



Faculteit Ingenieurswetenschappen  
Vakgroep Informatietechnologie  
Voorzitter: Prof. Dr. Ir. P. LAGASSE

# **Breedbandtoegangsnetwerk voor treinen door middel van Radio-over-Fiber**

door

Peter DEDECKER

Promotoren: Prof. Dr. Ir. I. MOERMAN, Dr. Ir. D. COLLE  
Scriptiebegeleiders: Ir. B. LANNOO, Lic. B. JOORIS, Ir. T. VAN LEEUWEN

Scriptie ingediend tot het behalen van de academische graad van  
BURGERLIJK INGENIEUR IN DE COMPUTERWETENSCHAPPEN  
OPTIE INFORMATIE- EN COMMUNICATIETECHNOLOGIE

Academiejaar 2005–2006

# Voorwoord

*De dag van vandaag is het belang van digitale communicatie toegenomen tot ongekende hoogten. We willen een alsmaar grotere bandbreedte en tegelijk een alsmaar grotere bewegingsvrijheid. Weinig breedbandgebruikers wensen nog herinnerd te worden aan de tijden van de modem, aangesloten op de telefoonlijn. Weinigen konden zich toen een wereld zoals nu voorstellen, waarbij mensen met een draagbare pc tot in de tuin toegang hebben tot het wereldwijde netwerk aan een snelheid die vele malen hoger ligt dan wat toen gangbaar was. De mens is echter gemaakt om grenzen te verleggen, en zal dat ook doen. We willen nu eenmaal steeds meer en er wordt ook steeds meer mogelijk. De volgende stap is het aanbieden van internet op de trein aan een redelijke snelheid. De mens is mobieler dan ooit, maar verliest ook niet graag veel tijd. Onderweg, tijdens een treinrit, toegang hebben tot het internet, tot het interne netwerk van je bedrijf, je mailbox,... Velen zouden het maar al te graag werkelijkheid zien worden. Ik ben een van hen. En ik zou enorm tevreden zijn mocht ik dat ook maar een miniem stapje dichterbij kunnen brengen hebben, vandaar mijn keuze voor dit onderwerp.*

*Dit werk is er niet alleen gekomen door mijn interesse in de materie, er zijn nog tal van mensen die, elk op hun manier, hun steentje bijgedragen hebben om dit werk te maken tot wat het nu is. Mensen, die mij in de loop der jaren gesteund hebben en mij gemaakt hebben tot wie ik nu ben.*

*Ik zou dan ook graag in de eerste plaats mijn thesisbegeleiders, Bart, Bart en Tom, willen bedanken voor de grote hoeveelheid tijd en energie die ze in mij en dit werk stopten, voor hun geduld met mij wanneer ik weer eens af kwam met de onnozelseste vragen als ik dacht dat het probleem weeral aan de software lag. Tevens wens ik ook mijn promotoren te bedanken voor hun vertrouwen en de geboden kans. Ook de taaljury verdient dank en respect voor het in de gaten houden van de gebruikte spelling en grammatica, consistentie,... Stefanie, Lynn, Jonas, Dries, pa: bedankt voor de muggenzifterij!*

*Mensen voor wie geen dank te veel kan zijn, zijn mijn ouders. Zij hebben mij in de loop der jaren alle kansen gegeven die ik maar wou en mij ten volle gesteund in mijn keuzes. Zij hebben mij de mogelijkheden gegeven om te staan waar ik nu sta, om mij te verdiepen in mijn interesses, mij te ontplooien, en uiteraard ook om enkele jaren goed te kunnen genieten in Gent.*

*Wie zeker aandacht verdient in dit stukje, is mijn vriendin, Lynn, die ook steeds klaarstond als ik haar nodig had, en die mij ondertussen toch ook weer enkele jaren een goed gevoel bezorgt.*

*Verder wil ik nog een aantal mensen bedanken die hier niet bij naam vermeld zijn maar toch op een of andere manier voor mij een uitlaatklep geweest zijn, mij gesteund hebben, en mij mee gemaakt hebben tot de mens die ik nu ben. Bedankt.*

*Peter Dedecker, juni 2006*

# Toelating tot bruikleen

“De auteur geeft de toelating deze scriptie voor consultatie beschikbaar te stellen en delen van de scriptie te kopiëren voor persoonlijk gebruik.

Elk ander gebruik valt onder de beperkingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit deze scriptie.”

Peter Dedecker, juni 2006

# Breedbandtoegangsnetwerk voor treinen door middel van Radio-over-Fiber

door

Peter DEDECKER

Scriptie ingediend tot het behalen van de academische graad van

BURGERLIJK INGENIEUR IN DE COMPUTERWETENSCHAPPEN:

OPTIE INFORMATIE- EN COMMUNICATIETECHNOLOGIE

Academiejaar 2005–2006

Promotoren: Prof. Dr. Ir. I. MOERMAN, Dr. Ir. D. COLLE

Scriptiebegeleiders: Ir. B. LANNOO, Lic. B. JOORIS, Ir. T. VAN LEEUWEN

Faculteit Ingenieurswetenschappen

Universiteit Gent

Vakgroep Informatietechnologie

Voorzitter: Prof. Dr. Ir. P. LAGASSE

## Samenvatting

In dit werk proberen we de traditionele trade off van bandbreedte en bewegingssnelheid te doorbreken. Een grote bandbreedte moet ook mogelijk zijn aan een hoge bewegingssnelheid, meer bepaald in (hogesnelheids)treinen. Door de introductie van een model op basis van Radio-over-Fiber en het *moving cell* concept, proberen we dit doel te bereiken. Dit model wordt gemapped op een simulatiemodel op basis van ethernet en standaard IEEE 802.11, dat uitvoerig getest wordt met behulp van simulaties in ns-click.

## Trefwoorden

Radio-over-Fiber, moving cell, netwerk, bewegingssnelheid, draadloze technologie

# Broadband access network for trains based on Radio-over-Fiber

Peter Dedecker

Supervisor(s): Ingrid Moerman, Didier Colle, Bart Lannoo, Bart Jooris, Tom Van Leeuwen

**Abstract**—This article tries to extend the limits of the trade off between bandwidth and velocity by proposing a new model for trains based on Radio-over-Fiber and the *moving cell* concept. This model has been tested by simulations with currently available technology.

**Keywords**—Radio-over-Fiber, moving cell, access network, fast moving users, wireless technologies

## I. INTRODUCTION

THE well known problem with broadband access networks is their trade off between bandwidth and velocity. While a WiFi-network can support up to 54 Mbps, you have to stay in the close neighborhood of the access point. Network technologies that allow you to move, like GSM, only support 9600 bps, or 170 kbps when using GPRS. Their problem is that small cells are needed to provide a high bandwidth, but with small cells you have to change to another cell very frequently, which is called the handover problem.

This problem typically arises on trains: one wants to have access to a broadband internet connection at a travelling speed up to 300 km/h for high speed trains. Today, this is possible when using a satellite connection, but there are lots of drawbacks: a long round trip time, need for a line of sight connection and a high price for a connection of e.g. only 2-4 Mbps.

In this abstract, we will first propose a model based on antennas near the railway and solve the handover problem. This model will be tested by simulations as proof of concept and compared to traditional communication systems.

## II. MOVING CELL BY RADIO-OVER-FIBER: CONCEPT

### A. Radio-over-Fiber

This model was proposed earlier in [1]. We divide the track of the train in cells. In each cell, base stations are placed near the railway and connected with a central node by an optical fiber. Each base station has a direct line to the central node by its own optical wavelength. In such an optical ring the central node needs to know where the train actually is so it can send packets to the right base station by modulating it on the right wavelength.

To reduce the cost of those base stations, they need to be really simple. This can be achieved by using *Radio-over-Fiber*: the base stations are just dumb antennas, capture a whole frequency band and place it integrally and immediately on the optical wavelength. The actual decoding takes place in the central node. Such antennas are called *Remote Antenna Units* (RAU's) in RoF-terminology. Those wireless signals can be decoded by the central node when the attenuation is low enough. This is normally a fact if the length of the ring is held under 10 km.

### B. Moving Cell

Normal handovers follow a fixed scenario by searching for an antenna at different frequencies. In most wireless network standards a handover needs also to be completed by exchanging management frames. After this, packets located in buffers at the previous antennas need to be rerouted which can take some time and obstructs fast handovers. We can eliminate this if we define fixed frequencies for all receivers.

This can be done by generating all wireless packets in the central node. Those packets are then placed on the right optical wavelength so they are sent out by the right RAU on the right frequency. When the train comes in the neighborhood of another RAU the central has to send those downlink packets to that new RAU.

This way, the train can keep its frequency the whole track in the same optical ring and no handovers are needed anymore. This principle, by holding the same frequency and keeping the train connected to the same logical access point in the central node, is called *moving cell*.

For simplicity reasons, all trains in the optical ring have their own dedicated frequency.

### C. Traffic Routing

To get downstream packets at the right RAU, the central node has to keep track of the train location. This can easily be done by monitoring upstream packets. The RAU that picks up the upstream packets coming from the train are likely the ones closest to the train. When the train moves and the signals are picked up by a new RAU, this RAU will soon be the only RAU that communicates with the train. The central node can then switch to the new RAU almost as soon as when the first packet reaches the central node by this new RAU. Of course the central node needs to remember the previous RAU in order to avoid switching back again in the overlap zone.

While all trains have their own dedicated channel, it is possible to support multiple trains or crossing trains at the same RAU's. It is even possible to place more than one sender/receiver on the train, each with their own channel, to increase bandwidth.

## III. SIMULATION MODEL

The operation of this model has to be demonstrated by simulations, however this is not evident with new technology. So we have to map it to currently available technology.

First of all, as wireless standard we choose IEEE 802.11, commonly known as WiFi, which is a cheap and easily available technology nowadays. To approach the concept of capturing a

whole frequency band, we install a number of wireless network cards in the nodes that act as RAU so we can capture a small selection of that frequency band. There is one WiFi card needed in each RAU for each train/channel we want to support.

As an optical network is not available, we have to fall back on traditional ethernet. By connecting the central node and the RAU's with a hub or switch and give all nodes a dedicated MAC-address, we can simulate the optical fiber where the central node can send packets to each RAU separately. To model the RoF, the RAU's encapsulate the whole captured packet including extra information like signal/noise ratio and received power into a new ethernet packet and send it to the central. The RAU's do not do anything with these packets except forwarding them, so they really act like dumb antennas.

#### IV. THE ACTUAL SIMULATIONS AND RESULTS

For the simulations themselves, we have chosen ns-click [2]. This way, we can run click-scripts [3] on nodes created and simulated by the ns Network Simulator [4].

##### A. Reference

Of course, we want to compare our system to the classical ones, so we need a reference model. We can compare our system to a hardware test as well as to a simulation test with the same ns-simulator. Results for the ns-one are shown in figure 1. Notice the gap where no communication is possible.

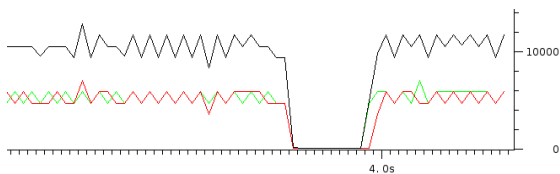


Fig. 1. Detailed capture of the stream at the moment of a handover between two simulated AP's with a strong signal. Notice the gap of 100 ms.

##### B. 1 train, 2 remote antenna units

A first test scenario is created by placing two RAU's at a distance of 200 m from each other. Each of them has a coverage range of 105 m so there is an overlap zone of 10 m. A train passes 10 m behind the two RAU's at a speed of 40 m/s ( $\dot{1}44$  km/h). A UDP test stream is sent in both directions.

Results are clear: the throughput of the UDP test stream is shown in figure 2 while the network traffic between the central node and the RAU's is shown in figure 3. It is clear the network connection is not broken anymore when changing to a new antenna. In figure 3 we can also clearly see the overlap zone of the two RAU's as the section where the blue and purple lines are not zero in together. We have to clarify that the red and green lines are straight vertical lines, but they are shown smooth because of the stretched out figure. You can also clearly see the traffic from the central node to the first RAU stops as soon as the first packet from the new RAU is received.

##### C. 2 trains, 3 remote antenna units

A second test is created by placing three RAU's at a distance of 200 m from each other and use two trains. The first one

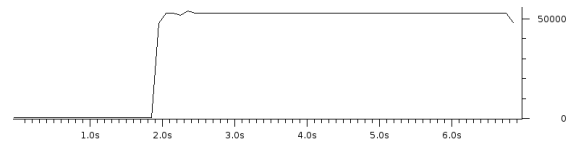


Fig. 2. Throughput of the UDP test stream in the RoF-model. The network connection is not broken anymore as there is no gap.

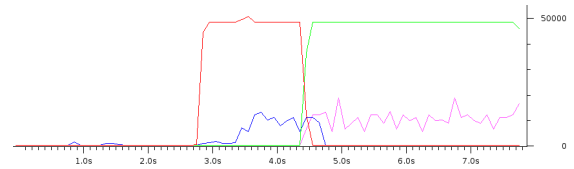


Fig. 3. Network streams in the RoF-model. Red and green lines show the traffic from the central node to RAU1 and RAU2, blue and purple show the traffic from RAU1 and RAU2 to the central node.

starts at RAU1 and drives towards RAU3 while the second one makes the opposite movement. A UDP throughput test delivers a graphs like figure 2. A more interesting result however is shown in figure 4. One can clearly see the two black peaks where both trains are in an overlap zone and their traffic is captured by two RAU's and sent to the central node together. The other graphs look the same as in the previous one but are slightly different when both trains are in the area of RAU2 where the traffic is twice as high.

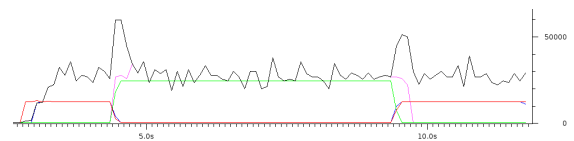


Fig. 4. Network streams in the RoF-model with two trains and three RAU's. Red, green and blue lines show the traffic from the central node to RAU1, RAU2 and RAU3. The purple one shows the traffic from RAU2 to the central node while the black line shows the total stream to the central node.

#### V. CONCLUSION

The simulation results show a nice advantage for the moving cell concept. The traditional handover problem can be avoided so a workable model for broadband access on trains seems realistic. But there needs to be done a lot of research to make RAU's and RoF as reliable and cheap as possible.

#### REFERENCES

- [1] Bart Lannoo, Didier Colle, Mario Pickavet, Piet Demeester, *Optical Switching Architecture to Implement Moveable Cells in a Multimedia Train Environment*, Proc. of ECOC 2004, 30th European Conf. on Optical Communication, vol. 3, pp. 344-345, Stockholm, Sweden, 5-9 Sep. 2004.
- [2] Michael Neufeld, Ashish Jain, Dirk Grunwald, *Nsclick:: bridging network simulation and deployment*, <http://systems.cs.colorado.edu/Networking/nsclick/>
- [3] *The Click Modular Router Project*, <http://www.read.cs.ucla.edu/click/>
- [4] *NS - Network Simulator*, <http://nsnam.isi.edu/nsnam/>

# Inhoudsopgave

Voorwoord	i
Toelating tot bruikleen	iii
Overzicht	iv
Extended abstract	v
Inhoudsopgave	vii
Gebruikte afkortingen	ix
<b>1 Inleiding</b>	<b>1</b>
1.1 Satellietverbinding . . . . .	3
1.2 Trackside oplossingen . . . . .	5
1.2.1 WiFi . . . . .	5
1.2.2 UMTS . . . . .	6
1.2.3 WiMax . . . . .	6
1.2.4 Andere . . . . .	7
1.3 Naar een verbeterde trackside oplossing . . . . .	7
<b>2 Radio-over-Fiber langs de spoorlijn: werking</b>	<b>8</b>
2.1 Radio-over-Fiber . . . . .	8
2.2 Moving cell . . . . .	9
2.3 Werking van het systeem . . . . .	11
2.3.1 Uplinkverkeer . . . . .	11
2.3.2 Downlinkverkeer . . . . .	12
2.4 Invloed op de netwerkeigenschappen . . . . .	13
<b>3 Simulatiemodel</b>	<b>17</b>
3.1 Wireless wordt WiFi . . . . .	17
3.2 Optisch wordt ethernet . . . . .	18



---

3.3	Beweging . . . . .	20
<b>4</b>	<b>Implementatie van het simulatiemodel</b>	<b>22</b>
4.1	Wat is ns-click? . . . . .	22
4.1.1	Click . . . . .	22
4.1.2	nsnam . . . . .	24
4.1.3	ns-click . . . . .	25
4.2	Implementatie . . . . .	27
4.2.1	De server . . . . .	27
4.2.2	De trein . . . . .	30
4.2.3	De Remote Antenna Units . . . . .	30
4.2.4	De centrale . . . . .	30
<b>5</b>	<b>Simulatieresultaten</b>	<b>38</b>
5.1	Wijzigingen aan de implementatie . . . . .	38
5.1.1	Het kickstartprobleem . . . . .	38
5.1.2	Verwisselen AP- en station-functionaliteit . . . . .	38
5.2	Referentie . . . . .	39
5.2.1	Hardwaretest . . . . .	40
5.2.2	Simulatie met conventionele AP's in ns-click . . . . .	41
5.3	Resultaten van het systeem op basis van Radio-over-Fiber . . . . .	43
5.3.1	1 trein, 2 remote antenna units . . . . .	43
5.3.2	2 treinen, 3 remote antenna units . . . . .	46
5.4	Evaluatie . . . . .	48
<b>6</b>	<b>Besluit en toekomstperspectieven</b>	<b>50</b>
<b>A</b>	<b>Gedetailleerde resultaten</b>	<b>52</b>
A.1	Pakketstroom tijdens gesimuleerde handover met conventionele AP's . . . . .	52
	<b>Bibliografie</b>	<b>56</b>
	<b>Lijst van figuren</b>	<b>58</b>
	<b>Lijst van tabellen</b>	<b>62</b>

# Gebruikte afkortingen

ACK	Acknowledgement
AP	Access Point
ARP	Address Resolution Protocol
Bps	Byte per seconde
BS	Base Station
BSS-ID	Basic Service Set Identification
EDGE	Enhanced Data Rates for GSM Evolution
FAMOUS	Fast Moving Users
Flash-OFDM	Fast Low-latency Access with Seamless Handoff
	Orthogonal Frequency Division Multiplexing
GPRS	General Packet Radio Service
GSM	Global System for Mobile telecommunication
HSDPA	High Speed Downlink Packet Access
HSUPA	High Speed Uplink Packet Access
IAPP	Inter Access Point Protocol
IP	Internet Protocol
kbps	kilobit per seconde
LAN	Local Area Network
MAC	Media Access Control
Mbps	Megabit per seconde
MIT	Massachusetts Institute of Technology

---

NAM	Network Animator
NS	Network Simulator
OS	Operating System
RAU	Remote Antenna Unit
RoF	Radio-over-Fiber
RTT	Round Trip Time
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
WiFi	Wireless Fidelity
WiMax	Worldwide Interoperability for Microwave Access

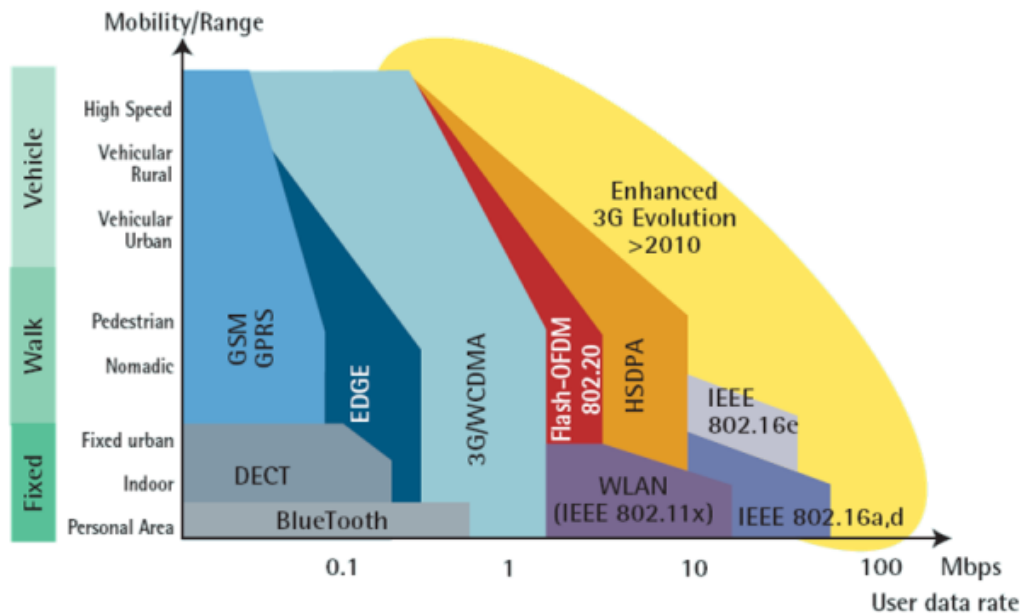
# Hoofdstuk 1

## Inleiding

Breedband internettoegang op treinen, het is vandaag de dag al (bijna) realiteit. Een voorbeeld is het pilootproject op de hogesnelheidstrein Thalys. Heeft een onderzoek als dit dan eigenlijk nog wel nut? Absoluut! De huidige technologieën, waaronder de opstelling ondermeer op de Thalys, hebben immers enkele grote nadelen. We bespreken in deze inleiding enkele hedendaagse technologieën met hun voor- en nadelen. Uiteraard zal dergelijk overzicht steeds onvolledig zijn aangezien er voortdurend onderzoek verricht wordt naar nieuwe methoden, bijvoorbeeld via DVB-T (Digital Video Broadcasting, Terrestrial). Hierdoor beperken we ons tot enkele voorbeelden waarvan de technologie momenteel beschikbaar is.

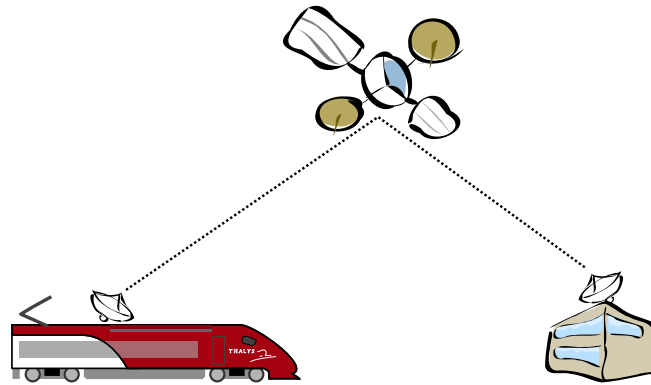
Het typische probleem van een mobiel netwerk is de trade-off tussen verbindingssnelheid en bewegingssnelheid. Hoe sneller je beweegt, hoe lager de verbindingssnelheid van je netwerk wordt. De snelste verbindingen zijn vaste verbindingen, waar de bewegingssnelheid vrijwel nul is. De gekende draadloze netwerkverbinding zoals deze tegenwoordig standaard op laptops geïnstalleerd is, is merkbaar trager dan een vaste verbinding, maar laat een grotere bewegingssnelheid toe, die echter nog niet al te hoog is. Meer hierover in 1.2.1. Andere technologieën, zoals UMTS en GSM maken een grote bewegingsvrijheid aan hoge snelheden mogelijk, helaas is de verbindingssnelheid niet om over naar huis te schrijven. Dit net terwijl we behoefte hebben aan grotere bandbreedtes aan steeds hogere bewegingssnelheden. Het is dan ook deze grens die we met dit onderzoek wensen te verleggen. Een en ander is schematisch weergegeven in figuur 1.1.

We kunnen de beschikbare technologie grotendeels opdelen in twee categorieën: satelliet- en trackside oplossingen. Een satellietverbinding is schematisch weergegeven in figuur 1.2.

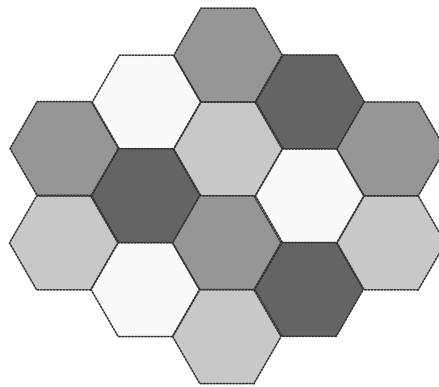


**Figuur 1.1:** Het doel van dit onderzoek is het verleggen van de grens van de combinatie van bandbreedte en bewegingssnelheid.

Hiermee kan je het volledige traject van de trein in één klap dekken, tot zelfs een heel land of werelddeel. De trackside oplossingen zijn een variant op celgebaseerde oplossingen: de gebruiker bevindt zich in een cel en is verbonden met een centrale installatie (zender, ontvanger en antenne) in deze cel. Een typisch voorbeeld hiervan is het GSM-netwerk, een lappendeken van allemaal cellen (zie figuur 1.3) die alzo vrijwel het volledige land bestrijken. Wanneer men echter enkel cellen naast elkaar langs het traject van een trein (langs de spoorlijn dus) gaat plaatsen, bekomen we een trackside oplossing zoals in figuur 1.4 die enkel een verbinding voorziet langs het traject van de trein en niet meteen over het hele land. De reden waarom we dit kunnen, is omdat de trein een gekend traject volgt. Bij een GSM-gebruiker is dit niet het geval: deze kan zich in alle richtingen bewegen, waardoor we overal dekking moeten voorzien. Bij trackside oplossingen kunnen we de dekking concentreren op de plaatsen waar de trein zich bevindt terwijl we de dekking daarbuiten minimaliseren, door bijvoorbeeld gebruik te maken van gerichte antennes. De overlappende cirkels van figuur 1.4 worden dan overlappende ovalen met de lange as volgens de richting van de spoorlijn. Trackside oplossingen zijn goed werkbaar omdat we weten welke cel de volgende is bij een beweging van de trein.



**Figuur 1.2:** Trein en grondstation zijn door middel van schotelantennes en een satellietverbinding met elkaar verbonden.

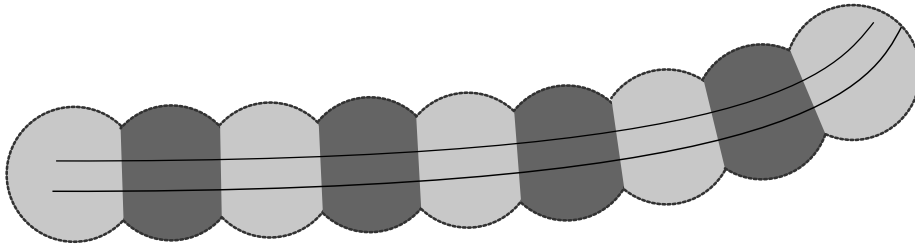


**Figuur 1.3:** Dekking van een heel gebied door middel van aangrenzende en overlappende cellen waarin verschillende frequenties gebruikt worden.

## 1.1 Satellietverbinding

Voor deze verbinding, die ondermeer tot voor kort uitgetest werd op de hogesnelheidstrein Thalys, maakt men gebruik van een schotelantenne op de trein en een telecommunicatiesatelliet die een verbinding tussen de trein en een vast grondstation mogelijk maakt. Deze verbinding laat toe om met een redelijke bandbreedte informatie uit te wisselen. Het grote voordeel aan deze manier van werken is dat er met een beperkte infrastructuurkost vrijwel overal dekking is. Er treden echter eveneens verschillende grote nevenverschijnselen op.

Een eerste uiterst belangrijk nadeel is dat deze verbinding een enorme vertraging introduceert. Er doen zich grote round-trip-times (RTT) voor. Een bericht verzonden door een



**Figuur 1.4:** Dekking van een traject waarlangs een gebruiker zich beweegt door middel van aangrenzende en overlappende cellen waarin verschillende frequenties gebruikt worden.

gebruiker op de trein moet immers een lange afstand afleggen, tot aan de satelliet en terug, vooraleer het door een vast netwerk op de eindbestemming kan afgeleverd worden. De hiervoor gebruikte geostationaire satellieten bevinden zich op een hoogte van 35.786 km [1] pal boven de evenaar, waardoor een pakket al snel 72.000 km dient af te leggen alvorens het ter bestemming is. De vertragingstijd bedraagt op die manier 0,25 s om een pakket van de ene kant naar de andere te brengen, wat in een round-trip-time van 0,5 s resulteert. Interactieve toepassingen zoals internetbellen en videoconferencing worden al snel onmogelijk, net terwijl deze alsmaar belangrijker worden. Denk maar aan het bellen met een satelliettelefoon waar hetzelfde probleem zich voordoet en men al snel door elkaar begint te praten.

Haar grote voordeel, namelijk de dekking van een vrij groot gebied vanuit de ruimte, is niet absoluut en overal vanzelfsprekend. Zo heb je voor satellietcommunicatie immers een vrije gezichtslijn (line of sight) nodig tussen een schotelantenne op de grond en de satelliet om een verbinding van goede kwaliteit te bekomen. Wanneer de gebruiker zich tussen hoge gebouwen bevindt, kan dit onmogelijk worden. De gebruiker kan de satelliet vanaf de grond dan niet "zien". Dit probleem doet zich eveneens voor in tunnels en onder overkappingen. Wanneer men zich in de buurt van een luchthaven bevindt, is men wettelijk verplicht de zender stil te leggen om de aanwezige radarinstallaties (opererend tussen 1 en 2 GHz) en automatische landingssystemen van de vliegtuigen niet te verstoren, hoewel de satellietcommunicatie gebeurt op een frequentie van 4-6 GHz (C-band), 12-18 GHz ( $K_u$ -band) of 18-40 GHz ( $K_a$ -band) [1].

Daar waar de infrastructuurkost voor de gebruiker beperkt is tot een schotelantenne, zender, ontvanger en apparatuur (robot) voor het richten van de schotel naar de satelliet, is de infrastructuurkost aan de kant van de exploitant een pak hoger. Het ontwikkelen en lanceren van een kunstmaan is niet vanzelfsprekend. Om deze reden is het huren van

bandbreedte voor een dergelijke verbinding ook vrij duur.

Het pilootproject op de Thalys is intussen stopgezet en ontmanteld. Voor meer informatie over satellietverbindingen verwijzen we naar [1].

## 1.2 Trackside oplossingen

### 1.2.1 WiFi

IEEE 802.11 is bij velen beter gekend als WiFi, de draadloze apparatuur die de dag van vandaag standaard op vrijwel elke laptop aanwezig is en een verbinding met een draadloos netwerk of een breedbandrouter mogelijk maakt. De verbindingssnelheid is vrij hoog, tot 54 Mbps, het bereik is echter eerder beperkt. Dit is te wijten aan het feit dat er gebruik wordt gemaakt van een licentievrije radioband in het gebied rond 2,4 GHz voor de b- en g-variant, of rond de 5 GHz voor de a-variant. Wettelijk heeft men beperkingen opgelegd op het uitgezonden vermogen (100 mW in Europa, 1000 mW in de VS [2]) zodat vrijwel iedereen een eigen residentieel netwerk kan aanleggen, waardoor de reikwijdte (tot 90 m in de vrije ruimte) dus beperkt is.

Wanneer we het bereik trachten te vergroten door langs de spoorlijn op regelmatige afstand toegangspunten ("access points" of kortweg "AP's") te voorzien, treden er echter problemen op. Een gebruiker ("mobile station" of kortweg "station") kan immers maar verbonden zijn met één AP. Hiervoor gaat hij eerst op zoek naar een AP door bepaalde pakketten (probe-requests) uit te zenden en te wachten op een antwoord van het AP. Komt er geen antwoord, dan probeert het station een andere frequentie. Is er een AP gevonden, dan dienen station en AP informatie uit te wisselen om het station toe te laten tot het netwerk (authenticate) en uiteindelijk dient het station met het betreffende AP verbonden te worden (associate). Deze procedure is vrij tijdrovend. Als een station op een trein telkens na 2,4 seconden (in het vrij optimistische scenario van antennes om de 200 m en een beweging aan 300 km/u) naar een nieuw AP op zoek moet, wordt dit al gauw onmogelijk. Bovendien kunnen pakketten afkomstig van het vaste netwerk naar de trein op het verkeerde moment op het verkeerde AP terechtkomen (de trein is intussen reeds voorbij) en verloren gaan. Tests tijdens een stage deze zomer (zie [3]) toonden aan dat een handover telkens na 10 seconden niet haalbaar is wanneer we het station zelf een AP laten selecteren. Wanneer we het station aansturen en meteen het juiste AP laten selecteren, is slechts een bandbreedte van 500 kbps mogelijk bij een handover elke 3 seconden.



### 1.2.2 UMTS

Een actuele celgebaseerde technologie is UMTS. UMTS, wat staat voor Universal Mobile Telecommunications System, wordt door velen aanzien als de opvolger van GSM wat betreft digitale datacommunicatie. Net als bij GSM wordt ook hier gebruik gemaakt van vaste antennes en mobiele gebruikers. Deze technologie belooft hoge verbindingssnelheden maar deze zijn vooral van toepassing op een beperkte oppervlakte, binnen een beperkte straal rond de antenne. Bovendien is de verbindingssnelheid ook hier een trade-off van de bewegingssnelheid van de gebruiker. In stedelijke gebieden, waar er vele antennes zijn, zal deze verbindingssnelheid vrij hoog zijn, tot 2 Mbps [1]. Dit laatste is echter enkel mogelijk wanneer je je aan een wandeltempo (maximum 10 km/u) verplaatst. In landelijk gebied, waar je je aan een hogere snelheid verplaatst, is de verbindingssnelheid heel wat minder, namelijk 384 kbps aan 120 km/u of 144 kbps aan een snelheid tot 500 km/u [1]. Ons doel, om een verbinding met grote bandbreedte te voorzien voor snel bewegende gebruikers zoals treinen, kan hiermee niet gerealiseerd worden. Dit is vooral te wijten aan de overgang tussen twee cellen, zoals vermeld bij de paragraaf over WiFi. UMTS kent weliswaar het principe van soft-handover, wat wil zeggen dat het mobiele station geconnecteerd kan zijn met twee toegangspunten (Node-B's in UMTS-terminologie) tegelijk in de overlapzone en alzo niet meteen de connectie verliest. Het aantal handovers per tijdseenheid blijft echter nog altijd te hoog om vlot werkbaar te zijn. Verder is de infrastructuurkost ontzettend hoog. Het land volzetten met antennes is niet goedkoop, om nog maar te zwijgen van esthetische problemen in de ruimtelijke ordening en het vergunningsbeleid.

### 1.2.3 WiMax

WiMax (IEEE 802.16), wat staat voor Worldwide Interoperability for Microwave Access, kent vooral een toepassing als draadloos distributienetwerk voor "the last mile". De dag van vandaag is een overbrugging van een afstand van 5 tot 8 kilometer praktisch mogelijk tot 2 Mbps. Theoretisch zou dit 70 Mbps [1] moeten zijn en belooft men tot 50 km te kunnen overbruggen, weliswaar beiden niet tegelijk. Een voordeel is dat hier, net als bij UMTS, geen line-of-sight nodig is, maar het bereik wordt dan wel kleiner. Net als bij UMTS treedt ook hier een trade-off tussen verbindingssnelheid en bewegingssnelheid op: hoe sneller men zich verplaatst, hoe trager de verbinding, net als bij UMTS. Voor hogesnelheidstreinen is dit al helemaal niet realistisch, hoewel men op dit vlak beterschap belooft voor de volgende generatie van de standaard.

### 1.2.4 Andere

Voor de volledigheid vermelden we ook nog opkomende technologieën zoals HSDPA (High-Speed Downlink Packet Access, tot 14,4 Mbps [1] voor alle gebruikers binnen één cel samen), EDGE (Enhanced Data Rates for GSM Evolution, 236,8 kbps [1]), Flash-OFDM (Fast Low-latency Access with Seamless Handoff Orthogonal Frequency Division Multiplexing, 1-1,5 Mbps [4]), HSUPA (High-Speed Uplink Packet Access, 5,76 Mbps [4]) of het momenteel gebruikte GPRS (General Packet Radio Service, tot 170 kbps [1]). Deze kennen echter voornamelijk dezelfde problemen die typisch zijn aan celgebaseerde oplossingen of zijn ontoereikend wat betreft bandbreedte. Voor meer informatie verwijzen we naar [1] of [4].

## 1.3 Naar een verbeterde trackside oplossing

Het lijkt ons opportuun om verder onderzoek te doen naar het verbeteren van een toegang via een trackside oplossing. Als we kunnen bepalen in de buurt van welke antenne de trein zich bevindt, kunnen we pakketten bestemd voor de gebruiker efficiënt naar de bestemming routeren. Als we bovendien het zoeken door de gebruiker naar een antenne kunnen elimineren, kan de tijd gedurende dewelke er geen verbinding is, gereduceerd worden en lijkt het aanbieden van een goede verbinding aan een hoge snelheid wel haalbaar. Een belangrijk punt dat echter niet mag verwaarloosd worden, is de kostprijs. Het is deze weg, met antennes langs de spoorlijn en gebruik makend van een bestaande technologie, die we in dit werk zullen behandelen om tot een werkbaar systeem te komen.

In dit werk gaan we vooreerst een voorstel poneren om deels met behulp van hedendaagse technologie een trackside oplossing uit te werken die bovenstaande neveneffecten zoveel mogelijk neutraliseert. Dit alles wordt uit de doeken gedaan in hoofdstuk 2. In hoofdstuk 3 gaan we dit model mappen op een simulatiemodel dat met de huidige bestaande apparatuur eenvoudig kan gesimuleerd worden om de werking van het model te verifiëren. Uiteraard worden in de volgende hoofdstukken dan ook de uitwerking, de gebruikte simulatiesoftware en de resultaten besproken, waarna we een besluit formuleren alsook even verder kijken welke weg er nog dient afgelegd te worden om een effectief breedbandtoegangsnetwerk op treinen te voorzien.

## Hoofdstuk 2

# Radio-over-Fiber langs de spoorlijn: werking

Zoals in de inleiding vermeld werd, hebben we gekozen voor het verbeteren van een trackside oplossing op basis van hedendaagse standaarden. We gaan op zoek naar een methode om de handovertijd (de tijd tussen het verliezen van de connectie met een antenne en de aanwezigheid van een verbinding met de volgende) te reduceren. Deze methode kan dan met grote waarschijnlijkheid eveneens toegepast worden bij toekomstige netwerktechnologieën. Verschillende modellen werden ontworpen, tegen elkaar afgewogen, uitgewerkt, bestudeerd en aangepast, maar de meesten voldeden niet aan de eisen wat betreft kostprijs, schaalbaarheid naar meerdere (kruisende) treinen, mogelijkheden van de beschikbare technologie of realistische celgroottes. Uiteindelijk werd volgend model geponeerd, waarvan we verwachten dat dit aan de vereisten voldoet.

Uiteraard dient dit model uitvoerig empirisch getest te worden. Voor een proef met behulp van een simulator verwijzen we naar de volgende hoofdstukken. Voor een testopstelling met hardwarematig onderzoek was helaas geen tijd meer.

### 2.1 Radio-over-Fiber

Vooreerst delen we het traject van de trein op in verschillende cellen. In elke cel plaatsen we langs de spoorlijn verschillende basisstations, die onderling verbonden zijn door middel van een optische vezelring. Het gebied dat door dergelijke ring bestreken wordt, noemen we eenvoudigweg *ring*, om verwarring te vermijden. Het begrip cel behouden we voor

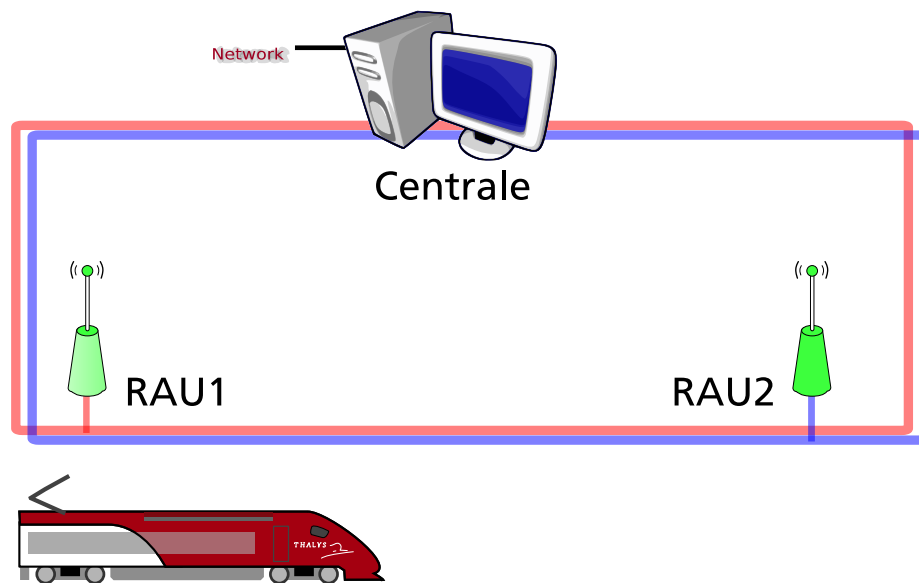
het gebied waarbinnen een basisstation actief is. We beschouwen enkel de werking en de handover binnen dergelijke ring, de handover tussen twee ringen is een andere zaak. Via deze ring staan de basisstations eveneens rechtstreeks in verbinding met een centrale die de toegang tot het bovenliggende netwerk verzorgt en telkens op de hoogte moet zijn ter hoogte van welk basisstation de trein zich bevindt. Het concept werd reeds voorgesteld in [5].

Voor elk basisstation langs de optische ring, voorzien we een eigen golflengte op de glasvezelkabel. Elk basisstation heeft dus een eigen, directe verbinding met de centrale en is ook het enige element op deze verbinding. De centrale kan op haar beurt signalen naar elk afzonderlijk basisstation sturen door dit signaal op de juiste golflengte te moduleren. Andere basisstations zullen de signalen niet ontvangen. Zie figuur 2.1.

Om de kostprijs te drukken, moeten de basisstations zo goedkoop mogelijk zijn. We hebben er immers veel nodig om een volledig traject te kunnen bedekken. Deze toestellen dienen dus zo eenvoudig mogelijk in elkaar te zitten, zo weinig mogelijk intelligentie te bevatten en zo onderhoudsvrij mogelijk te zijn. Dit kunnen we bekomen door deze basisstations louter te laten fungeren als antenne. Signalen die zij op hun golflengte binnenkrijgen, dienen zij na demodulatie onmiddellijk en zonder enige tussenbewerking uit te zenden als radiosignalen. De stations krijgen dus een analoog signaal binnen via de optische connectie en zenden dit, zonder omweg langs digitale kant, direct uit. Deze signalen dienen dan ook gemoduleerd te zijn op de juiste frequentie waarop de ontvanger op de trein luistert. Dit principe, waarbij we radiosignalen moduleren op een optische band, heet *Radio over Fiber* of kortweg *RoF*. Aangezien de basisstations de pakketten niet meer bewerken, spreken we ook niet meer van AP's (zoals in de WiFi wereld) of Base Stations (vanuit de GSM-wereld), maar van *Remote Antenna Units* (of *RAU's*) in RoF-terminologie. Doordat bij RoF de bron van het optisch signaal analoog is, is dit signaal veel gevoeliger voor attenuatie, waardoor het moeilijk wordt om het oorspronkelijke radiosignaal zo juist mogelijk te reconstrueren. Deze attenuatie is echter beperkt genoeg over een afstand tot 10 km.

## 2.2 Moving cell

Een handover gaat normaal gepaard met het zoeken naar een antenne op verschillende frequenties. Dit dienen we te elimineren. De ontvanger (kan zowel de trein als een vast toegangspunt zijn) moet op elk moment weten op welke frequentie de zender gaat zenden, of



**Figuur 2.1:** Optische ring: elk basisstation heeft een eigen toegewezen golflengte en staat rechtstreeks in contact met de centrale.

omgekeerd, zodat er niet meer moet gezocht worden. Hier werd tijdens een stage onderzoek naar gedaan voor wat betreft WiFi. We dwongen de client op regelmatige tijdstippen een andere ontvanger op een andere (vaste) frequentie te selecteren. De resultaten waren bedroevend. Pakketten, bestemd voor het draadloze station, hoopten zich immers op in de AP's, wat tot hoge vertragingwaarden of zelfs tot het uitvallen van de AP's leidde. We dienen dus best ook het zoeken naar een AP op die bepaalde frequentie en het associëren ermee, te vermijden.

Wanneer we nu elke trein een eigen radiokanaal toewijzen, kunnen we de trein langs zijn volledige traject binnen onze optische ring volgen. De signalen worden gegenereerd door de centrale, op de juiste golflengte op de optische ring geplaatst, komen terecht bij één enkele RAU en worden aldaar (en enkel daar) in de ether gestuurd. Verplaatst de trein zich en komt deze in de buurt van een volgende RAU, dan dient de centrale het radiosignaal op een andere golflengte te plaatsen om door dit nieuwe basisstation uitgezonden te worden. De trein behoudt zijn vaste frequentie en deze *cel* volgt hem, via de RAU's, langsheen het hele traject. Dit concept, waarbij we de cel laten meebewegen langs de spoorlijn en de trein de hele tijd (logische gezien) verbonden blijft met hetzelfde logische toegangspunt, weliswaar via een andere RAU, is het *moving cell* concept. Dankzij dit principe kunnen handovers worden vermeden. Dit logische toegangspunt bevindt zich trouwens in de centrale. Het is

de centrale die de pakketten afkomstig van het bovenliggende netwerk omzet in pakketten volgens de gebruikte standaard en voorziet van de juiste header. Het is ook die centrale die managementpakketten genereert. Voor de eenvoud voorzien we voor elke trein die zich tegelijk binnen onze ring kan bevinden, een eigen toegangspunt. We moeten er echter wel zeker van zijn dat het bovenliggende (bedrade) netwerk op elk moment weet waar de trein zich bevindt en naar welk punt de pakketten dus moeten gerouteerd worden.

## 2.3 Werking van het systeem

Hoe gaan we nu te werk om dataverkeer te versturen van de trein naar de centrale, en omgekeerd? Hoe komt het verkeer van de centrale (of het bovenliggende netwerk) op de juiste RAU terecht waar de trein binnen bereik is? Om hier enig inzicht in te verschaffen, behandelen we het up- en downlinkverkeer best afzonderlijk.

### 2.3.1 Uplinkverkeer

Onze RAU's gaan zuiver analoog te werk. Ze capteren voortdurend het volledige frequentiespectrum (beperkt rond de frequentieband die door de betreffende standaard gebruikt wordt) en moduleren dit signaal op de hen toegewezen golflengte en plaatsen dit op de optische ring. Het is pas in de centrale dat de binnengekomen signalen gedemoduleerd worden en de datapakketten uit de verschillende kanalen gereconstrueerd worden. Het is in deze opstelling perfect mogelijk dat meerdere treinen zich aan dezelfde RAU bevinden en pakketten naar de centrale sturen, elk op hun eigen toegewezen kanaal. De RAU ontvangt dan een superpositie van deze twee radiosignalen, moduleert ze op de juiste golflengte en stuurt ze naar de centrale alwaar deze superpositie terug uitgesplitst wordt in de afzonderlijke kanalen. Op die manier is het ook mogelijk kruisende treinen te ondersteunen: doordat ze elk gebruik maken van een eigen frequentie, hinderen ze elkaar niet, en doordat de RAU de volledige band capteert en deze pas in de centrale opgesplitst wordt, zijn er ook aan de ontvangtzijde geen problemen en volstaat één antenne en installatie (RAU) in het basisstation. Er kan eveneens opgemerkt worden dat er slechts modems en digitaal/anaaloog omzeters nodig zijn naargelang het aantal treinen dat men gelijktijdig in de ring wil ondersteunen. Dit aantal zal altijd een grootteorde lager liggen dan het aantal RAU's langs de ring. Niet elk basisstation dient over deze technologie te beschikken. Enkel de centrale dient over een beperkt aantal dergelijke modules te beschikken en dient deze in te zetten

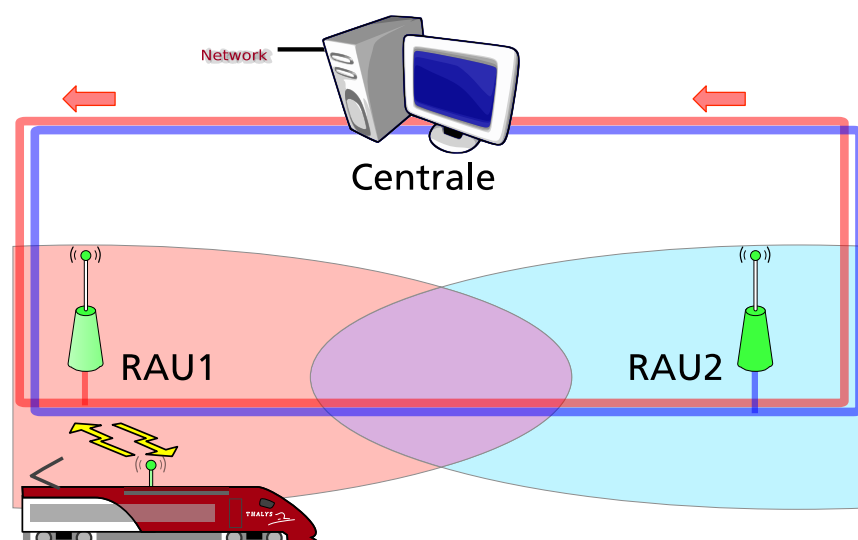
op de golflengtes waarvan de kans groot is dat de trein(en) zich in de buurt van de/het desbetreffende basisstation(s) bevind(en/t). Dit maakt de RAU's merkbaar eenvoudiger en kan alzo de kosten drukken, zoals vermeld in 2.1.

### 2.3.2 Downlinkverkeer

Om downlinkverkeer correct ter bestemming te krijgen, dient de centrale op de hoogte te zijn van de plaats waar de trein zich bevindt. Enkel op die manier kunnen pakketten bestemd voor de trein op de juiste golflengte gemoduleerd en aldus naar de juiste RAU gestuurd worden. Pakketten mogen niet door twee aangrenzende RAU's tegelijk uitgezonden worden, anders worden deze verstoord, treden collisions op en kan de ontvanger ze nooit eenduidig decoderen. Dit probleem kan echter betrekkelijk eenvoudig opgelost worden. De trein verstuurt immers vrijwel voortdurend informatie naar het netwerk. Mocht er toch een moment zijn dat er niets moet verstuurd worden, dan kunnen we nog altijd voorzien dat er random data verstuurd worden. Er zal dus op elk moment minstens één RAU zijn die deze signalen oppikt en naar de centrale verstuurt. Wanneer de centrale dergelijk pakket ontvangt, weet deze meteen waar de trein zich bevindt. Beweegt de trein zich verder, dan zal het signaal ook opgepikt worden door een naburige RAU. De centrale ontvangt ook dat pakket, en schakelt vanaf dan meteen over naar deze nieuwe RAU: alle pakketten bestemd voor de trein zullen vanaf nu enkel via deze RAU verstuurd worden. Door de *vorige RAU* in de centrale te onthouden, kunnen we vermijden dat wanneer de trein zich in de overlapzone tussen twee RAU's bevindt, de centrale het downlinkverkeer steeds afwisselend verzendt via de verschillende RAU's. We sturen enkel verkeer naar de nieuwe RAU en keren niet meer terug naar de vorige, eventueel houden we rekening met een time-outwaarde.

Deze werking wordt geïllustreerd in figuren 2.2 tem 2.5 in chronologische volgorde.

Tot slot kunnen we ook nog opmerken dat dit systeem uiterst flexibel is. We hebben reeds vermeld dat het mogelijk is om twee treinen elkaar te laten kruisen, hierbij gelijktijdig gebruikmakend van dezelfde RAU's. Dit doordat elke trein van een eigen toegewezen kanaal gebruik maakt. Deze redenering valt echter uit te breiden. Zo is het mogelijk om de capaciteit te verhogen door aan elke trein meerdere kanalen toe te kennen, die uiteraard niet overlappen met de kanalen toegekend aan andere treinen binnen dezelfde ring. Op een trein zijn dan meerdere zenders en ontvangers geïnstalleerd, die samen deze capaciteitsverhoging mogelijk maken.



**Figuur 2.2:** De trein bevindt zich in het gebied van RAU1, waarlangs alle communicatie met de centrale verloopt.

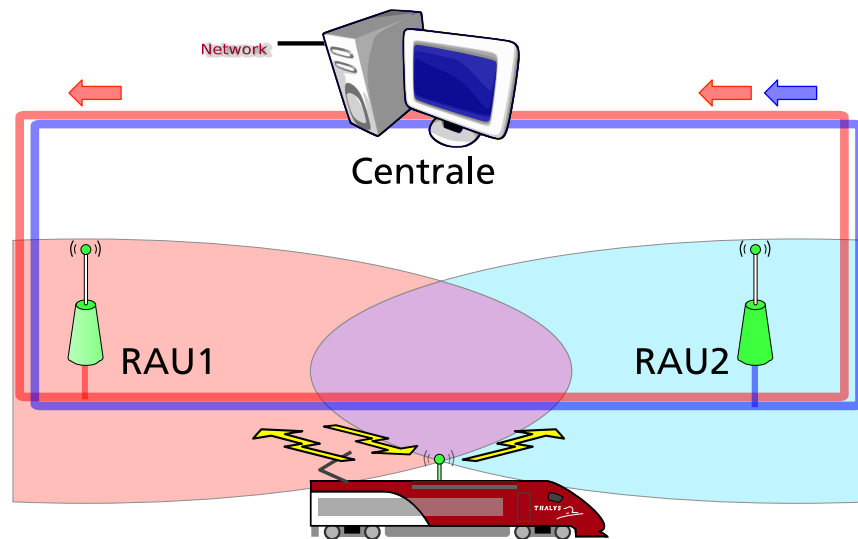
## 2.4 Invloed op de netwerkeigenschappen

Het nagaan van de invloed van ons systeem op netwerkeigenschappen zoals delay, packet loss en jitter, is vrij complex. Dit hangt namelijk allemaal af van de gebruikte standaard voor het draadloze netwerk, de omstandigheden, de afstand tussen de RAU's en hun vermogen, en de kwaliteit van de fiber en de gebruikte toestellen.

Wat het pakketverlies betreft, kunnen we vrij kort zijn: dit is vrijwel volledig afhankelijk van de kwaliteit van de apparatuur, het gekozen vermogen en de afstand tussen de antennes. We kunnen het zelfs omgekeerd stellen: deze parameters dient men te kiezen in functie van de gewenste verbindingkwaliteit om het pakketverlies tot een zeker niveau te beperken, uiteraard binnen de wettelijke en economische mogelijkheden.

De geïntroduceerde delay is al iets beter op voorhand ingeschat worden. We kunnen dit best illustreren aan de hand van een voorbeeld. Stel: de gekozen standaard voor het draadloze netwerk is IEEE 802.11g, met antennes om de 200 m die een bereik hebben tot 105 m zodat we een overlapzone van 10 m bekommen. Dit zijn dezelfde parameters die we verderop in dit werk zullen gebruiken voor de simulaties. De gemiddelde waarde voor de delay is dan de som van de tijd nodig voor het verzenden van een pakket (lengte van het pakket gedeeld door de bitrate van de standaard) en de gemiddelde transmissietijd tot aan



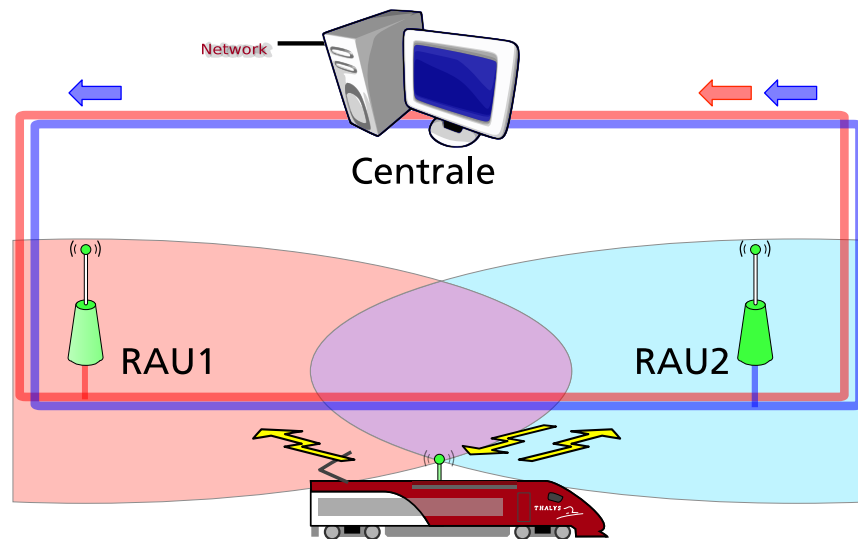


**Figuur 2.3:** De trein rijdt verder en komt terecht in de overlapzone van RAU1 en RAU2. Beide RAU's ontvangen de uplinkpakketten afkomstig van de trein en zenden deze over de optische ring naar de centrale.

de centrale. Er dient geen extra verzendingstijd gerekend te worden aan de RAU's, deze zetten immers het radiosignaal ogenblikkelijk op de optische ring, er komen geen buffers of omzettingen aan te pas. Nemen we voor de eenvoud aan dat de centrale fysiek gelocaliseerd is in het midden van de ring en dat de omzetting van een radiosignaal naar een optisch signaal ogenblikkelijk gebeurt, dan komen we voor ons voorbeeldje met een gemiddelde pakketgrootte van 1000 bytes aan volgende waarden:

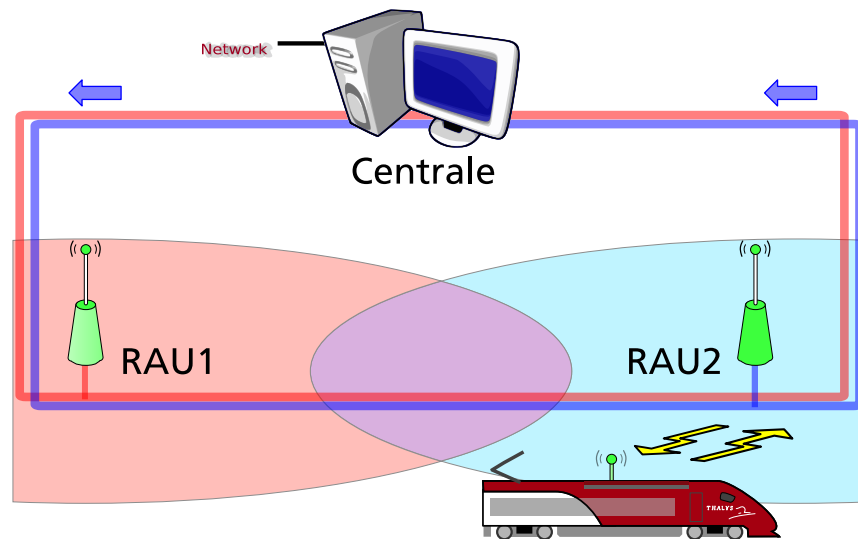
- Verzendingstijd IEEE 802.11g:  $1000 \text{ bytes} \times 8 \text{ b/B} : 54.000.000 \text{ Mbps} = 148,15 \mu\text{s}$
- Gemiddelde transmissietijd tot aan een RAU:  $50 \text{ m} : 300.000 \text{ km/s} = 0,17 \mu\text{s}$
- Gemiddelde transmissietijd van de RAU tot de centrale:  $2,5 \text{ km} : 200.000 \text{ km/s} = 12,5 \mu\text{s}$
- Totaal:  $160,82 \mu\text{s}$

Wat typisch is voor dit model, is dat de afstand tot de centrale en bijgevolg tot het core netwerk steeds verandert. Dit zou wel eens een grote invloed op de vertragsvariatie of jitter kunnen hebben. Ook hier kunnen we echter geen algemene waarden geven zonder parameters in rekening te brengen. Gaan we opnieuw uit van bovenstaand rekenvoorbeeldje,



**Figuur 2.4:** De centrale heeft een uplinkpakket afkomstig van de trein ontvangen via RAU2. Op dit moment schakelt de centrale over en wordt downlinkverkeer naar RAU2 gestuurd. RAU1 blijft de signalen van de trein ontvangen en geeft deze door aan de centrale, maar de centrale keert niet terug naar de  *vorige RAU*.

dan kunnen we op deze manier het verschil in delay bepalen wanneer de centrale overschakelt op een andere RAU. Het verschil in afstand tot de centrale zal het grootst zijn wanneer de centrale zich niet tussen de twee beschouwde RAU's bevindt, en is dan gelijk aan de afstand tussen twee RAU's. In ons geval dus 200 m. De overschakeling op een nieuwe RAU gebeurt wanneer de trein zich in het overlapgebied tussen twee RAU's bevindt. De afstand van de trein tot beide RAU's is dan vrijwel gelijk, zodat we enkel het verschil in afstand tussen de RAU's en de centrale in rekening moeten brengen. Een afstand van 200 m levert aan  $200.000 \text{ km/s}$  een tijdsduur op van  $1 \mu\text{s}$ , wat verwaarloosbaar klein is. Opdat een pakket, verzonden door de trein wanneer overgeschakeld wordt op een nieuwe RAU, eerder zou toekomen dan het vorige pakket dat enkel nog via de vorige RAU opgepikt werd, zou het zonet berekende tijdsverschil groter moeten zijn dan de tijd nodig om een nieuw pakket te versturen. Met de IEEE 802.11g standaard lijkt dit alvast niet mogelijk: de tijd die de trein nodig heeft om een (volgend) pakket te versturen bedraagt  $4.44 \mu\text{s}$ , indien we de backoff timer verwaarlozen en uitgaan van het kleinste mogelijke pakket van 30 bytes, enkel de WiFi-header. Dit is een pak langer dan de tijds winst die men bereikt via de nieuwe RAU van  $1 \mu\text{s}$  en is in werkelijkheid (met backoff timer en grotere pakketten) zelfs nog langer. Van jitter is dan ook nauwelijks sprake (slechts  $1 \mu\text{s}$ ) en van out-of-order-delivery



**Figuur 2.5:** De trein rijdt verder en verlaat de overlapzone. De uplinkpakketten worden enkel nog door RAU2 ontvangen en door deze laatste aan de centrale bezorgd. Alle verkeer verloopt via RAU2.

al zeker niet.

Deze berekening klopt echter enkel wanneer de RAU een pakket rechtstreeks naar de centrale kan sturen en omgekeerd. Dit is het geval zoals beschreven in dit hoofdstuk en zoals geïllustreerd in de figuren. Wanneer we echter de centrale tussen de RAU's plaatsen, kunnen we problemen krijgen. In onze lus sturen we het licht immers in één enkele richting. De RAU pikt dit licht op en plaatste een nieuw signaal op de fiber op dezelfde golflengte maar in de andere richting. Met de centrale tussen twee RAU's kunnen we dan andere resultaten bekomen. Dan is het immers mogelijk dat de RAU die zich net naast de centrale bevindt, het licht in de richting weg van de centrale moet sturen waarna het via een omweg van 10 km uiteindelijk op de centrale aankomt. Als onze trein dan verder beweegt in de richting van de volgende RAU, die zich aan de andere kant van de centrale bevindt, kunnen we jitterwaarden van  $50 \mu s$  bekomen, wat al een heel verschil is. We kunnen er echter van uitgaan dat elke RAU dan twee golflengtes toegewezen krijgt: één voor de uplink en één voor de downlink, zodat het licht steeds de kortste weg kan volgen. In ons model, waar de centrale zich niet tussen de RAU's bevindt, treden dergelijke problemen niet op en is één golflengte per RAU voldoende.

## Hoofdstuk 3

# Simulatiemodel

Het lijkt evident dat het model zoals geponoerd in hoofdstuk 2 zal werken. Om van enige wetenschappelijke waarde te kunnen spreken, dient dit proefondervindelijk aangetoond te worden. Dit is echter niet vanzelfsprekend. De voorgestelde apparatuur is nog onbestaande en zou allesbehalve goedkoop zijn om als testversie te bouwen. Bovendien hangt aan de fysieke opstelling ook een hoge kostprijs en zou men de grote middelen moeten inzetten om aan een hoge bewegingssnelheid metingen uit te voeren. We dienen bijgevolg onze toevlucht te zoeken tot simulaties of emulaties. De enige middelen die we hiervoor ter beschikking hebben, zijn gebaseerd op ethernet. Dit zowel voor hardware-emulaties als voor softwaresimulaties met behulp van een simulator zoals ns-click, waarover meer in hoofdstuk 4.1. In dit hoofdstuk stellen we een simulatiemodel op, waarbij we ons fysieke model mappen op bestaande apparatuur. We opteren voor een algemeen model, dat zowel toepasbaar is voor simulaties in software als voor emulaties.

### 3.1 Wireless wordt WiFi

Vooreerst kiezen we als gebruikte draadloze netwerkstandaard voor IEEE 802.11, beter bekend als WiFi. Hiervan is immers tal van goedkope apparatuur beschikbaar, alsook bestaan er simulators die het gebruik van deze standaard kunnen emuleren. Andere standaarden, zoals bijvoorbeeld GSM, zijn niet in die mate toegankelijk. Zoals eerder vermeld, is het reële model toepasbaar op meerdere standaarden, maar WiFi is veruit de gemakkelijkste om mee te werken om het concept aan te tonen.

Het capteren van de volledige frequentieband door de RAU's, zoals bij Radio-over-Fiber

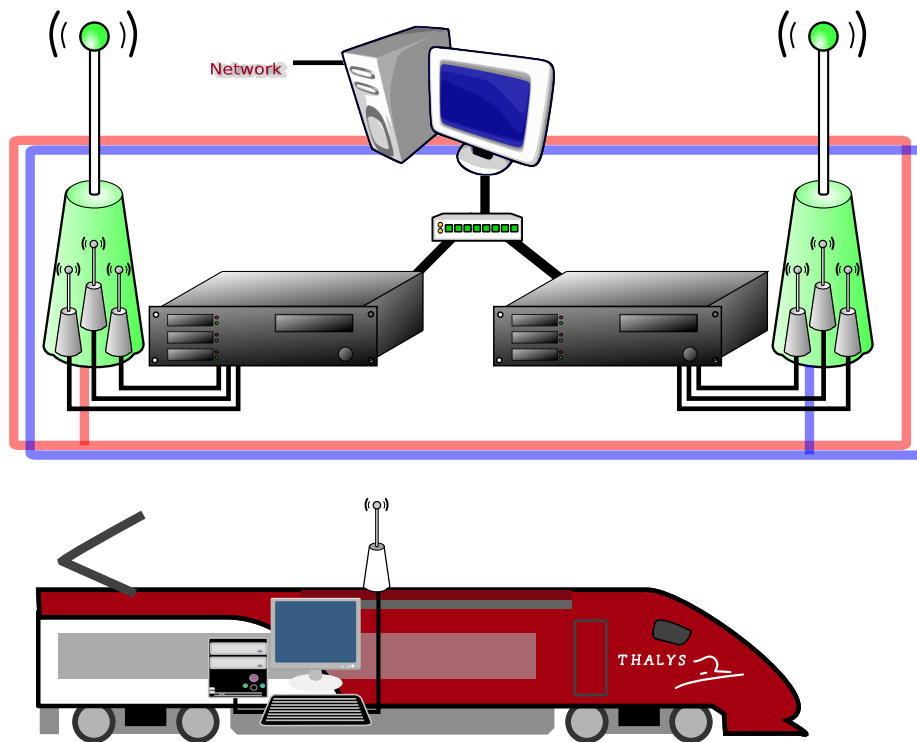
het geval is, is niet meteen realistisch in onze omgeving. Ook dit valt echter te simuleren. Aangezien we onze simulaties uitvoeren met de IEEE 802.11 standaard, kunnen we ons beperken tot het capteren van de kanalen gebruikt door deze standaard. Dit kan door de RAU's te voorzien van voldoende WiFi kaarten: één voor elk kanaal dat we willen capteren, of meer concreet: één voor elke trein die we simultaan in onze ring willen laten rijden, tenzij we meerdere kanalen per trein gebruiken zoals in de laatste paragraaf van hoofdstuk 2 vermeld. Dergelijke situatie wordt getoond in figuur 3.1.

## 3.2 Optisch wordt ethernet

Verder dienen we onze optische ring te vervangen door een conventioneel LAN. Zowel de RAU's als de centrale worden vervangen door een pc, voorzien van een ethernet interface, aangesloten op een gemeenschappelijke hub. Uiteraard heeft de centrale nog een tweede ethernetinterface die aangesloten wordt op het uplinknetwerk. Elke interface heeft een uniek MAC-adres, waardoor de centrale met elke RAU afzonderlijk en rechtstreeks kan communiceren. Bemerkt de analogie met de toegewezen golflengte aan elke RAU in het echte model, waar ook elke RAU rechtstreeks en afzonderlijk met de centrale in verbinding staat.

Uiteraard willen we dat onze basisstations zo goed mogelijk gelijken op de RAU's van het echte model en willen we vooral dat deze zo eenvoudig mogelijk in elkaar steken en de ontvangen signalen niet (of zo min mogelijk) aanpassen. Dit kunnen we verwezenlijken door het volledige ontvangen pakket, inclusief headers en bijkomende informatie zoals ontvangen vermogen, signaal/ruis-verhouding,... in een nieuw ethernetpakket te stoppen en te verzenden naar de centrale. We dienen er wel over te waken dat de RAU's elk ontvangen pakket doorsturen. Deze pakketten zijn normaal gericht aan het (logische) access point, dat zich in de centrale bevindt. Wanneer twee treinen zich ter hoogte van een specifieke RAU bevinden, zullen beide treinen pakketten uitzenden gericht aan een andere ontvanger, namelijk hun eigen toegewezen access point dat zich in de centrale bevindt. Dit kunnen we bekomen door de gebruikte draadloze netwerkkaarten in *promiscuous* mode te zetten. Dan worden alle pakketten binnengenomen en aan de software doorgegeven voor behandeling, ook unicastpakketten met een andere ontvanger. In standaard mode worden immers enkel multicastpakketten en unicastpakketten binnengenomen en doorgegeven waarbij het bestemmingsadres het hardware-adres is van de betrokken netwerkkaart.

Wat de downlinkpakketten betreft, zullen we deze ook zoveel mogelijk in de centrale aan-



**Figuur 3.1:** Mapping van de simulatie op het fysieke model: de RAU's die normaal de gehele band capteren en op de optische ring plaatsen, worden vervangen door een aantal draadloze netwerkkaarten aangesloten op een pc, die via een conventioneel LAN met de centrale in verbinding staat.

maken. De IEEE 802.11 pakketten worden gegenereerd door de centrale, alsook de extra informatie die normaal bestemd is voor de driver van de netwerkkaart, zoals zendvermogen en dergelijke. De centrale bundelt het IEEE 802.11 pakket samen met deze extra info alsook een aanduiding van het kanaal waarop het pakket moet verzonden worden in een nieuwe ethernetpakket met als bestemming de juiste RAU. Uiteraard is het afzenderadres dat van de centrale. De RAU krijgt het pakket binnen, verwijdert de ethernetheader en geeft het pakket door aan de draadloze netwerkkaart die ingesteld is op het kanaal dat aangegeven werd in het pakket.

Een voorbeeld van de headers van een dergelijk pakket tussen de centrale en de RAU met als inhoud UDP-verkeer naar de trein, is weergegeven in figuur 3.2. Vooreerst werd een UDP-pakket gegenereerd door een server ergens in het core network. Deze server capsuleert het in, in een IP-pakket voor verzending over het netwerk. De laatste node voor de centrale komt via een ARP-request te weten wat het IP is van de client op de

trein, en genereert vervolgens op basis van het IP-pakket een nieuw ethernetpakket met als doeladres dat van de trein. De centrale en de RAU's worden immers samen aanzien als één netwerkbrug, zonder eigen (publiek) MAC-adres. Voor de simulaties moeten we dus ook deze ethernetinterface van onze centrale in promiscuous mode zetten. De centrale neemt het pakket binnen, waarna het interne logische AP het omzet in een WiFi-pakket, met dezelfde bron en bestemming en het BSS-ID van het betrokken AP, dat door een RAU verzonden dient te worden. Zoals eerder vermeldt, wordt het volledige pakket gegenereerd zoals de driver het zou afgeven aan de firmware van de netwerkkaart. Hiertoe wordt een extra header toegevoegd met deze informatie. Omdat de RAU in ons geval gewoon uitzendt op verschillende kanalen, dienen we nog een header van 1 byte toe te voegen die aangeeft op welk kanaal (lees: via welke WiFi-kaart) het pakket verzonden dient te worden. Uiteindelijk wordt heel dit nieuwe pakket ingepakt in een normaal ethernetpakket om over het LAN naar de RAU verzonden te worden.

Eth 14	K 1	Extra 24	IEEE 802.11 30	IP 20	UDP 8	Data -
-----------	--------	-------------	-------------------	----------	----------	-----------

**Figuur 3.2:** Een voorbeeld van de headers van een UDP-pakket naar de client, gecapteerd tussen de centrale en de RAU. De lengte van de headers is vermeld in bytes. Header *K* staat voor het kanaal waarop het betreffende pakket dient uitgezonden te worden.

Wat de centrale betreft, blijven de werking en de toebedeelde taken dezelfde. Deze draait nog steeds een logisch AP per trein dat de pakketten genereert. Het enige verschil is dat deze pakketten niet meer uitgezonden worden (op de juiste golflengte), maar verpakt worden in een ethernetpakket en naar de juiste bestemming (RAU) gestuurd worden. Aan de werking van het mobiele station verandert er helemaal niets.

### 3.3 Beweging

Voor wat de beweging van de gebruiker betreft, zijn er twee mogelijkheden. Ofwel wordt er gewerkt met een simulator zoals ns-click, waarbij we de gebruiker virtueel aan gelijk welke snelheid over gelijk welke afstand kunnen laten bewegen, ofwel moeten we ook deze beweging gaan simuleren. Dit laatste zal zeker het geval zijn wanneer we metingen wensen uit te voeren op echte hardware.

Voor emulaties met echte hardware, zouden we het FAMOUS-testbed kunnen gebruiken.

Hierin wordt het signaal van de verschillende basisstations beurtelings gedempt met behulp van regelbare attenuatoren. De opstelling bevat standaard vier antennes, voorzien van een eigen attenuator. Door het signaal van 3 van de 4 antennes te dempen en één signaal ongedempt uit te zenden, lijkt het alsof de trein zich ter hoogte van die bepaalde antenne bevindt. Gaan we vervolgens het signaal van die antenne zachtjes dempen terwijl we de demping van het signaal van een naburige antenne verminderen, dan wordt het signaal van deze laatste sterker terwijl dat van de eerste zwakker wordt. We bewegen ons naar de nieuwe antenne. Door deze vier antennes elk om beurt te dempen, kunnen we beweging simuleren. De ervaring heeft echter geleerd dat deze demping nog niet voldoende is, het signaal blijft nog steeds vrij sterk aanwezig. Bovendien zou het aantal benodigde (dure!) attenuatoren vrij groot worden aangezien we er meerdere nodig hebben per basisstation.

We kunnen de beweging echter op een andere manier simuleren. Bij aanvang laten we alle RAU's, behalve één, alle binnenkomend draadloos verkeer blokkeren. De trein bevindt zich dan zogezegd enkel binnen het bereik van die RAU. Na een tijdsinterval bepaald door een exponentiële kansverdeling met parameters afgeleid van de bewegingssnelheid van de trein en afstand tussen twee RAU's, laat de naburige RAU de pakketten door naar de centrale. De eerste RAU ontvangt de pakketten nog steeds en stuurt ze nog steeds door. We bevinden ons virtueel in de overlapzone tussen twee RAU's. Enige tijd later stopt de vorige RAU met het doorsturen van de pakketten, de trein is dan immers buiten bereik. Dit model kan nog verbeterd worden door de nieuwe RAU gedurende de eerste tijd in de overlapzone een bepaald aandeel van de inkomende pakketten te laten droppen, de ontvangst zou op die afstand immers vrij slecht zijn. Hetzelfde geldt voor de oude RAU op het einde van de tijd in de overlapzone.

Initieel was het de bedoeling om voor alle nodes gebruik te maken van het *Click Modular Router Project* [6]. Dit zou uiterst generieke code opleveren. Click wordt immers gebruikt zowel voor hardwarematige simulaties, als in emulatie-omgevingen zoals ns-click. Op die manier kon de code, mits beperkte wijzigingen, hergebruikt worden voor beide simulaties. Wegens tijdsgebrek zijn er echter geen hardwaresimulaties doorgedaan. De simulaties in ns-click en meer informatie over ns-click zelf, zijn het onderwerp van volgend hoofdstuk.



## Hoofdstuk 4

# Implementatie van het simulatiemodel

### 4.1 Wat is ns-click?

Wat wij in dit werk aanduiden als ns-click is eigenlijk een samenvoeging van twee projecten. Enerzijds is er click [6], anderzijds is er ns-nam [7, 8].

#### 4.1.1 Click

Click is software waarmee op modulaire basis een router gebouwd kan worden. Er is een uitgebreide bibliotheek aan dergelijke click-elementen (meer dan 300) die een waaier aan functionaliteiten bieden. Deze elementen kunnen bewerkingen uitvoeren op een pakket, zowel op bitniveau als op een abstracter niveau, zoals bijvoorbeeld een wijziging van een adres, wat uiteraard ook een wijziging op de individuele bits van het pakket impliceert. Ook kunnen pakketten hiermee gegenereerd worden, zoals bijvoorbeeld beacon frames die aangemaakt worden door het element `BeaconSource`. Deze modules zijn elementaire bouwblokken waarmee je op betrekkelijk eenvoudige en overzichtelijke wijze een complex systeem kan bouwen door ingangen en uitgangen van verschillende elementen met elkaar te verbinden. De elementen zelf zijn geprogrammeerd in C++, om click-elementen te combineren in een click-script wordt de click-taal gebruikt. Zelf elementen toevoegen is uiteraard eveneens mogelijk, wat ook gebeurde in dit werk.

Click kan op verschillende manieren gebruikt worden: als userlevel software, als kernmo-

dule, of in een simulatoromgeving zoals ns-click. De derde methode wordt besproken in sectie 4.1.3. De eerste en tweede methode laten toe click te draaien op een reële machine, met een eigen netwerkkaart, draaiende Linux-kernel en gebruikersprogramma's. Click kan zowel als userlevel applicatie en als kernelmodule pakketten versturen via de netwerkkaart door gebruik te maken van het element `ToDevice(eth0)`, met eth0 de naam van de netwerkkinterface. Alle binnenkomende pakketten worden door de netwerkkaart zowel aan de kernel als aan click afgeleverd, waarbij click deze pakketten binnenkrijgt via het element `FromDevice(eth0)`. Een click-script kan echter ook communiceren met software die bovenop het besturingssysteem draait. Hiervoor wordt gebruik gemaakt van virtuele interfaces, genaamd tun- of tap-interfaces, waarbij de eerste IP-interfaces (level 3) zijn en de tweede ethernet-interfaces (level 2). Wanneer click gebruikt wordt als userlevel software dienen deze interfaces ondersteund te worden door de kernel. Dit is in vrijwel elke linux-kernel het geval, waar deze geselecteerd kunnen worden in de opties als "Universal TUN/TAP device driver support" en "Ethertap network tap". Ze worden als module ingeladen met de commando's `modprobe tun` en `modprobe kerneltap`. Communicatie met de andere software gebeurt dan via deze virtuele interfaces die in click gebruikt kunnen worden via de elementen `FromDevice(tap0)` en `ToDevice(tap0)`. Wanneer we echter gebruik maken van de click-kernelmodule en deze compileren tegen de headers van de draaiende kernel, hebben we geen extra ondersteuning nodig van de kernel en kan een dergelijke virtuele device aangesproken worden met de blokken `FromHost(kerneltap, IP, ETHER MAC)` en `ToHost(kerneltap)` waarbij kerneltap de naam van de device is. Let wel: enkel tap-devices worden hier ondersteund.

Click werd oorspronkelijk ontwikkeld aan het gerenommeerde *Massachusetts Institute of Technology* (MIT) [9]. Door de broncode vrij te geven als vrije software wordt het nu verder ontwikkeld door een grote gemeenschap van ontwikkelaars over de hele wereld, waaronder zowel vrijwilligers als mensen van onderzoeksgroepen aan universiteiten of bedrijven, zoals de MIT *Parallel and Distributed Operating Systems group* (PDOS [10], onderdeel van MIT's *Computer Science and Artificial Intelligence Laboratory* [11]), *Mazu Networks* [12], het *International Computer Science Institute* (ICSI) [13] aan de *Berkeley University of California* [14] en het *Computer Science Department* [15] van de *Los Angeles University of California* [16].

Click is uiteraard nooit afgewerkt, maar de ervaring ermee in ondermeer dit werk is uitermate positief. Het is een uiterst flexibel en eenvoudig te gebruiken systeem dat vrij goed werkt. Ook het toevoegen van een element gaat vrij vlot wegens de goede omgeving

die men ter beschikking krijgt. Tijdens het onderzoek waren er nauwelijks problemen met click en doken er geen bugs op. De gebruikte versie is click 1.4.2, die vrijgegeven werd op 29 december 2004. De meest recente versie is 1.5.0, die het levenslicht zag op 19 mei 2006.

### 4.1.2 nsnam

Nsnam is op zijn beurt ook weer een combinatie van de ns—Network Simulator en nam—Network ANimator, die beiden meestal in één naam genoemd worden omdat ze veelal samen gebruikt worden.

#### ns-2

Ns is een discrete event simulator bedoeld voor netwerkonderzoek. Ns-2 is de tweede grote release van ns, wat op zijn beurt gebaseerd was op de REAL netwerk simulator [17]. Ns-2 is geprogrammeerd in C++ en Object Tcl (OTcl) en werd voor het eerst vrijgegeven in 1996. Ns maakt het mogelijk om simulaties uit te voeren van TCP-verkeer, UDP-verkeer, routing, multicast protocols, zowel via bedrade als draadloze netwerken alsook via satellietnetwerken. De gesimuleerde nodes kunnen zowel vaste nodes of mobiele nodes zijn die we met de simulator laten bewegen.

Ns wordt momenteel ontwikkeld door het *Information Sciences Institute* [18] van de *University of Southern California* [19], het *International Computer Science Institute*, het *Collaborative Simulation for Education and Research* project [20] alsook tal van vrijwilligers, onderzoekers en bedrijven waaronder Sun Microsystems. Ook dit project is vrije software. De gebruikte versie is 2.26, die dateert van 26 februari 2003. De meest recente versie is ns-2.29, vrijgegeven op 19 oktober 2005.

Ns wordt gebruikt door het opstellen van een tcl-script. Hierin worden eerst de parameters ingesteld door waarden toe te kennen aan variabelen. Zo bijvoorbeeld de afmetingen van het speelveld, de fysische parameters (frequentie ingeval van draadloze netwerken), het uitgezonden vermogen, het gebruikte MAC, de files waarnaar de traces geschreven worden,... Het simulatorobject dient ook aangemaakt te worden door een nieuwe instantie van de simulatorklasse aan te maken. Vervolgens worden nodes aangemaakt, worden hun posities ingesteld, worden interfaces eraan gekoppeld, worden adressen (MAC en IP) ingesteld en worden links (wired of wireless) aan deze interfaces gekoppeld. Hierna kunnen we agents aan deze nodes koppelen waarop we een testapplicatie kunnen laten lopen, bijvoorbeeld

een applicatie die een UDP-stroom van constante bitrate genereert. Uiteindelijk wordt de simulatorinstantie ingesteld door de tijdstippen op te geven waarop bepaalde applicaties beginnen te lopen, wanneer de simulatie stopt, wanneer nodes beginnen te bewegen, alsook wordt de opdracht gegeven de simulatie effectief te starten.

Met ns hebben we echter wel heel wat problemen gehad, waarover meer in 4.1.3.

### **nam**

Nam wordt meestal gebruikt in combinatie met ns. Het is een op Tcl/Tk gebaseerde animatietool om de traces, gegenereerd met ns of door reële pakketstromen te capteren, op een visuele manier te bestuderen. Het is mogelijk om er de topologie in weer te geven, alsook met animaties de beweging van elk individueel pakket voor te stellen. Nam bevat verder ook nog allerlei tools om de data te onderzoeken. Dit werkt goed voor bedrade verbindingen, bij draadloze simulaties is dit jammer genoeg niet werkbaar in de door ons gebruikte versie 2.26. Nam wordt momenteel ontwikkeld door dezelfde groepen achter ns. Het werd niet gebruikt in dit project buiten enkele experimentjes om voeling te krijgen met ns, maar wordt hier vermeld voor de volledigheid.

### **4.1.3 ns-click**

Ns-click [21, 22] is de combinatie van bovenstaande applicaties. Het laat toe om op ns-nodes een click-script te draaien. Hierboven kan dan nog altijd een agent draaien die zich gedraagt als een gewoon programma bovenop een click-pc en met click communiceert via de tun-devices, op laag 3 dus. Aan de click-scripts moet er vrijwel niets gewijzigd worden. Het enige dat verandert, is dat er geen echte ethernetdevices gebruikt worden, maar door ns gesimuleerde devices. Pakketten versturen en ontvangen gebeurt dan met de elementen `FromSimDevice(eth0)` en `ToSimDevice(eth0)`. Dit werkt heel goed voor bedrade netwerken. Voor draadloze netwerken is de combinatie nog niet beschikbaar. Echter, op de vakgroep is er een patch beschikbaar die het mogelijk maakt dat click-nodes gebruikt kunnen worden in een draadloos netwerk in ns. Deze patch werkt vrij goed: we hebben een voorbeeld van werkende click-scripts die de werking van een AP en een draadloos station implementeren.

Met ns-click traden echter wel heel wat problemen op. De patch voor het toelaten van click-nodes in wireless simulaties, is enkel compatibel met ns versie 2.26. Deze versie werd

vrijgegeven begin 2003, wat een eeuwigheid geleden is in deze sector. Bovendien werd IEEE 802.11g pas gestandaardiseerd in juni 2003, na de release van ns-2.26 dus. Tijdens het onderzoek werd hier in eerste instantie niet bij stilgestaan. We merkten dat er slechts een heel lage doorstroom mogelijk was, ook al stonden ns-parameters zoals `dataRate_` en `basicRate_` op 54 Mb. Het internet werd afgeschuimd op zoek naar geschikte parameters om de doorstroom te verhogen, maar deze werden niet gevonden. Pas in latere instantie zagen we de mogelijkheid dat enkel de fysieke laag van 802.11b correct geïmplementeerd was en er bijgevolg nooit dataverkeer aan 54 Mbps zou kunnen door geraken. Er werd dan wel een patch gevonden om zowel 802.11g als 802.11a mogelijkheden toe te voegen aan ns, maar deze was niet compatibel met de gebruikte (gepatchte) versie.

Hiernaast hebben we ook nog een groot aantal andere problemen ondervonden. Een aantal hebben we kunnen omzeilen, andere niet. Zo was er een bug in de scheduler van ns, die we na wat spuurwerk op het net hebben kunnen oplossen.

Blijkbaar is ns-click met de voorziene patches ook niet meteen geschikt om een logisch, draadloos, AP te draaien op een node met enkel vaste verbindingen. Immers, wanneer het AP start, het juiste kanaal selecteert en begint beacons uit te zenden, roept ns een procedure `SwitchChannel` op die (op ns-niveau) het juiste medium aan de juiste interface vasthecht. Wanneer we ons AP echter als een logische entiteit beschouwen in een node met enkel bedrade interfaces (de centrale), geeft dit problemen. Dit kon omzeild worden door deze procedure aan te passen en de centrale daar expliciet als uitzondering in op te nemen met een if-constructie, maar dit is geen al te propere oplossing en is niet wat het hoort te zijn.

Een ander probleem dat in eerste instantie niet kon opgelost worden, is het door ons gedoopte *kickstartprobleem*. Zoals in hoofdstuk 2 besproken, is het de bedoeling dat de centrale geen enkel pakket verstuurt totdat het eerste pakket afkomstig van de trein binnengekomen is. De centrale weet dan waar de trein zich bevindt en naar waar downlinkpakketten dienen gestuurd te worden. Dit blijkt echter niet te werken om totnogtoe onbekende redenen. Wanneer we een testschema uitwerken waarbij de trein eerst authenticceert en associeert met het AP en pas even later een UDP-stroom probeert te sturen naar een server in het bovenliggend netwerk, voorafgegaan door een ARP-request, geraakt er geen verkeer door gedurende een bepaalde tijd. Het eerste pakket dat verstuurd wordt, is het ARP-request, ondanks het feit dat het station nog niet geassocieerd is met het AP. We hadden het vermoeden dat dit kwam omdat het ARP-pakket het eerste broadcast pakket is. Het meest courante scenario is immers dat er eerst een broadcast pakket gestuurd wordt: ofwel een

beaconframe ofwel een probe request. Mogelijks heeft men zich hierop gebaseerd. Dit klopt echter niet, want bij het testen geraakt het probe request er ook pas door nadat het ARP-request verstuurd is. We hebben ons dan ook initieel moeten beperken tot het testen met een aangepast scenario, waarbij de centrale weet aan welke RAU de trein vertrekt en de beacons reeds verstuurt via die RAU, ook al is er nog geen pakket van de trein binnengekomen. Naderhand is wel een manier gevonden om dit probleem te omzeilen, waarover meer in 5.1.

Meer details over opgetreden problemen vindt u bij de testresultaten. Het mag echter duidelijk zijn dat de gebruikte versie van ns-click allesbehalve een goedwerkend systeem is voor onderzoek betreffende draadloze netwerken, zeker met de IEEE 802.11g- of a-standaard.

## 4.2 Implementatie

Uit het simulatiemodel van hoofdstuk 3 volgt dat er drie soorten nodes ontwikkeld moeten worden: een centrale, een base station en een trein. Deze laatste is alvast vrij eenvoudig: hiervoor kan een gewoon standaard draadloos station gebruikt worden. Voor de volledigheid wordt toch een click-schema getoond in 4.2.2. De andere twee elementen zijn iets complexer. Beiden worden hieronder dan ook uitgebreid besproken. Tot slot dienen we ook nog een extra node te voorzien in het bovenliggend netwerk waarnaar we verkeer vanaf de trein kunnen sturen en omgekeerd. We noemen deze node simpelweg *server*.

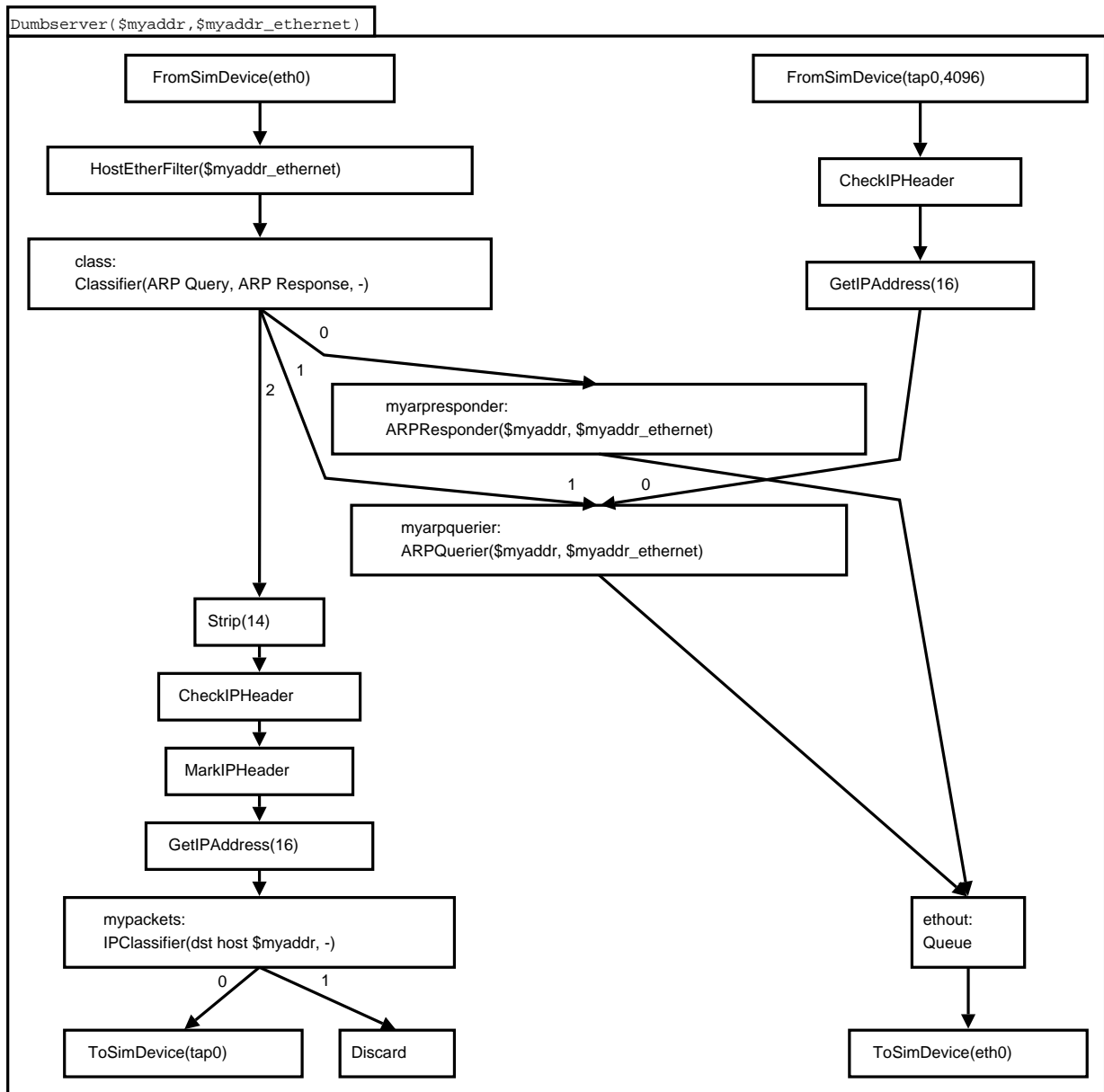
### 4.2.1 De server

Een click-schema is weergegeven in figuur 4.1. Deze server doet niets meer dan de binnengekomen IP-pakketten afkomstig van de kernel (of in ons geval de UDP-agent van onze simulator, ns) versturen. Hiervoor wordt eerst nagegaan of het MAC-adres gekend is, en indien nodig wordt eerst een ARP-query gestuurd en het antwoord afgewacht. In de omgekeerde richting worden pakketten afkomstig van de netwerkkaart eerst uitgesplitst in arp-replies, arp-request en andere pakketten, om vervolgens in deze laatste categorie na te gaan of de pakketten wel degelijk voor deze node bestemd zijn. Uiteraard worden ook de nodige controles op de headers uitgevoerd.

Voor de implementatie werd gekozen om een klasse dumbserver te maken, waarvan we nieu-

---

we instanties kunnen maken. Dit maakt het mogelijk om op eenvoudige wijze dergelijke functionaliteit te voorzien voor meerdere netwerkinterfaces. De voorziene functionaliteit in deze elementklasse omhelst het afhandelen van ARP-requests voor het versturen van IP-pakketten, filteren op IP- en MAC-adres, verwijderen van de ethernetheader voor binnenkomende pakketten en het toevoegen van de ethernetheader aan uitgaande pakketten.



**Figuur 4.1:** Click-schema van de server. Er wordt een instantie van de klasse `dumbserver` gecreëerd met `naam :: dumbserver(IP-adres, MAC-adres);`.



### 4.2.2 De trein

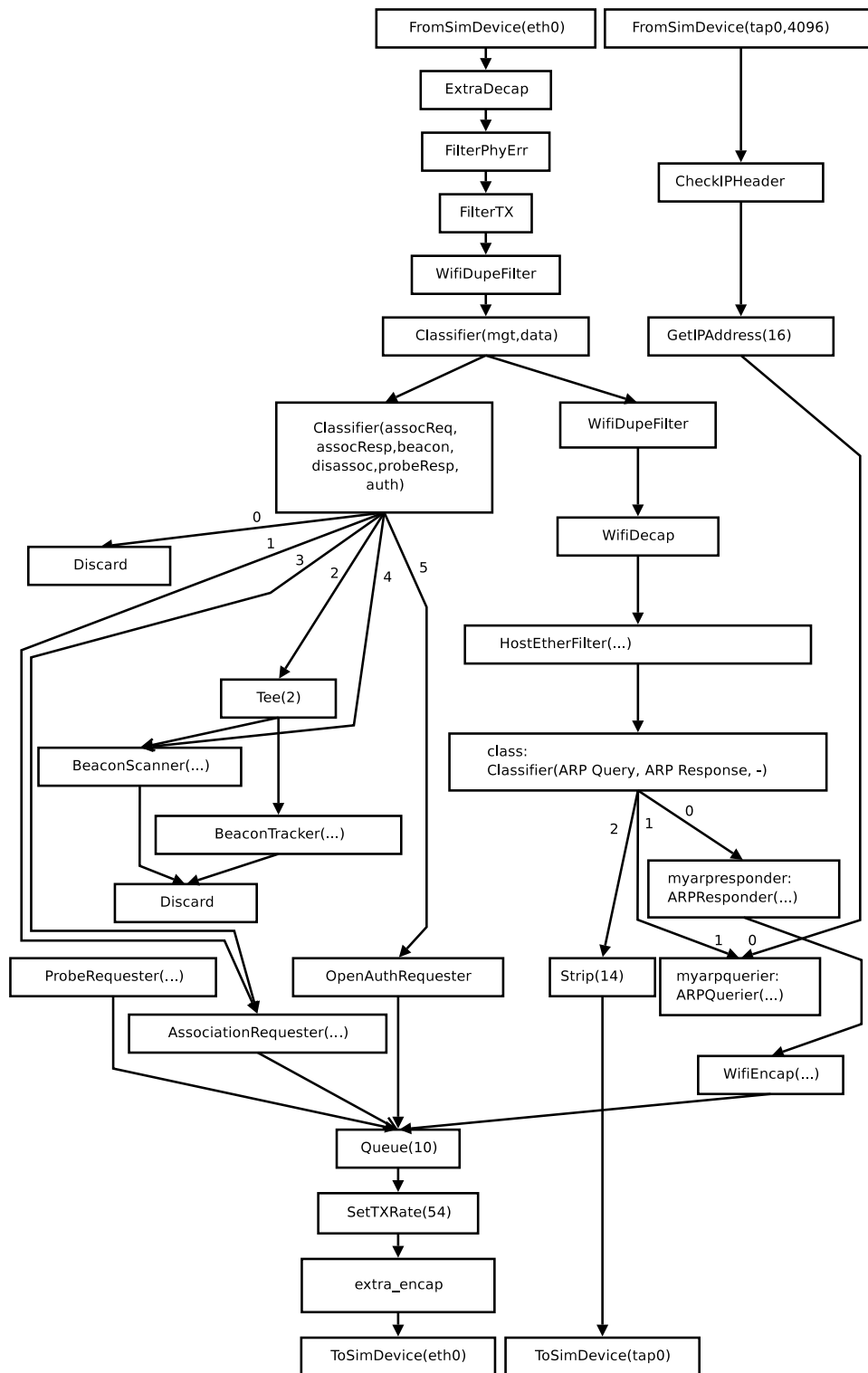
De trein is niet meer dan een standaard wireless station waarvoor een click-script ter beschikking gesteld werd. Dit werd uitgebreid met ARP-functionaliteiten zoals deze aanwezig zijn in de server. Een schematische voorstelling is weergegeven in figuur 4.2. Net als bij de server komen IP-pakketten van de kernel of bovenliggende software binnen via device tap0, en worden ook enkel IP-pakketten aldaar afgeleverd.

### 4.2.3 De Remote Antenna Units

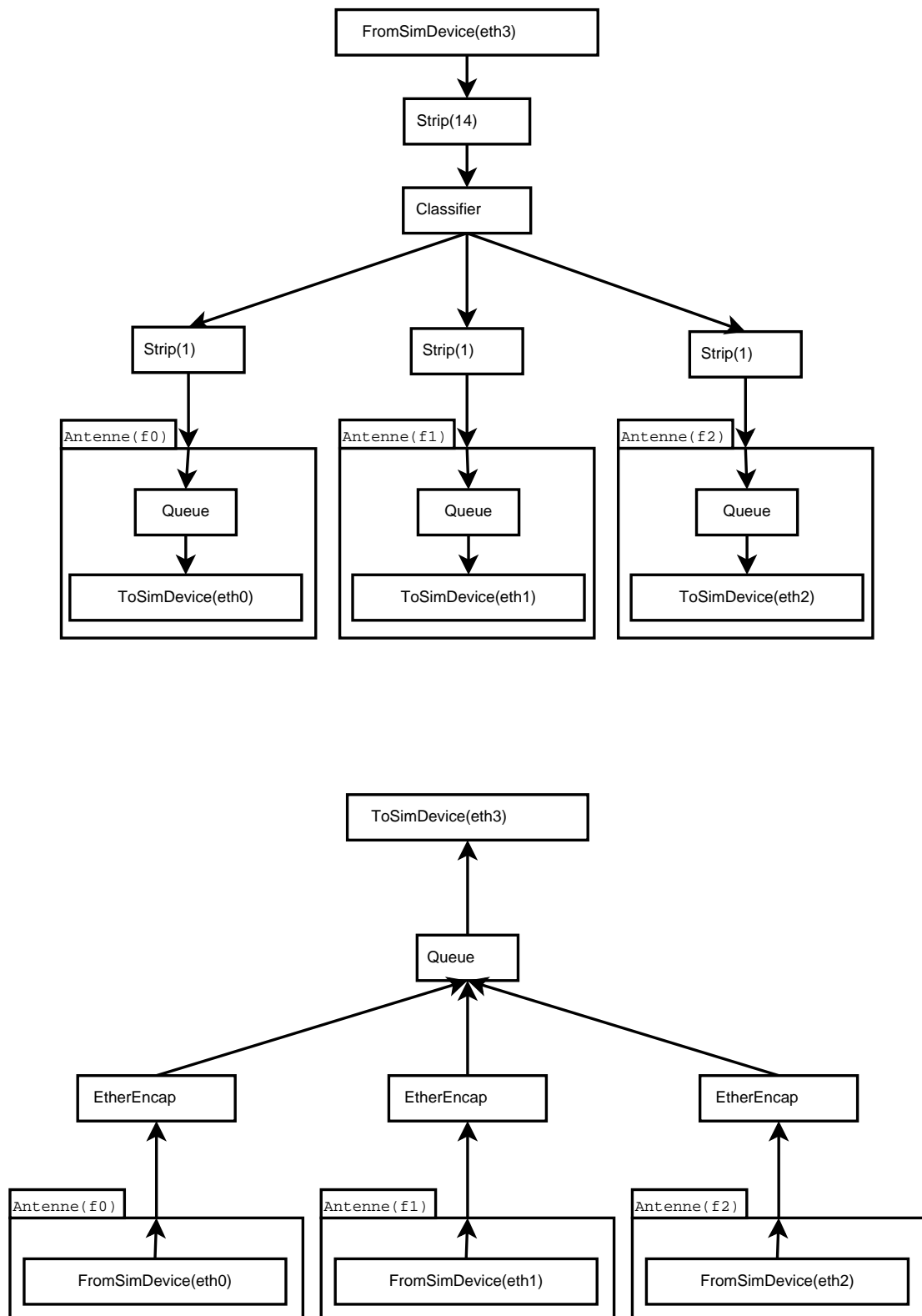
De RAU is een compleet nieuw element. Het is echter wel bijzonder eenvoudig, wat trouwens een vereiste is om een lage kostprijs te beogen. Uplinkpakketten worden geïncapsuleerd (samen met de ontvangstinformatie in de vorm van extra-headers) in een normaal ethernetpakket en naar de centrale verstuurd. Van downlinkpakketten wordt eerst de ethernetheader afgestript, waarna dit pakket, afhankelijk van de eerste byte in dit resulterend pakket, naar de juiste interface gestuurd wordt. Niet meer dan Strip-, EtherEncap- en Classifier-elementen dus, naast de interface-elementen. De enige extra elementen (Tee,...) zijn elementen die het mogelijk maken om een gestripte versie naar de dumpfiles te schrijven zodat deze herkenbaar zijn voor analysepakketten zoals ethereal. Een schematische voorstelling is te vinden in figuur 4.3.

### 4.2.4 De centrale

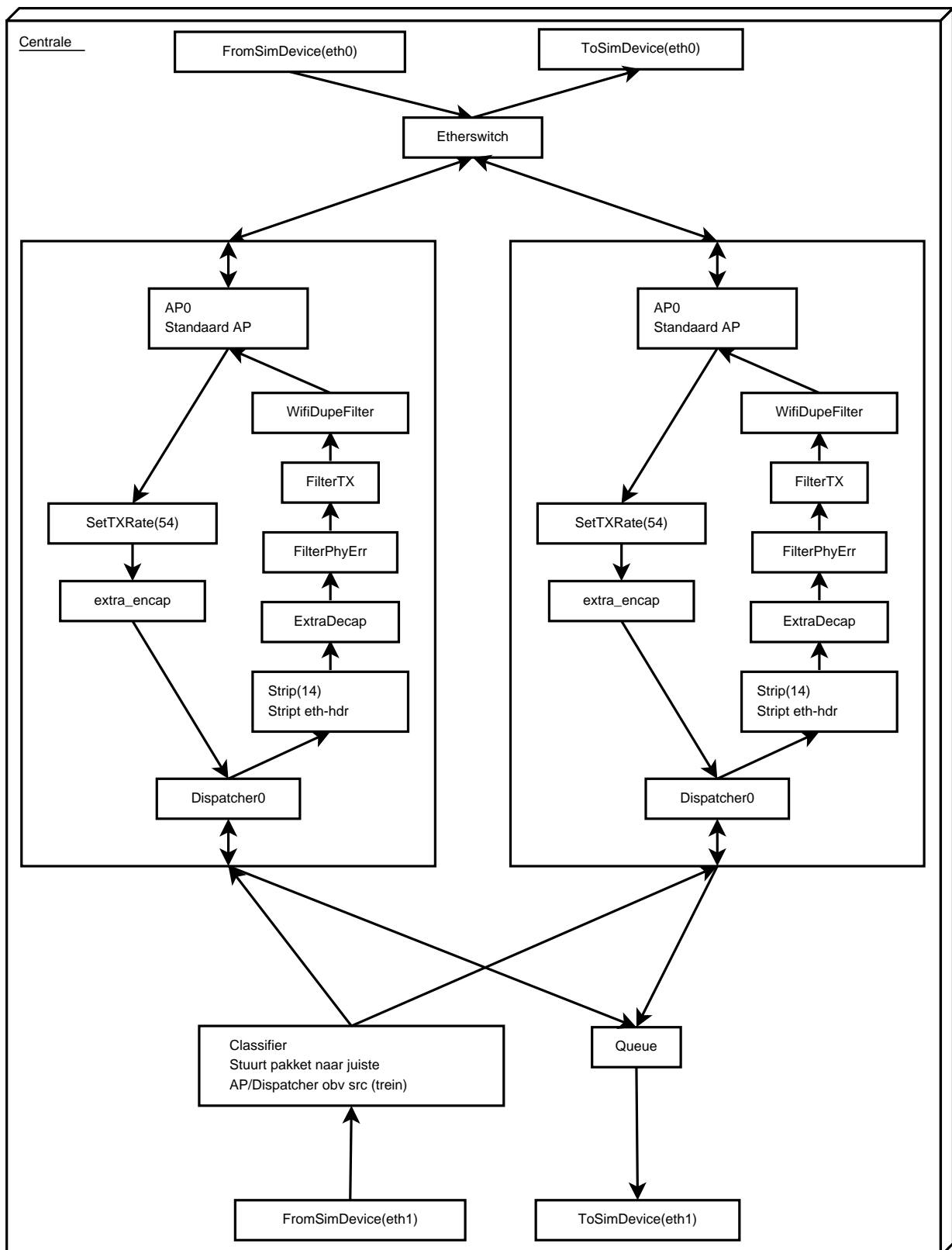
De centrale is eveneens een compleet nieuw element. Het schema van de implementatie met behulp van click-elementen is weergegeven in figuur 4.4. Dit is een implementatie met twee logische AP's, die dus twee treinen kan bedienen. Het valt dan ook meteen op in de figuur dat een bepaalde groep van samenhangende elementen zich twee maal voordoet. In deze groep van elementen is niet toevallig ook een standaard AP te vinden. We zullen deze groep van samenhangende elementen een *logisch blok* noemen. Elk logisch blok staat in voor het volgen van één trein.



Figuur 4.2: Click-schema van de trein.



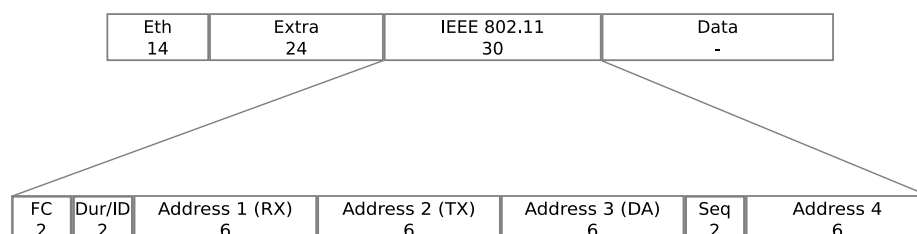
**Figuur 4.3:** Click-schema van een RAU. De constructies voor het bekomen van leesbare dumpfiles werden weggelaten.



**Figuur 4.4:** Click-schema van de centrale.

Bovenaan het schema bevindt zich de uplink-interface. Een element **EtherSwitch** verbindt de logische blokken met de uplink. Dit element werkt volgens hetzelfde principe als een gewone switch en zal het verkeer bestemd voor een bepaalde trein naar het juiste blok sturen, alsook uplinkverkeer van de verschillende blokken samenvoegen en naar de uplinkinterface sturen.

Onderaan het schema bevindt zich de downlink-interface die verbonden is met de base stations. Hier zorgt een classifier voor het verdelen van het binnenkomende verkeer, afkomstig van de RAU's, naar de juiste blokken. Aangezien elk blok één trein bedient, kunnen we de stroom gewoon verdelen op basis van het MAC-adres van de afzender van het draadloze pakket. Let op: dit is niet het pakket dat binnenkomt aan de eth1-interface, maar het pakket dat zich daarin bevindt. De draadloze pakketten werden door de RAU's immers, samen met de ontvangstinformatie, ingecapsuleerd in een ethernetpakket. Zie figuur 4.5 voor een voorbeeld van een dergelijk pakket. Het MAC-adres van de afzender is adres 2 (TX). Voor het downlinkverkeer volstaat het samenvoegen van de stroom van de logische blokken in een queue, waarna dit de centrale verlaat. Het adres van het RAU waar naar verstuurd wordt, dient reeds ingevuld te worden in het logische blok. Bemerkt dat we voor de upstreampakketten even goed konden filteren op basis van het BSSID, wat terug te vinden is als adres 1 (RX) in figuur 4.5.



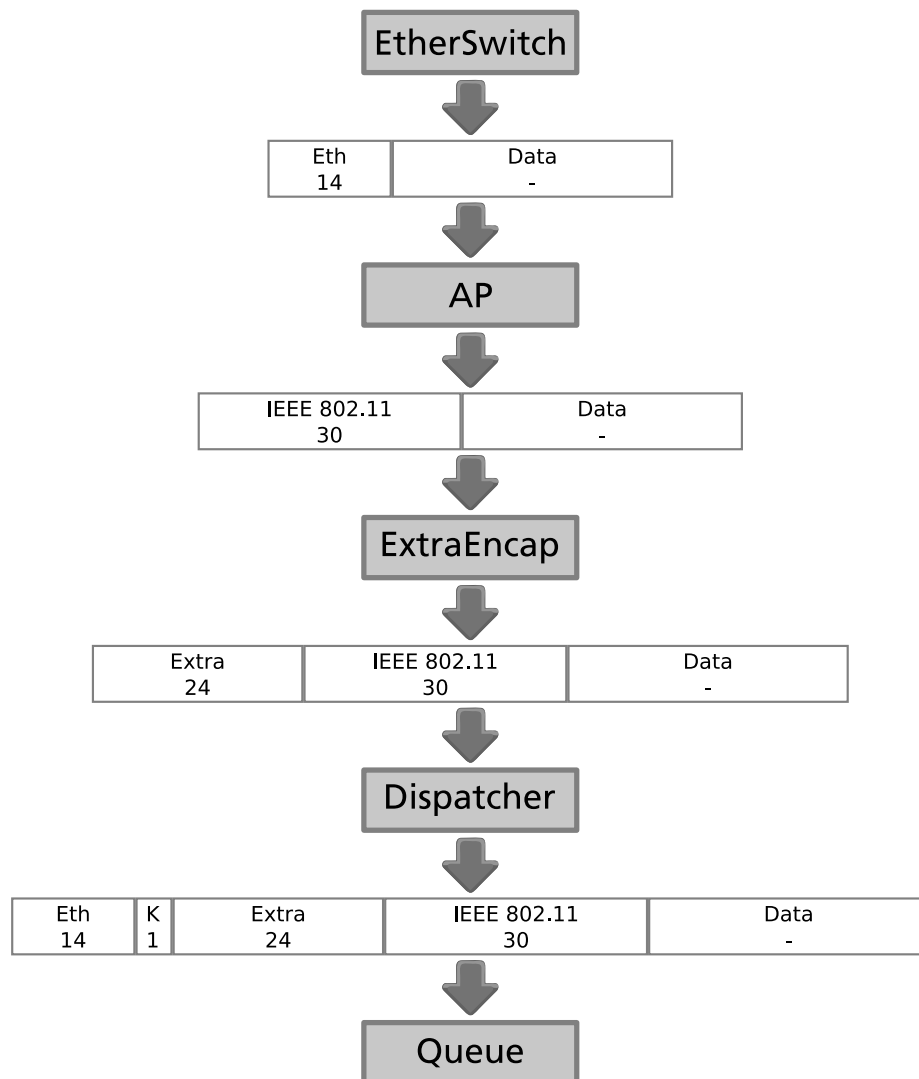
**Figuur 4.5:** Voorbeeld van een pakket verzonden door een RAU naar de centrale, met details van de WiFi-header. De lengte van de headers is vermeld in bytes.

Tijd om de logische blokken meer in detail te bekijken. Het eerste wat opvalt, is het nieuwe element **Dispatcher**. Dit element werd in dit werk gecreëerd en zorgt ervoor dat de downstreampakketten naar de juiste RAU gerouteerd worden. Zoals beschreven in 2.3.2 gebeurt dit door telkens de RAU van het laatst ontvangen upstreampakket bij te houden. Hier kunnen we gewoon het MAC-adres van deze RAU gebruiken als referentie. De dispatcher bevat intern dus twee variabelen: *vorige RAU* en *huidige RAU*, beiden een MAC-adres. Komt er een upstreampakket binnen, dan wordt *huidige RAU* vervangen door

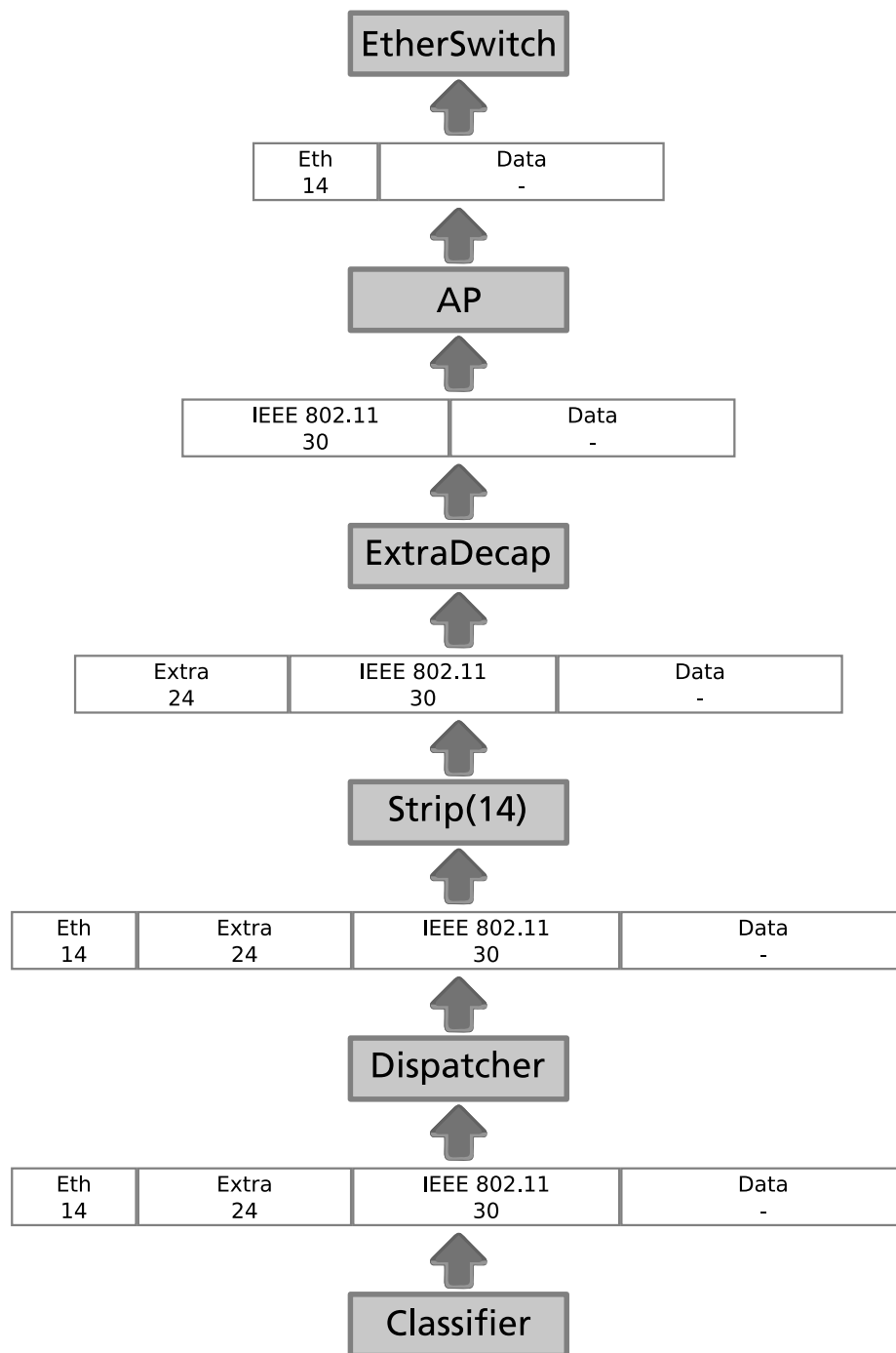
het afzenderadres van dit upstreampakket. Dit echter enkel indien *vorige RAU* van deze waarde verschilt. Dit om te vermijden dat wanneer de trein zich in de overlapzone bevindt, er telkens van RAU gewisseld wordt. Een downstreampakket wordt door de dispatcher voorzien van een header die het kanaal voorstelt waarop dit pakket dient uitgezonden te worden, om vervolgens ingecapsuleerd te worden in een ethernetpakket met als bestemming *huidige RAU*, zoals uitgelegd in hoofdstuk 3 en weergegeven in het onderste pakket van figuur 4.6. Het afzenderadres is uiteraard het MAC-adres van interface eth1. Indien de interne toestand nog leeg is omdat er nog geen uplinkpakket binnenkwam, wordt het pakket naar een aparte uitgang gestuurd waar het normaal gedropt wordt, maar desgewenst ook geanalyseerd kan worden. Merk op dat deze dispatcher geen weet hoeft te hebben van de gemeenschappelijke toestand van alle treinen. Op die manier is het gehele systeem makkelijk schaalbaar voor meerdere treinen, voor zover er verschillende kanalen voorhanden zijn natuurlijk.

Volgen we nu een downstreampakket dat binnenkomt aan interface eth0. Dit wordt door de switch naar het juiste blok geleid, alwaar het doorgegeven wordt aan het AP. Dit AP is een elementklasse die de werking van een standaard AP implementeert en ethernetpakketten omzet in IEEE 802.11 pakketten. Uiteraard kan dit element ook zelf pakketten genereren, zoals beacon frames of andere management frames. De extra informatie (voor de driver van de netwerkkaart *ed*) wordt aan het pakket toegevoegd door het element *ExtraEncap*, weliswaar nadat het vermogen ingesteld werd. Dit wordt dan doorgegeven aan de dispatcher die, volgens de beschrijving hierboven, de gegevens inpakt in een ethernetpakket en naar de juiste RAU stuurt. Deze bewerkingen van de headers door de click-elementen, is weergegeven in figuur 4.6.

Voor een upstreampakket dienen we net de omgekeerde weg te bewandelen, zoals uitgetekend in figuur 4.7. Het pakket komt binnen aan de eth1-interface en wordt door de classifier aan het juiste blok afgeleverd. Daar passeert het eerst langs de dispatcher die, indien nodig, zijn interne toestand aanpast en het pakket ongewijzigd weer uitspuwt. Vervolgens wordt het ontdaan van de ethernetheader, die aangebracht werd door de RAU, waarna met behulp van *ExtraDecap* de ontvangstinformatie in de click-annotations ingevuld wordt. Na controle van afzender, duplicaten en errors, wordt het aan het access point afgeleverd die het pakket verder behandelt en het indien nodig als regulier ethernetpakket via de switch naar de uplink-interface stuurt.



**Figuur 4.6:** Schematisch overzicht van de bewerkingen door de click-elementen op de headers van de pakketten bestemd voor de trein. De lengte van de headers is vermeld in bytes.



**Figuur 4.7:** Schematisch overzicht van de bewerkingen door de click-elementen op de headers van de pakketten afkomstig van de trein. De lengte van de headers is vermeld in bytes.



## Hoofdstuk 5

# Simulatieresultaten

### 5.1 Wijzigingen aan de implementatie

#### 5.1.1 Het kickstartprobleem

In principe zou bovenstaande uitwerking moeten werken. Helaas, dit was buiten ns-click gerekend waarin blijkbaar enkele rare implementatiekeuzes gemaakt werden of waar gewoon nog een aantal bugs in zitten. Zo hadden we de rare situatie dat wanneer de dispatcher geïmplementeerd was zoals het hoort en het downstreamverkeer dus geblokkeerd werd totdat een upstreampakket ontvangen werd, het geheel niet werkte. Pas vanaf 3.1 s kon het eerste upstreampakket verzonden worden. Wanneer we de dispatcher initialiseren, en deze vanaf het begin downstreamverkeer doorlaat naar een vooraf gekozen RAU, werkt alles naar behoren. Althans zo lang de trein zich ter hoogte van de eerste RAU bevindt. Vanaf er overgeschakeld wordt op de tweede RAU, geraakt er slechts 10% van het upstreamverkeer meer verzonden, terwijl er voor het downstreamverkeer geen probleem is. We hebben geen idee hoe dit komt, maar de kans bestaat dat de oorzaak in dezelfde richting moet gezocht worden als deze die problemen heeft bij de eerste RAU wanneer de dispatcher oorspronkelijk al het verkeer blokkeert. Dit probleem werd reeds aangebracht als het *kickstartprobleem* in sectie 4.1.3.

#### 5.1.2 Verwisselen AP- en station-functionaliteit

Zoals uit bovenstaande paragraaf blijkt, moet het eerste verkeer afkomstig zijn van het AP. Echter, als we aan het principe willen vasthouden dat het eerste verkeer afkomstig

moet zijn van de trein, dan lijkt de conclusie eenvoudig: plaats het AP op de trein en laat de centrale voor draadloze client doorgaan. In wezen mag dit geen verschil maken aangezien we toch met één station per AP werken. Station en AP vormen samen niet meer dan een tunnel waardoor gewoon ethernet- of IP-verkeer gestuurd kan worden. Het enige verschil is dat het AP normaal beacon frames verstuurt en dat het station zich moet authenticeren en associëren met het AP. Bovendien levert dit een bijkomend voordeel op. Om een goede werking mogelijk te maken, moet de trein regelmatig verkeer naar de centrale sturen. Indien er geen verkeer voorhanden is, gingen we random informatie versturen. Als de trein echter beacon frames uitzendt, dan hebben we meteen een periodieke bron van verkeer en moet daar ook niets meer voor voorzien worden.

Na het verwisselen van de AP- en station-functionaliteiten blijkt alles naar behoren te werken. Het verkeer start, met de correcte dispatcher, onmiddellijk ipv na 3.1 s en er zijn geen problemen meer nadat op een nieuw RAU werd overgegaan. De bespreking van de resultaten is dan ook van toepassing op deze kleine ingreep, die hoegenaamd niets verandert aan de werking van het geheel buiten deze simulatie-omgeving.

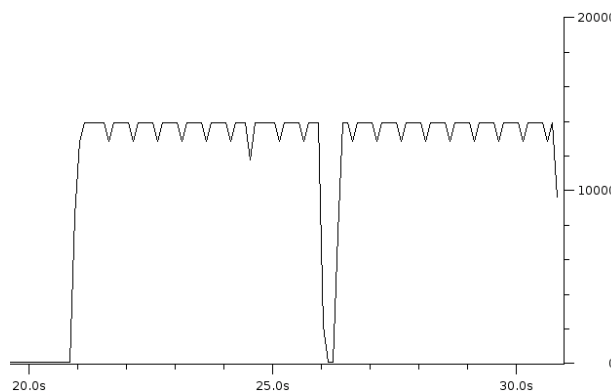
## 5.2 Referentie

Vooraleer over te gaan tot een gedetailleerde bespreking van de resultaten, is het best eerst over referentiewaarden te beschikken van een klassiek systeem. Het is onzinnig om een vrijwillige handover als basis voor onze vergelijking te nemen. Dit is immers afhankelijk van de implementatie van het mechanisme dat bepaalt wanneer het signaal van het oude AP te zwak wordt en dat van het nieuwe sterk genoeg geworden is, en er tot een handover overgegaan wordt. Bovendien wacht een standaard client nog op een ACK-bericht na het sturen van de disassociate, wat bij een slecht signaal wel heel lang kan duren na enkele retries. In onze simulaties beslissen we zelf wanneer we overschakelen en is het signaal tijdens het overschakelen altijd goed genoeg. Van zodra we de handover uitgevoerd hebben, wordt het signaal enkel nog beter. We vergelijken dus best met een test met gedwongen handovers, waarbij we zelf beslissen om het station te laten associëren met een nieuw AP terwijl de signaalkwaliteit van beide AP's constant blijft op een goede waarde. Testresultaten met gedwongen handovers op het goede moment leveren betere resultaten op dan wanneer de beslissing overgelaten wordt aan het station dat geen idee heeft waar het zich bevindt en welk signaal er sterker of zwakker zal worden. Als ons systeem even goed of beter scoort dan een vergelijkbaar systeem met gedwongen handovers, dan weten we ook

waar we staan ten opzichte van de situatie met door het station geïnitieerde handovers.

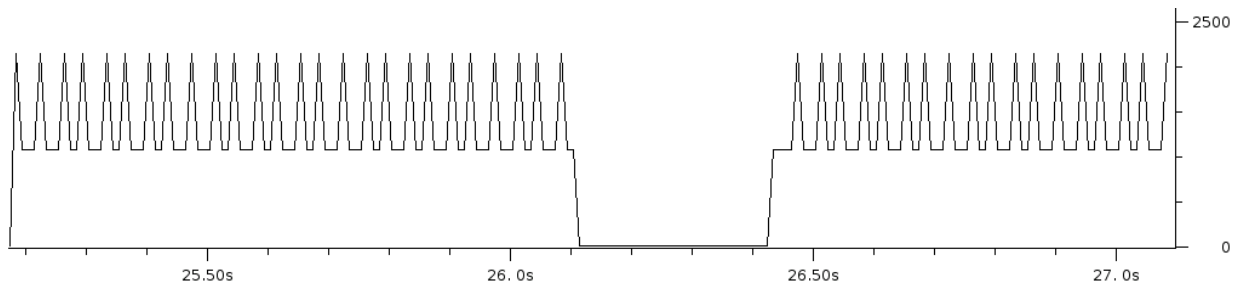
### 5.2.1 Hardwaretest

Een eerste test waarmee we kunnen vergelijken, is een hardwaretest die deze zomer uitgevoerd werd tijdens een stage. Hiervoor hernemen we grafieken C.8 en C.9 van pagina's 43 en 44 van het stageverslag [3] over als figuren 5.1 en 5.2. Beiden zijn het resultaat van het sturen van een UDP-teststroom van +/- 1 Mbps vanaf een server in het bovenliggend netwerk naar het station. Figuur 5.2 is een detailopname van de handover. 5 s na het begin van de datastroom werd tot een handover overgegaan, die duidelijk te herkennen is aan de *gap* van +/- 300 ms in de grafiek gedurende dewelke geen verkeer door de draadloze brug mogelijk was. De AP's zijn Chantry Beaconpoints, die sinds de overname door Siemens door het leven gaan als Siemens HiPath. Het station is een door Linux aangedreven laptop met Intel IPW2200 interne WiFi-kaart.



**Figuur 5.1:** Analyse van een UDP-teststroom aan +/- 1 Mbps met na de 26<sup>e</sup> seconde (5 s na het begin van de datastroom) een handover. De horizontale as is de tijdsas, verticaal staat de doorstroom in *Bytes per tick*. Eén tick is 0,1 s waardoor de waarde 10.000 op de grafiek overeenkomt met 100.000 Bps of 800 Kbps.

Bij deze vergelijkingsbasis dienen uiteraard de nodige kritische bedenkingen aangebracht te worden. Deze hardwaresimulatie werd uitgevoerd op een platform dat ontwikkeld werd voor snelle handovers. Bovendien werd hier gewerkt met de IEEE 802.11g standaard in plaats van de b-standaard die wij gebruiken. Verder is ns-click echt niet geschikt voor doorstroomtesten aangezien er een beperking zit op de verkeersstroom die men met ns door een draadloze brug kan sturen.

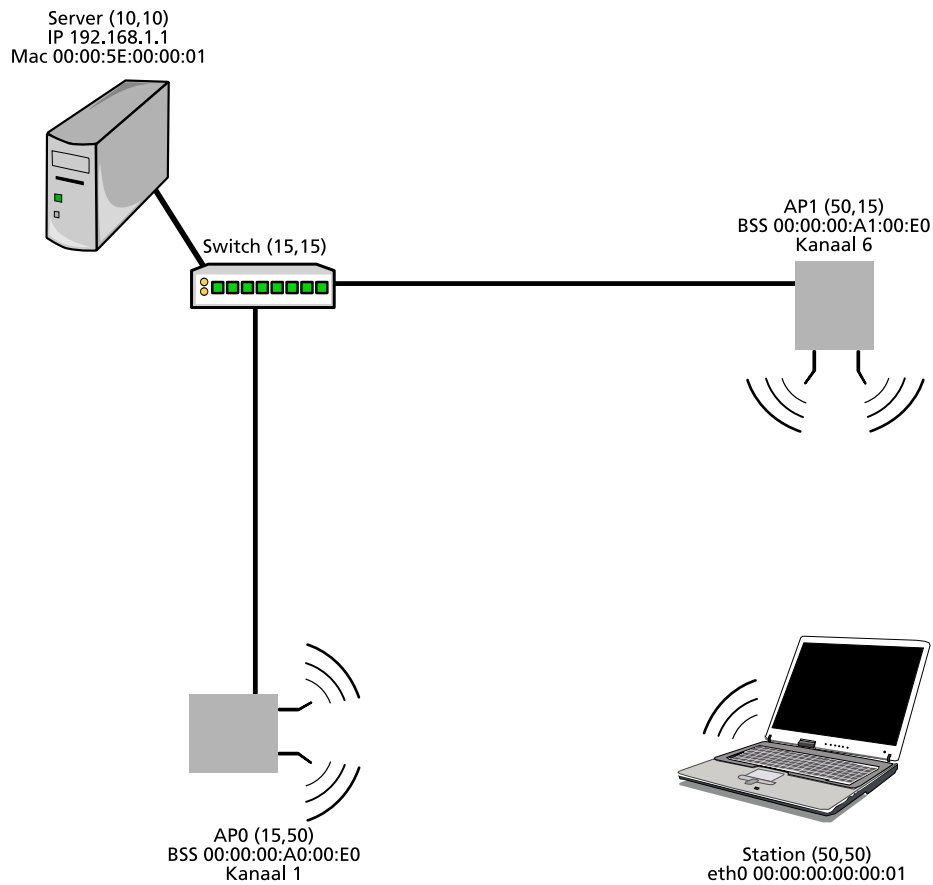


**Figuur 5.2:** Detail van de handover van figuur 5.1: tijdens een UDP-teststroom aan +/- 1 Mbps werd na de 26<sup>e</sup> seconde (5 s na het begin van de stroom) tot een handover overgegaan. De horizontale as is de tijdsas, verticaal staat de doorstroom in *Bytes per tick*. Eén tick is 0,01 s waardoor de waarde 1.000 op de grafiek overeenkomt met 100.000 Bps of 800 Kbps.

### 5.2.2 Simulatie met conventionele AP's in ns-click

Een betere basis om mee te vergelijken, zou het nabootsen van een klassieke situatie zijn in ns-click. Op die manier worden simulatieresultaten afkomstig van dezelfde simulator, met dezelfde eigenschappen en beperkingen, met elkaar vergeleken. Dit is echter ook niet evident. De modellen voor een AP en station die gebruikt worden in ns-click zijn slechts basismodellen. De standaard wordt (grotendeels) gevolgd, maar het blijft toch mogelijk om pakketten te versturen over de draadloze brug ook zonder dat het station geassocieerd is met het AP. Een voorbeeld van een dergelijke gecapteerde pakketstroom is te vinden in A.1. Als het station overschakelt op een nieuw AP, dan kan deze al onmiddellijk upstreampakketten sturen via het nieuwe AP, nog voor het geassocieerd is, waarna de bovenliggende switch onmiddellijk de downstreampakketten naar het nieuwe AP stuurt. De pakketten in de buffers van het oude AP gaan dan verloren, tenzij deze door de node in het bovenliggend netwerk opnieuw verstuurd worden. Indien het station geen verkeer naar een node in het bovenliggende netwerk stuurt, zal de switch ook nooit weten dat het station bereikt moet worden via een nieuw AP en zullen de pakketten nog steeds enkel bij het oude AP toekomen alwaar ze de buffers laten overlopen en verloren gaan. In bovenstaande hardwaretest gaan geen pakketten verloren: het oude AP stuurt deze naar het nieuwe AP via het Inter Access Point Protocol (IAPP), dat momenteel gestandaardiseerd wordt, of via een eigen IAPP. Het is echter niet omdat men direct pakketten kan versturen na het overschakelen op een nieuw AP, dat er geen handovertijd zou zijn. Het omschakelen op een nieuwe frequentie en het synchroniseren met een zender op die frequentie vergt ook enige

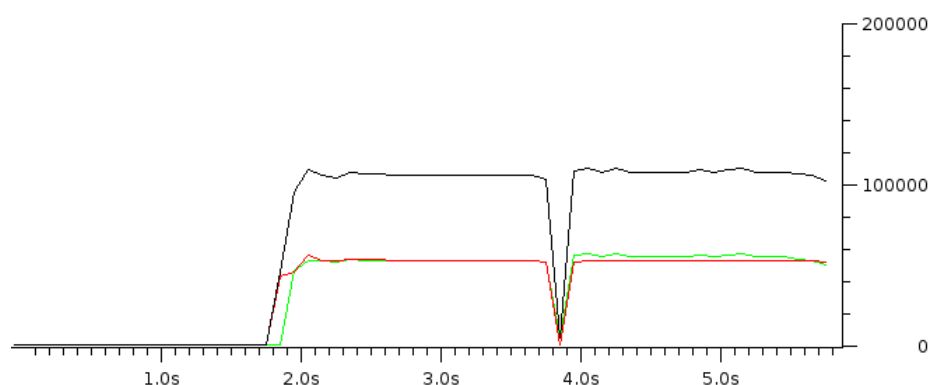
tijd.



**Figuur 5.3:** Testopstelling voor het bepalen van een vergelijkingsbasis.

Er werd een testopstelling aangemaakt in ns-click zoals afgebeeld in figuur 5.3. De nodes zijn vast gepositioneerd op de opgegeven coördinaten. Na 1 s stelt het station zich in op kanaal 1 en authenticereert het zich bij AP0 om 10 ms later ermee te associëren. 2 s na het begin van de simulatie wordt een UDP-teststroom aan een bitrate van 3,6 Mbps gestuurd van de server naar het station. 100 ms later begint een gelijkaardige bitstroom in de omgekeerde richting. Na 2 s verandert het station naar kanaal 6 om zich te authenticeren bij AP1. Dat er ondertussen al pakketten gestuurd worden, is af te lezen in bijlage A: gedetailleerde resultaten. De simulatie wordt afgebroken na 6 seconden. De resultaten zijn grafisch uitgezet in figuur 5.4. De zwarte lijn stelt de totale doorstroom door de draadloze link voor. Groen en rood staan voor de stroom van het station naar de server, respectievelijk van de server naar het station. Er is duidelijk een gap te zien op het moment

dat overgeschakeld wordt op de nieuwe frequentie. Dit moment is uitvergroot in figuur 5.5. Deze gap is 100 ms breed voor de rode lijn en 90 ms voor de andere lijnen. Dit verschil is te verklaren door het feit dat pas vanaf het eerste upstream UDP-pakket de switch bereikt heeft, deze overschakelt en aankomende downlink UDP-pakketten naar het nieuwe AP stuurt.



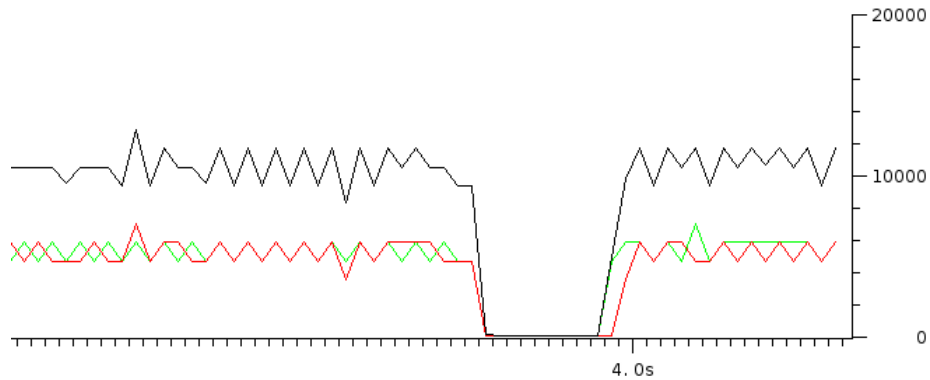
**Figuur 5.4:** Doorstroom van een UDP-teststroom bij 2 gewone AP's. De horizontale as is de tijdsas, verticaal staat de doorstroom in *Bytes per tick*. Eén tick is 0,1 s waardoor de waarde 100.000 op de grafiek overeenkomt met 1.000.000 Bps of 8 Mbps.

## 5.3 Resultaten van het systeem op basis van Radio-over-Fiber

Zoals vermeld in 2.3, is dit een uiterst flexibel systeem. Meerdere (kruisende) treinen, meerdere kanalen per trein, er is heel wat mogelijk. We zullen in deze sectie een paar van deze voorbeelden uitwerken en de simulatieresultaten bespreken. We beginnen uiteraard bij een basisgeval om de werking van het moving cell principe te illustreren.

### 5.3.1 1 trein, 2 remote antenna units

De testopstelling is geïllustreerd in figuur 5.6. Het bereik van de draadloze nodes is afgesteld op 105 m, zodat we een overlapzone van 10 m bekommen. Het grote voordeel aan deze manier van werken, is dat wanneer er overgeschakeld wordt op een nieuwe RAU, eventuele data die nog onderweg is van de centrale naar de oude RAU alsnog afgeleverd kan worden op

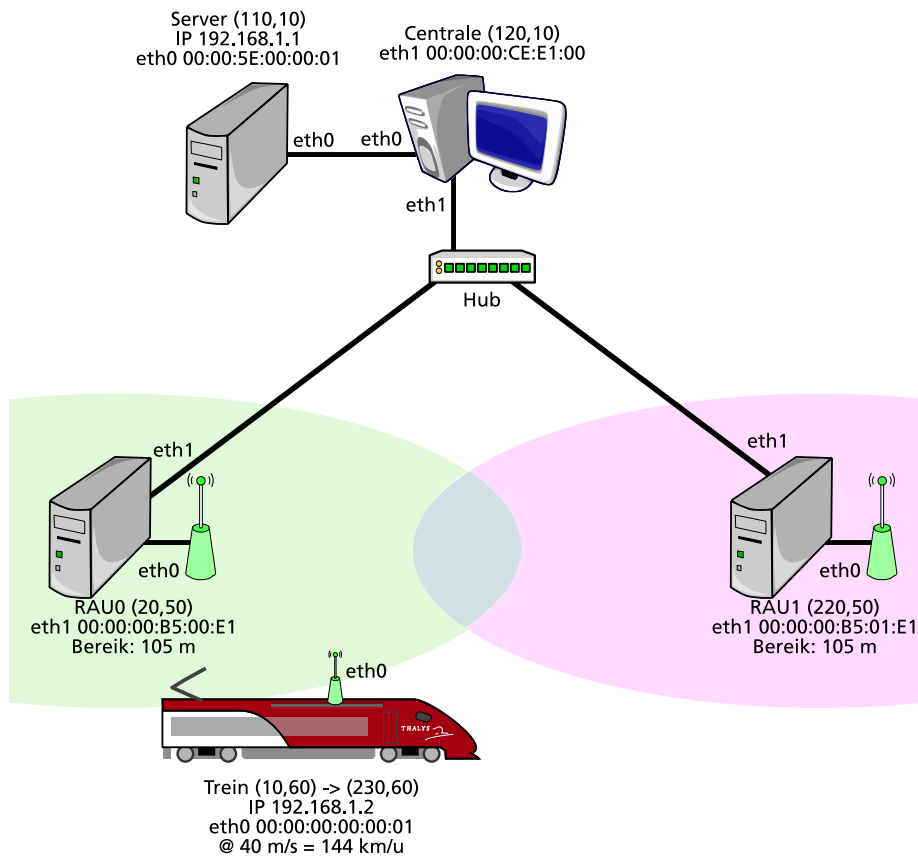


**Figuur 5.5:** Detailopname van de handover tijdens de doorstroom van een UDP-teststroom bij 2 gewone AP's, zoals weergegeven in figuur 5.4. De horizontale as is de tijdsas, verticaal staat de doorstroom in *Bytes per tick*. Eén tick (hier één klein streepje op de horizontale as) is 0,01 s waardoor de waarde 10.000 op de grafiek overeenkomt met 1.000.000 Bps of 8 Mbps.

de trein. De trein is ondertussen overgeschakeld op de nieuwe RAU en de signaalkwaliteit zal alleen maar verbeteren, tot wanneer de trein de RAU voorbijrijdt.

Initieel staat de trein stil op de beginpositie. Het AP in de trein is reeds actief vanaf het begin en zendt op regelmatige tijdstippen beacon frames uit. Na 1 s gaat het station in de centrale een probe request uitsturen naar het AP dat zich op de trein bevindt, van waaruit een antwoord teruggestuurd wordt. 500 ms later authenticceert het station zich, om 100 ms later te associëren met het AP. 2 seconden na het begin van de simulatie, start de trein zijn beweging aan 40 m/s, wat overeenkomt met 144 km/u. Op seconde 3 start vervolgens een bitstroom aan 3,6 Mbps van de server naar de trein. Om een nog onbekende reden is de trougput van de trein naar de centrale bijzonder laag. De omgekeerde richting, van de server naar de trein, is echter het interessantst van al. Het is deze stroom die naar de juiste RAU gerouteerd moet worden om op de trein terecht te komen. Aan de trein zelf verandert er niets: het is steeds hetzelfde station (in de centrale) dat geassocieerd blijft met het AP in de trein en deze laatste verandert ook niet van frequentie of andere parameters. Enkel in de centrale gebeurt de omschakeling naar de nieuwe RAU. Na 8 s is het experiment voorbij.

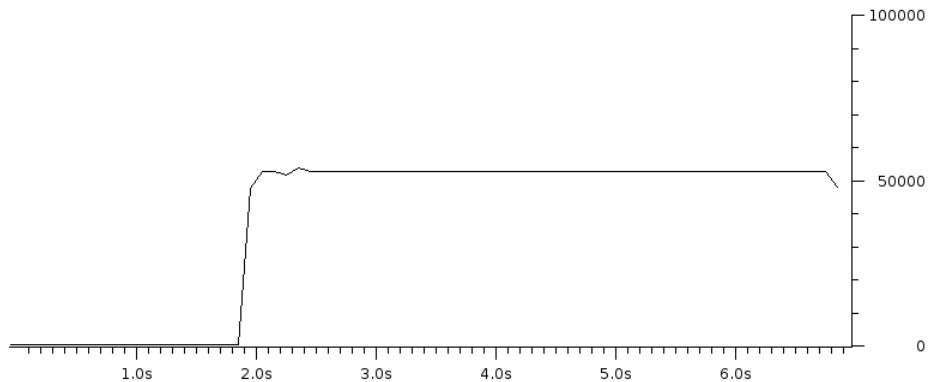
Een analyse van de bitstroom levert bijzonder interessante informatie op. Het resultaat is grafisch weergegeven in figuur 5.7. In tegenstelling tot de referentiemetingen is hier helemaal geen gap of dipje te merken in de doorstroom. De stroom is continu.



**Figuur 5.6:** Testopstelling met 1 trein en 2 RAU's.

Een ander interessant experiment, is het meten van de bitstroom aan de centrale, meer bepaald het verkeer van en naar de RAU's. Als we dat verkeer opsplitsen per RAU, zien we hoe lang de trein zich in de overlapzone bevindt en wanneer de centrale omgeschakeld is naar de nieuwe RAU. Om een en ander beter zichtbaar te maken, sturen we ook een kleine UDP-teststroom van 76 kbps van de trein naar de server. Op die manier hebben we iets meer verkeer en bekomen we kwalitatievere metingen. De resultaten zijn te zien in figuur 5.8. We zien dat vanaf het eerste pakket van de tweede RAU (paarse lijn) binnenkomt op seconde 4,35, de centrale overschakelt en het verkeer naar RAU0 (de rode lijn) volledig naar nul gaat. Op de grafiek lijkt dit niet onmiddellijk te gaan, maar ongeveer 200 ms te duren. Dit klopt niet: dit zou moeten een rechte verticale lijn zijn. Dit is beter zichtbaar in een detailopname van de handover, weergegeven in figuur 5.9. De trein bevindt zich in de overlapzone als de uitgezonden pakketten opgepikt worden door zowel RAU0 en RAU1, wat op de grafiek te zien is als de zone waar de paarse en blauwe lijn samen verschillen van nul. Dit is het geval vanaf 4,3 s tot 4,75 s. De centrale is echter al vanaf het begin





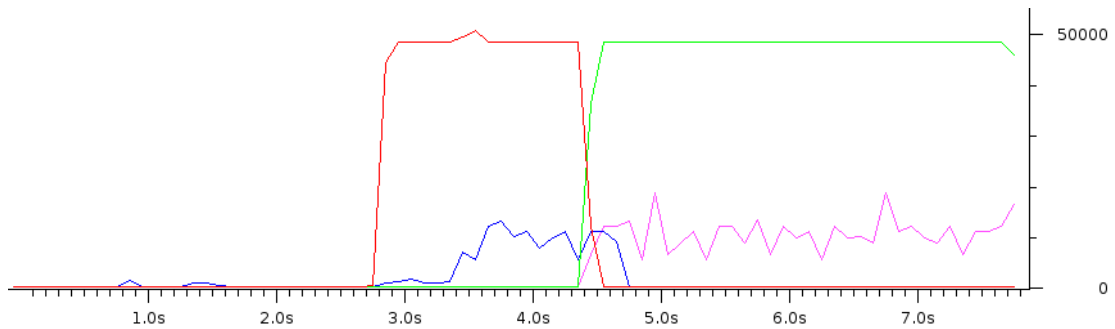
**Figuur 5.7:** Doorstroom van een UDP-teststroom bij het RoF-model. De horizontale as is de tijdsas, verticaal staat de doorstroom in *Bytes per tick*. Eén tick is 0,1 s waardoor de waarde 50.000 op de grafiek overeenkomt met 500.000 Bps of 4 Mbps.

overgeschakeld, daar waar de groene lijn (het verkeer naar RAU1) de tijdsas verlaat. Er is duidelijk te zien dat het verkeer, zowel van de server naar de trein als omgekeerd, niet onderbroken wordt.

Figuur 5.9 lijkt grilliger, wat komt door de sterke uitvergroting van de tijdschaal. Elke lage piek komt dan overeen met een binnenkomend of vertrekkend pakket, UDP-pakketten worden verzonden met een datainhoud van 1000 bytes. Hier is de omschakeling van de centrale op RAU1 beter te zien: de rode en groene lijn zijn steiler en het gebied waar beiden van de tijdsas afwijken, is in verhouding tot de overlapzone (waar de blauwe en paarse lijn van de tijdsas afwijken) veel kleiner dan op figuur 5.8, wat duidelijk aangeeft dat de breedte van de groen-rode overlap een zuiver grafische oorzaak heeft.

### 5.3.2 2 treinen, 3 remote antenna units

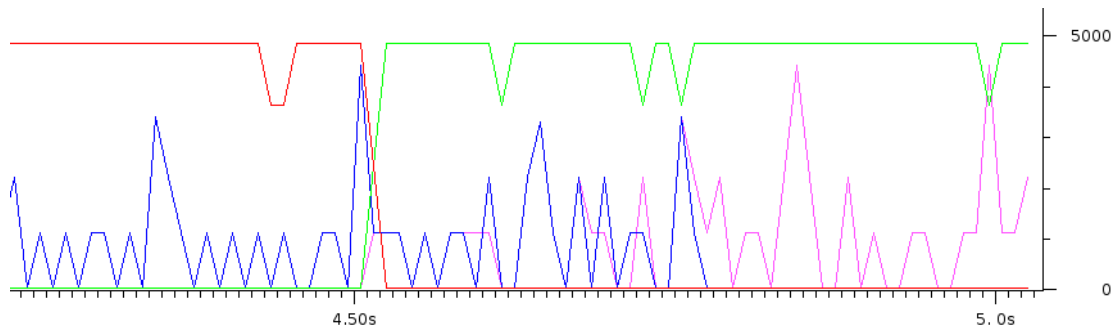
Zoals reeds meermaals in dit werk vermeld, is ons model bijzonder flexibel. Er kunnen meerdere kruisende treinen zijn, of treinen die behoefte hebben aan een hogere bandbreedte, kunnen een bijkomend kanaal toegewezen krijgen. Uiteraard is het aangewezen ook dit even experimenteel na te gaan. Hiervoor creëren we een opstelling zoals weergegeven in figuur 5.10. Wanneer we naar beide treinen afzonderlijk maar simultaan een teststroom willen sturen van 3,2 Mbps, komen we al snel een beperkende factor tegen. Na +/- 7 seconden, als de eerste trein zich in het overlapgebied van RAU0 en RAU1 bevindt en de tweede in dat van RAU1 en RAU2, loopt de eerste wachtlijn in een RAU reeds over, namelijk



**Figuur 5.8:** Pakketstromen tussen de centrale en de RAU's in het RoF-model. De horizontale as is de tijdsas, verticaal staat de doorstroom in *Bytes per tick*. Eén tick is 0,1 s waardoor de waarde 50.000 op de grafiek overeenkomt met 500.000 Bps of 4 Mbps. De rode en groene lijnen geven het verkeer van de centrale naar RAU0, respectievelijk RAU1, weer, terwijl de blauwe en paarse lijnen de stroom van RAU0 respectievelijk RAU1 naar de centrale voorstellen.

deze aan de vaste netwerkverbinding van RAU1 die 10 pakketten kan bevatten. Dit is vrij logisch: de centrale en de RAU's bevinden zich in onze opstelling in één collision domain, en als de treinen zich in het overlapgebied bevinden, worden hun signalen telkens door twee RAU's opgevangen en naar de centrale gestuurd. Bij een reële uitvoering zou dit niet mogen voorvallen. Daar heeft elke RAU immers een directe link naar de centrale die niet met andere RAU's gedeeld wordt en bovendien up- en downlinkverkeer gescheiden houdt.

Het is geweten dat het met ns redelijk moeilijk is om throughputmetingen te doen. Echter, het belangrijkste doel in dit werk is het demonstreren van het concept *moving cell* en de mogelijkheden ermee. Wanneer we de bitrate van de UDP-teststromen verminderen, lopen de wachlijnen dan ook niet meer over. Als we nu bij elke trein individueel gaan kijken naar de doorstroom, dan levert dat gelijkaardige figuren op als in de vorige sectie met één trein, alleen ligt de bitrate lager. Een interessantere meting om het concept te illustreren, bestaat er weer in om de pakketten op de link tussen de centrale en de RAU's te meten. Een dergelijke meting is grafisch weergegeven in figuur 5.11 alwaar de rode lijn de hoeveelheid verkeer weergeeft naar RAU0, terwijl de groene en blauwe dit doen voor RAU1 respectievelijk RAU2. Bijkomend is ook een paarse lijn afgebeeld die een indicatie geeft van het verkeer afkomstig van RAU1, waar beide treinen zich op een bepaald moment samen bij bevinden. De grootte van de totale stroom naar de centrale wordt weergegeven door de zwarte lijn. De rode en blauwe lijn overlappen elkaar grotendeels aangezien de treinen een identieke maar tegengestelde beweging maken. Ze komen beiden terzelfdertijd

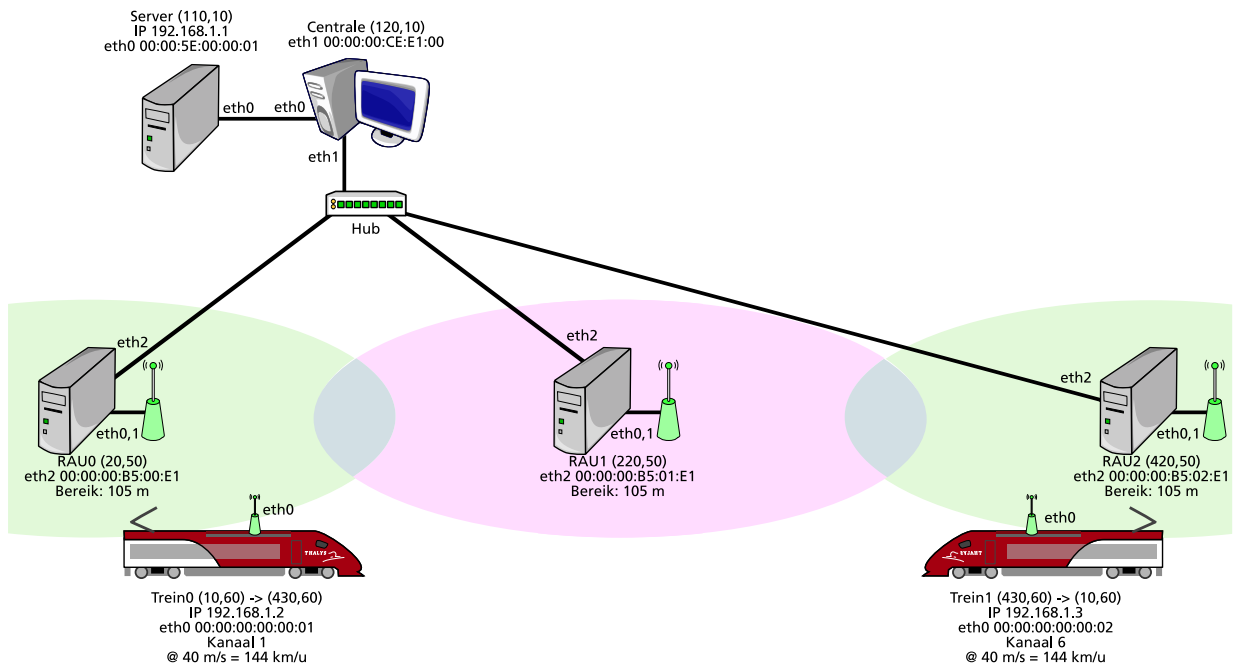


**Figuur 5.9:** Detailopname van de pakketstromen tussen de centrale en de RAU's in het RoF-model. De horizontale as is de tijdsas, verticaal staat de doorstroom in *Bytes per tick*. Eén tick is 0,01 s waardoor de waarde 5.000 op de grafiek overeenkomt met 500.000 Bps of 4 Mbps. De rode en groene lijnen geven het verkeer van de centrale naar RAU0, respectievelijk RAU1, weer, terwijl de blauwe en paarse lijnen de stroom van RAU0 respectievelijk RAU1 naar de centrale voorstellen.

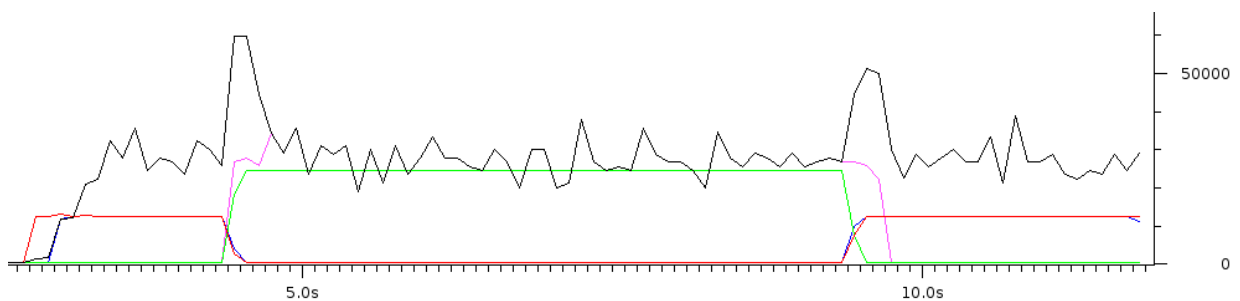
binnen bereik van een nieuw RAU en verlaten het ook samen. Alleen wordt de stroom naar de tweede trein iets later gestart dan de eerste trein, wat het verschil tussen de blauwe en rode lijn helemaal links verklaart. De groene lijn ligt tussen de 4<sup>e</sup> en 9<sup>e</sup> seconde ongeveer dubbel zo hoog als de rode en blauwe lijn daarbuiten, aangezien beide treinen zich nu in het gebied van dezelfde RAU bevinden en de traffic in hun richtingen dus opgeteld wordt. Wanneer beide treinen zich in een overlapgebied bevinden, worden hun signalen door twee RAU's opgepikt, wat resulteert in een traffic die het viervoudige bedraagt van de traffic afkomstig van één trein, wat resulteert in de twee pieken van de zwarte lijn. Tussen deze twee pieken vallen de paarse en zwarte lijn samen, het verkeer naar de centrale is dan enkel en alleen afkomstig van RAU1, waar beide treinen zich tegelijk bevinden.

## 5.4 Evaluatie

Met deze resultaten hebben we aangetoond dat het moving cell concept een werkbare benadering kan zijn. Het grote voordeel ten opzichte van de klassieke manier is dat de connectie nergens verloren gaat, de datastroom wordt nergens onderbroken. Aan een dergelijke bewegingssnelheid is een connectie met traditionele WiFi-oplossingen helemaal onmogelijk, maar dit kan via het hier voorgestelde model verholpen worden.



**Figuur 5.10:** Testopstelling met 2 treinen en 3 RAU's.



**Figuur 5.11:** Pakketstromen tussen de centrale en de RAU's in het RoF-model. De horizontale as is de tijdsas, verticaal staat de doorstroom in *Bytes per tick*. Eén tick is 0,1 s waardoor de waarde 50.000 op de grafiek overeenkomt met 500.000 Bps of 4 Mbps. De rode lijn geeft de hoeveelheid verkeer weer naar RAU0, terwijl de groene en blauwe dit doen voor RAU1 respectievelijk RAU2. De paarse lijn geeft een indicatie van het verkeer afkomstig van RAU1, waar beide treinen zich op een bepaald moment samen bij bevinden. De grootte van de totale stroom naar de centrale wordt weergegeven door de zwarte lijn.

## Hoofdstuk 6

# Besluit en toekomstperspectieven

In dit werk werd een model vooropgesteld dat, met behulp van radio-over-fiber en het moving cell concept, een belangrijk probleem van celgebaseerde oplossingen, namelijk het handoverprobleem, oplost. Door dit probleem van de baan te helpen, is een belangrijke hinderpaal voor het verwezenlijken van een breedbandtoegangsnetwerk voor snel bewegende gebruikers weggenomen. De conceptuele werking werd aangetoond met behulp van simulaties.

Dit werk is echter slechts het topje van de ijsberg, en van een ijsberg is geweten dat de grootste hoeveelheid zich onder water bevindt. Het werk is dan ook nog lang niet af. Om een beter zicht te hebben op het geheel, zijn betere metingen nodig. In eerste instantie denken we aan emulaties, met echte draadloze kaarten, waar de effecten van de verschillende signalen op elkaar en hun omgeving beter in kaart gebracht kan worden. Een mogelijkheid hiervoor is de FAMOUS-opstelling, echter deze houdt nog steeds met verschillende parameters geen rekening, zoals bijvoorbeeld het doppler-effect. Ook de aanwezigheid van de hoogspanningslijnen bij treinen kan een rol spelen, om nog maar te zwijgen van het effect in tunnels. Een emulatie met een echte beweging is dus zeker gewenst. Verder dienen ook nog een heleboel optimalisaties te gebeuren, die ingegeven kunnen worden door dergelijke emulaties. Een voorbeeld is het moment dat de centrale wisselt van RAU. Bij een simulatie bekomen we steeds dezelfde waarden voor de signaalsterkte, en zal het signaal van zodra we het bereik van een RAU binnentreden, alleen maar verbeteren. In realiteit is de signaalsterkte geen stabiel gegeven. Een eenmalige opstoot van een signaal kan de centrale doen overschakelen terwijl het signaal op die plaats helemaal nog niet sterk genoeg is data op een goede manier te transporteren. Een mogelijkheid kan hier zijn om bij het oppikken van het eerste signaal nog een bepaalde tijdspanne te wachten alvorens over te schakelen. Ook

andere gegevens kunnen geoptimaliseerd worden, zoals de keuze van de antennes. In onze simulaties werd gebruik gemaakt van een omnidirectionele antenne die in alle richtingen een even sterk signaal uitzendt. Met richtantennes kan dat signaal sterk verbeterd worden, wat een impact heeft op het bereik ervan.

En zelfs als dergelijke experimenten uitgevoerd zijn en betere parameterwaarden opgeleverd hebben, is de kous nog lang niet af. Het capteren van een volledige frequentieband met radio-over-fiber en het nadien reconstrueren van de individuele kanalen is immers geen evidente zaak. Ook hier dient nog heel wat onderzoekswerk verricht te worden vooraleer remote antenna units op een efficiënte manier gebouwd kunnen worden.

De hamvraag is echter: is het sop de kool wel waard? Conceptueel en technisch kunnen grenzen doorbroken worden, maar wat wil de consument hiervoor betalen? De fabricage van speciaal voor deze doeleinden ontworpen RAU's, de installatie van vele RAU's met stroomvoorziening en dergelijke, de glasvezelkabel, de centrales,... Zaken waaraan zeker geen laag kostenplaatje hangt, hoe eenvoudig de RAU's ook mogen ontworpen zijn. Is de consument wel bereid hiervoor genoeg te betalen om de investering rendabel te maken? Ook dit dient onderzocht te worden.

We kunnen dan ook besluiten dat er technische grenzen doorbroken zijn of kunnen worden, het is conceptueel haalbaar, maar er dient nog veel onderzoek verricht te worden in dit onderwerp. Onderzoek dat niet alleen voer is voor ingenieurs, maar ook voor economen. De toekomst ziet er echter bijzonder rooskleurig uit.

## Bijlage A

### Gedetailleerde resultaten

#### A.1 Pakketstroom tijdens gesimuleerde handover met conventionele AP's

Zoals vermeld in 5.2.2, laten de simulaties in ns-click toe om pakketten te sturen van en naar een AP zonder dat het station met dit AP geassocieerd is. In tabel A.1 is een opsomming te vinden van de pakketten die gecapteerd werden tijdens een dergelijke gesimuleerde handover. De adressen stemmen overeen met deze in 5.2.2.

Tabel A.1: Pakketten tijdens een handover.

No.	Time (s)	Src	Dst	Protocol	Info
1808	03.995077	192.168.1.1	192.168.1.2	UDP	Destination address: 00:00:00:00:00:01 BSS Id: 00:00:00:a0:00:e0 Source address: 00:00:5e:00:00:01
1809	03.995556	192.168.1.2	192.168.1.1	UDP	BSS Id: 00:00:00:a0:00:e0 Source address: 00:00:00:00:00:01 Destination address: 00:00:5e:00:00:01
1810	03.996980	192.168.1.1	192.168.1.2	UDP	Destination address: 00:00:00:00:00:01 BSS Id: 00:00:00:a0:00:e0 Source address: 00:00:5e:00:00:01

Tabel A.1: Pakketten tijdens een handover.

No.	Time (s)	Src	Dst	Protocol	Info
1811	03.997778	192.168.1.2	192.168.1.1	UDP	
		BSS Id: 00:00:00:a0:00:e0			
		Source address: 00:00:00:00:00:01			
		Destination address: 00:00:5e:00:00:01			
1812	03.999622	192.168.1.1	192.168.1.2	UDP	
		Destination address: 00:00:00:00:00:01			
		BSS Id: 00:00:00:a0:00:e0			
		Source address: 00:00:5e:00:00:01			
1813	03.1000000	192.168.1.2	192.168.1.1	UDP	
		BSS Id: 00:00:00:a0:00:e0			
		Source address: 00:00:00:00:00:01			
		Destination address: 00:00:5e:00:00:01			
1814	04.003347	00:00:00:a1:00:e0	Broadcast	IEEE 802.11	Beacon frame
		Destination address: Broadcast (ff:ff:ff:ff:ff:ff)			
		Source address: 00:00:00:a1:00:e0			
		BSS Id: 00:00:00:a1:00:e0			
1815	04.093686	00:00:00:00:00:01	00:00:00:a1:00:e0	IEEE 802.11	Authentication
		Destination address: 00:00:00:a1:00:e0			
		Source address: 00:00:00:00:00:01			
		BSS Id: 00:00:00:a1:00:e0			
1816	04.095153	192.168.1.2	192.168.1.1	UDP	
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:00:00:00:01			
		Destination address: 00:00:5e:00:00:01			
1817	04.096973	192.168.1.2	192.168.1.1	UDP	
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:00:00:00:01			
		Destination address: 00:00:5e:00:00:01			
1818	04.098376	00:00:00:a1:00:e0	00:00:00:00:00:01	IEEE 802.11	Authentication
		Destination address: 00:00:00:00:00:01			
		Source address: 00:00:00:a1:00:e0			
		BSS Id: 00:00:00:a1:00:e0			



Tabel A.1: Pakketten tijdens een handover.

No.	Time (s)	Src	Dst	Protocol	Info
1819	04.098773	192.168.1.2	192.168.1.1	UDP	
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:00:00:00:01			
		Destination address: 00:00:5e:00:00:01			
1820	04.100732	192.168.1.2	192.168.1.1	UDP	
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:00:00:00:01			
		Destination address: 00:00:5e:00:00:01			
1821	04.102847	00:00:00:a1:00:e0	Broadcast	IEEE 802.11	Beacon frame
		Destination address: Broadcast (ff:ff:ff:ff:ff:ff)			
		Source address: 00:00:00:a1:00:e0			
		BSS Id: 00:00:00:a1:00:e0			
1822	04.103047	00:00:00:00:00:01	00:00:00:a1:00:e0	IEEE 802.11	Association Request
		Destination address: 00:00:00:a1:00:e0			
		Source address: 00:00:00:00:00:01			
		BSS Id: 00:00:00:a1:00:e0			
1823	04.104976	192.168.1.2	192.168.1.1	UDP	
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:00:00:00:01			
		Destination address: 00:00:5e:00:00:01			
1824	04.107136	192.168.1.2	192.168.1.1	UDP	
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:00:00:00:01			
		Destination address: 00:00:5e:00:00:01			
1825	04.107332	192.168.1.1	192.168.1.2	UDP	
		Destination address: 00:00:00:00:00:01			
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:5e:00:00:01			
1826	04.109051	192.168.1.1	192.168.1.2	UDP	
		Destination address: 00:00:00:00:00:01			
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:5e:00:00:01			

Tabel A.1: Pakketten tijdens een handover.

No.	Time (s)	Src	Dst	Protocol	Info
1827	04.109255	192.168.1.2	192.168.1.1	UDP	
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:00:00:00:01			
		Destination address: 00:00:5e:00:00:01			
1828	04.110539	00:00:00:a1:00:e0	00:00:00:00:00:01	IEEE 802.11	Association Response
		Destination address: 00:00:00:00:00:01			
		Source address: 00:00:00:a1:00:e0			
		BSS Id: 00:00:00:a1:00:e0			
1829	04.111015	192.168.1.2	192.168.1.1	UDP	
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:00:00:00:01			
		Destination address: 00:00:5e:00:00:01			
1830	04.112590	192.168.1.1	192.168.1.2	UDP	
		Destination address: 00:00:00:00:00:01			
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:5e:00:00:01			
1831	04.112654	192.168.1.2	192.168.1.1	UDP	
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:00:00:00:01			
		Destination address: 00:00:5e:00:00:01			
1832	04.114570	192.168.1.1	192.168.1.2	UDP	
		Destination address: 00:00:00:00:00:01			
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:5e:00:00:01			
1833	04.114834	192.168.1.2	192.168.1.1	UDP	
		BSS Id: 00:00:00:a1:00:e0			
		Source address: 00:00:00:00:00:01			
		Destination address: 00:00:5e:00:00:01			

## Bibliografie

- [1] Jochen Schiller. *Mobiele Communicatie*. Pearson Education, 2005.
- [2] Matthew Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly, april 2002.
- [3] Peter Dedecker. Stageverslag Siemens NV. 2005.
- [4] Wikipedia, the free encyclopedia. <http://en.wikipedia.org/>.
- [5] Bart Lannoo, Didier Colle, Mario Pickavet, Piet Demeester, *Optical Switching Architecture to Implement Moveable Cells in a Multimedia Train Environment*, proc. of ECOC 2004, 30th European Conf. on Optical Communication, vol. 3, pp. 344-345, Stockholm, Sweden, 5-9 Sep. 2004.
- [6] The Click Modular Router Project. <http://www.read.cs.ucla.edu/click/>.
- [7] Steven McCanne en Sally Floyd. ns—Network Simulator. <http://nsnam.isi.edu/nsnam/>.
- [8] Steven McCanne en Van Jacobson. nam—Network AniMator. <http://nsnam.isi.edu/nsnam/>.
- [9] Massachusetts Institute of Technology. <http://www.mit.edu/>.
- [10] Parallel and Distributed Operating Systems group. <http://pdos.csail.mit.edu/>.
- [11] Computer Science and Artificial Intelligence Laboratory. <http://www.csail.mit.edu/>.
- [12] Mazu Networks. <http://www.mazunetworks.com/>.
- [13] International Computer Science Institute. <http://www.icsi.berkeley.edu/>.
- [14] Berkeley University of California. <http://www.berkeley.edu/>.

- 
- [15] Computer Science Department. <http://www.cs.ucla.edu/>.
- [16] University of California, Los Angeles. <http://www.ucla.edu/>.
- [17] The REAL network simulator. <http://www.cs.cornell.edu/skeshav/real/>.
- [18] Information Sciences Institute. <http://www.isi.edu/>.
- [19] University of Southern California. <http://www.usc.edu/>.
- [20] Collaborative Simulation for Education and Research. <http://www.isi.edu/conser/>.
- [21] Michael Neufeld, Ashish Jain, and Dirk Grunwald. Nsclick:: bridging network simulation and deployment. In *Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*, pages 74–81. ACM Press, 2002.
- [22] The *nsclick* simulation environment. <http://systems.cs.colorado.edu/Networking/nsclick/>.

## Lijst van figuren

1.1	Het doel van dit onderzoek is het verleggen van de grens van de combinatie van bandbreedte en bewegingssnelheid. . . . .	2
1.2	Trein en grondstation zijn door middel van schotelantennes en een satellietverbinding met elkaar verbonden. . . . .	3
1.3	Dekking van een heel gebied door middel van aangrenzende en overlappende cellen waarin verschillende frequenties gebruikt worden. . . . .	3
1.4	Dekking van een traject waarlangs een gebruiker zich beweegt door middel van aangrenzende en overlappende cellen waarin verschillende frequenties gebruikt worden. . . . .	4
2.1	Optische ring: elk basisstation heeft een eigen toegewezen golflengte en staat rechtstreeks in contact met de centrale. . . . .	10
2.2	De trein bevindt zich in het gebied van RAU1, waarlangs alle communicatie met de centrale verloopt. . . . .	13
2.3	De trein rijdt verder en komt terecht in de overlapzone van RAU1 en RAU2. Beide RAU's ontvangen de uplinkpakketten afkomstig van de trein en zenden deze over de optische ring naar de centrale. . . . .	14
2.4	De centrale heeft een uplinkpakket afkomstig van de trein ontvangen via RAU2. Op dit moment schakelt de centrale over en wordt downlinkverkeer naar RAU2 gestuurd. RAU1 blijft de signalen van de trein ontvangen en geeft deze door aan de centrale, maar de centrale keert niet terug naar de <i>vorige RAU</i> . . . . .	15
2.5	De trein rijdt verder en verlaat de overlapzone. De uplinkpakketten worden enkel nog door RAU2 ontvangen en door deze laatste aan de centrale bezorgd. Alle verkeer verloopt via RAU2. . . . .	16

---

3.1	Mapping van de simulatie op het fysieke model: de RAU's die normaal de gehele band capteren en op de optische ring plaatsen, worden vervangen door een aantal draadloze netwerkkaarten aangesloten op een pc, die via een conventioneel LAN met de centrale in verbinding staat. . . . .	19
3.2	Een voorbeeld van de headers van een UDP-pakket naar de client, gecaptureerd tussen de centrale en de RAU. De lengte van de headers is vermeld in bytes. Header $K$ staat voor het kanaal waarop het betreffende pakket dient uitgezonden te worden. . . . .	20
4.1	Click-schema van de server. Er wordt een instantie van de klasse <code>dumbserver</code> gecreëerd met <code>naam :: dumbserver(IP-adres, MAC-adres);</code> . . . . .	29
4.2	Click-schema van de trein. . . . .	31
4.3	Click-schema van een RAU. De constructies voor het bekomen van leesbare dumpfiles werden weggelaten. . . . .	32
4.4	Click-schema van de centrale. . . . .	33
4.5	Voorbeeld van een pakket verzonden door een RAU naar de centrale, met details van de WiFi-header. De lengte van de headers is vermeld in bytes. . . . .	34
4.6	Schematisch overzicht van de bewerkingen door de click-elementen op de headers van de pakketten bestemd voor de trein. De lengte van de headers is vermeld in bytes. . . . .	36
4.7	Schematisch overzicht van de bewerkingen door de click-elementen op de headers van de pakketten afkomstig van de trein. De lengte van de headers is vermeld in bytes. . . . .	37
5.1	Analyse van een UDP-teststroom aan +/- 1 Mbps met na de 26 <sup>e</sup> seconde (5 s na het begin van de datastroom) een handover. De horizontale as is de tijdsas, verticaal staat de doorstroom in <i>Bytes per tick</i> . Eén tick is 0,1 s waardoor de waarde 10.000 op de grafiek overeenkomt met 100.000 Bps of 800 Kbps. . . . .	40
5.2	Detail van de handover van figuur 5.1: tijdens een UDP-teststroom aan +/- 1 Mbps werd na de 26 <sup>e</sup> seconde (5 s na het begin van de stroom) tot een handover overgegaan. De horizontale as is de tijdsas, verticaal staat de doorstroom in <i>Bytes per tick</i> . Eén tick is 0,01 s waardoor de waarde 1.000 op de grafiek overeenkomt met 100.000 Bps of 800 Kbps. . . . .	41
5.3	Testopstelling voor het bepalen van een vergelijkingsbasis. . . . .	42

---

5.4	Doorstroom van een UDP-teststroom bij 2 gewone AP's. De horizontale as is de tijdsas, verticaal staat de doorstroom in <i>Bytes per tick</i> . Eén tick is 0,1 s waardoor de waarde 100.000 op de grafiek overeenkomt met 1.000.000 Bps of 8 Mbps. . . . .	43
5.5	Detailopname van de handover tijdens de doorstroom van een UDP-teststroom bij 2 gewone AP's, zoals weergegeven in figuur 5.4. De horizontale as is de tijdsas, verticaal staat de doorstroom in <i>Bytes per tick</i> . Eén tick (hier één klein streepje op de horizontale as) is 0,01 s waardoor de waarde 10.000 op de grafiek overeenkomt met 1.000.000 Bps of 8 Mbps. . . . .	44
5.6	Testopstelling met 1 trein en 2 RAU's. . . . .	45
5.7	Doorstroom van een UDP-teststroom bij het RoF-model. De horizontale as is de tijdsas, verticaal staat de doorstroom in <i>Bytes per tick</i> . Eén tick is 0,1 s waardoor de waarde 50.000 op de grafiek overeenkomt met 500.000 Bps of 4 Mbps. . . . .	46
5.8	Pakketstromen tussen de centrale en de RAU's in het RoF-model. De horizontale as is de tijdsas, verticaal staat de doorstroom in <i>Bytes per tick</i> . Eén tick is 0,1 s waardoor de waarde 50.000 op de grafiek overeenkomt met 500.000 Bps of 4 Mbps. De rode en groene lijnen geven het verkeer van de centrale naar RAU0, respectievelijk RAU1, weer, terwijl de blauwe en paarse lijnen de stroom van RAU0 respectievelijk RAU1 naar de centrale voorstellen. . . . .	47
5.9	Detailopname van de pakketstromen tussen de centrale en de RAU's in het RoF-model. De horizontale as is de tijdsas, verticaal staat de doorstroom in <i>Bytes per tick</i> . Eén tick is 0,01 s waardoor de waarde 5.000 op de grafiek overeenkomt met 500.000 Bps of 4 Mbps. De rode en groene lijnen geven het verkeer van de centrale naar RAU0, respectievelijk RAU1, weer, terwijl de blauwe en paarse lijnen de stroom van RAU0 respectievelijk RAU1 naar de centrale voorstellen. . . . .	48
5.10	Testopstelling met 2 treinen en 3 RAU's. . . . .	49

- 5.11 Pakketstromen tussen de centrale en de RAU's in het RoF-model. De horizontale as is de tijdsas, verticaal staat de doorstroom in *Bytes per tick*. Eén tick is 0,1 s waardoor de waarde 50.000 op de grafiek overeenkomt met 500.000 Bps of 4 Mbps. De rode lijn geeft de hoeveelheid verkeer weer naar RAU0, terwijl de groene en blauwe dit doen voor RAU1 respectievelijk RAU2. De paarse lijn geeft een indicatie van het verkeer afkomstig van RAU1, waar beide treinen zich op een bepaald moment samen bij bevinden. De grootte van de totale stroom naar de centrale wordt weergegeven door de zwarte lijn. . . . . 49



## Lijst van tabellen

A.1 Pakketten tijdens een handover. . . . .	52
A.1 Pakketten tijdens een handover. . . . .	53
A.1 Pakketten tijdens een handover. . . . .	54
A.1 Pakketten tijdens een handover. . . . .	55