt is voor een drone netwerk. In [Subparagraaf 4.2.5](#) is vastgesteld dat de payload berichten van 21 byte moet kunnen bevatten. Criteria [PH1](#) eist dat het bereik van de modules minimaal 500 meter is. Vervolgens stelt criteria [PH5](#) stelt dat het netwerk zuinig in gebruik moet zijn.

#### 4.3.1 Wat is een meshnetwerk?

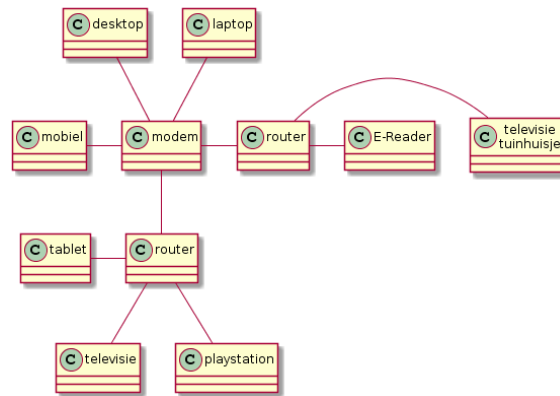
Een meshnetwerk is een term die gebruikt wordt voor een topologie techniek in netwerken. Om een voorbeeld beter toe te kunnen lichten worden eerst de tegenhangers besproken de ster en boom topologie.

Een doorsnee huishouden maakt zeer waarschijnlijk gebruik van een ster topologie. Er staat een modem bij de aansluiting in het huis die een Wifi signaal uitzendt waar vervolgens alle mobiele apparaten mee verbinden en vaak nog een fysieke poorten aanbiedt voor een bekabelde verbinding. Dat levert een topologie op zichtbaar in [Figuur 4.3](#)



Figuur 4.3: Topologie van ster netwerk

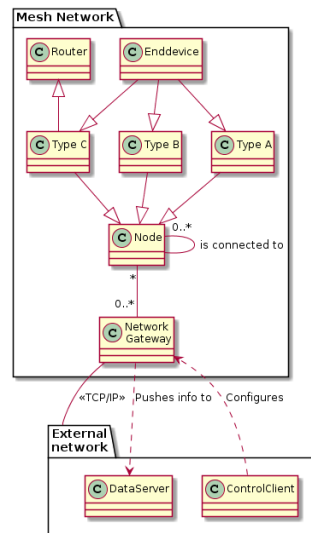
In de meeste huishoudens volstaat dit aangezien de router bereikbaar is vanaf elk punt in het huis. Op het moment dat dit niet zo is het gangbaar om uit te wijken tot een boom topologie door een of meerdere routers toe te voegen in het netwerk. Vertaalt naar een topologie zou dit er als [Figuur 4.4](#) uitzien.



Figuur 4.4: Topologie van boom netwerk

Een probleem wat uit deze twee voorbeelden duidelijk wordt is de grens van het bereik. Deze is namelijk alleen uit te breiden in door het toevoegen van extra netwerkpunten. De mesh netwerk techniek is een oplossing voor dit probleem. In een meshnetwerk zijn de apparaten in staat om direct met elkaar te verbinden. Zo is elk apparaat in het netwerk direct een uitbreiding in het bereik van het netwerk doordat het informatie kan doorgeven.

Er zitten aan een meshnetwerk ook nadelen. Omdat er nu meerdere routes van en naar een apparaat zijn komt er een extra protocol bij kijken welke route genomen moet worden wat extra complexiteit toevoegt. Daarnaast zal het draadloze verkeer verhoogt worden waardoor uiteindelijk apparaten elkaar zullen verstoren. Tenslotte voegt het ook een afhankelijkheid toe omdat bijvoorbeeld de televisie in het tuinhuisje de laptop in de keuken gebruikt als stap naar het modem.



Figuur 4.5: Verbanden tussen de rollen in een mesh netwerk

In een meshnetwerk zijn er twee rollen te verdelen (Saha, Mandal, Mitra, Banerjee & Saha, 2017). Er is de rol van gateway waarmee het netwerk verbonden wordt naar een extern

punt. En er is de rol van node welke in staat is om met een gateway te verbinden of met een andere node.

Een node is vervolgens weer te verdelen in drie types, de namen die hier aan gegeven worden verschillen per aanbieder voor het gemak worden die van LoRa gebruikt. In LoRa wordt gesproken over type A, B en C waarbij elke type gericht is op een verschillende visie van stroomconsumptie (Adelantado et al., 2017).

- **Type A** is de energiezuinigste variant doordat het alleen online komt op het moment dat het data moet versturen. Het grote nadeel van dat het communiceren naar dit type onmogelijk is en het dus ook niet bijdraagt het uitbreiden van het netwerk.
- **Type B** is een variant van type A welke na het zenden van een bericht enige tijd online blijft om een berichten te kunnen ontvangen. Een voordeel is dat deze dus wel op afstand aan te sturen is maar nog steeds is deze alleen een gebruiker van het netwerk.
- **Type C** is de minst zuinige variant van de nodes doordat hij altijd aan staat. Het nadeel is dus dat deze het meest stroom verbruikt daarin tegen kan deze node wel de belangrijke rol krijgen van router en biedt vergroot daarmee het mesh netwerk.

In het geval van de opdracht wordt alleen gekeken naar type C nodes omdat er interesse is naar de dynamische plaatsing van dit type.

#### 4.3.2 LPWAN LoRa

Deze twee eisen zijn een logische stap om een verdieping uit te voeren in low-power wide-area network (LPWAN) netwerken.

In de publicatie van (Vangelista & Centenaro, 2019) wordt het LoRaWAN netwerk onderzocht. Ook worden daar de grootste concurrenten van het LoRaWAN netwerk benoemd namelijk **SigFox** en Narrowband IoT (NB-IoT). Ondanks het feit dat deze protocollen niet interessant zijn voor het project omdat ze in de verkeerde OSI-laag vallen namelijk de netwerklaag is het wel interessant om te kijken welke techniek deze gebruiken. NB-IoT valt direct af als interesse doordat het gebruikt maakt van mobiele communicatie en dus niet zou voldoen aan criteria **PH4**. Daarna valt SigFox ook af simpelweg omdat het gebruik maar beperkt openbaar is gemaakt waardoor het niet geschikt is om een meshnetwerk mee te bouwen.

Dan blijft uit deze publicatie het LoRaWAN netwerk over. Deze is gebaseerd op de LoRa fysieke laag (PHY)(Semtech, 2015) implementatie om de communicatie af te handelen. LoRaWAN is een netwerkstandaard opgezet voor telecom providers om het mogelijk te maken netwerken op te zetten voor communicatie over afstanden tot 15km. Alleen het LoRaWAN maakt gebruik van een ster topologie doordat nodes maar 1 hop kunnen maken naar een gateway van het netwerk. Een andere nadeel die het LoRa netwerk heeft is dat ondanks de grote afstand die het kan afleggen door de spreidingsfactor te vergroten en de gevoeligheid van de ontvanger te verhogen dit ten koste gaat van de snelheid van de datadoorvoer. Lee en Ke (2018) hebben in onderzoek gedaan naar de mogelijkheid om het LoRa PHY netwerk te gebruiken als mesh netwerk en ook aangetoond dat dit werkt. Het doel van hun onderzoek was om door middel van mesh technieken LoRa geschikter te maken voor verstedelijk gebied zonder het toevoegen van extra gateways. Een nadeel

wat ze beschrijven in de conclusie is het aantal nodes wat ze kunnen verwerken verkleind wordt. Dit komt door de degradatie van timing prestaties waarin ze stellen dat periode  $p$  die een bericht nodig heeft om verstuurd te worden naar de gateway de volgende regel volgt ( $p = 2h \cdot t$ ) waarbij  $h$  het aantal hops is die het bericht maakt en  $t$  de frametijd. Om de frametijd te berekenen geldt het volgende:  $\text{frametijd} = \frac{\text{framelengte}}{\text{bitsnelheid}}$ . Voordeel van het gebruik die een mesh kan hebben ten opzichte van een ster netwerk is dat het een hogere bitsnelheid kan behouden omdat er kleinere afstanden overbrugt hoeven te worden waardoor de frametijd alsnog klein kan blijven. Een voorwaarde van het gebruik van dit netwerk is dus wel dat de framelengte klein moet zijn waardoor het niet geschikt is voor het verwerken van grote berichten zoals afbeeldingen video of andere grote bestanden.

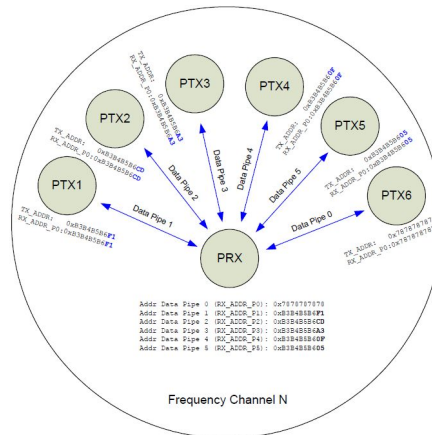
## LPWAN LoRa samengevat

Als er gekeken wordt naar LPWAN oplossingen is LoRa PHY de enige noemenswaardige oplossing voor dit project. Lee en Ke (2018) hebben al aangetoond dat een mesh netwerk op basis van LoRa mogelijk is waardoor het risico afneemt in de haalbaarheid van dit onderzoek. Een voordeel van dit netwerk is de afstand die de netwerkmodules van elkaar kunnen staan die met deze techniek kan oplopen tot meer dan 10 kilometer afhankelijk van de hoeveelheid en frequentie van de data die verstuurd moet worden. Een ander voordeel van deze techniek is dat het signaal enorme afstanden kan halen bij een vrije lijn of sight waarbij het wereldrecord zelf op 702 kilometer staat door een weerballon op 38 kilometer hoogte te gebruiken (Telkamp & Slats, 2017). Hoewel deze extreme usecase niet zal voorkomen worden de netwerk modules wel gemaakt om aan drone gekoppeld te worden. Hierdoor is het makkelijk mogelijk voor een netwerkmodule om geplaatst te worden op hoge punten. Een nadeel van dit netwerk is de lage data snelheid waardoor het alleen geschikt is voor toepassingen die weinig data versturen zoals sensor netwerken.

### 4.3.3 nRF24l01+

De nRF24l01 long range is een transceiver van Nordic Semiconductor die werkzaam is op de 2,4ghz band (Nordic Semiconductor, 2008). De module kan gebruik maken van verschillende standen van datasnelheid: 250kbps, 1Mbps, 2Mbps. Hoewel het een 3,3 volt apparaat is zijn de I/O pinnen 5 volt tolerant waardoor het een veilige interface biedt aan microcontrollers zoals die van Arduino.

Er zijn van de module twee varianten met een gelijke interface. Er is een module (nRF24l01) voor korte afstand en een module (nRF24l01+pa+lna) voor lange afstanden vaak aangeduid als nRF24l01+. De module voor lange afstanden beschikt over een power amplifier en een low noise amplifier wat het een beter zend en ontvangst vermogen geeft. De module heeft volgens de specificaties een theoretisch bereik van 1100 meter bij vrij zicht. Er zijn verschillende video's op youtube waar aangetoond wordt dat deze afstand gehaald wordt. Zo wordt in de video (<https://www.youtube.com/watch?v=IR60toEjHl8>) van iforce2d een afstand overbrugt van 700m met een data rate van 250kbps (iforce2d, 2014). Het zal in ieder geval voldoende zijn om de eis van 500 meter te overbruggen.



Figuur 4.6: Werking van verbindingen nRF24l01. Afkomstig van <https://www.instructables.com/id/NRF24L01-Multiceiver-Network>, 18 maart 2019

Hij kan tot zes kanalen tegelijk open houden die kunnen schakelen tussen een RX (ontvang) en TX (zend) stand. Figuur 4.6 laat het gebruik van adressen zien, hierin is te zien dat een adres is opgebouwd uit 5 bytes. Dit levert een totaal op van 10782039375 ( $255^5$ ) unieke adressen. Het grote aantal zal waarschijnlijk nooit bereikt worden door de complexiteit die de topologie van een dusdanig netwerk geeft. De modules versturen pakketten met een maximale grootte van 32 byte.

Als er gekeken wordt naar de populaire library genaamd nRF24 dan wordt er gebruikt gemaakt van 1 byte voor het adres wat 255 unieke mogelijkheden geeft. De samenstelling van een pakket is af te lezen in Tabel 4.6 NRF24 library pakket

	FROM	TO	FORWARD	ACK	PAYLOAD
Aantal Bytes	1	1	1	1	28

Tabel 4.6: NRF24 library pakket

In deze toepassing blijft er dus ruimte over voor 28 bytes als payload wat nog steeds voldoende zou zijn voor de netwerkmodules van de drones.

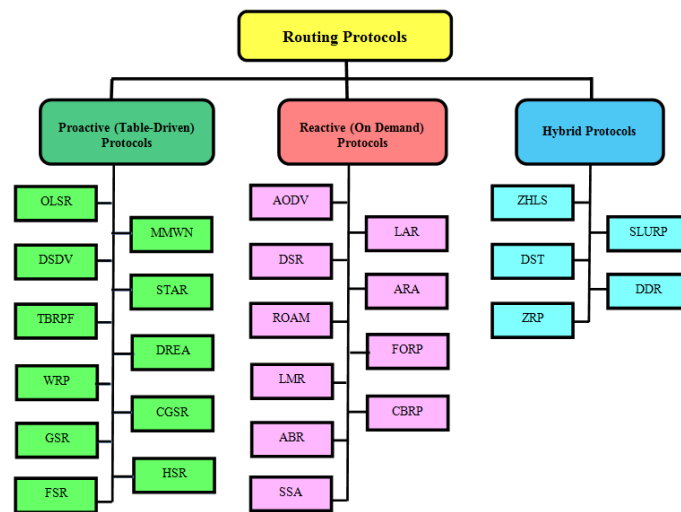
#### 4.3.4 Conclusie keuze hardware: LoRa vs nRF

Er zijn nu twee kandidaten voor het netwerk. De LoRa PHY module die gigantische afstanden kan afleggen maar als keerzijde een langzame verbinding heeft. Als andere kandidaat is er de nRF24l01+ een module die vele malen sneller is maar dan ook flink inlevert op het bereik ten opzichte van LoRa.

Omdat beide modules voldoen aan de gestelde eisen en in staat zijn de gewenste payload te versturen zijn ze voorgelegd aan de technisch manager van Alten Nederland. Hij ziet een betere business case in een netwerk met een hogere dichtheid maar dan ook een hogere snelheid. Daarom zal er in dit project gebruik gemaakt worden van de nRF24l01+.

## 4.4 Welke routingstechniek is geschikt voor de te gebruiken hardware?

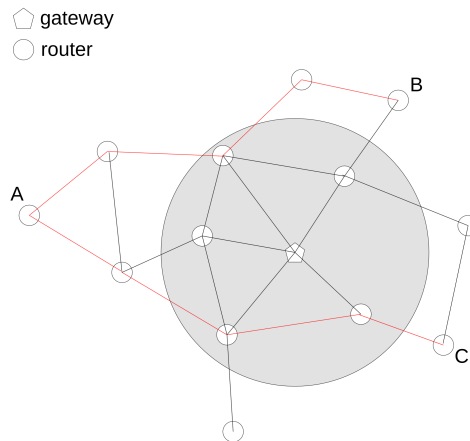
Het netwerk wordt opgebouwd door het toepassen van een meshnetwerktechniek. Wat een meshnetwerk is wordt al toegelicht in [Subparagraaf 4.3.1](#). Een kenmerkend eigenschap van een meshnetwerk is het hebben van meerdere verbindingpunten per router of gateway. Dit wordt opgezet door een routingstechniek. Er zijn verschillende technieken met elk voor en nadelen. De technieken zijn te verdelen in drie groepen: proactieve, reactief en hybride (Abdulghani & Salleh, 2013). Het verschil tussen deze technieken zit hem in ontdekken van andere nodes in het netwerk. In de hierop volgende paragrafen worden deze technieken kort toegelicht. Vervolgens zal besloten worden welke van deze drie het meest geschikt is voor de te gebruiken hardware. Tenslotte zal gekeken worden binnen die groep gekozen worden welk techniek uitgewerkt wordt.



Figuur 4.7: Bestaande routeringsprotocollen. Afkomstig van (Abdulghani & Salleh, 2013)

### 4.4.1 Proactieve routingstechniek

Een proactief protocol is een protocol die zelf constant bezig is met het zoeken naar nieuwe punten in het netwerk. De nodes zorgen dat veranderingen meteen kenbaar gemaakt worden aan andere punten in het netwerk. Dit zorgt ervoor dat het netwerk een zelf herstellend vermogen heeft en geschikt is voor het opsporen van fouten in het netwerk. Het protocol is geschikt voor situaties waar de posities van nodes zelden tot nooit veranderen. In situaties waar paden vaak veranderen door het veranderen van node posities zal het protocol veel meer bandbreedte vragen van het netwerk.



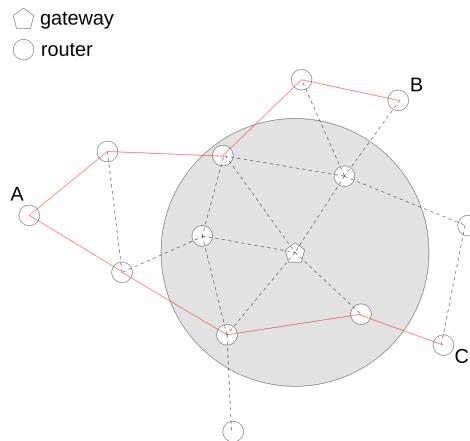
Figuur 4.8: Voorbeeld proactieve routeringstechniek.

In [Figuur 4.8](#) wordt een voorbeeld getoond waar punt A een bericht wil versturen naar punt B en C. Punt A is op de hoogte van alle routes in het netwerk net als de andere punten in het netwerk dat zijn. In de huidige situatie zijn er 21 verbindingen in het netwerk tussen de nodes. Er zijn 14 punten aanwezig in het netwerk minimaal zullen er dus voor het op de hoogte brengen van bestaande verbindingen  $294$  ( $14 \times 21$ ) berichten rondgestuurd moeten worden.

Vervolgens kan er een pad gemaakt worden waarover een bericht verstuurd kan worden. Dit voorbeeld laat goed zien dat deze routingstechniek zijn voordeel haalt wanneer de nodes stil blijven staan omdat er dan niet opnieuw berichten verstuurd hoeven te worden.

#### 4.4.2 Reactieve routeringstechniek

Een reactieve routeringstechniek is een methode waarbij de nodes geen routes opslaan maar op aanvraag routes opbouwen. Dit scheelt in het gebruik van geheugen en het heeft minder berichten bij het initialiseren van het netwerk. Daarentegen kost het deze techniek meer tijd om een bericht te versturen omdat er voordat er een bericht verstuurd kan worden eerst een route opgebouwd moet worden.

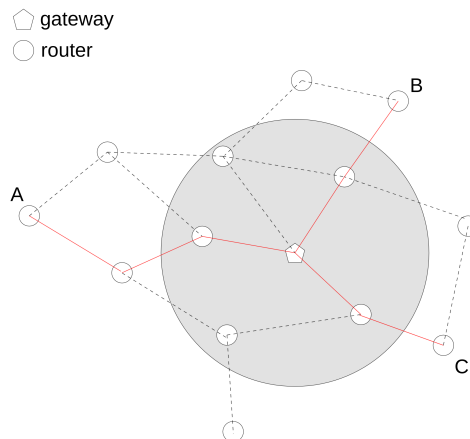


Figuur 4.9: Voorbeeld proactieve routeringstechniek.

Figuur 4.9 laat een voorbeeld zien waarin er gebruik gemaakt wordt van een reactieve routeringstechniek. Er zal in dit voorbeeld voordat een route gezocht wordt er een route gemaakt moeten worden tussen de twee punten. Vervolgens als deze route gevonden is zal er een bericht hierover gestuurd worden. Deze route wordt tijdelijke opgeslagen tot de node klaar is met het versturen van zijn data.

#### 4.4.3 Hybride routeringstechniek

Een hybride routeringstechniek is een techniek die een proactieve en reactieve techniek deels combineert om zo ongewenste nadelen te elimineren van elkaar.



Figuur 4.10: Voorbeeld hybride routeringstechniek.

In dit voorbeeld is een hybride routeringstechniek uitgewerkt er zijn veel meer varianten aanwezig het voorbeeld is om te laten zien hoe reactief en proactief geactiveerd wordt. In deze situatie is de gateway voorzien van een proactief protocol waardoor het alle routes kent. De routers hoeven daardoor alleen een route te berekenen naar een gateway toe.



#### 4.4.4 Conclusie

Het dronenetwerk is een netwerk die vaak te maken zal hebben met verplaatsingen. De hele eigenschap om autonoom nodes te kunnen verplaatsen is zelfs de aanleiding van het project. Daardoor vallen proactieve routingstechnieken af. Toch is de tegenhanger reactieve routing ook geen ideale oplossing. Het niet opslaan van routes maakt het versturen van berichten sloom en er is geheugen aanwezig op de microcontroller voor het opslaan van routes.

Daarom is de conclusie om een zelf een hybride variant te maken van het reactieve Lightweight Mobile Routing(LMR) protocol. Dit protocol wordt gebruikt omdat het een protocol is die een hop routingstechniek gebruikt waardoor de payload klein blijft. De techniek is niet afhankelijk van een gateway en nodes hoeven alleen te bewust te zijn van de burens en met wie die burens verbonden zijn. Het protocol hoeft alleen maar aangepast te worden zodat routes opgeslagen blijven. Doordat deze techniek geen routes hoeft op te slaan van het volledige netwerk maar allen die van hun aangesloten burens en hun aangesloten punten zullen er geen grote routingstabellen gecreëerd worden.

### 4.5 Wat is minimaal benodigd in een simulatie om abstracte drone te representeren?

In [Paragraaf 4.1](#) is geconcludeerd dat als simulatie software ROS melodic met Gazebo9 wordt gebruikt. Hierin kwam naar voren dat niet elke drone package gebruikt kan worden doordat ze specifiek gemaakt zijn voor de versie van ROS. Omdat het project de focus heeft op de werking van netwerkmodules voldoet een abstracte versie van een drone. Zo hoeft deze bijvoorbeeld op dit moment nog niet realistisch te vliegen.

Daarom is besloten dat de drone het volgende moet kunnen in de simulatie:

- Een drone moet verticaal kunnen opstijgen.
- Een drone moet verticaal kunnen landen.
- Een drone moet zich kunnen voortbewegen.
- Een drone verplaatsing begint altijd met opstijgen en eindigt altijd met een landing.
- Een drone moet voorzien zijn van module die een locatie kan teruggeven.
- Een drone moet mag een basisvorm gebruiken om zich zichtbaar te maken in de simulatie

## 5 — Oplossingsrichtingen

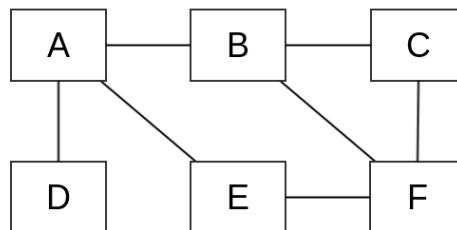
In dit hoofdstuk worden oplossingsrichtingen beschreven die bijdragen aan het beantwoorden van de hoofdvraag. Een logisch vervolg voor elke oplossingsrichting is dat deze vervolgt wordt in het [Hoofdstuk 6 Experimenten](#) om te bewijzen dat deze oplossingsrichting werkt.

### 5.1 Hybride Lightweight Mobile Routing

[Welke routingstechniek is geschikt voor de te gebruiken hardware?](#) heeft uitgewezen dat het implementeren van een aangepast versie van het LMR protocol aangeraden wordt als implementatie. In deze paragraaf wordt verdiepend behandeld hoe dit protocol werkt en wat er aangepast moet worden.

#### 5.1.1 Hoe werkt Lightweight Mobile Routing?

Lightweight Mobile Routing is een techniek die gebruik maakt van een link reversal protocol. Dit houdt in dat een pad zich achterstevoren opbouwt.



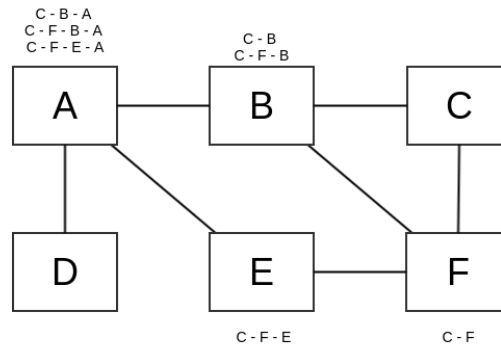
Figuur 5.1: Voorbeeld link reversal.

Om dit beter uit te leggen wordt [Figuur 5.1](#) gebruikt. In de meeste gevallen wanneer A een bericht naar C wil sturen gaat hij zoeken naar een pad. Dit doet A door aan B en D en E te vragen of zij C kennen wat vervolgens een heel netwerk zou doorlopen. In elke bericht wordt de stap van route opgeslagen waardoor in het voorbeeld C uiteindelijk een drie berichten zou krijgen namelijk A-B-C, A-B-F-C en A-E-F-C. Deze stuurt C vervolgens weer terug waarop A kan bepalen welke route hij wil nemen waarbij A-B-C aannemelijk is aangezien dit de kortste route is.

Bij link reversal werkt dit dus andersom (Vainio, z. j.). Hier maakt A bekend dat hij een route wil opbouwen naar C op dezelfde wijze als hierboven alleen begint C nu met het

bouwen van een route. A zal dus uiteindelijk drie routes ontvangen C-B-A, C-F-B-A en C-F-E-A.

Op zich lijkt dit niet zo veel anders dan wat er normaal gebeurt maar dat is waar het LMR protocol komt kijken. Na het versturen van een bericht naar C zijn er namelijk routes opgeslagen bij de punten te zien in [Figuur 5.2](#)



Figuur 5.2: Voorbeeld LMR na het versturen van een bericht.

Nu op het moment dat A een bericht naar C wil sturen weet hij op basis van de informatie die hij heeft dat hij het bericht moet versturen naar B om het bij C terecht te laten komen. Stel dat de verbinding B - C wegvalt dan weet B dat als hij het bericht aan F geeft dat het bericht nog steeds aan zal komen bij C. Dit houdt dus in dat wanneer de verbinding B - C weg is dat A dit niet hoeft te weten want een bericht kan nog steeds succesvol verzonden worden via B.

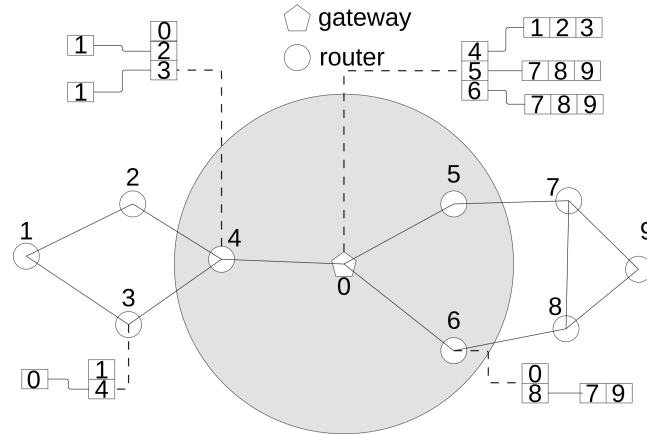
Er hoeven dus alleen berichten verstuurd te worden over verloren verbindingen op het moment dat alle wegen verloren zijn. In het geval van een dronenetwerk waar punten vaak veranderen zal dit aanzienlijk schelen in de hoeveelheid berichten die rondgestuurd zullen worden.

### 5.1.2 Eigen implementatie idee voor de techniek

Wat opvalt in de LMR techniek is dat er routes worden opgeslagen naar nodes toe terwijl waarop de richtingen worden bepaald. De routes worden opgeslagen terwijl er alleen maar interesse is in de eerste stap van de route. In het voorbeeld van [Figuur 5.2](#) slaat A bijvoorbeeld route C-B-A en C-F-B-A op. Beiden geven het zelfde effect bij het zoeken voor een route naar C, het bericht moet aan B gegeven worden.

Het idee om ruimte te bewaren is om twee soorten tabellen bij te houden. Een tabel van direct aangesloten punten waarbij elke aangesloten punt ook een eigen tabel heeft met punten waar deze directe buur mee kan communiceren. In het bovenstaande voorbeeld zou dit betekenen dat A alleen hoeft te weten dat hij via B met C en F kan communiceren. Dit bespaart ruimte in het opslaan van routes maar ook in de grootte van berichten, nu hoeven er immers geen routes meer rondgestuurd te worden alleen notificaties van nieuwe verbindingen.

Om te laten zien hoe deze implementatie zich vertaalt naar een praktisch voorbeeld van een netwerk met een gateway en routers is er een extra afbeelding gemaakt te zien in [Figuur 5.3](#).



Figuur 5.3: Toelichting routeringstabel

In [Figuur 5.3](#) wordt geïllustreerd hoe de routeringstabellen van een gateway en verschillende routers er uitzien. Een nadeel van de huidige implementatie is dat er geen rekening wordt gehouden met het kortste pad door het netwerk. Als in [Figuur 5.3](#) bijvoorbeeld de gateway een bericht wil versturen naar router nummer 8 zal dit gebeuren via de weg 0-5-7-8 in plaats van de kortste weg 0-6-8. Dit komt omdat er eerst gekeken wordt of het bericht voor de geadresseerde een direct aangesloten punt is, als dat niet zo is worden de tabellen van de indirecte punten nagelopen. Omdat deze op numerieke volgorde de tabellen van externe punten nazoeft zal de tabel van punt 5 als eerste een routepunt geven.

### 5.1.3 Netwerk ondersteunen met heartbeats

Een wens van Alten is dat het netwerk snel is en ook snel achter verloren verbindingen komt. De huidige implementatie van LMR zal dit beiden niet zijn om twee redenen. LMR is event driven waardoor het alleen achter verloren verbindingen zal komen op het moment dat er een bericht geen route weet te vinden tot een punt in het netwerk. Daarnaast worden routes pas opgebouwd op het moment dat er een bericht over het netwerk verstuurd wordt. Tenslotte hebben de routers een belang bij de kennis of ze verbinding hebben met een gateway in het netwerk van Alten omdat dit hun verbindingspunt is met de buitenwereld.

Een oplossing voor al deze problemen is het implementeren van een functie in elke router dat hij periodiek moet proberen een bericht naar een gateway te sturen. Als response op deze heartbeat zal de router ook weer een heartbeat terug moeten sturen. Deze oplossing zou zorgen dat het netwerk zijn routes blijft onderhouden door ze periodiek te controleren. Het zou ook zorgen dat een gateway altijd bekend is met alle mogelijke routes omdat hij weet welke node er in elke richting beschikbaar is. Bij het versturen van een bericht hoeft er dus niet eerst een route opgebouwd te worden. Tenslotte zou het sturen van een response de router meteen bewust maken of er een verbinding bestaat naar een gateway.

### 5.1.4 Conclusie

Het veranderen van de LMR door er heartbeat aan toe te voegen maakt de techniek deels proactief. Zoals al eerder beschreven is een routeringstechniek die deels proactief is en deels

reactief een hybride techniek. Vandaar dat deze techniek onder naam zal vallen hybride lightweight mobile routing (H-LMR). In de huidige implementatie of die van de originele LMR wordt geen rekening gehouden met het kortste pad. Op het moment dat dit wel wenselijk wordt kunnen de heartbeat uit het netwerk gebruikt worden om het aantal hops te tellen die een bericht heeft ondergaan om zo te registreren hoever een node per punt verwijderd is. Een tegenzijde hiervan zou zijn dat er wel weer extra rekenkracht vereist wordt voor het zoeken van een pad.

## 5.2 Drone aansturing: Een stap terug voor netwerkherstel

Elke netwerkmodule zit aangesloten op een drone en heeft de mogelijkheid om deze aan te sturen. Alten wil gebruik maken van deze combinatie om zo het herstellend vermogen te vergroten van het netwerk. In de stuk wordt een uitwerking beschreven waar netwerkmodules zich een stap terug verplaatsen in het netwerk om zo een poging te doen een verloren verbinding te herstellen.

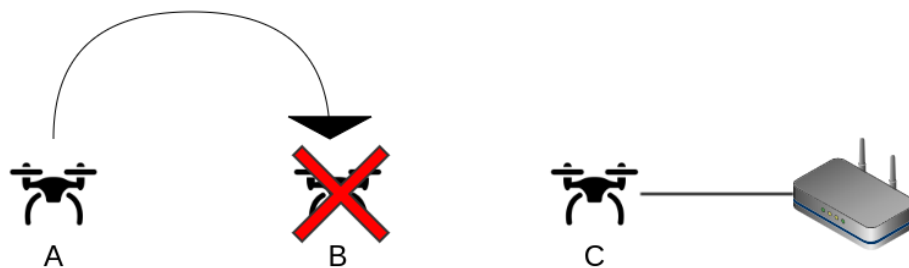
### 5.2.1 Een enkele drone die verloren is

[Figuur 5.4](#) laat een lus van drones zien die met elkaar verbonden zijn en op die wijze een netwerk verspreiden. Deze situatie wordt gebruikt om het gedrag te omschrijven die uitgevoerd moet worden door de drones op het moment dat er een verbinding verloren gaat.



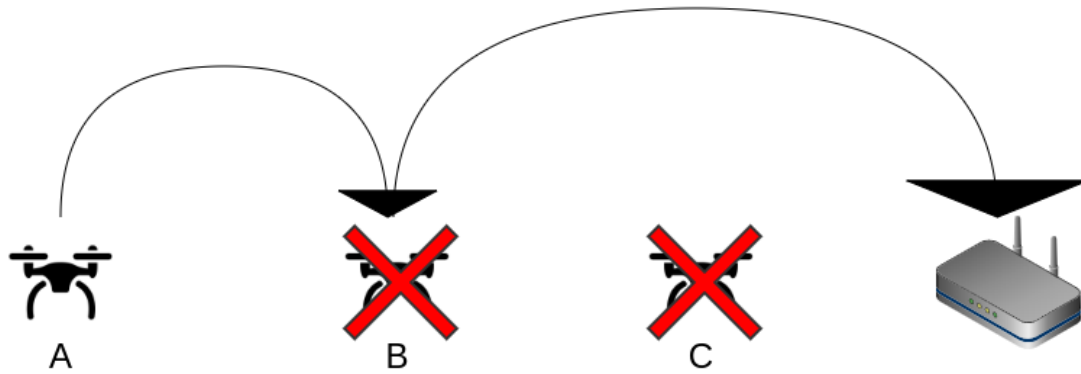
Figuur 5.4: Situatie waar drie drones met elkaar in verbinding staan

**Verlies van een enkel punt** Op het moment dat er een enkel punt wegvalt waardoor er een drone alleen overblijft moet hij zich verplaatsen naar het weggefallen punt. Dit wordt geïllustreerd in [Figuur 5.5](#) waar B uitvalt waardoor A zich zal herpositioneren op de positie van B.



Figuur 5.5: Situatie waar drone B is uitgevallen

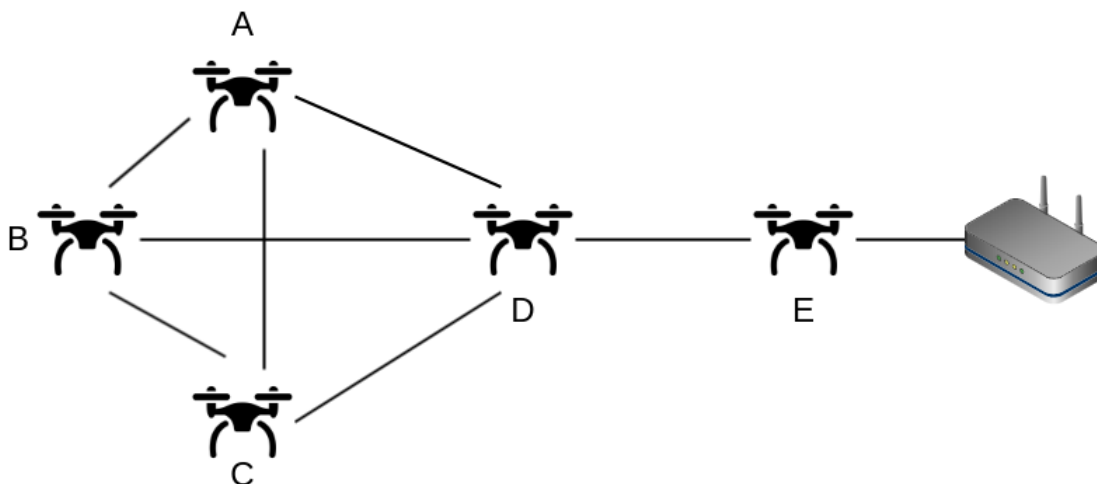
**Verlies van een twee punten** Wanneer er twee punten wegvallen zoals te zien in [Figuur 5.6](#) wordt er verwacht van de drone dat hij eerst naar punt B vliegt. Zodra de drone op punt B is gaat de netwerkmodule opnieuw zoeken naar een nieuwe verbinding. In deze situatie zal de netwerkmodule geen nieuw punt vinden waardoor de volgende stap is dat de drone terugkeert naar de gateway.



Figuur 5.6: Situatie waar drone B en C zijn uitgevallen

### 5.2.2 Een groep drones die verloren is

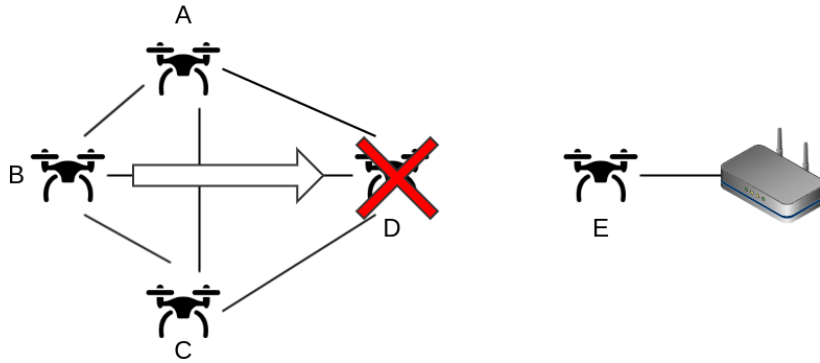
Wanneer er een groep verloren gaat komt er meer kijken bij herverdelen van de drones. Het zou namelijk geen zinloos zijn om alle drones te verplaatsen naar een stap terug in het netwerk op het moment dat deze plek hetzelfde is voor meerdere drones. Om de herverdeling die uitgevoerd wordt toe te lichten wordt er vanuit gegaan dat de drones verdeelt zijn als in [Figuur 5.7](#).



Figuur 5.7: Situatie waar drie drones met elkaar in verbinding staan

### **Uitval van een enkele drone die het directe aanknooppunt is voor de groep**

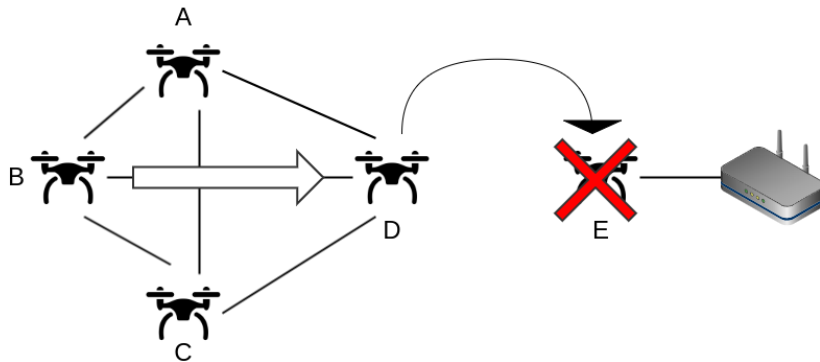
De eerste situatie die wordt toegelicht is wanneer er een drone uitvalt die de laatste drone is die de groep met de rest van het netwerk verbindt. Deze situatie wordt geschetst in [Figuur 5.8](#). In dit figuur valt punt D uit waardoor punt A, B, en C geen verbinding meer hebben. Vervolgens moeten deze drie punten met elkaar communiceren wie er gaat bewegen naar punt D. Zij zullen alle drie de afstand communiceren die zij hebben tot de locatie van de gateway. Vervolgens zal de drone die de grootste afstand heeft, in dit geval B, zich verplaatsen naar locatie D. Als de drone op locatie D staat kan hij een verbinding leggen met drone E waardoor A en C ook via hem weer verbinding krijgen.



Figuur 5.8: Situatie waar drone D is uitgevallen

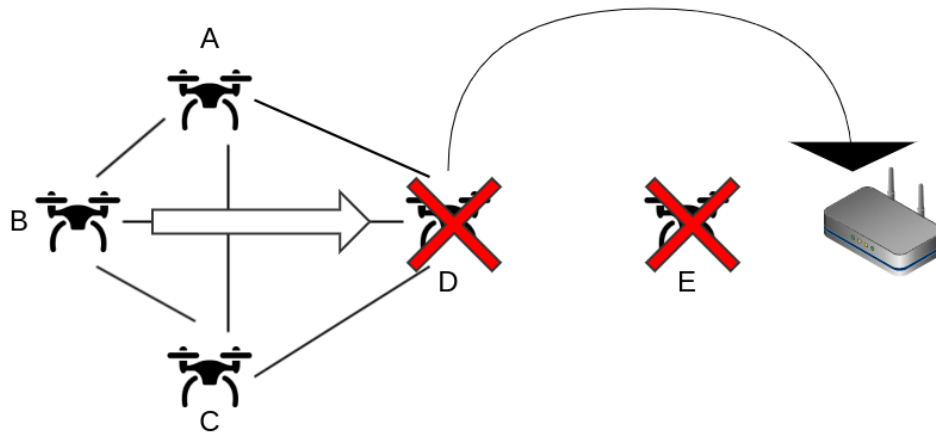
### **Uitval van een enkele drone die het indirecte aanknooppunt is voor de groep**

Als punt E uitvalt zoals te zien in [Figuur 5.9](#) zou alleen een berekening wie het verste van de gateway afstaat betekenen dat drone B zich zou verplaatsen naar de plek van drone D. Dit terwijl drone D nog gewoon werkt. Daarom zal er in de onderhandeling over wie zich moet verplaatsten worden meegenomen of er al een andere drone op de plek staat. Als dit zo is sluit de drone zichzelf buiten van de onderhandeling. In de situatie waar drone E uitvalt zal dit dus betekenen dat drone D zich naar de positie van E verplaatst. Vervolgens zullen de drones A-B-C alsnog geen verbinding hebben waardoor er opnieuw onderhandeld moet worden. Het resultaat moet dan zijn dat B zich verplaatst naar de oude positie van B.



Figuur 5.9: Situatie waar drone E is uitgevallen

**Uitval van twee drones die het aankooppunt zijn voor de groep** Op het moment dat allebei de punten D en E uitvallen zullen de drones uiteindelijk allemaal terugkeren naar de gateway. Het effect zal zijn dat punt A, B en C zich eerste naar punt D zullen verplaatsen en vervolgens terugkeren naar de gateway.



Figuur 5.10: Situatie waar drone B en C zijn uitgevallen

### 5.2.3 Conclusie

Het protocol om de drone een stap terug te laten zetten in het netwerk is een zwak protocol die wel gemakkelijk te implementeren is. Het is bedacht met het idee om te laten zien dat er potentie zit in het gebruiken van drones voor het bevorderen van het netwerken. De implementatie zal geen optimale verdeling bevorderen en geeft het netwerk alleen maar een kans om zich te herstellen op het moment dat er een enkele drone uitvalt.



## 6 — Experimenten

### 6.1 Test hybride lightweight mobile routing protocol

Om het aangepaste protocol van [Paragraaf 5.1](#) te testen zal er gebruik gemaakt worden van de simulatie. Uit het protocol kunnen de volgende stukken pseudo code geleerd worden.

**Ontvangen van een bericht** Met het H-LMR protocol leert een netwerkpunt van elk bericht die langskomt over andere zenders in het netwerk. Hierin maakt het niet uit of het bericht bestemd is voor het ontvangende punt of dat het doorgestuurd moet worden.

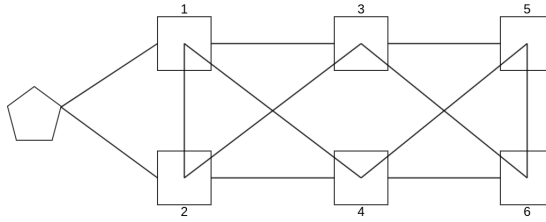
```
recieveMessage(message){
    message[destination] == this->address ? processMessage(message) :
        forwardMessage(message);

    add the sender of the message to the direct connections
    if(message[sender] != message[creator])
    {
        add the creator of the message to the indirect connection table of the
        sender
    }
}
```

**Doorsturen van een bericht** Het H-LMR stelt dat wanneer een bericht aankomt die doorgestuurd moet worden dat de ontvanger zelf zoekt wie de volgende stap neemt. Pas als er geen volgende stap mogelijk is mag de node een informatie bericht versturen over een verloren verbinding.

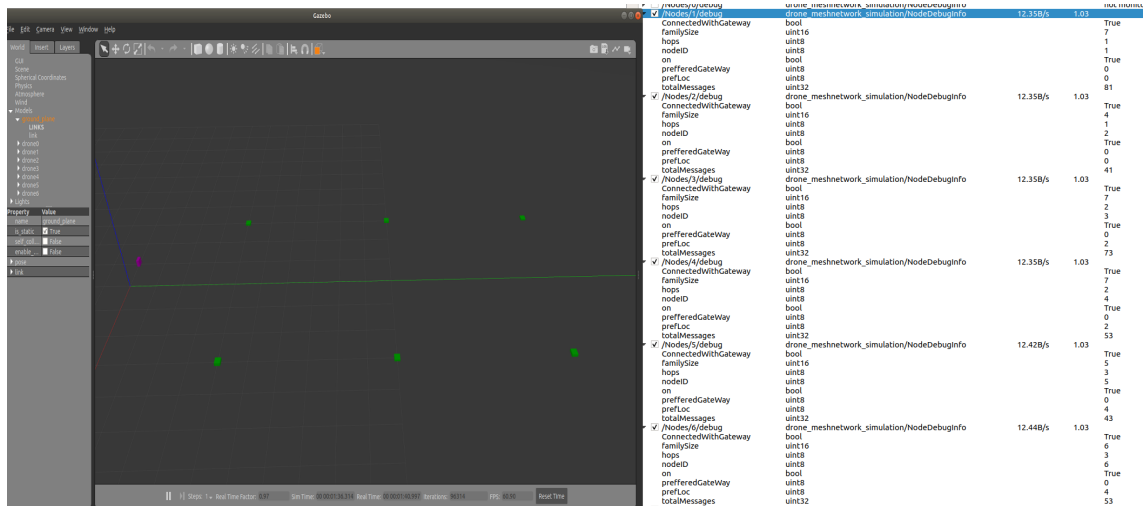
```
forwardMessage(message){
    int direction = getDirection(message[destination]);
    if (direction == NOTFOUND || direction == message[sender])
    {
        inform the sender of the message about a invalid path
        return;
    }
    else
    {
        Forward the message to the found direction
    }
}
```

**Test met zes routers en een gateway** Om het protocol te testen wordt de testopstelling gebruikt getoond in [Figuur 6.1](#).



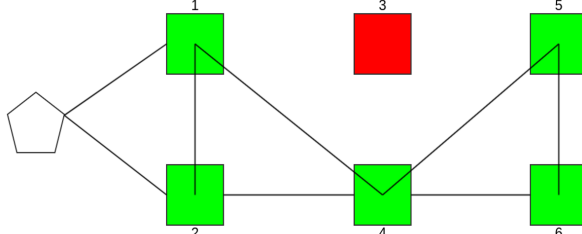
Figuur 6.1: Testopstelling met links een gateway waarna zes routers volgen

In deze opstelling is de begin positie het netwerk zoals die getoond wordt op de afbeelding. In de applicatie ziet dit er als volgt uit.



Figuur 6.2: Applicatie gazebo waar zes routers en gateway getoond wordt samen met debug informatie

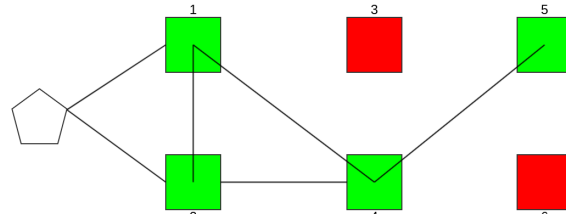
In [Figuur 6.2](#) is rechts debug informatie zichtbaar. De eerste waarde van elke router laat zien of verbinding is met een gateway. In het huidige geval heeft dus elke router verbinding met een gateway.



✓ /Nodes/1/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.428/s	1.03	not monitor
ConnectedWithGateway	bool			True
familySize	uint16			5
hops	uint8			0
nodeID	bool			True
on	bool			0
preferredGateway	uint8			0
prefLoc	uint8			0
totalMessages	uint32			240
✓ /Nodes/1/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.438/s	1.03	True
ConnectedWithGateway	bool			True
familySize	uint16			5
hops	uint8			1
nodeID	bool			1
on	bool			True
preferredGateway	uint8			2
prefLoc	uint8			0
totalMessages	uint32			291
✓ /Nodes/2/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.438/s	1.03	True
ConnectedWithGateway	bool			True
familySize	uint16			3
nodeID	uint8			2
on	bool			2
preferredGateway	uint8			0
prefLoc	uint8			0
totalMessages	uint32			130
✓ /Nodes/3/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.438/s	1.03	False
ConnectedWithGateway	bool			False
familySize	uint16			0
hops	uint8			2
nodeID	bool			False
on	bool			3
preferredGateway	uint8			0
prefLoc	uint8			0
totalMessages	uint32			216
✓ /Nodes/4/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.438/s	1.03	True
ConnectedWithGateway	bool			True
familySize	uint16			4
hops	uint8			2
nodeID	bool			2
on	bool			True
preferredGateway	uint8			0
prefLoc	uint8			2
totalMessages	uint32			194
✓ /Nodes/5/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.448/s	1.03	True
ConnectedWithGateway	bool			True
familySize	uint16			4
hops	uint8			3
nodeID	bool			5
on	bool			True
preferredGateway	uint8			0
prefLoc	uint8			4
totalMessages	uint32			123
✓ /Nodes/6/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.448/s	1.03	True
ConnectedWithGateway	bool			True
familySize	uint16			3
hops	uint8			6
nodeID	bool			True
on	bool			0
preferredGateway	uint8			4
prefLoc	uint8			128
totalMessages	uint32			

Figuur 6.3: Debug informatie van testopstelling waar punt drie is uitgezet

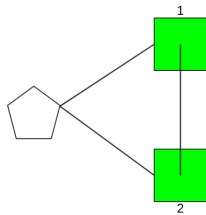
Op het moment dat punt 3 wordt uitgezet blijven punt 5 en 6 verbonden met de gateway zoals verwacht. In de debug informatie wordt ook een variabele getoond met de naam familySize. Deze variabele is een som van alle tabellen in een node. Wat opvalt na het uitzetten van punt 3 is dat punt 5 zijn familySize verkleint van 7 naar 4. Dit houdt dus in dat punt 5 drie routes had opgebouwd via punt 3 waarvan één route natuurlijk 5-3 was. De andere routes waren naar alle waarschijnlijkheid 5-3-1 en 5-3-0.



✓ /Nodes/1/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.448/s	1.04	True
ConnectedWithGateway	bool			True
familySize	uint16			4
hops	uint8			1
nodeID	bool			1
on	bool			True
preferredGateway	uint8			0
prefLoc	uint8			0
totalMessages	uint32			142
✓ /Nodes/2/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.448/s	1.04	True
ConnectedWithGateway	bool			True
familySize	uint16			3
hops	uint8			1
nodeID	bool			2
on	bool			True
preferredGateway	uint8			0
prefLoc	uint8			0
totalMessages	uint32			73
✓ /Nodes/3/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.448/s	1.04	False
ConnectedWithGateway	bool			False
familySize	uint16			0
hops	uint8			2
nodeID	bool			False
on	bool			0
preferredGateway	uint8			2
prefLoc	uint8			103
totalMessages	uint32			
✓ /Nodes/4/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.448/s	1.03	True
ConnectedWithGateway	bool			True
familySize	uint16			5
hops	uint8			2
nodeID	bool			4
on	bool			True
preferredGateway	uint8			0
prefLoc	uint8			2
totalMessages	uint32			103
✓ /Nodes/5/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.458/s	1.04	True
ConnectedWithGateway	bool			True
familySize	uint16			2
hops	uint8			3
nodeID	bool			5
on	bool			True
preferredGateway	uint8			0
prefLoc	uint8			4
totalMessages	uint32			63
✓ /Nodes/6/debug	drone_meshnetwork_simulation/NodeDebugInfo	12.468/s	1.04	False
ConnectedWithGateway	bool			False
familySize	uint16			0
hops	uint8			3
nodeID	bool			6
on	bool			False
preferredGateway	uint8			0
prefLoc	uint8			4
totalMessages	uint32			66

Figuur 6.4: Debug informatie van testopstelling waar punt drie en zes is uitgezet

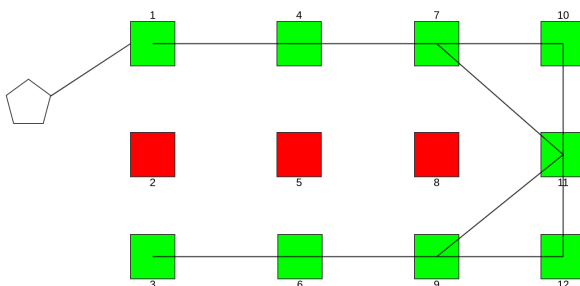
Figuur 6.4 laat zien hoe het netwerk zich gedraagt als punt zes ook wordt uitgezet. Zoals verwacht heeft punt 5 nog steeds een verbinding met een gateway. In de informatie is te zien dat punt 5 nog maar bewust is van twee routes. Er kan met zekerheid gezet worden dat dit route 5-4 en 5-4-0 zullen zijn.



✓ /Nodes/1/debug	drone_meshnetwork_simulation/NodeDebugInfo	unknown	0.00	True
ConnectedWithGateway	bool	unknown	0.00	True
familySize	uint16	unknown	0.00	2
hops	uint8	unknown	0.00	1
nodeID	uint8	unknown	0.00	1
on	bool	unknown	0.00	True
preferredGateway	uint8	unknown	0.00	0
prefLoc	uint8	unknown	0.00	0
totalMessages	uint32	unknown	0.00	163
✓ /Nodes/2/debug	drone_meshnetwork_simulation/NodeDebugInfo	unknown	0.00	True
ConnectedWithGateway	bool	unknown	0.00	True
familySize	uint16	unknown	0.00	2
hops	uint8	unknown	0.00	1
nodeID	uint8	unknown	0.00	2
on	bool	unknown	0.00	True
preferredGateway	uint8	unknown	0.00	0
prefLoc	uint8	unknown	0.00	0
totalMessages	uint32	unknown	0.00	77
✓ /Nodes/3/debug	drone_meshnetwork_simulation/NodeDebugInfo	unknown	0.00	False
ConnectedWithGateway	bool	unknown	0.00	False
familySize	uint16	unknown	0.00	0
hops	uint8	unknown	0.00	2
nodeID	uint8	unknown	0.00	3
on	bool	unknown	0.00	False
preferredGateway	uint8	unknown	0.00	0
prefLoc	uint8	unknown	0.00	2
totalMessages	uint32	unknown	0.00	105
✓ /Nodes/4/debug	drone_meshnetwork_simulation/NodeDebugInfo	unknown	0.00	False
ConnectedWithGateway	bool	unknown	0.00	False
familySize	uint16	unknown	0.00	0
hops	uint8	unknown	0.00	2
nodeID	uint8	unknown	0.00	4
on	bool	unknown	0.00	False
preferredGateway	uint8	unknown	0.00	0
prefLoc	uint8	unknown	0.00	2
totalMessages	uint32	unknown	0.00	106
✓ /Nodes/5/debug	drone_meshnetwork_simulation/NodeDebugInfo	unknown	0.00	False
ConnectedWithGateway	bool	unknown	0.00	False
familySize	uint16	unknown	0.00	0
hops	uint8	unknown	0.00	3
nodeID	uint8	unknown	0.00	5
on	bool	unknown	0.00	True
preferredGateway	uint8	unknown	0.00	0
prefLoc	uint8	unknown	0.00	4
totalMessages	uint32	unknown	0.00	67
✓ /Nodes/6/debug	drone_meshnetwork_simulation/NodeDebugInfo	unknown	0.00	False
ConnectedWithGateway	bool	unknown	0.00	False
familySize	uint16	unknown	0.00	0
hops	uint8	unknown	0.00	3
nodeID	uint8	unknown	0.00	6
on	bool	unknown	0.00	False
preferredGateway	uint8	unknown	0.00	0
prefLoc	uint8	unknown	0.00	4
totalMessages	uint32	unknown	0.00	66

Figuur 6.5: Debug informatie van testopstelling waar punt drie, vier en zes is uitgezet

In deze opstelling zijn punten 3, 4 en 6 uitgezet. In de debug informatie is te zien dat punt 5 zijn verbinding verloren is. Zowel punt 1 en punt 2 geven aan twee routes te kennen. Dit is zoals verwacht.



✓ /Nodes/1/debug	drone_meshnetwork_simulation/NodeDebugInfo	unknown	0.00	True
ConnectedWithGateway	bool	unknown	0.00	True
familySize	uint16	unknown	0.00	10
hops	uint8	unknown	0.00	1
nodeID	uint8	unknown	0.00	1
on	bool	unknown	0.00	True
preferredGateway	uint8	unknown	0.00	0
prefLoc	uint8	unknown	0.00	0
totalMessages	uint32	unknown	0.00	1738
✓ /Nodes/2/debug	drone_meshnetwork_simulation/NodeDebugInfo	unknown	0.00	False
ConnectedWithGateway	bool	unknown	0.00	False
familySize	uint16	unknown	0.00	0
hops	uint8	unknown	0.00	2
nodeID	uint8	unknown	0.00	1
on	bool	unknown	0.00	False
preferredGateway	uint8	unknown	0.00	0
prefLoc	uint8	unknown	0.00	0
totalMessages	uint32	unknown	0.00	819
✓ /Nodes/3/debug	drone_meshnetwork_simulation/NodeDebugInfo	unknown	0.00	True
ConnectedWithGateway	bool	unknown	0.00	True
familySize	uint16	unknown	0.00	2
hops	uint8	unknown	0.00	7
nodeID	uint8	unknown	0.00	3
on	bool	unknown	0.00	True
preferredGateway	uint8	unknown	0.00	0
prefLoc	uint8	unknown	0.00	5
totalMessages	uint32	unknown	0.00	903
✓ /Nodes/4/debug	drone_meshnetwork_simulation/NodeDebugInfo	unknown	0.00	False
ConnectedWithGateway	bool	unknown	0.00	False
familySize	uint16	unknown	0.00	0
hops	uint8	unknown	0.00	2
nodeID	uint8	unknown	0.00	5
on	bool	unknown	0.00	False
preferredGateway	uint8	unknown	0.00	0
prefLoc	uint8	unknown	0.00	1
totalMessages	uint32	unknown	0.00	1145
✓ /Nodes/5/debug	drone_meshnetwork_simulation/NodeDebugInfo	unknown	0.00	False
ConnectedWithGateway	bool	unknown	0.00	False
familySize	uint16	unknown	0.00	0
hops	uint8	unknown	0.00	3
nodeID	uint8	unknown	0.00	8
on	bool	unknown	0.00	False
preferredGateway	uint8	unknown	0.00	4
prefLoc	uint8	unknown	0.00	4
totalMessages	uint32	unknown	0.00	1127

Figuur 6.6: Debug informatie van testopstelling met 12 routers waar punt twee, vijf en acht is uitgezet

Als laatste opstelling is een opstelling van twaalf routers gebruikt. Hier zijn routers 2, 5, en 8 uitgezet om router nummer 3 te forceren via een grote omweg te communiceren naar de gateway. In de debug informatie is te zien dat hij hierin geslaagd is en dat er 7 hops nodig zijn voor de gateway om een route naar de router te vinden. Dit betekent dat er de volgende route wordt gebruikt 0-1-4-7-11-9-6-3.

**Conclusie** Het experiment waarin de aangepaste versie van LMR wordt getest bewijst dat de techniek bruikbaar is voor het dronenetwerk. Er zijn genoeg scenario's getest om aan te tonen dat het netwerk zichzelf kan herstellen wanneer er een verbinding verloren is gegaan.

- 6.2 Is het mogelijk voor netwerkmodule om een netwerk te herstellen door het de mogelijkheid te geven zich vrij te bewegen?
- 6.3 Hoe gedraagt een fysieke node uit het mesh netwerk zich bij slecht of geen bereik?

## 7 — Conclusie

In dit hoofdstuk wordt een conclusie getrokken op de vraag: Hoe kunnen een netwerkmodules een groep van ongeveer 100 drones voorzien van een onderling meshnetwerk die snel reageert op uitval van netwerkpunten?

Dit wordt vervolgt met een advies die voortvloeit uit de kennis van dit verslag.

### 7.1 Antwoord op de hoofdvraag

Uit de experimenten is gebleken dat door het toevoegen van een mircobord die zelfstandig kan opereren aan drone een meshnetwerk opgezet kan worden van ongeveer 100 drones. Als microbord is voor de experimenten een arduino uno gebruikt dit als mircocontroller een Atmel328 gebruikt. De microcontroller van het mircobord moet toegang hebben tot een antenne voor onderlinge communicatie waarbij het advies is om gebruik te maken van een [nRF24l01+](#) vanwege zijn bereik en snelheid. Om het netwerk te kunnen opbouwen en onderhouden wordt een aangepaste variant van de reactieve routingstechniek LMR toegepast. Belangrijk bij het gebruiken van deze techniek is de mogelijkheid voor de individuele netwerkpunten dat zij in staat zijn te kunnen verifiëren dat een verbinding bestaat. De routingstechniek is aangepast zodat deze zich enigszins proactief gedraagt door het versturen van heartbeats naar de gateway.

### 7.2 Advies

Het advies uit dit onderzoek luidt als volgt:

Op het moment dat Alten besluit om drones in te gaan zetten voor het verspreiden van netwerken zal zij zich bewust moeten zijn van haar plan voor dit netwerk. Het aangepast LMR protocol is geschikt voor een netwerk waarbij de drones zich verspreiden over een gebied en zich maar incidenteel verplaatsen naar een ander punt. Dit is zo gemaakt omdat dit zo aangegeven is door Alten. Op het moment dat Alten besluit dat de drone een andere gedrag moeten krijgen zoals bijvoorbeeld swarming dan is het protocol ongeschikt en is het advies om te kijken naar een reactief protocol die altijd nieuwe paden berekend. Hetzelfde geldt als Alten besluit dat de drones altijd op een zelfde positie zullen staan, dan zal een proactief protocol een beter alternatief zijn om zo berichten over het kortste pad te kunnen sturen.

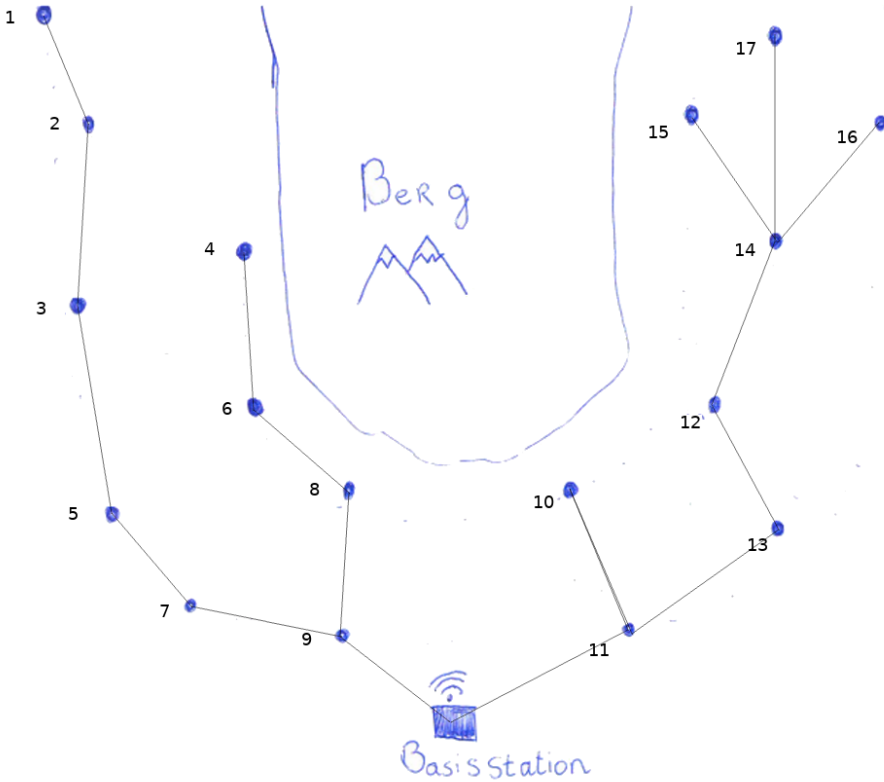
## Literatuur

- Abdulghani, S. & Salleh, M. F. M. (2013, 09). Overview of wireless mesh networks. *Journal of Communications*, 8, 586-599. Op 19 maart 2019 verkregen van [https://www.researchgate.net/publication/275594880\\_Overview\\_of\\_Wireless\\_Mesh\\_Networks](https://www.researchgate.net/publication/275594880_Overview_of_Wireless_Mesh_Networks) doi: 10.12720/jcm.8.9.586-599
- Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melià-Seguí, J. & Watteyne, T. (2017, Juni). Understanding the limits of lorawan. *IEEE Communications Magazine*, 55. Op 13 maart 2019 verkregen van [https://www.researchgate.net/publication/312627168\\_Understanding\\_the\\_limits\\_of\\_LoRaWAN](https://www.researchgate.net/publication/312627168_Understanding_the_limits_of_LoRaWAN) doi: 10.1109/MCOM.2017.1600613
- Castillo, D. C., Rosales, J. & Blanco, G. A. T. (2018, 03). Optimizing binary serialization with an independent data definition format. *International Journal of Computer Applications*, 180, 15-18. Op 1 maart 2019 verkregen van [https://www.researchgate.net/publication/323893287\\_Optimizing\\_Binary\\_Serialization\\_with\\_an\\_Independent\\_Data\\_Definition\\_Format](https://www.researchgate.net/publication/323893287_Optimizing_Binary_Serialization_with_an_Independent_Data_Definition_Format) doi: 10.5120/ijca2018916670
- iforce2d. (2014, 7 dec). *nrf24l01 range test (arduino)*. [videobestand]. Op 18 maart 2019 verkregen van <https://www.youtube.com/watch?v=1R60toEjHl8>
- Kulhavy, D. L., Hung, I.-K., Unger, D. R. & Zhang, Y. (2017, 19 april). *Feature and performance comparison of the v-rep, gazebo and argos robot simulators*. Nacogdoches, Verenigde Staten: Arthur Temple College of Forestry and Agriculture. Op 11 maart 2019 verkregen van <http://lenkaspace.net/tutorials/programming/robotSimulatorsComparison>
- Lee, H.-C. & Ke, K.-H. (2018, 03). Monitoring of large-area iot sensors using a lora wireless mesh network system: Design and evaluation. *IEEE Transactions on Instrumentation and Measurement*, PP, 1-11. Op 13 maart 2019 verkregen van <http://mx.nthu.edu.tw/~huclee/paper/Monitoring%20of%20Large-Area%20IoT%20Sensors%20using%20a%20LoRa%20Wireless%20Mesh%20Network%20System%20Design%20and%20Evaluation.pdf> doi: 10.1109/TIM.2018.2814082
- Nordic Semiconductor. (2008, september). *nrf24l01+ single chip 2.4ghz transceiver product specification v1.0* (1e dr.). Trondheim, Noorwegen. Op 14 maart 2019 verkregen van [https://www.nordicsemi.com/-/media/DocLib/Other/Product\\_Spec/nRF24L01PPSv10.pdf](https://www.nordicsemi.com/-/media/DocLib/Other/Product_Spec/nRF24L01PPSv10.pdf)
- Oates, B. J. (2005, november). *Researching information systems and computing*. Sage Publications Inc.
- Pitonakova, L., Giuliani, M., Pipe, A. & Winfield, A. (2018, januari). Feature and performance comparison of the v-rep, gazebo and argos robot simulators. *Proceedings of the 19th Towards Autonomous Robotic Systems Conference (TAROS 2018)*. Op 19 februari 2019 verkregen van <http://lenkaspace.net/tutorials/programming/robotSimulatorsComparison>

- Saha, H., Mandal, S., Mitra, S., Banerjee, S. & Saha, U. (2017, 07). Comparative performance analysis between nrf24l01+ and xbee zb module based wireless ad-hoc networks. *International Journal of Computer Network and Information Security*, 9, 36-44. Op 19 maart 2019 verkregen van [https://www.researchgate.net/publication/319421687\\_Comparative\\_Performance\\_Analysis\\_between\\_nRF24L01\\_and\\_XBEE\\_ZB\\_Module-Based\\_Wireless\\_Ad-hoc\\_Networks](https://www.researchgate.net/publication/319421687_Comparative_Performance_Analysis_between_nRF24L01_and_XBEE_ZB_Module-Based_Wireless_Ad-hoc_Networks) doi: 10.5815/ijcnis.2017.07.05
- Semtech. (2015, mei). *Lora modulation basics* (2e dr.). Op 13 maart 2019 verkregen van <https://www.semtech.com/uploads/documents/an1200.22.pdf>
- Tauro, C., Ganesan, N., Mishra, S. & Bhagwat, A. (2012, 05). Article: Object serialization: A study of techniques of implementing binary serialization in c++, java and .net. *International Journal of Computer Applications*, 45, 25-29. Op 1 maart 2019 verkregen van [https://www.researchgate.net/publication/235719668\\_Article\\_Object\\_Serialization\\_A\\_Study\\_of\\_Techniques\\_of\\_Implementing\\_Binary\\_Serialization\\_in\\_C\\_Java\\_and\\_NET](https://www.researchgate.net/publication/235719668_Article_Object_Serialization_A_Study_of_Techniques_of_Implementing_Binary_Serialization_in_C_Java_and_NET)
- Telkamp, T. & Slats, L. (2017, september). *Ground breaking world record! lorawan packet received at 702 km (436 miles) distance*. Op 14 maart 2019 verkregen van <https://www.thethingsnetwork.org/article/ground-breaking-world-record-lorawan-packet-received-at-702-km-436-miles-distance>
- Van Turnhout, K., Craenmehr, S., Holwerda, R., Menijn, M., Zwart, J. & Bakker, R. (2014, september). De informatieprofessional 3.0. In (p. 163-174). Den Haag, Nederland: Academic Service BV. Op 11 februari 2019 verkregen van [https://www.researchgate.net/publication/272421605\\_De\\_methodenkaart\\_praktijkonderzoek](https://www.researchgate.net/publication/272421605_De_methodenkaart_praktijkonderzoek)
- Van Turnhout, K., Mulholland, C., Kamp, I., Jacobs, M., Neumann, A., Rouwhorst, S. & Van der Vlies, L. (z. j.). Cmd methods pack: Find a combination of research methods that suit your needs. Nederland: HAN University of Applied Sciences - Amsterdam University of Applied Sciences. Verkregen van <http://cmdmethods.nl>
- U.S. Government. (2008, september). *Global positioning system standard positioning service performance standard* (4e dr.). Arlington, Verenigde Staten: The Office of the Secretary of Defense. Op 11 maart 2019 verkregen van <https://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf>
- Vainio, M. (z. j.). *Link reversal routing*. Op 21 maart 2019 verkregen van <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=37CA0D61991EAF1DE9E4048BB2162C66?doi=10.1.1.19.1663&rep=rep1&type=pdf>
- Vangelista, L. & Centenaro, M. (2019, maart). Worldwide connectivity for the internet of things through lorawan. *Future Internet*, 11, 57. Op 13 maart 2019 verkregen van [https://www.researchgate.net/publication/331493728\\_Worldwide\\_Connectivity\\_for\\_the\\_Internet\\_of\\_Things\\_Through\\_LoRaWAN](https://www.researchgate.net/publication/331493728_Worldwide_Connectivity_for_the_Internet_of_Things_Through_LoRaWAN) doi: 10.3390/fi11030057



## A



Figuur A.1: Schets van casus netwerk