



**HWI HAMBURG**  
HOCHSCHULÜBERGREIFENDER STUDIENGANG  
WIRTSCHAFTSINGENIEURWESEN HAMBURG

# BACHELORARBEIT

## Implementierung von Relaisstationen in IMTAPhy

**Bereich:** Wirtschaftsinformatik

**Verfasser:** Tillmann Radmer

**1. Gutachter:** Prof. Dr. Stefan Voß

**2. Gutachter:** Dr. Silvia Schwarze

**vorgelegt am:** 3.2.2013

AM LEHRANGEBOT BETEILIGTE HOCHSCHULEN:

**Universität Hamburg**

**Hochschule für Angewandte Wissenschaften Hamburg**

**Technische Universität Hamburg-Harburg**

Hiermit erkläre ich, dass ich die vorliegende Bachelorarbeit ohne fremde Hilfe selbständig verfasst habe. Ich habe keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellverzeichnis nicht benannten Internet-Quellen – benutzt. Ich habe die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht. Die schriftliche Fassung entspricht der auf dem elektronischen Speichermedium.

Hamburg, den

Unterschrift

# Inhaltsverzeichnis

Abbildungsverzeichnis	V
Tabellenverzeichnis	V
Abkürzungsverzeichnis	VI
1 Einleitung	1
1.1 Thematische Einführung . . . . .	1
1.2 Zielsetzung und Vorgehensweise . . . . .	2
2 Kurzer Überblick über die Entwicklung der relevanten Standards in Funknetzwerken	4
2.1 GSM, GPRS und EDGE . . . . .	4
2.2 UMTS und HSPA . . . . .	5
2.3 LTE . . . . .	7
2.4 LTE-A . . . . .	8
3 Relaisstationen	9
3.1 Arten von Relaisstationen in Mobilfunknetzen . . . . .	9
3.2 Relais im LTE Standard . . . . .	11
4 openWNS	13
4.1 Aufbau von OpenWNS . . . . .	13
4.2 Konfiguration von Szenarien . . . . .	13
4.3 Node-Component Modell . . . . .	14
4.4 Functional Units . . . . .	16
4.5 Multiplexing . . . . .	18

## *Inhaltsverzeichnis*

4.6	Compounds . . . . .	20
4.6.1	Commands . . . . .	21
4.6.2	Command Pool . . . . .	21
5	Implementierung . . . . .	23
5.1	StationPhy . . . . .	24
5.2	Layer2 . . . . .	26
5.2.1	Das Functional Unit Network . . . . .	27
5.2.2	CompoundRelaySwitch . . . . .	28
5.3	LinkManager . . . . .	30
5.3.1	Links innerhalb von Relaisstationen . . . . .	31
5.3.2	Links zwischen Relaisstationen . . . . .	31
5.3.3	Indirekt verbundene Mobilstationen . . . . .	32
6	Evaluation . . . . .	34
6.1	Simulationsparameter . . . . .	36
6.2	SINR . . . . .	38
6.3	Durchsatz . . . . .	40
7	Fazit . . . . .	44
A	Die determineIndirectlyServedLinks Funktion . . . . .	46
B	Angepasstes Kanal-Modell für das Test-Szenario . . . . .	48
	Literaturverzeichnis . . . . .	49

# Abbildungsverzeichnis

4.1	Aufbau von openWNS . . . . .	14
4.2	Screenshot von dem imtaphyViewer. . . . .	15
4.3	Übersicht über das Node-Component-Modell. . . . .	17
4.4	Die Schnittstellen einer FU. . . . .	19
4.5	Multiplexing in einem FUN. . . . .	20
5.1	Die Components in eNB und UE . . . . .	24
5.2	Components und FUs in Relaisstationen . . . . .	25
6.1	Das Urban Macro Szenario. . . . .	35
6.2	SINR Verteilungsfunktionen . . . . .	40
6.3	Durchschnittliches SINR . . . . .	41
6.4	Verteilungsfunktion des aggregierten Durchsatzes pro Zelle. . . . .	41
6.5	Durchsatz pro Mobilgerät . . . . .	42
6.6	Zuordnung der Mobilstationen zu den Basis- und Relaisstationen . . . . .	43

# Tabellenverzeichnis

3.1	Vor- und Nachteile verschiedener Relaisstationen. . . . .	11
6.1	Parameter für das Urban Macro Szenario. . . . .	39
6.2	Durchsatzsteigerung durch Relaisstationen. . . . .	42

# Abkürzungsverzeichnis

## Abkürzungen

Abkürzung	Bedeutung
3GPP	3rd Generation Partnership Project
AF	Amplify-and-Forward
ARQ	Automatic Repeat Request
CDMA	Code Division Multiple Access
CoMP	Coordinated Multi-Point
CSE	Cell Spectral Efficiency
DCI	Downlink Control Information
DF	Decode-and-Forward
EDGE	Enhanced Data rates for Global Evolution
eNB	evolved Node B
ETSI	European Telecommunications Standards Institute
FDD	Frequency-Division Duplex
FDMA	Frequency-Division Multiple Access
FU	Functional Unit
FUN	Functional Unit Network
GPRS	General Packet Radio Service
HARQ	Hybrid Automatic Repeat Request
HSDPA	High Speed Downlink Packet Access
HSPA	High Speed Packet Access
HSUPA	High Speed Uplink Packet Access
IP	Internet Protocol
ISD	Inter Site Distance
ISI	Inter-Symbol Interference

## *Abkürzungsverzeichnis*

Abkürzung	Bedeutung
ITA-2000	Information Technology Act 2000
ITU	International Telecommunication Union
LGPL	Lesser General Public License
LOS	line-of-sight
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
MAC-Adresse	Media-Access-Control-Adresse
MIMO	Multiple Input, Multiple Output
NLOS	non-line-of-sight
OFDMA	Orthogonal Frequency-Division Multiple Access
OSI	Open Systems Interconnection
PCI	Physical Cell ID
QAM	Quadrature Amplitude Modulation
RN	Relay Node
RTT	round-trip time
SINR	Signal to Interference plus Noise Ratio
TDD	Time-Division Duplex
TDMA	Time-Division Multiple Access
UE	User Equipment
UMTS	Universal Mobile Telecommunication System
W-CDMA	Wideband Code Division Multiple Access

# 1 Einleitung

## 1.1 Thematische Einführung

Das Datenaufkommen in den mobilen Netzen steigt stetig, global allein 2012 um 70% gegenüber dem Vorjahr. Die treibende Kraft ist dabei nicht die steigende Anzahl an Mobiltelefonen, sondern die Art und Weise wie diese verwendet werden. So können Smartphones Bilder anzeigen, Voice-over-IP Telefonate mit Video führen und Videos in hoher Qualität aus dem Internet abspielen. Diese Dienste verschlingen ungleich mehr Bandbreite im Vergleich zu einfachen Sprach-, SMS- oder MMS-Nachrichten. Zusätzlich zu der steigenden Verbreitung von Smartphones, ist in den letzten Jahren eine neue Klasse von Geräten, die Tablets, entstanden. Tablets werden, im Gegensatz zu Notebooks, vermehrt mit mobilem Internetzugang verkauft und das Nutzerverhalten mit diesen Geräten ist noch datenintensiver, als an einem Smartphone. Insgesamt soll sich das Datenaufkommen bis 2017 so verdreizehnfachen. [Cisco, 2013]

Das alles ist an sich kein Problem, so lange die Netzbetreiber zusätzliche Einnahmen durch die Datendienste erwirtschaften. Das ist jedoch nicht der Fall. Die Netzbetreiber verkaufen zu einem Großteil Flatrate-Tarife, bei denen die Benutzer unbegrenzt, zu einem festen monatlichen Preis, im Internet surfen können. Dadurch sind die Einnahmen nur schwach an das Datenaufkommen gebunden. Die Kosten für den Betrieb und den Ausbau der Netze explodieren, während der Umsatz stagniert. [UMTS Forum, 2009]

Abhilfe können neue Technologien in Long Term Evolution (LTE) und LTE-Advanced (LTE-A) schaffen, welche die Kosten pro übertragenem Bit senken (Kapitel 2). Eine davon sind Basisstationen, welche nicht über eine eigene Kabelverbindung mit dem Kernnetz verbunden sind, sondern über die Luftschnittstelle einer anderen Basisstation. Dadurch verspricht die Technik eine kostengünstige und flexible Alternative zu herkömmlichen



## 1 Einleitung

Sendestationen zu werden (Kapitel 3). Eine solche Basisstation heißt Relaisstation oder Relay Node (RN). Um die Leistungsfähigkeit von Netzwerken mit RNs zu bewerten und den Einfluss alternativer Protokolle oder Scheduler zu testen, wird eine offene Plattform benötigt. Diese muss die physikalische Schicht, als auch die darüber liegenden Protokollschichten simulieren können. Im folgenden Abschnitt wird der quelloffene Simulator openWNS und alternative Simulationsplattformen vorgestellt.

### 1.2 Zielsetzung und Vorgehensweise

Eine Implementierung von Relaisstationen ermöglicht es Strategien für eine optimale Verteilung von RNs zu finden, andere Einstellungen (z.B. für die Sendestärke) zu testen oder die Anzahl an Platzierten RNs zu variieren. Der Simulator openWNS und die Erweiterung IMTAphy sind quelloffen, wodurch die Implementierung erweitert werden kann. So können neue Strategien für die automatische Konfiguration von RNs oder kooperative, intelligente Scheduler getestet werden, welche sich mit benachbarten RNs absprechen. Da kein Open Source Simulator existiert, welcher Relaisstationen unterstützt, soll im Rahmen dieser Arbeit das IMT-A Modul IMTAphy für den Simulator openWNS um diese Funktion erweitert werden und das Ergebnis evaluiert werden.

IMTAphy erweitert openWNS um eine LTE-A Simulation auf der System-Ebene. Das heißt, neben einem Modell der Luftschnittstelle, werden auch die Basisstationen und Endgeräte, sowie der Fluss von Daten-Paketen zwischen diesen simuliert. Wird nur das Kanal-Modell simuliert, spricht man von der Kanal-Ebene. Die Simulationsplattform openWNS wurde von der Communication Networks (ComNets) Research Group der Universität Aachen mit dem Ziel entwickelt, eine offene Plattform für die Forschung zu schaffen. Darum ist der Quellcode unter der GNU LGPL (Lesser General Public License) Version 2.1 veröffentlicht worden. Durch den modularen Aufbau der Plattform, ist es relativ einfach, Erweiterungen zu implementieren und dabei vorhandene Module wieder zu verwenden (Kapitel 4).

Eine Alternative zu openWNS ist der etablierte Open Source Simulator ns-3. Wie openWNS ist auch ns-3 in C++ programmiert und Szenarien werden in der Programmiersprache Python beschrieben. Die Weiterentwicklung von ns-3 wird von einem Konsortium geleitet, welches vom *Institut national de recherche en informatique et en automatique* (INRIA) und der University of Washington gegründet wurde. Weniger bekannt und relativ neu ist LTE-Sim.

## *1 Einleitung*

Dieser Simulator bietet ausschließlich LTE Simulationen und wird von der University of Bari entwickelt. Schließlich gibt es einen kommerziellen Simulator von Nomor Research. Anders als die quelloffenen Alternativen, kann dieser Relaisstationen in einem LTE-A Netzwerk simulieren [Research, 2012]. Es ist jedoch nicht möglich die Software um neue Algorithmen z.B. für den Paket-Scheduler zu erweitern, da der Quellcode nicht öffentlich zugänglich ist.

Um die Technik hinter Relaisstationen einordnen zu können, wird in Kapitel 2 zuerst ein Überblick über die Entwicklung der Mobilfunkstandards bis hin zu LTE-A gegeben. Der Fokus liegt dabei auf den Technologien, welche den Durchsatz oder die Netzabdeckung steigern, da die RNs eben diesem Zweck dienen sollen. Anschließend werden verschiedene Arten von RNs vorgestellt und die spezifizierten RNs aus dem LTE-A Standard nach diesen kategorisiert (Kapitel 3). Kapitel 4 gibt eine Einführung in die Struktur von openWNS und erläutert die Konzepte und Schnittstellen, welche für die Implementierung wichtig sind. Aufbauend auf diesem Wissen, wird in Kapitel 5 ein Konzept für die Implementierung von RNs vorgestellt und die wichtigsten Details der Implementierung werden erläutert. Abschließend wird ein Test-Szenario mit entsprechender Konfiguration beschrieben und die Ergebnisse aus dem Test werden verglichen (Kapitel 6).

## 2 Kurzer Überblick über die Entwicklung der relevanten Standards in Funknetzwerken

In diesem Kapitel wird eine kurze Übersicht über die Entwicklung mobiler Kommunikationssysteme gegeben. Das Hauptaugenmerk liegt dabei auf den von dem *3rd Generation Partnership Project* (3GPP) spezifizierten Standards. Entstanden ist das 3GPP in Folge des Information Technology Act 2000 (ITA-2000) um das 2G Netzwerk *Global System for Mobile Communications* (GSM) zu einem 3G-kompatiblen Standard weiter zu entwickeln. GSM wurde von dem *European Telecommunications Standards Institute* (ETSI) entwickelt. Analog zur 3GPP entstand auch die 3GPP2 welche das auf Code Division Multiple Access (CDMA) basierte 2G Netzwerk fortführte [Ali-Yahiya, 2011]. Letzteres ist vor Allem in den USA weit verbreitet.

### 2.1 GSM, GPRS und EDGE

Gegenüber der ersten, analogen Generation mobiler Funknetzwerke, bietet das GSM eine höhere Kapazität und bessere Sprachqualität durch den Einsatz von digitaler Übertragungstechnik. Zur Ressourcenaufteilung wird *Time-Division Multiple Access* (TDMA) verwendet. Dabei werden den Benutzern einzelne Zeitschlitzte zugeteilt, in denen der gesamte Funkkanal zur Übertragung verwendet werden kann. Der Funkkanal ist 200 kHz breit.

Wie die erste Generation ist auch GSM noch ein leitungsvermittelltes Netzwerk. Wird eine Verbindung zwischen zwei Teilnehmern des Netzwerks hergestellt, werden alle benötigten Ressourcen auf dem Weg reserviert. In GSM heißt das, dass ein Zeitschlitz für diese

## 2 Kurzer Überblick über die Entwicklung der relevanten Standards in Funknetzwerken

Verbindung fest eingenommen wird, auch wenn gerade keine Daten zwischen den beiden Endpunkten anfallen. Damit eignen sich leitungsvermittelte Netzwerke nur begrenzt für die Übertragung von Daten-Paketen. Das gilt insbesondere für das Surfen im Internet. Dabei wird die Bandbreite schlagartig benötigt, wenn z.B. eine Internetseite geladen wird. Danach kann es vorkommen, dass minutenlang gar keine Pakete übermittelt werden. Während dieser Zeit bleibt der reservierte Kanal für andere Benutzer unzugänglich. Die maximale Übertragungsgeschwindigkeit in GSM liegt bei 14,4 kbit/s.

Um die Paketübertragung zu verbessern, wurde im Jahr 2000 die GSM Erweiterung *General Packet Radio Service* (GPRS) eingeführt. Hier wird jedem Nutzer nicht mehr ein fester Zeitschlitz zugeteilt. Vielmehr werden die verfügbaren Ressourcen dynamisch auf die Teilnehmer aufgeteilt. Dadurch wird die maximale Bandbreite pro Nutzer erheblich gesteigert. Theoretisch sind bis zu 170 kbit/s möglich. *Enhanced Data rates for Global Evolution* (EDGE) ist die nächste Erweiterung zu GSM und führt das neue Modulationsverfahren 8-PSK (*Phase shift-keying*) ein. Statt wie bisher 1 Bit pro Übertragungsschritt, können nun bis zu 3 Bits übertragen werden. [Sauter, 2010]

## 2.2 UMTS und HSPA

Das *Universal Mobile Telecommunication System* (UMTS) ist der erste Standard der 3GPP, spezifiziert im Release 99. Um die Ressourcen auf der Luftschnittstelle noch flexibler nutzen zu können, wird ein neues Verfahren zur Ressourcenaufteilung eingeführt. Anstelle von TDMA wird nun *Wideband Code Division Multiple Access* (W-CDMA) verwendet. Anstatt einzelnen Frequenz- oder Zeitabschnitten werden Benutzern Codes zugeteilt, mit welchen ein Bit multipliziert wird. Anhand dieser Codes kann der Empfänger die einzelnen Signal unterscheiden und aus dem gesamten Signal herausfiltern. Da der Code ein Vektor mit bis zu 512 Stellen ist, wird ein zu übertragendes Bit außerdem um den sogenannten Spreizfaktor vergrößert. Die einzelnen Stellen des so generierten Signals heißen Chips. Da die Luftschnittstelle mit einer konstanten Chip-Rate arbeitet, sinkt die effektiv erreichbare Bandbreite bei höheren Spreizfaktoren. Auf der anderen Seite wird das Signal so weniger anfällig für Interferenzen, wodurch Übertragungen auch bei vielen Teilnehmern in einer Zelle möglich sind. Die Bandbreite wird effizient ausgenutzt, indem zu jedem Zeitpunkt möglichst niedrige Spreizfaktoren gewählt werden, sodass ein Signal gerade noch erfolgreich übertragen wird. Zusätzlich zu W-CDMA steigt auch die

## 2 Kurzer Überblick über die Entwicklung der relevanten Standards in Funknetzwerken

Bandbreite des Kanals auf 5 MHz. Insgesamt sind so Übertragungsraten von bis zu 384 kbit/s möglich. [Sauter, 2010]

Wie schon bei GSM mit der Einführung von GPRS, wird auch UMTS weiter flexibilisiert, um die maximal verfügbare Bandbreite pro Teilnehmer weiter zu steigern und die Ressourcen effektiver zu nutzen. So ist UMTS zwar schon recht flexibel, dahingehend, dass bei wenigen Teilnehmern die Bandbreite für einzelne Nutzer steigt. Jedoch können ungenutzte Codes nicht von anderen Benutzern vorübergehend verwendet werden, um die Datenrate zu steigern. Um dies zu ermöglichen, werden mit dem in Release 5 spezifizierten *High Speed Downlink Packet Access* (HSDPA) Kanäle hinzugefügt, welche von allen Teilnehmern geteilt werden. Über einen Kontrollkanal werden die Endgeräte informiert, auf welchem Zeitschlitz Daten für sie auf dem geteilten Kanal übertragen werden. Zusätzlich wird in Release 5 die Modulationsart 16-QAM (*Quadrature Amplitude Modulation*) eingeführt. Statt wie bisher mit *Quadrature Phase Shift Keying* (QPSK) 2 Bits können nun 4 Bits pro Übertragungsschritt gesendet werden. Eine ähnliche Erweiterung namens *High Speed Uplink Packet Access* (HSUPA) wird mit Release 6 für die andere Senderichtung eingeführt. HSDPA und HSUPA werden zusammen auch *High Speed Packet Access* (HSPA) genannt. [Sauter, 2010] In 3GPP Release 7 werden 64-QAM, für 6 Bits pro Übertragungsschritt, und MIMO (Multiple Input, Multiple Output) spezifiziert. Letzteres erlaubt es mehrere räumlich getrennte Datenströme auf derselben Frequenz zu übertragen. Dafür muss sowohl im Sender als auch im Empfänger die gleiche Anzahl an Antennen vorhanden sein. Für 2x2 MIMO also jeweils zwei. 3GPP Release 8 kombiniert schließlich diese beiden Verfahren um die Transferraten weiter zu steigern. [Holma et al., 2007] Auch in den danach folgenden Spezifikationen wird HSPA weiterentwickelt und teilweise mit Funktionen aus LTE ausgestattet. Der Grund dafür ist die weite Verbreitung von UMTS Hardware, sowohl auf der Seite der Netzbetreiber, als auch auf der Seite der Konsumenten. Zu den wichtigsten Neuerungen gehören Dual-Cell HSDPA bzw. HSUPA und Multi-Carrier HSPA. Bei ersterem können Mobilgeräte zu zwei Basisstationen gleichzeitig verbunden sein, um die Bandbreite theoretisch zu verdoppeln. Letzteres verbindet mehrere nicht benachbarte Frequenzbänder zu einem und bildet so einen breiteren Kanal [Ericsson, 2011].

### 2.3 LTE

LTE beschreibt allgemein die vierte Generation mobiler Netzwerke nach der 3GPP Release 8 Spezifikation, wobei erst LTE-A (siehe Abschnitt 2.4) die Spezifikationen nach der *International Telecommunication Union* (ITU) [Astély et al., 2009] erfüllt. Um die Kapazität gegenüber HSPA zu steigern, wird die Kanalbreite weiter erhöht. Sie beträgt nunmehr bis zu 20 Mhz, wodurch der Einsatz von W-CDMA problematisch wird. So würde durch die große Kanalbreite die Chip-Rate steigen und die Zeit für die Übertragung eines Bits immer geringer. Dadurch werden die Signale anfälliger für sogenannte Mehrpfadinterferenzen, d.h. verzögerte Kopien desselben Signals [Sauter, 2010]. Deshalb wird in LTE *Orthogonal Frequency-Division Multiple Access* (OFDMA) verwendet. Dabei wird der Kanal nicht in Codes, sondern in viele kleine Frequenzblöcke unterteilt. Anders als beim klassischen *Frequency-Division Multiple Access* (FDMA) werden orthogonale Frequenzschlitze verwendet, wodurch sich die Frequenzbänder teilweise überlappen können ohne die Übertragung zu stören. Durch die schmalen Träger sinkt die Chip-Rate jedes einzelnen Trägers im Vergleich zu W-CDMA und die Übertragungsdauer eines Bits steigt. [Ali-Yahiya, 2011] Die hohe spektrale Effizienz von W-CDMA wird mit der hohen Störsicherheit von FDMA kombiniert.

Ein weiterer Vorteil von OFDMA ist, dass der Aufwand zum Entzerren des Signals relativ gering ist. Das liegt daran, dass fast keine *Inter-Symbol Interference* (ISI) auftritt. Die Empfängerhardware muss nur zeitlich verzögerte Kopien des selben Signals herausfiltern [Ali-Yahiya, 2011, S. 59f.]. Das senkt den Energieverbrauch und die Komplexität der Hardware und erleichtert somit den Einsatz von MIMO-Techniken, weil dabei mehrere Kanäle parallel (de-)kodiert werden müssen. [Astély et al., 2009] Breite 20 MHz Kanäle in Verbindung mit OFDMA, 64-QAM und 4x4 MIMO ermöglichen Transferraten von bis zu 300 Mbit/s. Eine weitere wichtige Neuerung ist die Umstellung auf ein fast komplett IP (Internet Protocol) basiertes Kernnetz. Dadurch kann ein höherer Grad an Integration zwischen Luftschnittstelle, Radionetzwerk und Kernnetzwerk erreicht werden, wodurch die Kosten für Komponenten sinken. [Sauter, 2010] Außerdem ist die Hardware für IP-Router weiter verbreitet.

## 2.4 LTE-A

Die Weiterentwicklung von LTE, LTE-A, hat das Ziel die 4G Anforderungen der ITU-R zu erfüllen. So werden maximale Übertragungsraten von 1500 Mbit/s in Downlink- und 270 Mbit/s in Uplink-Richtung gefordert. Das soll durch eine hohe spektrale Effizienz von über 15 Bit/s/Hz in Downlink- und 6,75 Bit/s/Hz in Uplink-Richtung, sowie große Bandbreiten von bis zu 100 MHz erreicht werden. Bereits LTE nach Release 8 erreicht eine maximale spektrale Effizienz von 16 Bit/s/Hz in Downlink-Richtung. Die größte Kanalbreite von 20 MHz liegt jedoch weit unter den Anforderungen für 4G. [M.2134, 2008]

In LTE-A wird durch *Carrier Aggregation* eine Bandbreite von 100 MHz erreicht. Dabei werden bis zu fünf Kanäle mit jeweils 20 MHz zu einem Kanal zusammengeschlossen [Wannstrom, 2012]. Mit den bestehenden Mitteln aus LTE (64-QAM, 4x4 MIMO) sind so bis zu 1500 MBit/s erreichbar [Sauter, 2010]. Durch den Einsatz von 8x8 MIMO liegt die maximale Transferrate bei 3000 MBit. Ein großer Vorteil von Carrier Aggregation ist, dass die Netzbetreiber keinen zusammenhängenden Frequenzblock brauchen, was hinsichtlich der Frequenzknappheit in vielen Ländern unwahrscheinlich ist. Neben Carrier Aggregation und erweitertem MIMO, werden in 3GPP Release 10 auch Relaisstationen eingeführt, um die Bandbreite an den Zellrändern zu erhöhen und eine günstige Alternative zu Basisstationen zu liefern [3GPP, 2009].

## 3 Relaisstationen

Allgemein empfangen Relaisstationen Signale von einer Sendestation und leiten diese an eine weiter entfernte Empfängerstation weiter, ohne dabei auf eine kabelgebundene Anbindung an das Kernnetz angewiesen zu sein. Da die Kosten für Aufbau, Installation und Verlegung der Kabel einen Großteil der einmaligen Kosten ausmachen, können Relaisstationen wesentliche Einsparungen ermöglichen, obwohl die Hardware- und Betriebskosten teilweise höher liegen [Lang et al., 2009]. Durch die drahtlose Anbindung der Relaisstationen versprechen diese die Netzabdeckung kostengünstig zu steigern. Auch der flexible Einsatz an unzugänglichen Orten oder an besonders kostengünstigen Standorten, wie bspw. Straßenlaternen, ist denkbar. Weitere Einsatzmöglichkeiten sind unter anderem die Verbesserung des Durchsatzes in geschlossenen Räumen, die Erschließung von unterirdischen Regionen oder ein stabilerer Empfang in schnellen Verkehrsmitteln [Delbecke et al.]. Die drahtlose Anbindung an die Basisstation kann dabei technisch unterschiedlich komplex umgesetzt werden, wobei sich die Implementierungen hinsichtlich dem Entwicklungsaufwand, der Anschaffungskosten und der Leistung teilweise stark unterscheiden [Iwamura et al., 2010][Nagata et al., 2011]. Um zu verstehen, wie die Implementierung für den LTE-A Standard funktioniert, werden im Folgenden die drei grundlegenden Arten von Relaisstationen vorgestellt und die Vor- und Nachteile erläutert. Anschließend wird noch auf die Problematik gleicher, bzw. verschiedener Empfänger- und Senderfrequenzen in Relais und auf die Spezifikation von Relaisstationen in LTE Rel. 10 eingegangen.

### 3.1 Arten von Relaisstationen in Mobilfunknetzen

In der einfachsten Form, empfängt eine Relaisstation ein Signal, verstärkt dieses und sendet es weiter an den Empfänger. Diese Technik wird *Amplify-and-Forward-Relais* (AF-Relais) oder Layer1-Relais genannt. Wenn die Sendeleistung der Senderstation nicht mehr



### 3 Relaisstationen

stark genug ist um das Signal erfolgreich zu einer weit entfernten Empfängerstation zu senden, kann so die Signalstärke bei der Empfängerstation gesteigert werden. AF-Relais versprechen dadurch eine ausreichende Netzabdeckung bei geringen Kosten zu erreichen. Je nach Senderleistung müssen jedoch bis zu 30 Relaisstationen eingesetzt werden, um eine vergleichbare Netzabdeckung zu einer einzigen Basisstation zu erreichen [Lang et al., 2009]. Der wesentliche Nachteil von AF-Relais ist, dass das Rauschen und die Interferenzen zusammen mit dem Signal verstärkt und weitergeleitet werden. Wenn dadurch das *Signal to Interference plus Noise Ratio* (SINR) zu gering ist, kann das Signal nicht mehr erfolgreich dekodiert werden.

Um nicht nur die Signalstärke, sondern auch das SINR zu erhöhen, kann das von der Relaisstation empfangene Signal dekodiert und mit einem Fehlerkorrekturmechanismus rekonstruiert werden. Diese Art wird *Decode-and-Forward-Relais* (DF-Relais) oder Layer2-Relais genannt. Durch das Dekodieren der Pakete wird das Signal vom Rauschen und von Interferenzen getrennt. Liegt ein Paket in digitaler Form in der Relaisstation vor, wird es wieder kodiert und mit Fehlerkorrekturinformationen versehen. Der Empfänger kann aufgrund des deutlich höheren SINR das Signal öfter erfolgreich dekodieren. Aus Sicht der Netzbetreiber haben DF-Relais drei Nachteile gegenüber den einfachen AF-Relais. Erstens, verzögert das Dekodieren das Signal und erhöht die *round-trip time* (RRT), was allgemein in einem Telekommunikationsnetzwerk unerwünscht ist. Zweitens, ist die Hardware und Software komplexer. Es entstehen zusätzliche Kosten für die Entwicklung und Erprobung einer neuen Technologie. AF-Relais wurden dagegen schon in LTE Rel. 8 spezifiziert und sind kompatibel zu entsprechenden Mobilgeräten [3GPP, 2012]. Drittens, ist diese Implementierung nicht transparent für das mobile Endgerät da abgesehen von (De-)Modulation und (De-)Kodierung keine Funktionen von der Relaisstation übernommen werden. Das heißt zum Beispiel, dass ein Paket, welches fehlerhaft von dem Endgerät empfangen wird, von der Basisstation erneut gesendet werden muss. Die Relaisstation bietet keinen *Automatic Repeat Request* (ARQ) oder *Hybrid Automatic Repeat Request* (HARQ). [Iwamura et al., 2010]

Damit die Relaisstation transparent gegenüber der Mobilstation ist, können alle wichtigen Dienste einer Basisstation auch in der Relaisstation implementiert werden. Diese Art wird Layer3-Relais genannt. Dabei wird in der Relaisstation neben der Dekodierung auch die Verarbeitung von Benutzerdaten zur erneuten Übertragung (ARQ/HARQ) implementiert und die Relaisstation hat eine eigene *Physical Cell ID* (PCI). Insgesamt erscheint eine Layer3-Relaisstation aus Sicht der Endgeräte wie eine Basisstation. Das ist vor allem für

### 3 Relaisstationen

**Tabelle 3.1:** Vor- und Nachteile verschiedener Relaisstationen.

Art	Vorteile	Nachteile
Layer 1	<ul style="list-style-type: none"><li>• kostengünstig</li><li>• kompatibel zu LTE Rel. 8 Mobilgeräten</li></ul>	<ul style="list-style-type: none"><li>• SINR wird schlechter</li></ul>
Layer 2	<ul style="list-style-type: none"><li>• Rauschen wird eliminiert</li></ul>	<ul style="list-style-type: none"><li>• Verzögerung des Signals durch Dekodierung</li><li>• Implementierung relativ komplex</li><li>• inkompatibel zu LTE Rel. 8 Mobilgeräten</li></ul>
Layer 3	<ul style="list-style-type: none"><li>• Rauschen wird eliminiert</li><li>• kompatibel zu LTE Rel. 8 Mobilgeräten</li></ul>	<ul style="list-style-type: none"><li>• Verzögerung des Signals durch Dekodierung</li><li>• Verzögerung durch Verarbeitung von Benutzerdaten</li></ul>

Netzbetreiber von Vorteil, da bestehende Mobilgeräte von den Relaisstationen profitieren, ohne dafür eine neue Software zu benötigen. Wie bei Layer2-Relais entsteht auch in Layer3-Relais eine Verzögerung. Zusätzlich müssen die Benutzerdaten bearbeitet werden, so dass eine zusätzliche Latenz entsteht. Alle Vor- und Nachteile der drei Arten sind in Tabelle 3.1 zusammengefasst. [Iwamura et al., 2010]

## 3.2 Relais im LTE Standard

Wie im vorherigen Abschnitt bereits erwähnt wurde, sind Layer1-Relaisstationen bereits seit der LTE Rel. 8 Spezifikation standardisiert, wurden aber schon in 2G- und 3G-Netzwerken erforscht und eingesetzt [3GPP, 2012, Patwary et al., 2005]. Für LTE Rel. 10 wurden schließlich zwei Typen von Layer3-Relais spezifiziert [Akyildiz et al., 2010].

### 3 Relaisstationen

Die Art des RN wird je nach der Frequenz der Verbindung zwischen *enhanced Node B* (eNB) und RN unterschieden. Ein Typ 1 RN kommuniziert mit dem eNB auf derselben Frequenz, die auch von dem eNB für die Kommunikation mit dem *User Equipment* (UE) verwendet wird. Bei einem dedizierten Frequenzband für die RN-eNB-Verbindung, spricht man dagegen von einem Typ 1a RN. Letzteres hat den Vorteil die Gefahr von zusätzlichen Interferenzen in der Zelle zu minimieren. Dafür wird ein zusätzliches Frequenzband benötigt. Diese sind auf Grund der hohen Nachfrage jedoch sehr begrenzt. Für die RN-eNB Verbindung wird mindestens *Time-Division Duplex* (TDD) und *Frequency-Division Duplex* (FDD) zur Ressourcenaufteilung zwischen Uplink- und Downlink-Richtung vorausgesetzt. [Akyildiz et al., 2010]

Beide Typen treten als unabhängige Zelle mit eigener Physical ID auf, können HARQ- und Signalisierungs-Daten direkt zwischen RN und UE austauschen und erscheinen für LTE Rel. 8 UEs als LTE Rel. 8 eNB. Aufgrund dieser Eigenschaften sind beide Typen nach der Kategorisierung aus Abschnitt 3.1 den Layer 3 RNs zu zuordnen. Es ist einem RN auch erlaubt sich gegenüber moderneren UEs anders zu identifizieren. So sollen zukünftig weitere Verbesserungen möglich gemacht werden [Akyildiz et al., 2010]. Beispiele sind das *Joint Processing* und das *Coordinated Beamforming*. Beide Techniken gehören zu den *Coordinated Multi-Point*-Übertragungen (CoMP-Übertragungen). Bei ersterem ist das Ziel, die Signalqualität zu verbessern und Interferenzen aktiv zu eliminieren, indem ein Signal von mehreren eNBs und RNs aus an ein UE gesendet wird. Bei letzterem, sendet nur ein eNB oder RN. Alle benachbarten Zellen versuchen ihre Übertragungen so zu koordinieren, dass die Signalqualität bei dem UE maximiert wird [Nam et al., 2010].

# 4 openWNS

## 4.1 Aufbau von OpenWNS

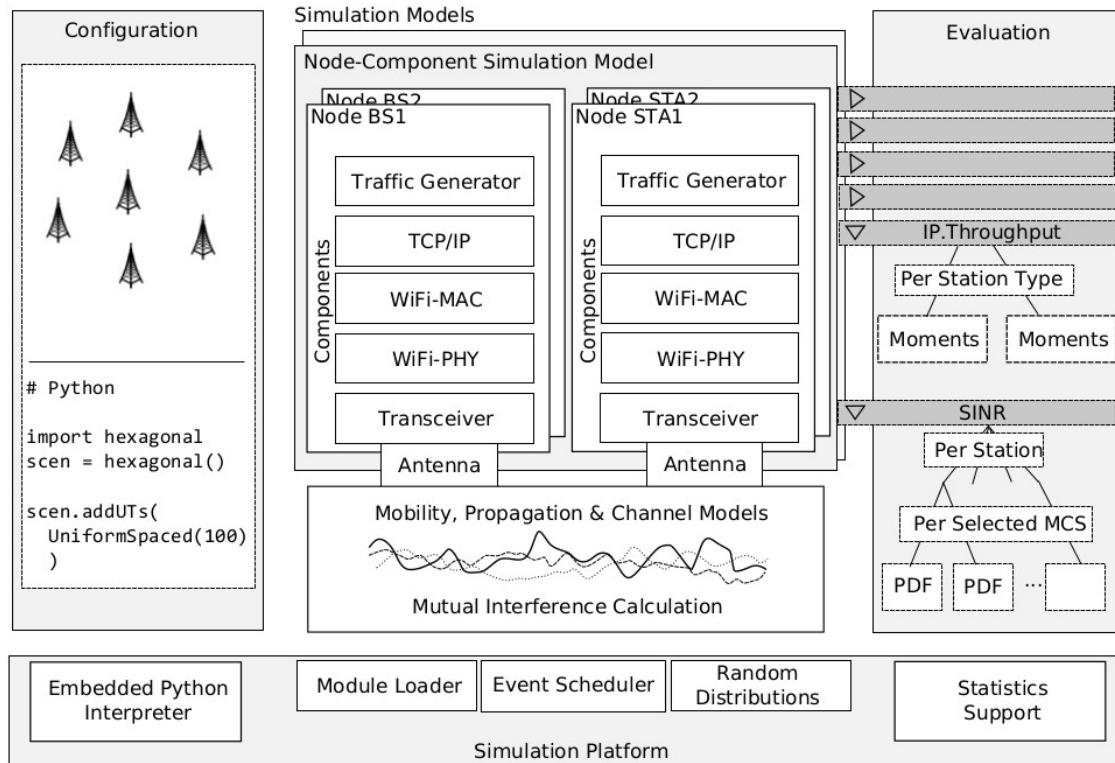
Der Netzwerksimulator openWNS ist modular aufgebaut, wodurch die Plattform einfach erweitert werden kann und bestehende Module ausgetauscht werden können. Die Simulationsplattform kann die benötigten Module bei Bedarf aus Bibliotheken laden. Unter anderem liefert openWNS Module für den Netzwerkstandard IEEE 802.16, zur Simulation von Interferenzen (RISE), für die Internet Protokolle TCP und IP, sowie ein Modul welches das Datenaufkommen simuliert (Constanze). Alle Module implementieren das Node-Component-Modell (Abschnitt 4.3), wodurch Nodes oder Components aus verschiedenen Modulen in der selben Simulation verwendet werden können.

## 4.2 Konfiguration von Szenarien

Szenarien werden in einer Python-Konfiguration beschrieben, welche von dem integrierten Python-Interpreter gelesen wird. Als objektorientierte Sprach, bietet Python einige Vorteile gegenüber datenzentrischen Sprachen wie XML. Erstens, können wiederkehrende Aufgaben als Funktionen umgesetzt werden. Gerade bei großen Szenarien mit vielen Einheiten, wie eNBs und UEs, vereinfacht dies das Aufsetzen einer Simulation. Zweitens, können Klassen und abgeleitete Klassen gebildet werden. So ist es z.B. möglich eine Basisklasse für alle Arten von Funktstationen zu implementieren und davon eine Klasse für eNBs und eine Klasse für UEs abzuleiten. [Bultmann et al., 2009]

Um die Konfiguration weiter zu vereinfachen, gibt es zu jeder Komponente in openWNS eine Klasse in Python. Neben wichtigen Funktionen zur Konfiguration dieser Komponente, stellt diese eine sinnvolle Ausgangskonfiguration bereit. So soll es neuen Nutzern der Plattform erleichtert werden eine grundlegende Simulation zu erstellen, an welcher die

## 4 openWNS



**Abbildung 4.1:** Aufbau von openWNS. Die wichtigsten Komponenten von openWNS sind die Konfiguration in Python, das Simulationsmodell, die Evaluation und die Simulationsplattform. [Wehrle and Gross, 2010]

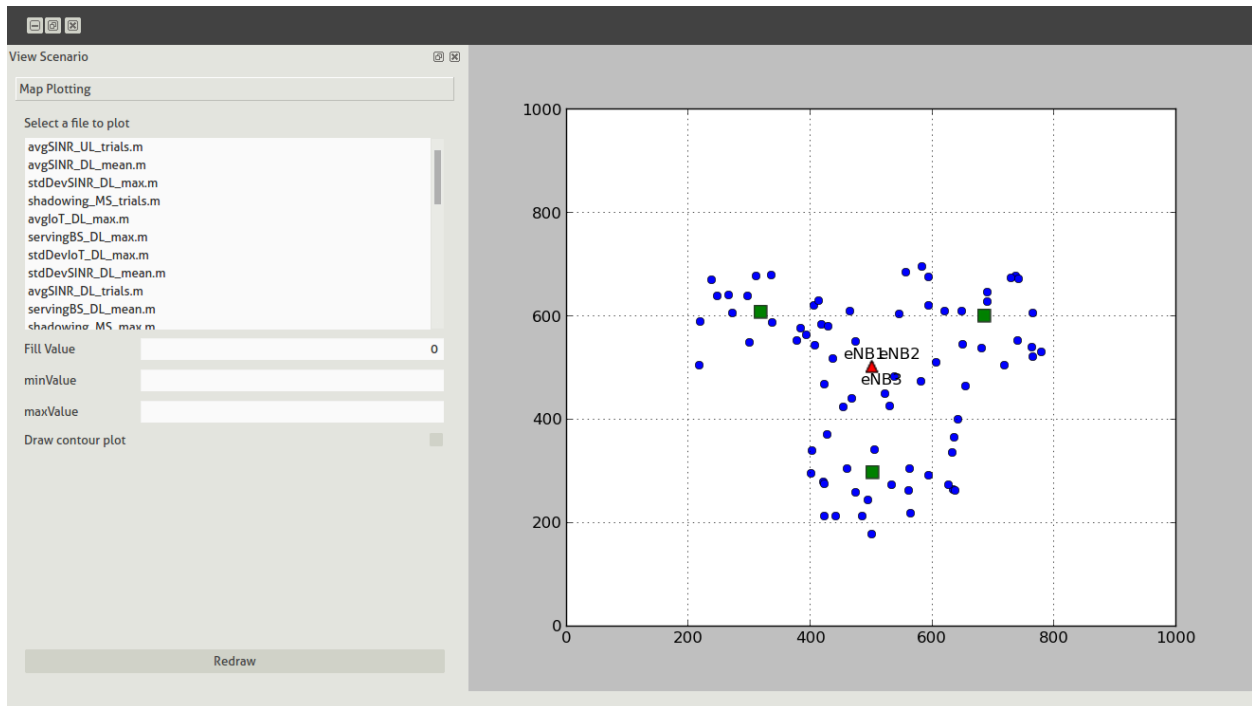
gewünschten Anpassungen vorgenommen werden können. Letztlich bieten Programme wie PyTree, welches im Rahmen von openWNS entwickelt wurde, und imtaphyViewer, welcher zu IMTaphy gehört, die Möglichkeit die Konfiguration grafisch darzustellen.

Insgesamt bietet Python damit eine Grundlage um Szenarien verschiedenster Komplexität relativ einfach zu konfigurieren und der Benutzer wird durch grafische Programme weiter unterstützt.

### 4.3 Node-Component Modell

Um einzelne Komponenten einer Software unabhängig voneinander weiterentwickeln und austauschen zu können, muss eine einheitliche Schnittstelle festgelegt werden. So

## 4 openWNS



**Abbildung 4.2:** Screenshot von dem imtaphyViewer: Der Benutzer kann sich die Positionen der eNBs (rotes Dreieck), RNs (grüne Vierecke) und UEs (blaue Kreise) anzeigen lassen.

können Module untereinander über die gegebenen Wege kommunizieren und die Simulationsplattform kann Module ohne weitere Kenntnis über die innere Struktur laden und initialisieren.

Intern orientiert sich der Aufbau des Node-Component-Modells an der Struktur von realen Funknetzwerken und den eingesetzten Protokollschichten. So wird das zu simulierende Netzwerk aus verschiedenen *Nodes* für eNBs und UEs aufgebaut, die alle von der Klasse *Node* abgeleitet werden. Ein *Node* stellt die Infrastruktur, in die *Components* eingesetzt werden, um so das Verhalten und die Eigenschaften einer Station abzubilden. Alle *Nodes* implementieren eine einheitliche Schnittstelle, über welche sie von dem Simulationsmodell angesprochen werden. Unter anderem werden folgende Funktionen von einem *Node* implementiert:

- `addService()` : Fügt einen Service zu der Registrierung hinzu, welcher von einer *Component* bereitgestellt wird.
- `startup()` : Wird direkt nach dem Konstruktor aufgerufen.

## 4 openWNS

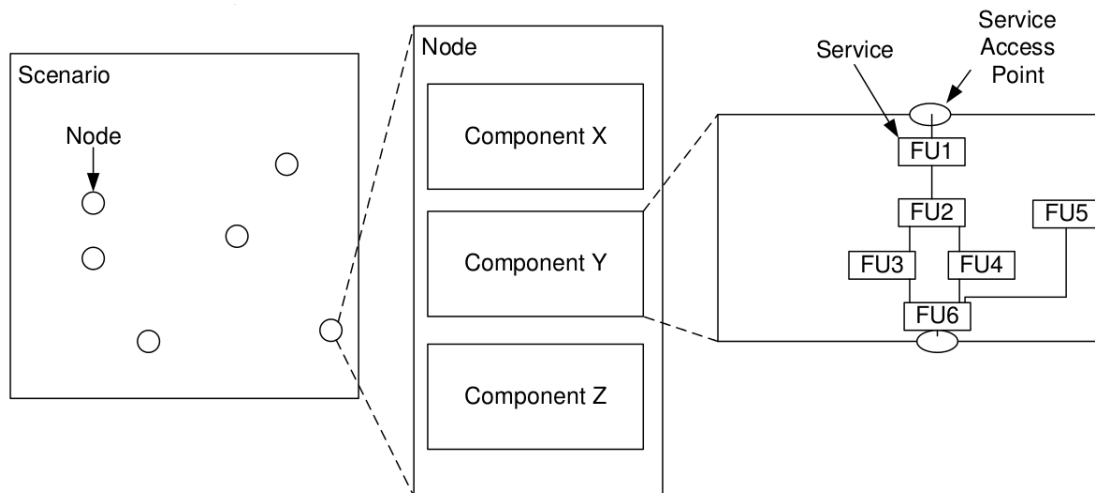
- `onWorldCreated()` : Wird aufgerufen, nachdem alle Nodes initialisiert wurden.
- `onShutdown()` : Wird aufgerufen, wenn die Simulation beendet ist.

Die Funktion `addService()` wird von Components innerhalb des Nodes aufgerufen, wenn diese einen Dienst für andere Components in dem selben Node bereit stellen wollen. `onWorldCreated()` und `onShutdown()` rufen die entsprechenden Funktionen in den einzelnen Components des Nodes auf. Besonders erstere ist wichtig, da beim Aufrufen dieser Funktion alle Nodes und Components initialisiert sind. Hier können also Interdependenzen zwischen Components aufgelöst werden. So können die Dienste, welche vorher von anderen Components registriert wurden, abgefragt werden. [Bultmann et al., 2009]

Components repräsentieren annähernd die Protokollschichten, wie sie im *Open Systems Interconnection* (OSI) Referenzmodell beschrieben werden. Das heißt, eine Component implementiert z.B. die Funktion der Bitübertragungsschicht (eng. Physical Layer) und eine weitere die Datensicherungsschicht (eng. Data Link Layer) [Zimmermann, 1980]. Untereinander kommunizieren Components nicht über eine feste Schnittstelle. Stattdessen, können Components Dienste unter einem Namen bei ihrem Node registrieren (siehe oben). Die Schnittstelle wird in einer separaten Klasse definiert, welche einen Teil der Funktionen der eigentlichen Klasse für andere Components verfügbar macht. Dieses Vorgehen ist flexibler als eine statische Schnittstelle. Da ein Node eine abgegrenzte Einheit darstellt und meist als eine Einheit weiterentwickelt wird, ist dies sinnvoll, auch wenn Änderungen an den Schnittstellen ggf. zu notwendigen Anpassungen an anderer Stelle im Node führen. Da Dienste durch einen Name identifiziert werden, ist es auch möglich den gleichen Dienst durch mehrere Components unter anderem Namen zugänglich zu machen. Dies hilft z.B. bei der Implementierung von Relaisstationen.

### 4.4 Functional Units

Wie in 4.3 zu sehen ist, setzen sich Components aus *Functional Units* (FUs) zusammen. Diese werden wiederum verbunden und bilden so ein *Functional Unit Network* (FUN). Die FUs übernehmen die Paketverarbeitung in einer Component. Um Informationen mit anderen FUs im selben und anderen Nodes auszutauschen, enthält jedes Paket zusätzliche Daten. (siehe Abschnitt 4.6).



**Abbildung 4.3:** Übersicht über das Node-Component-Modell. Ein Modell stellt unterschiedliche Nodes zur Verfügung, welche wiederum aus mehreren Components aufgebaut sind. Die Components verarbeiten die ein- und ausgehenden Pakete in einem Netzwerk aus FUs. In der Konfiguration werden Instanzen der Nodes beschrieben, um so ein Szenario aufzubauen. [Bultmann et al., 2009]

Anders als die Components, welche durch registrierte Dienste kommunizieren, verwenden FUs eine statische Schnittstelle, welche bei Bedarf um weitere Funktionen erweitert werden kann (siehe Abb. 4.4). Um Pakete auszutauschen implementieren alle FUs folgende vier Funktionen:

- `sendData()`
- `isAccepting()`
- `onData()`
- `wakeup()`

Die Methode `sendData()` einer FU wird von einer anderen FU aufgerufen, wenn diese ein Paket in ausgehender Richtung weiterleiten möchte. Hat eine andere FU dagegen ein eingehendes Paket verarbeitet, ruft sie `onData()` auf. Diese beiden Methoden reichen theoretisch aus, um den kompletten Paketfluss zu realisieren. Ein Problem entsteht jedoch, wenn es zu Engpässen in einer FU kommt, meist durch die limitierte Übertragungsgeschwindigkeit der physikalischen Schicht. Da die meisten FUs keine Pakete zwischenspeichern, würden Pakete in der limitierenden FU verloren gehen. Auch ein Zwischenspeicher in den entsprechenden FUs würde zu Problemen führen. So könnte ein Paket durch eine



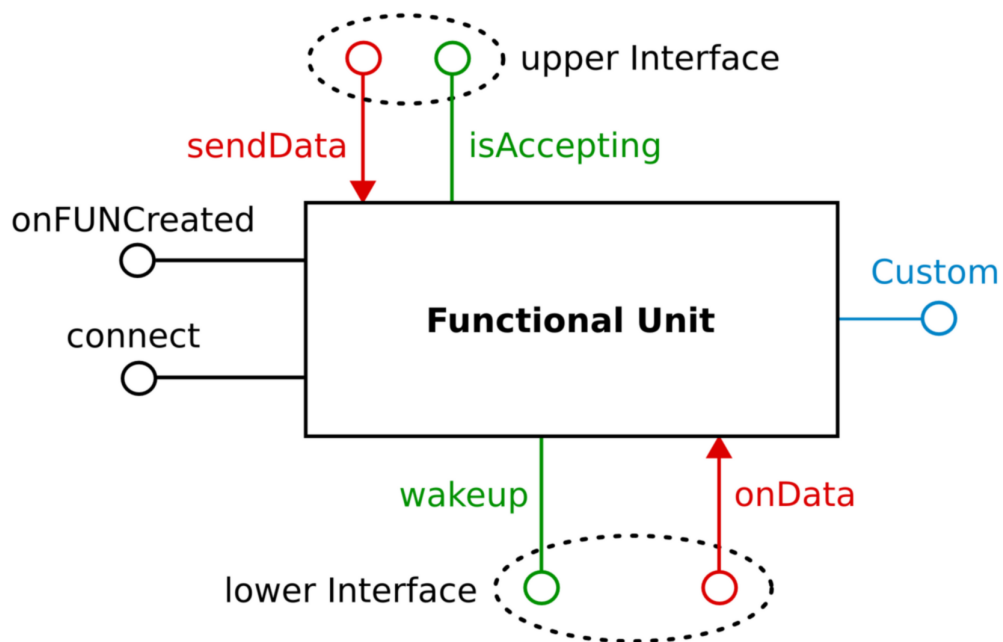
Zeitüberschreitung im HARQ-Mechanismus für eine erneute Übertragung gesetzt werden, ohne dass die vorherige Übertragung jemals den Zwischenspeicher in der physikalischen Schicht verlassen hätte, also noch gar nicht übertragen worden ist [Schinnenburg et al., 2005]. Das geschieht, wenn der Zwischenspeicher in der limitierenden FU viele Elemente enthält. Pakete die neu in den Speicher kommen müssen dann zu lange warten bis die vorher eingetroffenen Pakete abgearbeitet worden sind. Dies könnte zwar durch eine begrenzte Größe eingeschränkt werden, jedoch zu dem Preis, dass das erste Problem zurückkehrt, wenn das Limit erreicht ist. Damit ein reibungsloser Fluss in allen Lastfällen gewährleistet werden kann, wird zusätzlich eine Schnittstelle für die Signalisierung zwischen den FUs benötigt. Da die physikalische Schicht nur in der ausgehenden Richtung begrenzt, ist die Flusskontrolle auch nur für diesen Fall ausgerichtet. Dabei stellt jede FU je zwei Funktionen bereit. `isAccepting()` gibt Auskunft darüber, ob die FU bereit ist ein weiteres Paket zu verarbeiten. `wakeup()` kann aufgerufen werden um verbundenen FUs mitzuteilen, dass Kapazitäten frei geworden sind. Essentiell ist, dass alle FUs `isAccepting()` aufrufen, bevor `sendData()` aufgerufen wird. Wie in 4.4 dargestellt, bilden diese beiden Funktionen zusammen das *upper Interface*, `onData()` und `wakeup()` das *lower Interface*. [Schinnenburg et al., 2007]

Zusätzlich zu der allgemeinen Schnittstelle können weitere Funktionen definiert werden. Diese können von anderen FUs verwendet werden, um zusätzliche Informationen über eine FU abzufragen, bzw. um diese an eine andere FU weiterzuleiten. So kann die Entscheidung zu welcher FU ein Paket geschickt wird, basierend auf den Eigenschaften der möglichen Ziele getroffen werden.

## 4.5 Multiplexing

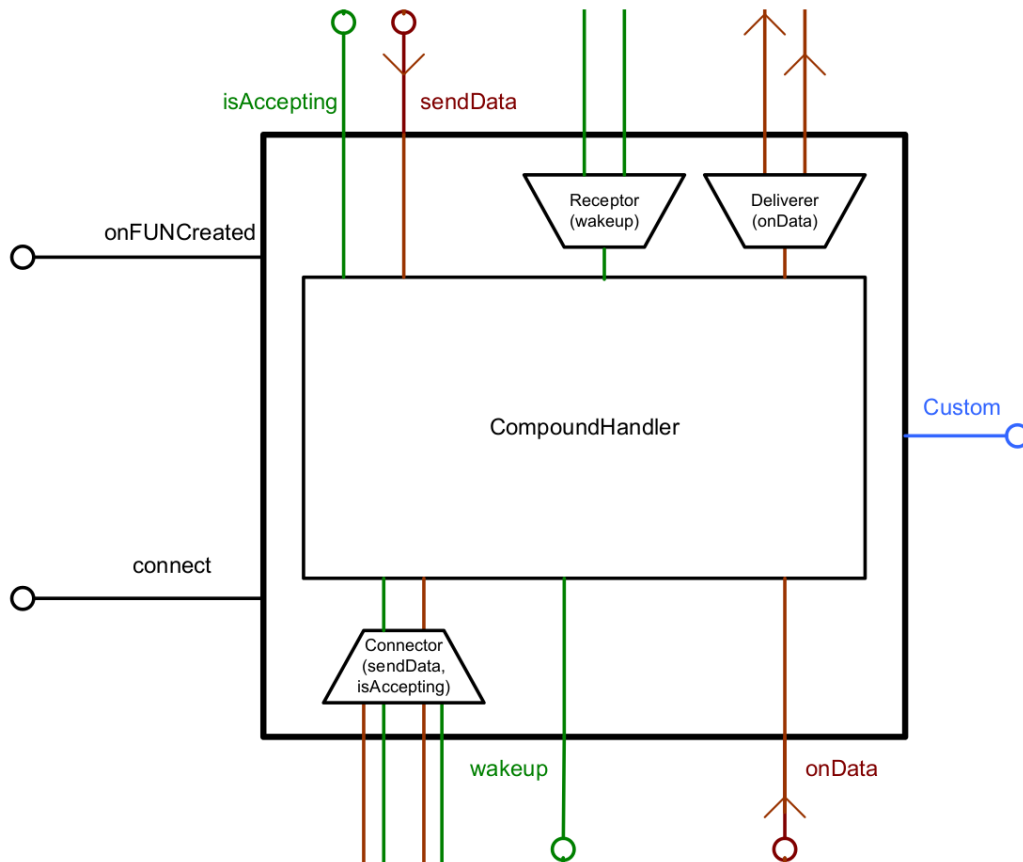
In der Praxis sind Protokollschichten selten linear aufgebaut. Vielmehr durchlaufen Pakete verschiedene Wege durch ein FUN, abhängig z.B. von Start oder Ziel des Pakets. Um dieses Multiplexing zu ermöglichen verwaltet jede FU drei Listen:

1. Connector-Liste
2. Deliverer-Liste
3. Receptor-Liste



**Abbildung 4.4:** Die Schnittstelle einer FU: Grundsätzlich bieten alle FUs eine obere und eine untere Schnittstelle, Funktionen für die Verwaltung, sowie die Möglichkeit Erweiterungen zu definieren. [Schinnenburg et al., 2007]

Die Connector- und Deliverer-Liste werden mit einem Filter verknüpft. Dieser kann nach Namen der FUs, nach MAC-Adressen oder nach aktivierten Commands unterscheiden und zutreffende Compounds durchlassen. Zusätzlich sind Filter möglich, welche alle bzw. keine Pakete durchlassen. Ein Filter muss demnach nicht eindeutig sein, sondern kann ein Paket auch an beliebig viele FUs weiterleiten. Wie in 4.5 zu sehen ist, rufen die ersten beiden Listen entsprechend der Richtung `isAccepting()` und dann `sendData()` bzw. `onData()` aller FUs auf, die auf den Filter zutreffen. Damit können FUs zu einem komplexen FUN vernetzt werden. Einzelne FUs in einem Netzwerk können als Ausgang gekennzeichnet werden. Pakete verlassen das FUN durch diese FUs zu höher gelegenen Components. Im Unterschied dazu, können Pakete durch eine beliebige FU in das Netzwerk fließen. Dazu ruft eine andere Component die `sendData()` oder `onData()` Funktion einer FU in dem Netzwerk auf. [Schinnenburg et al., 2007]



**Abbildung 4.5:** Multiplexing in einem FUN. Eine FU kann anhand der Connector-, der Deliverer- und der Receptor-Liste und entsprechenden Regeln entscheiden wohin ein Paket fließt. [Schinnenburg et al., 2007]

## 4.6 Compounds

In openWNS heißt ein Paket *Compound* und stellt hauptsächlich den Datenverkehr in dem virtuellen Netzwerk dar. Zusätzlich werden sie von FUs mit Informationen zur Signalisierung versehen. So kann eine HARQ-FU ein neues Compound als Antwort auf den eingehenden Datenverkehr generieren. Dieses Compound wird mit Informationen versehen und in die entgegengesetzte Richtung geschickt. In diesem Compound wird die HARQ-FU in dem anderen Node über Erfolg oder Misserfolg der Übertragung informiert, in dem sie die angehängten Daten des Compounds liest. Ein weiteres Beispiel ist ein RN, der die Media-Access-Control-Adresse (MAC-Adresse) des Ziels braucht um das Compound weiterzuleiten. Diese Zusatzinformationen heißen *Commands*.

### 4.6.1 Commands

Jede FU in einem FUN definiert einen *Command-Type*. Die Command-Types aller FUs werden an jedes Compound angehängt. Eine FU kann direkt nur auf den Command mit ihrem eigenen Command-Type zugreifen. Durch diesen Mechanismus des Informationsaustausches, brauchen andere FUs keinerlei Kenntnis über die angehängten Commands haben. Das Compound fließt einfach durch alle FUs durch und jede FU modifiziert ihren Command ohne die Funktion der anderen FUs zu stören. In dem Beispiel der HARQ-Übermittlung, braucht keine FU über oder unter der HARQ-FU Kenntnis von dem Command dieser FU haben. Die HARQ-FU ist vollkommen transparent für andere FUs. Lediglich die FU in einem anderen Node, welche ebenfalls den HARQ-Command-Type implementiert, greift auf die Informationen zu und reagiert entsprechend.

In dem zweiten Beispiel kann es jedoch sein, dass eine FU die Informationen einer anderen FU braucht. So wird z.B. die Ziel-MAC-Adresse in dem Command der PDCP-FU (Packet Data Convergence Protocoll) gespeichert. Der Scheduler, welcher Compounds zur Übermittlung an die physikalische Schicht weiterleitet, muss jedoch diese Adresse kennen und als Ziel-Adresse weitergeben. Diese FU muss also neben ihrem eigenen Command, den *Downlink Control Informations* (DCI), auch den Command der PDCP-FU kennen.

Da keine einzelne FU die Struktur aller Commands kennt, wird diese Aufgabe von dem FUN übernommen. Diese Einheit kennt alle Commands die von FUs in einem FUN registriert wurden und kann diese für alle anderen verbundenen FUs bereitstellen. Dafür muss die FU den Namen des gesuchten Commands kennen, anhand dessen dieser eindeutig in der Registrierung identifiziert wird. [Schinnenburg et al., 2005]

### 4.6.2 Command Pool

Die Menge aller Commands aus einem FUN wird an jedes Compound angehängt. Diese Commands bilden den *Command Pool*. Ein Compound besteht also aus den Nutzdaten und dem Command Pool. Zu Beginn sind alle Commands in einem Command Pool inaktiv. Wenn das Compound durch eine FU fließt, aktiviert diese ihren zugehörigen Command. Dieses Vorgehen stellt sicher, dass FUs in einem anderen FUN nicht auf einen

#### 4 *openWNS*

Command zugreifen, welcher keine relevanten Informationen enthält, da er nie von der entsprechenden FU bearbeitet wurde.

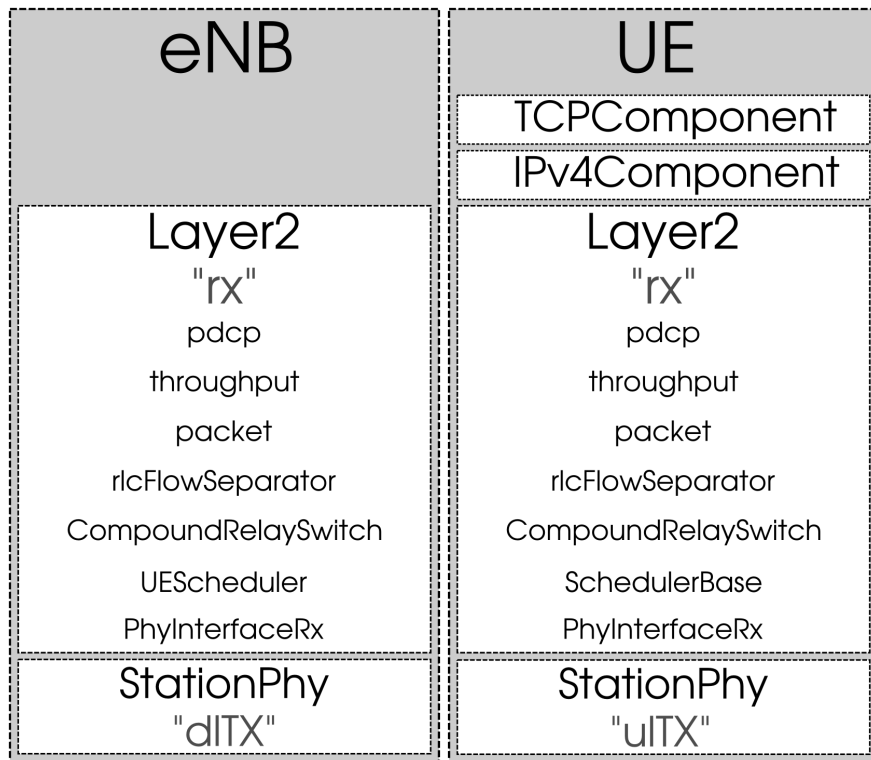
Dadurch entsteht jedoch ein Problem für RNs [Pabst et al., 2004]. Wird ein Compound von einem eNB durch ein RN an ein UE weitergeleitet, kann es passieren, dass ein Command, der schon aktiviert ist, nochmals aktiviert wird. Diese Gefahr besteht nur bei FUs, die unter dem Wendepunkt liegen. Die in [Schinnenburg et al., 2005, 2007] vorgeschlagene Lösung wird in Kapitel 5.2.2 erläutert.

## 5 Implementierung

Die Implementierung basiert auf der Idee von Quaium [2011, S. 40], einen RN als einen eNB und ein UE zu sehen. Mit diesem Ansatz können viele bestehende Komponenten aus openWNS und IMTAphy wiederverwendet werden. Potentiell sinkt dadurch der Aufwand erheblich, da weniger neue Elemente umgesetzt werden müssen und die vorhandenen Komponenten bereits getestet und integriert sind. Ein Nachteil ist jedoch, dass Teile unter Umständen angepasst werden müssen, um auch mit RNs zu arbeiten.

Aus dem vorherigen Abschnitt ist bekannt, dass die eNBs und RNs jeweils als Node implementiert sind. Daher können eNBs und UEs nur über die vorgegebenen Schnittstellen, genauer über Compounds, kommunizieren. Das heißt auch, dass ein Verbund aus einem eNB und einem UE über Compounds kommunizieren muss. Die Compounds werden jedoch über die Luftschnittstelle übertragen, was zusätzliche Interferenzen für andere, echte Übertragungen verursacht. Ein weiteres Problem ist, dass ein Compound ein zusätzliches mal kodiert und dekodiert wird. Dafür werden zusätzliche Verzögerungen berechnet, was eine weitere Anpassung nötig machen würde.

Statt ein eNB und ein UE zu koppeln, wird der modulare Aufbau von openWNS und IMTAphy ausgenutzt, um *einen* RN-Node aus den einzelnen Components und FUs aus eNB und UE aufzubauen. Der Aufbau von eNB- und UE-Nodes, mit Anpassungen für die Unterstützung von RNs, ist in Abbildung 5.1 zu sehen. In beiden Nodes bietet die Layer2-Component den Dienst “rx” an. Die StationPhy registrieren ihren Dienst unter “dlTX” oder “ulTX”, abhängig davon ob sie in einem eNB oder UE sind. Dabei gilt allgemein: Der Name entspricht der Sende-Richtung der StationPhy. Da eNBs in Downlink-Richtung senden, heißen deren StationPhys “dlTX”. Des Weiteren wird die FU “CompoundRelaySwitch” zum Layer2-FUN hinzugefügt. In dem RN übernimmt sie die Aufgabe Compounds weiterzuleiten. In den eNBs und UEs übernimmt sie keine Aufgabe und lässt alle Compounds



**Abbildung 5.1:** Die Components in eNB und UE. Zusätzlich zu StationPhy und Layer2, umfasst ein UE zusätzlich IPv4Component und TCPComponent. Dadurch kann ein UE im Netzwerk adressiert werden.

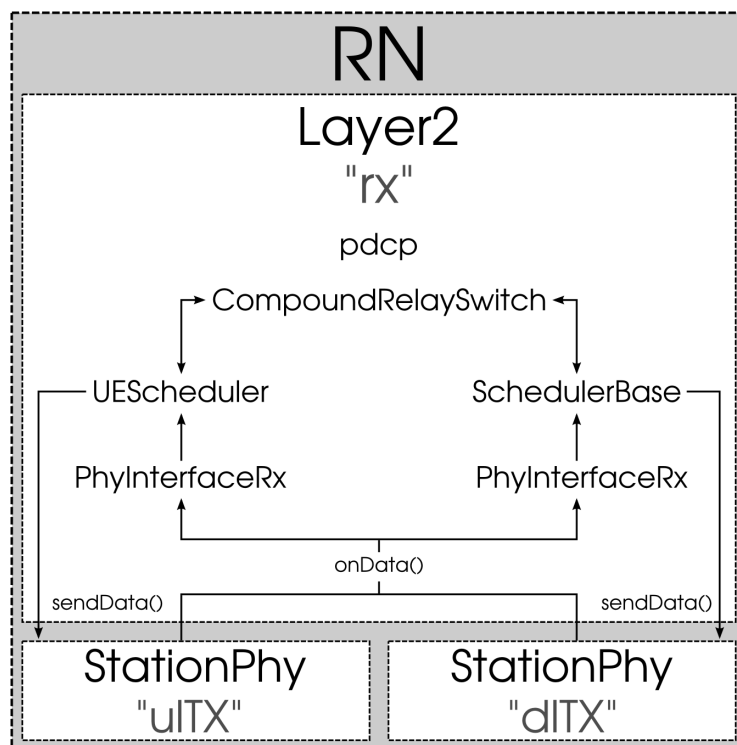
durchlaufen, aktiviert jedoch ihren Command. Auf die Einzelheiten dieser FU wird in Abschnitt 5.2.2 eingegangen.

## 5.1 StationPhy

In einem normalen Node ist es nicht notwendig die StationPhy in Uplink- oder Downlink-Richtung zu unterscheiden. Ein RN braucht jedoch zwei StationPhy-Components. Eine für jede Richtung. Diese registrieren sich bei der Channel-Klasse und ermöglichen so Daten über die Luftschnittstelle zu senden und zu empfangen. Jede StationPhy verwaltet außerdem die Antennen und einen Empfänger. Erstere bestimmt unter anderem die Verstärkung des Signals durch die physikalischen Eigenschaften der Antenne und letztere das Rauschen im Empfänger. Beide Größen müssen für UE- und eNB-Teil eines RN unabhängig konfiguriert werden können (siehe Abschnitt 6.1). Die StationPhy auf der UE-Seite

## 5 Implementierung

registriert den Dienst "ulTX", da über diese Einheit in Uplink-Richtung gesendet wird. Entsprechend heißt die StationPhy auf der anderen Seite "dlTX" (Abbildung 5.2). Ausgehende Pakete, welche den Layer2 in Richtung physikalischer Schicht verlassen, können so vom Layer2 an die richtige Component geschickt werden. Auch die StationPhys selber sind auf diese Differenzierung angewiesen. So muss eine StationPhy beim Registrieren einer neuen Übertragung beim Channel die Empfänger-StationPhy weitergeben. Diese kennt die StationPhy jedoch nicht direkt. Stattdessen fragt sie den Empfänger-Node nach dem entsprechenden Dienst. Wenn also eine StationPhy in einer UE ein Paket an ein RN sendet fragt sie nach dem Dienst "dlTX". Der RN gibt daraufhin die StationPhy auf der eNB-Seite zurück (rechte Seite in Abbildung 5.2).



**Abbildung 5.2:** Components und FUs in einem RN. Die StationPhys rufen `onData()` um Compounds zum Layer2 zu schicken. Ausgehende Compounds werden von den Schemulern direkt an die jeweilige StationPhy weitergegeben.

Damit Pakete bei einem UE ankommen, welches über ein RN angebunden ist, muss die Ziel-Adresse im Compound angepasst werden. Die Adresse des UEs wird dabei durch die Adresse des jeweils nächsten Nodes auf dem Weg zum Ziel ersetzt. Die Einheit die diese Aufgabe übernimmt, muss dazu Kenntnis über die gesamte Struktur des Netzwerks haben



## 5 Implementierung

oder erhalten. Daher bietet sich hierfür die StationPhy der Nodes an. Diese sind direkt mit dem Channel verbunden, über welchen sie den LinkManager erreichen. Dieser verwaltet alle möglichen, sowie alle aktuellen Zuordnungen von UEs zu eNBs bzw. RNs. Zusätzlich zu den direkten Verbindungen, erhalten die StationPhy nun eine Liste mit allen indirekt verbundenen UEs und dem RN über welchen ein Paket zu dem Ziel gelangt (siehe dazu Abschnitt 5.3). Wie im dazugehörigen Listing 5.1 zu sehen ist, wird der LinkManager nur befragt, wenn die StationPhy in einem eNB oder auf der eNB-Seite eines RN ist. StationPhy welche als UE agieren, haben nur eine zugeordnete Basisstation. Diese bekommen sie über eine separate Funktion des LinkManagers.

Es sei darauf hingewiesen, dass auch mehrere RN zwischen dem Sender und dem Empfänger liegen dürfen. In der Liste wird auf den jeweils *nächsten* RN auf dem Weg verwiesen, wodurch das Paket von RN zu RN fließt, bis letztlich das Ziel erreicht ist. Im Rahmen dieser Arbeit werden jedoch nur Szenarien mit einfacher Verkettung behandelt und die Implementierung nur anhand solcher Netzwerke evaluiert. Grundsätzlich sollte diese Implementierung aber auch mit komplexeren Netzwerken fehlerfrei funktionieren.

Auch andere Components innerhalb eines Nodes fragen die StationPhy nach den zugeordneten UEs, da diese selber keine Kenntnis von dem LinkManager haben. Die StationPhy geben hier nicht mehr nur alle direkt, sondern auch alle indirekt verbundenen UEs weiter. In eNBs werden auf diesem Weg alle UEs bis zum *Evolved Packet Core gateway* (EPCgw) weitergeleitet. In dem EPCgw wird das Datenaufkommen für die Simulation generiert. Dafür ist dieser mit allen eNBs verbunden und bekommt von diesen alle zugeordneten UEs. Wenn ein Compound zu einem UE gesendet werden soll, sucht der EPCgw nach dem eNB welcher dieses UE gemeldet hat. Würden indirekt verbundene UEs nicht von eNBs gemeldet werden, würden diese nicht gefunden und könnten keine Pakete erhalten.

### 5.2 Layer2

Wie eingehend erwähnt, meldet die Layer2-Component nur einen Service mit dem Name "rx" an. Der Grund hierfür ist, dass diese Component auch in einem RN nur einmal vorhanden ist. Dafür ändert sich der Aufbau des FUN, welches die Compounds verarbeitet und eine neue FU übernimmt die Aufgabe Compounds in die jeweils andere

## 5 Implementierung

Listing 5.1: Aus der Funktion `StationPhy::getAssociatedNodes()` :

```
1 std::vector<wns::node::Interface*> associations;
2 if(stationType == BASESTATION)
3 {
4     // Direkt verbundene UEs werden in associations gespeichert:
5     LinkMap servedLinks = channel->getLinkManager()
6                             ->getServedLinksForBaseStation(this);
7     for(LinkMap::const_iterator iter = servedLinks.begin();
8         iter != servedLinks.end();
9         iter++)
10    {
11        associations.push_back(iter->first->getNode());
12    }
13
14    StationMap indirectlyServedLinks = channel->getLinkManager()
15                                         ->getIndirectlyServedLinksForBaseStation(this);
16
17    // Indirekt verbundene UEs werden in associations, der direkte
18    // Nachfolger in resolveDestination gespeichert:
19    for(StationMap::const_iterator iter = indirectlyServedLinks.begin();
20        iter != indirectlyServedLinks.end();
21        iter++)
22    {
23        StationPhy* servedStation = iter->first;
24        associations.push_back(servedStation->getNode());
25        resolveDestination[servedStation->getNode()] = iter->second
26                                                         ->getNode();
27    }
28 }
```

Richtung weiterzuleiten. Der komplette Aufbau des FUN ist in Abbildung 5.2 zu sehen.

### 5.2.1 Das Functional Unit Network

Eintreffende Compounds werden von den StationPhys an die Layer2-Component geschickt und von dieser weiter an die PhyInterfaceRx-FU. Entsprechend der Anzahl von StationPhys, sind in einem RN zwei dieser FUs vorhanden. Damit die Layer2-Component entscheiden kann, an welche der beiden FUs ein Compound weiterzuleiten ist, übergeben die StationPhy zusätzlich zu dem Compound ihre Art, d.h. UE oder eNB (Listing 5.2 Zeile 4).

## 5 Implementierung

Listing 5.2: Aus der Definition von Layer2:

```
1      void onData(std::vector<wns::ldk::CompoundPtr> transportBlocks ,
2                  wns::node::Interface* source ,
3                  imtaphy::interface::TransmissionStatusPtr status ,
4                  imtaphy::StationType callerType);
```

Wenn das Paket erfolgreich dekodiert worden ist, leitet die `PhyInterfaceRx-FU` das Compound an den Scheduler weiter. In eingehender Richtung hat dieser keine Aufgaben und schickt das Compound weiter. In ausgehender Richtung sammelt der Scheduler Compounds von allen zugeordneten Nodes und entscheidet welche Compounds im nächsten *Transmission Time Interval* (TTI) übermittelt werden. Auch von dieser FU sind in einem RN zwei vorhanden. Auf der UE-Seite wird die *UEScheduler-FU*, auf der eNB-Seite die *SchedulerBase-FU* verwendet. Diese unterscheiden sich unter anderem in der Strategie nach welcher Compounds ausgewählt werden, da in Uplink-Richtung nur eine Verbindung verwaltet werden muss.

Damit die Scheduler mit der richtigen physikalischen Schnittstelle kommunizieren, werden diese mit dem Namen des entsprechenden Dienstes initialisiert. Durch die direkte Kommunikation dieser FUs mit der *StaionPhy-Component* wird die `PhyInterfaceRx-FU` in ausgehender Richtung übergangen. Über den beiden Schemulern liegt die *CompoundRelaySwitch-FU*, welche im nächsten Abschnitt beschrieben wird.

Die letzte FU in dem FUN ist die *PDCP-FU*. In RNs ist sie mit keiner anderen FU verbunden. Die Compounds werden schon vorher weitergeleitet. In dem Command dieser FU werden aber die Start- und Ziel-Adressen jedes Paketes gespeichert. Diese werden von dem Scheduler auf der eNB-Seite gebraucht, um Pakete zu verschicken. Da Commands nur ausgelesen werden können, wenn die dazugehörige FU in dem FUN registriert ist, muss die *PDCP-FU* auch im Layer2 von RNs vorhanden sein.

### 5.2.2 CompoundRelaySwitch

Die *CompoundRelaySwitch-FU* verwendet die Multiplexing-Schnittstelle, wie sie in Abschnitt 4.5 beschrieben ist. In der Connector-Liste befinden sich die beiden Scheduler, in Listing 5.4 sind das “*ueScheduler*” und “*nbDLScheduler*”. Die Deliverer-Liste bleibt leer,

## 5 Implementierung

denn anders als die anderen FUs, leitet der CompoundRelaySwitch eingehende Compounds nicht an `onData()` der FUs in der Deliverer-List weiter, sondern wieder zurück an `sendData()` der FUs in der Connector-Liste. Dabei entsteht das Problem, dass vorher aktivierte Commands erneut aktiviert werden, wenn sie die tiefer liegenden FUs durchlaufen. So wird z.B. der Command "Scheduler" im Sender aktiviert, bevor ein Compound übermittelt wird. In einem RN würde dieser Command nun erneut aktiviert bevor das Compound weitergeleitet wird, was nicht zulässig ist. Damit ein Compound erneut die FU-Kette durchlaufen kann, wird eine Kopie des Compounds erstellt, wobei der CommandPool, welcher alle aktivierten Commands enthält, nur teilweise übernommen wird. Die Commands aller FUs, die unterhalb des CompoundRelaySwitches liegen, werden dabei nicht wieder aktiviert. Dafür speichert der CommandPool die Reihenfolge, in der die Commands aktiviert wurden. Beim Kopieren werden dann alle Commands aktiviert bis in der Kette der Command der FU erreicht ist, von welcher der Kopiervorgang ausgeht. In diesem Fall ist das der CompoundRelaySwitch. Damit der CompoundRelaySwitch-Command in der Kette der aktivierten Commands auftaucht, muss dieser auch in den eNBs und UEs aktiviert werden. Aus diesem Grund ist die CompoundRelaySwitch-FU auch in eNB und UE vorhanden (Abbildung 5.1). Abschließend aktiviert der CompoundRelaySwitch noch seinen eigenen Command in der neuen Kopie des Compounds.

Um die Compounds entsprechend ihrer Senderichtung an die richtige FU zurück leiten zu können, wird ein neuer Filter, *FilterDirection*, eingesetzt. Dieser wird mit der Richtung konfiguriert in der Pakete durchgelassen werden sollen, also "ul" oder "dl". Der konfigurierte Wert wird mit der Senderichtung verglichen, welche in den DCI des Compounds gespeichert ist. Die DCI werden von dem Command der Scheduler bereitgestellt. Da die Scheduler unterhalb des CompoundRelaySwitches liegen, wird dieser Command bei dem Kopiervorgang des Command Pools nicht erneut aktiviert. Damit der Filter trotzdem die Senderichtung kennt, wird diese vorher lokal in der CompoundRelaySwitch-FU gespeichert. Listing 5.3 zeigt die Funktion des Filters, welche darüber entscheidet welche Compounds durchgelassen werden. Für einen RN werden zwei Instanzen des Filters erstellt. Einer wird mit dem Scheduler auf der UE-Seite des RN verbunden und leitet Compounds weiter, wenn dieses in Uplink-Richtung fließt. Der zweite Filter wird mit dem Scheduler auf der eNB-Seite verbunden und lässt alle Pakete in Downlink-Richtung durch.

## 5 Implementierung

Listing 5.3: Die Funktion entscheidet welche Compounds weitergeleitet werden.

```
1 bool
2 FilterDirection::filter(const wns::ldk::CompoundPtr& compound) const
3 {
4     if(compoundSwitch_ ->getDirection() == friends_.passInDirection)
5         return true;
6
7     return false;
8 }
```

Listing 5.4: Aus der Python-Konfiguration der Layer2-Component:

```
1 switch = dll.CompoundSwitch.CompoundRelaySwitch()
2 switch.setFUN(self.fun)
3
4 ulFilter = dll.CompoundSwitch.FilterDirection("ulFilter", "ul")
5 switch.connectSendDataFU(ueScheduler, ulFilter)
6 dlFilter = dll.CompoundSwitch.FilterDirection("dlFilter", "dl")
7 switch.connectSendDataFU(nbDLScheduler, dlFilter)
8
9 self.fun.add(switch)
```

### 5.3 LinkManager

Wie oben beschrieben, ist der LinkManager unter anderem dafür zuständig, UEs zu eNBs zuzuordnen. Dafür verwaltet er eine Liste von Links. Jeder Link stellt eine mögliche Verbindung dar, über welche Pakete übertragen oder Interferenzen berechnet werden können. Bei der Initialisierung erstellt der LinkManager normalerweise Links von jedem UE zu jedem eNB. Um die direkten Zuordnungen zu bestimmen, werden für jedes UE alle Links betrachtet und derjenige mit der höchsten spektralen Effizienz ausgewählt. Fortan ist das UE direkt mit dem entsprechenden eNB verbunden. Dem EPCgw wird dies mitgeteilt, so dass alle Datenpakete für ein UE an den entsprechenden eNB gesendet werden. Durch RNs entsteht die weitere Anforderung auch indirekt verbundene UEs zu erkennen und entsprechend an die eNB-StationPhys zu melden. Dafür wird der LinkManager um eine weitere Datenstruktur und eine Funktion erweitert, welche die indirekten Verbindungen findet und speichert.

### 5.3.1 Links innerhalb von Relaisstationen

Bei der Initialisierung des LinkManager werden alle Links zwischen allen StationPhys betrachtet. Das heißt auch solche zwischen der UE-StationPhy und der eNB-StationPhy desselben RN. Als Folge, würde die UE-StationPhy eines RN mit großer Wahrscheinlichkeit der eNB-StationPhy desselben RNs zugewiesen und den RN somit zwecklos machen. Da diese Links auch bei der Berechnung der Interferenzen keine Bedeutung haben und sogar unerwünscht sind, werden sie gar nicht erst erstellt. Dafür werden wie zuvor alle Links generiert. Anschließend werden aber nur die Links hinzugefügt für welche die Start- und die Ziel-Position nicht gleich ist (5.5).

Listing 5.5: Aus dem Konstruktor des LinkManager:

```
1 // Beziehe Links von allen UEs zu allen eNBs ein:
2 // Erstelle einen Link wenn die Position der beiden StationPhy
3 // NICHT gleich ist:
4 if( msPosition.getX() != bsPosition.getX() ||
5     msPosition.getY() != bsPosition.getY() )
6 {
7     wns::Position wrappedPosition;
8     wrappedPosition = getWrapAroundPosition(bsPosition,
9                                             msPosition);
10
11     linktype::LinkTypes linktype;
12     linktype = linkClassifier->classifyLink(*bsIter, **msIter,
13                                           wrappedPosition);
14
15     Link* link = new Link(*bsIter, *msIter, linktype.scenario,
16                          linktype.propagation,
17                          linktype.outdoorPropagation,
18                          linktype.userLocation);
19     links.push_back(link);
20
21     link->wrappedMSposition = wrappedPosition;
22     cntLinks++;
23 }
```

### 5.3.2 Links zwischen Relaisstationen

Im Unterschied zu den Links zwischen den StationPhy *eines* RN, werden die Links zwischen den StationPhy *verschiedener* RN für die Berechnung der Interferenzen benötigt. Sie

## 5 Implementierung

müssen daher vom LinkManager allgemein berücksichtigt werden. Da Verkettungen von RNs nicht vorkommen sollen, müssen sie ausgelassen werden, wenn der LinkManager die Zuordnung von UE-StationPhy zu eNB-StationPhy bestimmt. Wenn der LinkManager den besten Link für ein UE-StationPhy ermittelt, werden jetzt nurnoch diejenigen Links betrachtet, für die mindestens eine der beiden Nodes *nicht* in einem RN ist (Listing 5.6).

Listing 5.6: Aus der Funktion `determineServingLinks()` :

```
1 for (LinkMap::const_iterator linkIter = linksForThisMS.begin();
2     linkIter != linksForThisMS.end();
3     linkIter++)
4 {
5     // Wenn eine Start oder Ziel des Links nicht in einem RN sind:
6     if(!linkIter->second->getBS()->isInRelayNode() ||
7         !linkIter->second->getMS()->isInRelayNode())
8     {
9         sortedLinks.insert(std::make_pair<wns::Power, Link*>(
10             (linkIter->second->getRSRP(),
11             linkIter->second));
12     }
13 }
```

### 5.3.3 Indirekt verbundene Mobilstationen

In der Datenstruktur *indirectlyServedLinks* werden die zu RNs verbundenen UEs gespeichert. Sie ist wie folgt definiert:

Listing 5.7: Definition der Datenstruktur *indirectlyServedLinks*

```
1 typedef std::map<StationPhy*,
2                 StationPhy*,
3                 StationPhyPtrCompare> StationMap;
4 std::map<StationPhy*,
5         StationMap,
6         StationPhyPtrCompare> indirectlyServedLinks;
```

Die Datenstruktur ist eine Map, welche jeder StationPhy in der Map genau eine StationMap zuordnet. Die StationMap wiederum ordnet jeder StationPhy in der StationMap eine andere StationPhy zu. In der ersten Map wird einer eNB-StationPhy eine StationMap

## 5 Implementierung

zugeordnet, welche alle indirekt verbundenen UEs beinhaltet. In der zweiten Map wird zu jeder UE der direkt mit der eNB-StationPhy verbundene RN gespeichert, über welchen dann direkt oder wiederum indirekt diese UE erreichbar ist. Fragt eine StationPhy also nach ihren indirekt verbundenen UEs, bekommt der LinkManager von der Datenstruktur eine StationMap. Diese enthält die gesuchten UEs und die diesen zugeordneten RNs. Der LinkManager gibt den StationPhy diese StationMap zurück.

Der Algorithmus, welcher die UEs bestimmt, ist in der Funktion `determineServingLinks()` implementiert. Hier werden die Zuordnungen aller UEs aktualisiert. Dementsprechend müssen anschließend die indirekten Zuordnungen angepasst werden. In dem Listing 1 ist der Pseudocode des Algorithmus zu sehen. Für eine beliebige eNB-StationPhy werden zuerst alle direkten Verbindungen nach RNs durchsucht und diese zu einer Liste hinzugefügt. Anschließend werden in einer Schleife die Verbindungen der RNs in dieser Liste nach UEs durchsucht. Diese werden dann mit ihrem direkten Nachfolger in die oben beschriebene Datenstruktur eingefügt. Wird nicht ein UE sondern noch ein RN gefunden, wird dieser hinten an die Liste der RNs angefügt. Der gesamte Programmcode ist in Appendix A zu finden.

### Algorithm 1: Pseudocode:

For some eNB or RN baseStation:

```
for all nodes associated with baseStation do
  if a node is a RN then
    add baseStation to the parents of the node
    add the node to the list of RNs that need to be checked
  end if
end for

for all relays in the list of RNs that need to be checked do
  for all served nodes of a relay do
    if a served node is not a relay then
      find the direct successor for the served node on the path
      add the served node to the list of indirectly served nodes fo baseStation
    else
      add the node to the list of RNs that need to be checked
      add the relay as the parent of that node
    end if
  end for
end for
```

---



## 6 Evaluation

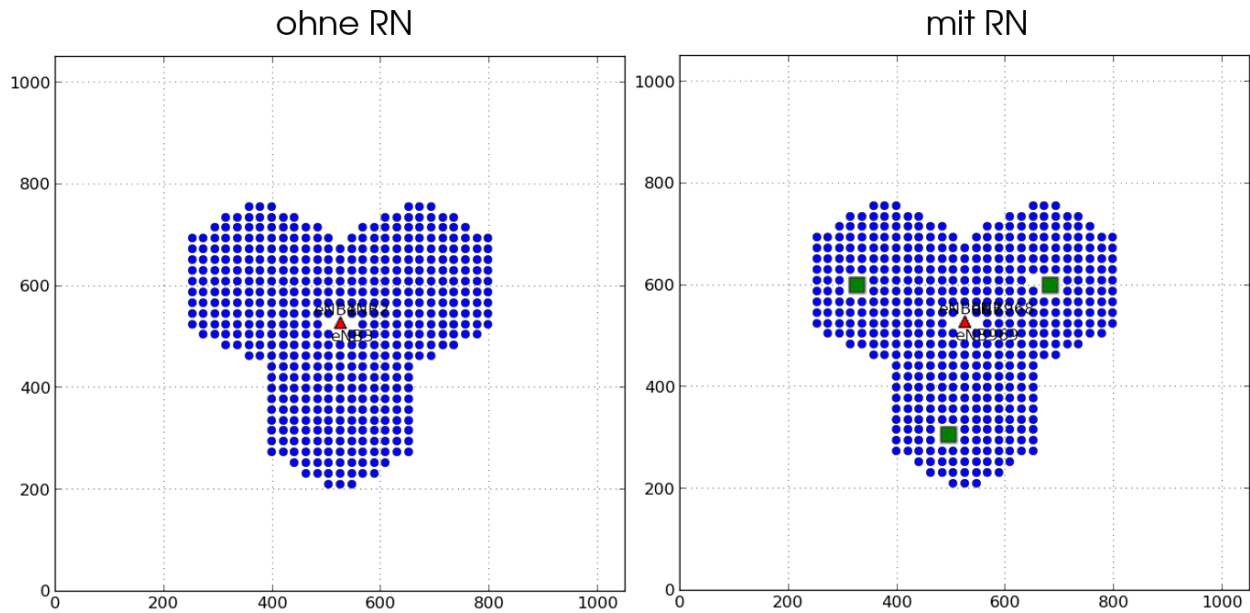
Da der zeitliche Rahmen dieser Arbeit begrenzt ist, wird die Implementierung nur anhand des 3GPP Case 1 Szenarios (Urban Macro) [3GPP, 2010] evaluiert. Das ist vor allem damit zu begründen, dass es auch in anderen Arbeiten zu dem Thema RNs verwendet wird, wodurch die Ergebnisse vergleichbar sind. Da die Simulation sehr viele Ressourcen braucht, wird die Simulation auf die Downlink-Richtung beschränkt. Diese Beschränkung ist angemessen, da oft ein FDD-Schema mit zusätzlichem Frequenzband für die zweite Senderichtung angenommen wird. Die Uplink- und Downlink-Richtung arbeiten dann weitgehend unabhängig ohne sich zu beeinflussen.

Das 3GPP Case 1 Szenario nähert die Verhältnisse in einer Vorstadt an. Es wird dabei angenommen, dass die UEs in Automobilen sind. Das Test-Szenario besteht aus einem zentralen eNB mit drei Sektoren. Die UEs werden gleichmäßig mit einem Abstand von 20m platziert. In jeder Zelle wird ein RN platziert, welcher

$$\begin{aligned}D_{RN} &= \frac{ISD}{2} * d \\ ISD &= 500m \\ d &= 0,8\end{aligned}$$

von dem zentralen eNB entfernt ist (siehe Abbildung 6.1). Die *Inter Site Distance* (ISD) ist der Abstand zwischen zwei benachbarten eNBs. Der Faktor  $d$  bestimmt wie weit entfernt die RNs von dem eNB liegen. Für den Test wurde  $d = 0,8$  gewählt. Die RNs liegen somit 200 m von dem eNB entfernt. Das ist relativ weit entfernt von dem eNB, wo das SINR niedriger ist als nah an dem eNB. Dadurch werden die UEs von den RNs versorgt, die ansonsten wenig Durchsatz erreicht hätten. Würden die RN viel weiter weg platziert, könnten die Interferenzen von benachbarten eNB die RNs stören.

## 6 Evaluation



**Abbildung 6.1:** Das Urban Macro Szenario mit einem eNB (rotes Dreieck) und drei Zellen. Die blauen Kreise stellen Testpunkte dar. In jeder Zelle wird ein RN platziert (grüne Rechtecke).

Eine optimale Platzierung der RNs wurde für die Tests nicht ermittelt. Es ist jedoch bekannt, dass eine Platzierung mit Blick auf optimalen Durchsatz oder maximale Abdeckung erhebliche Steigerungen ermöglicht [Elgendy et al., 2012] und auch der Abstand zwischen eNB und RN kann einen großen Einfluss auf die Effektivität von Relaisstationen haben [Bulakci et al., 2010]. Da sich die RNs gegenüber dem eNB wie ein UE verhalten, verbinden sie sich automatisch dem eNB mit der besten Kanalqualität. Das kann Vorteile haben, wenn die RNs am Rand zwischen zwei Zellen platziert werden [Bulakci et al., 2010]. In dem einfachen Szenario und auf Grund des sicheren Abstands zum Zell-Rand, spielt dies jedoch keine Rolle. Die RNs verbinden sich mit dem eNB der Zelle in welcher sie stehen.

Es sollte auch beachtet werden, dass die Platzierung der UEs nicht dazu geeignet ist reale Bedingungen abzubilden. Die große Anzahl verringert den Durchsatz pro Nutzer erheblich und erhöht die Interferenzen in den Funkzellen. Normalerweise finden sich Ansammlungen von Mobilgeräten, z.B. entlang von Straßenzügen. Dafür eignet sich die gleichmäßige Verteilung dazu, SINR-Werte an allen Stellen in den Zellen zu erhalten. Dadurch kann der Einfluss der RNs auf verschiedene Positionen bewertet werden.

## 6 Evaluation

Im nächsten Abschnitt, werden die wichtigsten Simulationsparameter erläutert. Anschließend werden die Simulationsergebnisse für das Test-Szenario vorgestellt und Ergebnissen anderer Untersuchungen verglichen. Speziell, wird auf das SINR und die *Cell Spectral Efficiency* (CSE) eingegangen. Letztere erlaubt es bei bekannter Kanalbreite die absolute Performance und die spektrale Effizienz des Netzwerkes in einem Wert zusammenzufassen.

### 6.1 Simulationsparameter

Die Simulationsparameter sind nach den Vorgaben der 3GPP gesetzt [3GPP, 2010]. Für den eNB-RN Link wird eine gerichtete 2x2 Antennen-Konfiguration angenommen, d.h. zwei Empfänger- und zwei Sender-Antennen. Die Antenne zeigt Richtung dem eNB. Der Antennengewinn ist 7 dBi. Dieser gibt an, um wie viel die Antenne das Signal gegenüber einer isotropischen Antenne, welche in alle Richtungen gleich stark sendet, verstärkt. Der Gewinn durch die Antennengeometrie und Ausrichtung für diesen Link ist durch

$$A(\theta) = -\min \left\{ 12 \left( \frac{\theta}{\theta_{3dB}} \right)^2, A_m \right\}$$
$$\theta_{3dB} = 70^\circ$$
$$A_m = 20dB$$

gegeben. Diese Formel wird auch zur Berechnung der Absorptionsfläche der eNB-Antenne verwendet. Der Unterschied liegt in dem eingesetzten Winkel. Für die Simulation wurde eine Abwärtsneigung von  $1^\circ$  gewählt. Die Abwärtsneigung eNB Antenne ist  $12^\circ$ . Der geringe Winkel der RN Antenne erhöht die Interferenzen zugunsten einer verbesserten Verbindung zum eNB. Der RN-UE Link ist ebenfalls eine 2x2 Konfiguration. Der Antennengewinn ist mit 5 dBi etwas geringer und die Antenne ist nicht gerichtet, sondern omnidirektional. Die Absorptionsfläche ist unabhängig von dem Neigungswinkel. Also gilt entsprechend:

$$A(\theta) = 0$$

## 6 Evaluation

Alle weiteren Parameter für die RNs, eNBs und UEs sind in Tabelle 6.1 zusammengefasst. [Yuan, 2013, S. 37]

Neben den entsprechenden Einstellungen der Komponenten, ändert sich auch das Kanal-Modell für den eNB-RN- und den RN-UE-Link. Diese geben die Dämpfung des Signals für den Abstand zwischen zwei Knoten wieder. Nach abwägen der Vor- und Nachteile ist das angepasste Kanal-Modell für die RNs nicht implementiert worden. Der Aufwand für die Umsetzung und die Tests des Modells überwiegt die Vorteile. Das Modell verändert zwar die Ergebnisse hinsichtlich der Leistung des Netzwerkes, jedoch wird die grundlegende Funktionalität der hier vorgestellten Implementierung nicht verändert. Bei der Bewertung der Ergebnisse werden die Unterschiede der Kanal-Modelle berücksichtigt. Dafür wird im Folgenden das Kanal-Modell wie es für den eNB-UE-Link implementiert ist vorgestellt. Das selbe Modell wird hier auch für die beiden RN-Links verwendet. Die angepassten Modelle für die RN-Links und das 3GPP Case 1 Szenario ist in Appendix B zu finden [Yuan, 2013, S. 32].

Ein Kanal-Modell nimmt, mit einer vom Abstand abhängigen Wahrscheinlichkeit, entweder eine line-of-sight (LOS) oder non-line-of-sight (NLOS) Ausbreitung des Signals an. Diese Wahrscheinlichkeit hängt von dem Szenario, hier 3GPP Case 1, und von den Eigenschaften der verbundenen Knoten ab. Für den eNB-UE Links in der Urban Macro Umgebung gilt folgendes Modell:

Die Wahrscheinlichkeit, dass für die Übertragung der LOS-Fall angenommen wird ist durch

$$Prob_{LOS}(R) = \min\left(\frac{18}{R}; 1\right) * \left(1 - e^{-\frac{R}{63}}\right) + e^{-\frac{R}{63}}$$

gegeben.  $R$  ist der Abstand zwischen den eNB und UE in Metern. Die Dämpfungen  $PL_{LOS}$  und  $PL_{NLOS}$  werden in dB berechnet. Die LOS-Dämpfung errechnet sich wie folgt:

$$\begin{aligned} d'_{BP} &= 4 * h_{BS} * h_{UE} * \frac{f_c}{c} \\ h_{BS} &= 25m \\ h_{UE} &= 2m \\ f_c &= 2GHz \\ c &= 3 * 10^8 \frac{m}{s} \end{aligned}$$

## 6 Evaluation

Dabei sind  $h_{BS}$  und  $h_{UE}$  die Antennenhöhe der eNBs bzw. der UEs,  $f_c$  ist die mittlere Frequenz für die Übertragungen und  $c$  ist die Ausbreitungsgeschwindigkeit des Signals im freien Raum. Wenn für den Abstand  $R$  zwischen zwei Knoten  $R < d'_{BP}$  gilt, wird das Signal um

$$PL_{LOS}(R) = 22,0 * \log_{10} R + 20,0 * \log_{10} f_c + 28,0$$

gedämpft. Ist dagegen  $R \geq d'_{BP}$ , wird die Dämpfung durch

$$PL_{LOS}(R) = 40,0 * \log_{10} R + 7,8 - 18,0 * \log_{10} h'_{BS} - 18,0 * \log_{10} h'_{UE} + 2,0 * \log_{10} f_c$$

berechnet. Die NLOS-Dämpfung ist

$$\begin{aligned} PL_{NLOS}(R) = & 161,04 - 7,1 * \log_{10} W + 7,5 * \log_{10} h \\ & - \left( 24,37 - 3,7 * \left( \frac{h}{h_{BS}} \right)^2 \right) * \log_{10} h_{BS} \\ & + (43,42 - 3,1 * \log_{10} h_{BS}) * (\log_{10} R - 3,0) \\ & + 20,0 * \log_{10} f_c - \left( 3,2 * (\log_{10} 11,75 * h_{UE})^2 - 4,97 \right). \end{aligned}$$

$$W = 20m$$

$$h = 20m$$

$W$  ist die Breite der Straßen und  $h$  die Höhe der Gebäude (jeweils in Metern). [3GGP, 2010]

## 6.2 SINR

Das mittlere SINR ist Vergleich zu dem Szenario ohne RNs nur leicht, von 17,7 dB auf 17,0 dB gesunken. Die Schwankungen des SINR sind jedoch stark gestiegen. Ohne RNs liegt der Variationskoeffizient bei 0,35, mit bei 0,54. Ein Teil der UEs haben mit RNs besseren Empfang, während andere mehr Interferenzen empfangen. Der Großteil der UEs hat vergleichbaren Empfang (Abbildung 6.2).

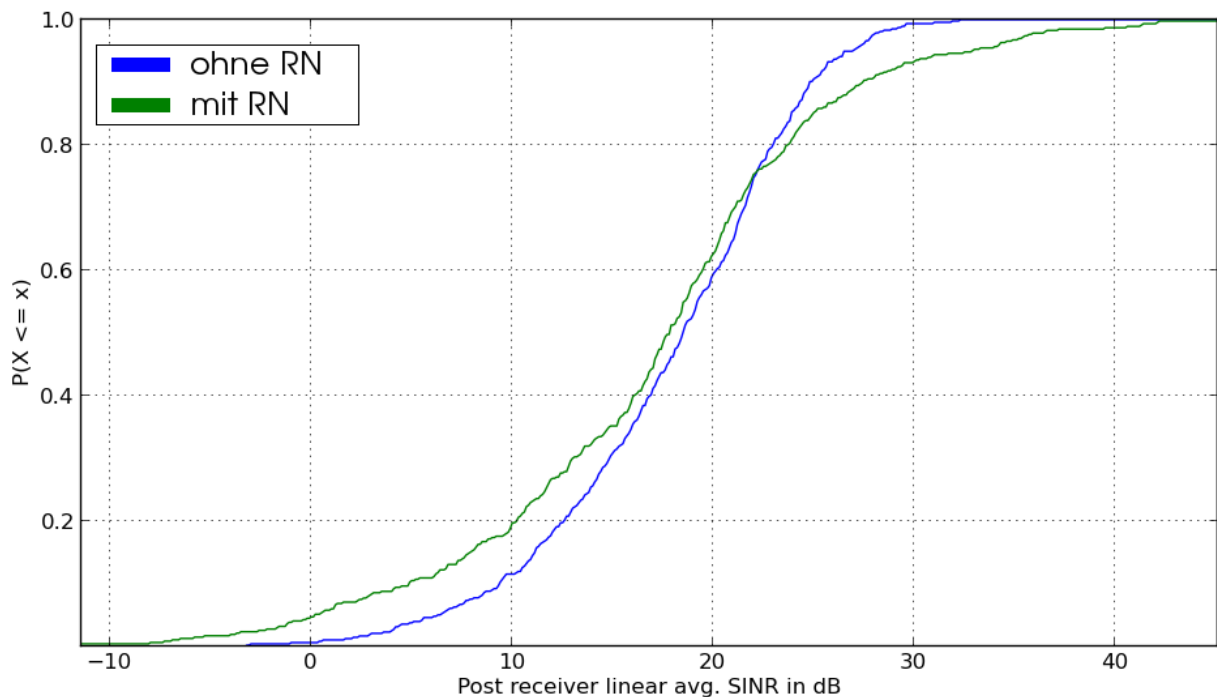
In den Extremen profitieren erwartungsgemäß die UEs in der direkten Umgebung der RNs von besseren Empfangsbedingungen (Abbildung 6.3). Auffällig ist außerdem, dass einzelne UEs stark verringerte SINR-Werte erfahren (dunkel Blau in Abbildung 6.3). Diese

## 6 Evaluation

**Tabelle 6.1:** Parameter für das Urban Macro Szenario.

Parameter	Wert
Trägerfrequenz	2 Ghz
Bandbreite	10 Mhz
eNB	
Antennenkonfiguration	Tx-2, Rx-2
Neigungswinkel	12 °
Sendeleistung	46 dBm
Antennenhöhe	25 m
Antennengewinn	14 dBi
Empfängerrauschen	5 dB
UE	
Antennenkonfiguration	Tx-1, Rx-2
Neigungswinkel	omnidirektional
Sendeleistung	24 dBm
Antennenhöhe	2,0 m
Antennengewinn	0 dBi
Empfängerrauschen	9 dB
RN	
Antennenhöhe	5 m
Empfängerrauschen	5 dB
eNB-RN Link	
Antennenkonfiguration	Tx-2, Rx-2
Neigungswinkel	1 °
Sendeleistung	30 dBm
Antennengewinn	7 dBi
RN-UE Link	
Antennenkonfiguration	Tx-2, Rx-2
Neigungswinkel	omnidirektional
Sendeleistung	30 dBm
Antennengewinn	5 dBi

## 6 Evaluation



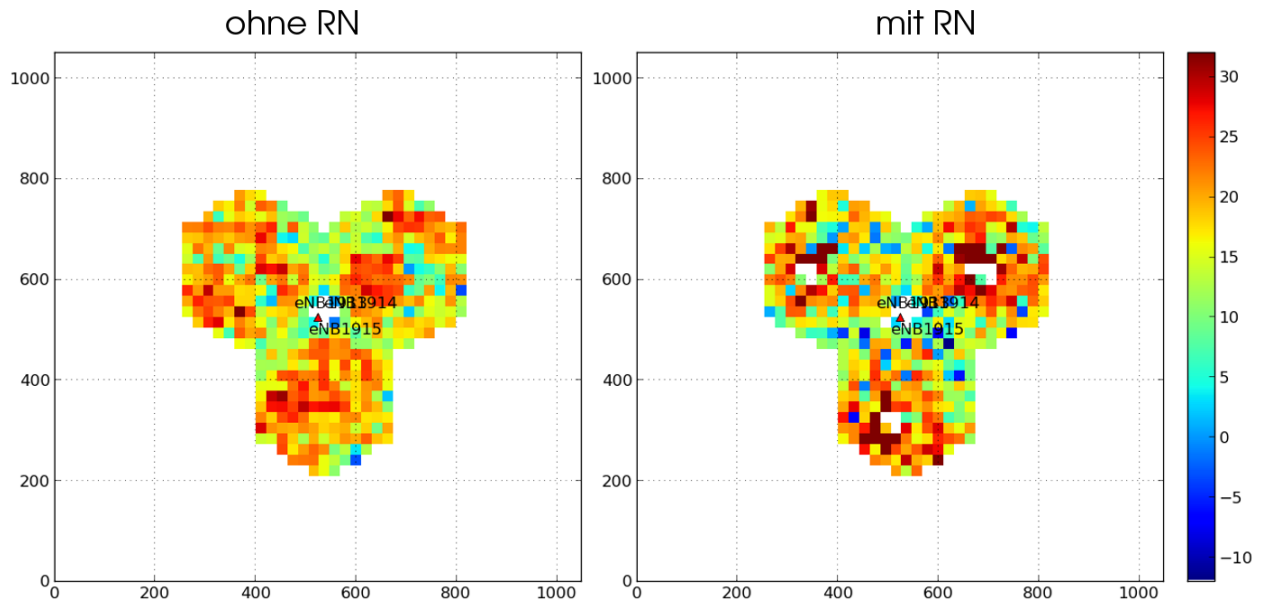
**Abbildung 6.2:** SINR Verteilungsfunktionen: In dem Szenario mit RN (grüne Linie) erfahren einige UEs verbesserte, andere verschlechterte SINR-Werte.

liegen hauptsächlich im Grenzgebiet zwischen zwei Zellen, wo das SINR auch ohne RN niedriger als in der Mitte der Zelle ist. Diese Mobilstationen profitieren nicht von den RNs, empfangen aber trotzdem mehr Interferenzen durch die Funkverbindung der RNs mit dem eNB.

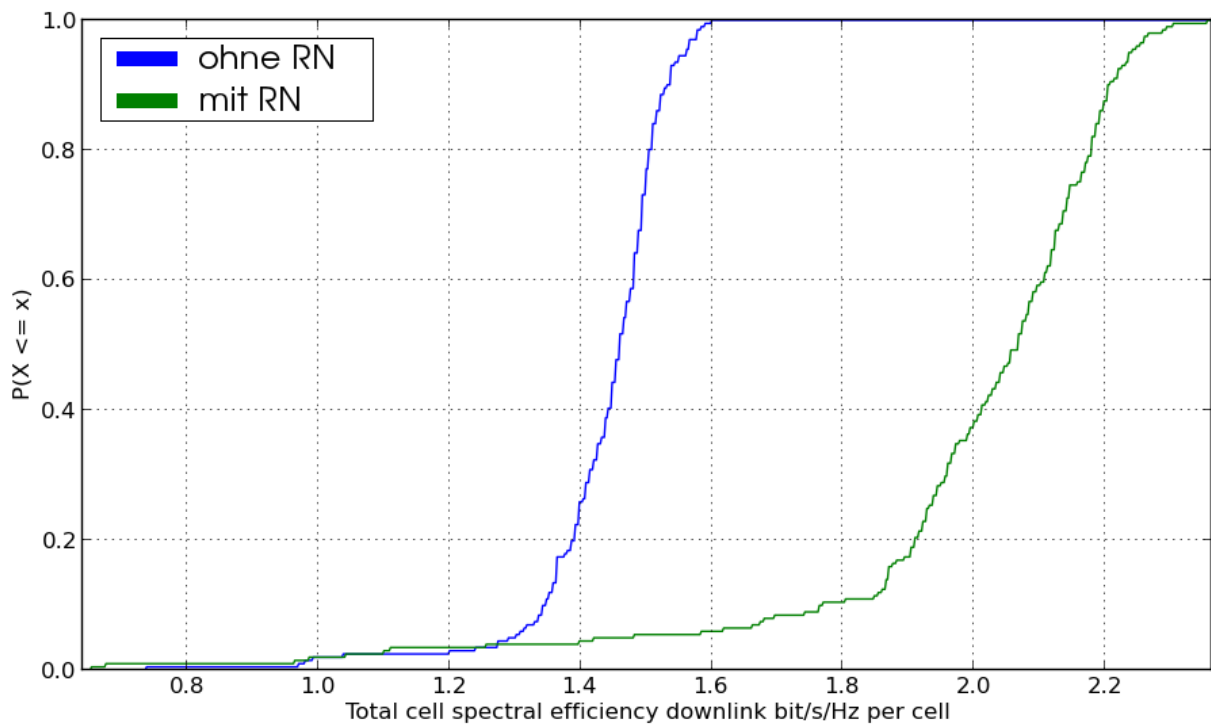
### 6.3 Durchsatz

Die CSE und der Durchsatz steigen durch die RNs erheblich. Für das vorgestellte Szenario steigt der mittlere Durchsatz pro Zelle von 1,44 auf 2,00 Bit/s/Hz (Abbildung 6.4). Ferner, ist der minimale Durchsatz pro Zelle nur unerheblich gesunken und nur zwei Prozent mehr UEs erreichen eine spektrale Effizienz von 0.01 Bit/s/Hz (Abbildung 6.5). Im Unterschied dazu, liegen 16,5% der UEs über einem Durchsatz von 0.02 Bit/s/Hz, einem Wert der ohne RNs an keiner Stelle überschritten wird. Der allgemein niedrige Durchsatz pro Nutzer ist durch die oben begründete Wahl vieler, homogen verteilter Test-Punkte zu erklären.

## 6 Evaluation



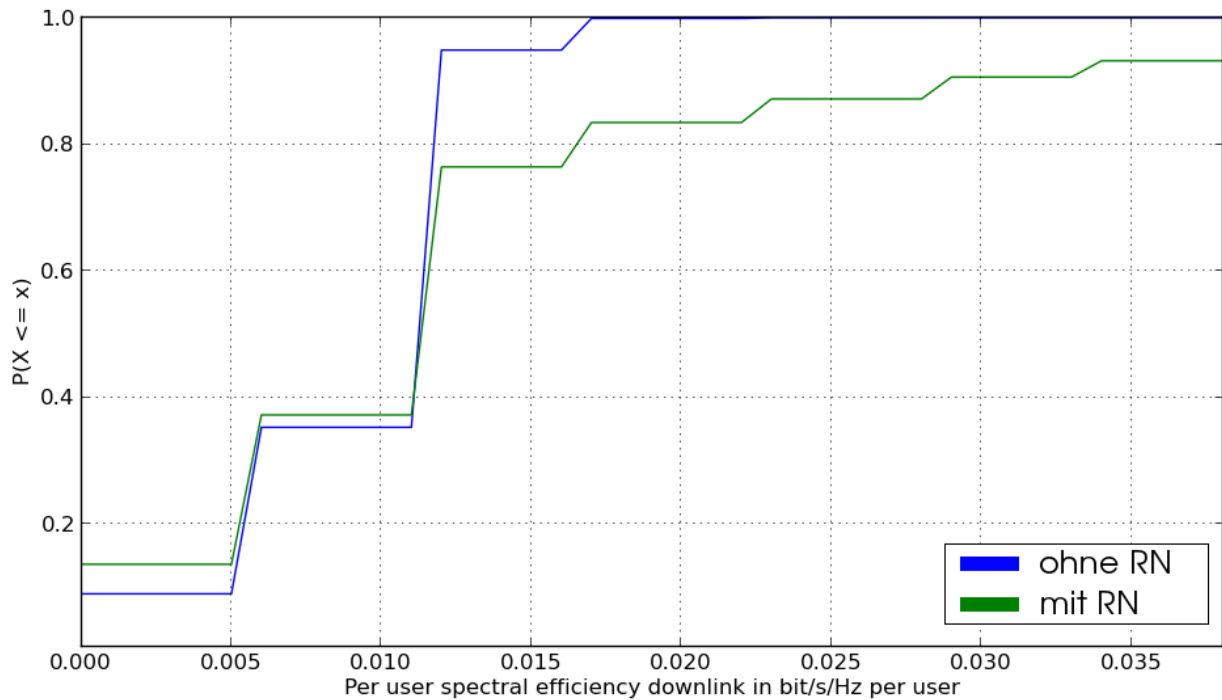
**Abbildung 6.3:** Durchschnittliches SINR: In der Umgebung der RNs werden höhere SINR-Werte erreicht (rechts) als im Szenario ohne RNs (links).



**Abbildung 6.4:** Aggregierter Durchsatz pro Zelle in Bit/s/Hz: Durch den Einsatz von RNs wird der mittlere Durchsatz um 38.9% gesteigert.



## 6 Evaluation



**Abbildung 6.5:** Durchsatz pro Mobilgerät: 16.5% der UEs erreichen einen Durchsatz über 0.02 Bit/s/Hz.

**Tabelle 6.2:** Durchsatzsteigerung durch Relaisstationen.

Quantil	ohne RN	mit RN	Änderung
10%	1,3461000	1,7682000	+31,36%
50%	1.4591250	2.0666000	+41,63%
90%	1.5318000	2.2039600	+43,88%

Andere Simulationen von Relaisstationen sind zu vergleichbaren Ergebnissen gekommen. In [Schoenen et al., 2010] wird eine CSE von 1,804 Bit/s/Hz für drei, bzw. 1,920 Bit/s/Hz für sechs RNs in einem Urban Macro Szenario mit drei Sektoren pro eNB erreicht. Ohne RN sind es nur 1,254 Bit/s/Hz. In [Bou Saleh et al., 2009] wird eine Steigerung von 48% (10% Quantil) bzw. 19% (50% Quantil) berichtet. Dabei sind 5 RN an am Rand der Zelle platziert worden. Auffällig ist, dass hier weniger Nutzer von den RN profitieren, dafür jedoch relativ stark. Zum Vergleich sind die Quantile der Verteilung aus Abbildung 6.4 in Tabelle 6.2 aufgeführt.

In den Ergebnissen ist der Einfluss des nicht angepassten Kanal-Modells zu sehen. Aufgrund der wesentlich niedrigeren Dämpfung des Signals werden weit mehr UEs in einem

## 6 Evaluation

relativ großen Umkreis um die RNs zu diesen zugeordnet (Abbildung 6.6). Diese UEs profitieren dann von der besseren Signalqualität. Es ist jedoch klar erkennbar, dass in größerem Abstand der RNs vereinzelt UEs zugeordnet werden. Das lässt darauf schließen, dass der Unterschied in der Signalqualität zwischen eNB und RN nicht sehr groß ist. Unterstützt wird diese Beobachtung durch die gestiegene Standardabweichung des SINR. Durch die relativ hohe Signalstärke der RNs werden einige UEs erheblich gestört. In diesem Fall wäre die geringere Signalausbreitung in dem angepassten Modell vorteilhaft. UEs nah an den RNs würden stärker profitieren und UEs weiter entfernt hätten weniger Störsignale durch die RNs. Insgesamt gesehen, liegen die Ergebnisse jedoch in dem Rahmen der Resultate von [Bou Saleh et al., 2009] und [Schoenen et al., 2010].

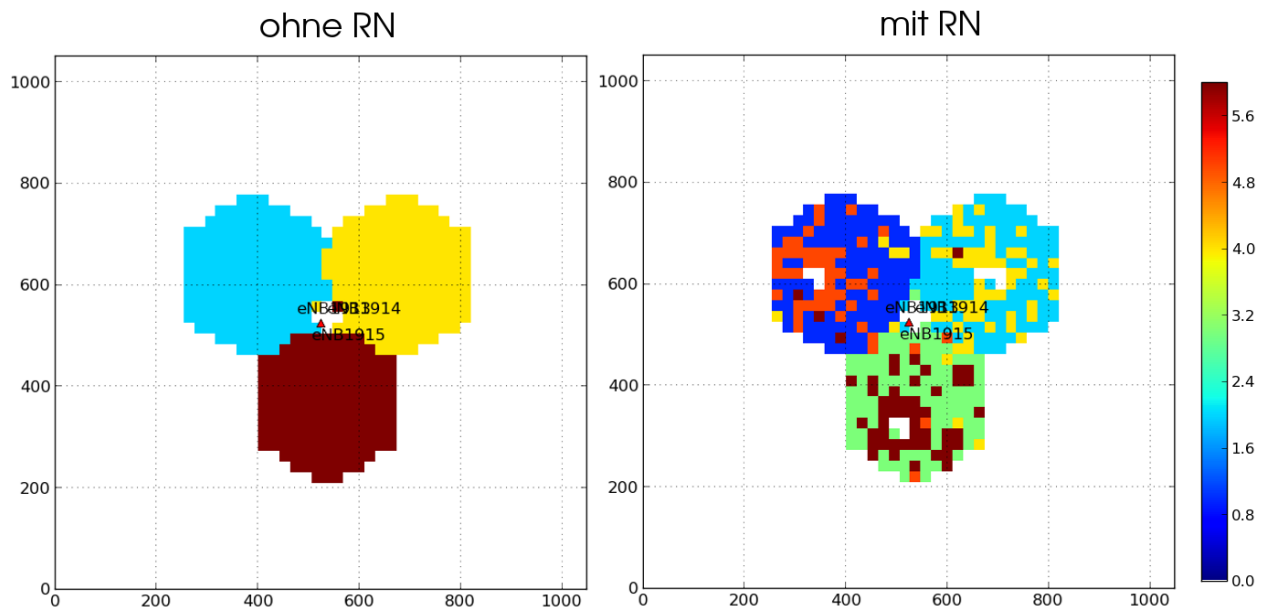


Abbildung 6.6: Zuordnung der UEs zu den eNBs bzw. RNs.

## 7 Fazit

In Rahmen dieser Arbeit ist eine Implementierung von RN in dem IMT-A Modul IMTAphy der Simulationsplattform openWNS entstanden. Dabei wird auf das Node-Component-Modell aus openWNS zurückgegriffen, wodurch viele Teile aus dem bestehenden Quellcode wiederverwendet werden können. Neu ist insbesondere eine Funktion in der Verwaltung der Verbindungen zwischen den Knoten, welche der physikalischen Schicht in eNB und RN auch indirekt verbundene UEs zurückgibt. So können diese Stationen eintreffende Pakete an das entsprechende Ziel weiterleiten, auch wenn dieses nicht von ihnen direkt bedient wird. Außerdem wurde eine neue Functional Unit implementiert. Diese sorgt dafür, dass erfolgreich dekodierte Pakete in der RN in die andere Richtung weitergeleitet werden. Damit dies möglich ist, wird das Paket kopiert und die darin gespeicherten Befehle nur teilweise wieder aktiviert. Abschließend wurde gezeigt, dass die mit der Implementierung ermittelten Ergebnisse, mit denen anderer Veröffentlichungen vergleichbar sind, auch wenn das Kanal-Modell für die Verbindungen zwischen eNB und RN, sowie zwischen RN und UE nicht angepasst wurde.

Die nächsten Schritte zur Erweiterung der Implementation sind also, die angepassten Kanal-Modelle für das getestete und andere Szenarien umzusetzen. Aufbauend auf dieser Arbeit, kann dann das Verhalten von Netzwerken mit mehreren, verketteten RN getestet werden. Die Verkettungen sollte die Implementation theoretisch unterstützen, was hier aber nicht überprüft wurde. Bei diesen Netzwerken ist vor allem die Zuordnung von RNs zu anderen RNs interessant. In der aktuellen Umsetzung, würden sich die RNs naiv die Station mit dem besten Signal suchen, ohne zu berücksichtigen, ob das ein eNB oder RN ist. Das kann negative Auswirkungen auf die Leistung haben, da die Verbindung zwischen eNB und RN bzw. zwischen RN und RN schnell zu dem limitierenden Faktor wird, wenn zu viel Datenverkehr über ein RN fließt. Daher ist eine intelligente Zuordnung denkbar bei der die maximale Kapazität, auf dem Weg über alle RNs bis zur eNB, berücksichtigt wird. Eine andere Erweiterung sind die CoMP-Übertragungen, die auch

## *7 Fazit*

in Kapitel 3 angesprochen wurden. Dabei koordinieren die RNs und eNB ihre Sendestärke und Übertragungen derart, dass die Signalqualität für die UEs steigt. Es außerdem möglich, die RNs mit einer automatischen Konfiguration zu versehen, so dass diese ihre Sendestärke oder den Neigungswinkel selbstständig einstellen. Schließlich ergibt sich die Möglichkeit, Algorithmen zu verwenden, welche die optimale Anzahl und Position von RNs zu ermitteln. Dabei sollte besonders Wert auf eine intelligente Umsetzung gelegt werden, da die Simulation, je nach Anzahl der eNBs, RNs und UEs, viel Zeit in Anspruch nehmen kann.

# A Die determineIndirectlyServedLinks Funktion

Listing A.1: Aus determineServingLinks():

```
1 typedef std::vector<StationPhy*> StationVector;
2 StationVector RelaysToCheck;
3 // Speichert RN, die mit der baseStation (in-)direkt verbunden sind.
4
5 StationMap parentStation;
6 // Speichert den direkten Nachfolger einer UE aus Sicht der
7 // baseStation.
8
9 StationPhy* baseStation = (*bsIter);
10
11 // Initialisiere RelaysToCheck mit den direkt verbundenen RNs:
12 LinkMap servedByBS = servedLinks[baseStation];
13 for(LinkMap::const_iterator iter = servedByBS.begin();
14      iter != servedByBS.end();
15      iter++)
16 {
17     StationPhy* servedStation = iter->first;
18     if(servedStation->isInRelayNode())
19     {
20         StationPhy* relayStation =
21             servedStation->getNode()
22                 ->getService<StationPhy*>("u1TX");
23
24         parentStation[relayStation] = baseStation;
25         RelaysToCheck.push_back( relayStation );
26     }
27 }
28
```

## A Die *determineIndirectlyServedLinks* Funktion

```
29 // Unter allen RN in RelaysToCheck:
30 for(StationVector::const_iterator iter = RelaysToCheck.begin();
31      iter != RelaysToCheck.end();
32      iter++)
33 {
34     LinkMap* served = &servedLinks[(*iter)];
35     for(LinkMap::const_iterator servedIter = served->begin();
36         servedIter != served->end();
37         servedIter++)
38     {
39         StationPhy* servedStation = servedIter->first;
40
41         // Wenn die StationPhy kein RN ist,
42         if(!servedStation->isInRelayNode())
43         {
44             // ist sie ein indirekt bedientes UE.
45             StationPhy* parent = parentStation[(*iter)];
46             StationPhy* current = (*iter);
47             while(parent != baseStation)
48             {
49                 current = parent;
50                 parent = parentStation[current];
51             }
52             indirectlyServedLinks[baseStation][servedStation]
53                                     = current;
54         } else {
55             // eine indirekt verbundene RN, welche auch noch zu
56             // untersuchen ist.
57             RelaysToCheck.push_back(servedStation);
58             parentStation[servedStation] = (*iter);
59         }
60     }
61 }
```

---

## B Angepasstes Kanal-Modell für das Test-Szenario

Im Folgenden wird kurz das Kanal-Modell für die beiden Verbindungen der Relaisstationen für das Urban Urban Macro Case 1 Szenario beschrieben [Yuan, 2013, S. 32]. Für den Pfadverlust zwischen eNB und RN in Abhängigkeit von der Distanz  $R$  gilt

$$PL_{LOS}(R) = 100,7 + 23,5 \log_{10} R$$

$$PL_{NLOS}(R) = 125,2 + 36,3 \log_{10} R.$$

Die Wahrscheinlichkeit dass  $PL_{LOS}$  eintritt ist

$$Prob(R) = \min\left(\frac{0,018}{R}; 1\right) * \left(1 - e^{-\frac{R}{0,072}}\right) + e^{-\frac{R}{0,072}}.$$

Der Pfadverlust zwischen RN und UE ist durch

$$PL_{LOS}(R) = 103,8 + 20,9 \log_{10} R$$

$$PL_{NLOS}(R) = 145,4 + 37,5 \log_{10} R$$

gegeben und die Wahrscheinlichkeit für den ersten Fall beträgt

$$Prob(R) = 0,5 - \min\left(0,5; 5 * e^{-\frac{0,156}{R}}\right) + \min\left(0,5; 3 * e^{-\frac{R}{0,03}}\right).$$

# Literaturverzeichnis

- 3GPP. Further Advancements for E-UTRA Physical Layer Aspects. Technical Report 3GPP TR 36.814 V9.0.0, March 2010.
- 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN). Technical Report 3GPP TS 36.300 V8.11.0, December 2009.
- 3GPP. Evolved universal terrestrial radio access (e-UTRA); FDD repeater radio transmission and reception. Technical Report 3GPP TS 36.106 V11.0.1 Release 11, October 2012.
- Ian F Akyildiz, David M Gutierrez-Estevez, and Elias Chavarria Reyes. The evolution to 4G cellular systems: LTE-Advanced. *Physical Communication*, 3(4):217–244, 2010.
- Tara Ali-Yahiya. *Understanding LTE and its Performance*. Springer, January 2011.
- David Astély, Erik Dahlman, Anders Furuskar, Ylva Jading, Magnus Lindstrom, and Stefan Parkvall. LTE: the evolution of mobile broadband. *Communications Magazine, IEEE*, 47(4):44–51, 2009.
- A Bou Saleh, Simone Redana, Bernhard Raaf, and Jyri Hamalainen. Comparison of relay and pico eNB deployments in LTE-advanced. In *IEEE 70th Vehicular Technology Conference (VTC 2009-Fall)*, September 2009.
- O Bulakci, Simone Redana, Bernhard Raaf, and J Hamalainen. Performance Enhancement in LTE-Advanced Relay Networks via Relay Site Planning. In *IEEE 71st Vehicular Technology Conference (VTC 2010-Spring)*, May 2010.
- D. Bultmann, M. Muhleisen, K. Klagges, and M. Schinnenburg. OpenWNS - open wireless network simulator. In *European Wireless Conference*, pages 205 –210, May 2009.



## Literaturverzeichnis

- Cisco. Cisco Visual Networking Index: Global Mobile data Traffic Forecast Update, 2012–2017. White Paper, February 2013.
- Thomas Delbecke, Ranjith Kumar Duraisamy, Houda Rizqi, and Xing Ge Xu. LTE-Advanced – Release 10 relaying. *Cell*, 2(3.7):1–2.
- O.A. Elgendy, M.H. Ismail, and K. Elsayed. On the relay placement problem in a multi-cell LTE-Advanced system with co-channel interference. In *IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 300–307, October 2012.
- Ericsson. HSPA evolution – beyond 3GPP release 10. White Paper, July 2011. Von <http://www.ericsson.com/res/docs/whitepapers/WP-HSPA-Evolution.pdf>.
- Harri Holma, Antti Toskala, Karri Ranta-Aho, and Juho Pirskanen. High-Speed Packet Access Evolution in 3GPP Release 7. *IEEE Communications Magazine*, 45(12):29–35, 2007.
- Mikio Iwamura, Hideaki Takahashi, and Satoshi Nagata. Relay technology in LTE-Advanced. *NTT DoCoMo Technical Journal*, 12(2):29–36, 2010.
- Ekkehard Lang, Simone Redana, and Bernhard Raaf. Business impact of relay deployment for coverage extension in 3GPP LTE-Advanced. In *LTE Evolution Workshop, IEEE ICC*, pages 1–5, 2009.
- ITU-R Rep. M.2134. Requirements related to technical performance for IMT-Advanced radio interface(s), 2008.
- Satoshi Nagata, Yuan Yan, Xinying Gao, Anxin Li, Hidetoshi Kayama, Tetsushi Abe, and Takehiro Nakamura. Investigation on system performance of L1/L3 relays in LTE-Advanced downlink. In *IEEE 73rd Vehicular Technology Conference (VTC 2011-Spring)*, pages 1–5, 2011.
- Young-Han Nam, Lingjia Liu, Yan Wang, Charlie Zhang, Joonyoung Cho, and Jin-Kyu Han. Cooperative communication technologies for LTE-advanced. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 5610–5613, 2010.
- Ralf Pabst, Bernhard H Walke, Daniel C Schultz, Patrick Herhold, Halim Yanikomeroglu, Sayandev Mukherjee, Harish Viswanathan, Matthias Lott, Wolfgang Zirwas, Mischa Dohler, et al. Relay-based deployment concepts for wireless and mobile broadband radio. *IEEE Communications Magazine*, 42(9):80–89, 2004.

## Literaturverzeichnis

- Mohammad N Patwary, Predrag B Rapajic, and Ian Oppermann. Capacity and coverage increase with repeaters in umts urban cellular mobile communication environment. *IEEE Transactions on Communications*, 53(10):1620–1624, 2005.
- Adnan Quaium. *Evaluation of Relay-Enhanced LTE-Advanced Networks*. PhD thesis, Delft University of Technology, April 2011.
- Nomor Research. Nomor Research completes calibration of 3GPP LTE-A relay system-level simulation environment. Press Release, June 2012.
- Martin Sauter. *Grundkurs Mobile Kommunikationssysteme: UMTS, HSDPA und LTE, GSM, GPRS und Wireless LAN*. Vieweg+Teubner Verlag, 4., überarb. und erw. Aufl. 2011 edition, October 2010.
- Schinnenburg, F. Debus, A. Otyakmaz, L. Berlemann, and R. Pabst. A Framework for Reconfigurable Functions of a Multi-Mode Protocol Layer. In *SDR Forum*, page 6, November 2005.
- M. Schinnenburg, R. Pabst, K. Klagges, and B. Walke. A software architecture for modular implementation of adaptive protocol stacks. In *MMBnet Workshop*, pages 94–103, Hamburg, Germany, September 2007.
- Rainer Schoenen, Carlos Teijeiro, and D Bultmann. System level performance evaluation of LTE with MIMO and relays in reuse-1 IMT-Advancedscenarios. In *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, pages 1–5, 2010.
- UMTS Forum. Mobile Broadband Evolution: the roadmap from HSPA to LTE. White Paper, February 2009.
- Jeanette Wannstrom. LTE-Advanced, May 2012. Für das 3GPP. Verfügbar unter [http://www.3gpp.org/IMG/pdf/lte\\_advanced\\_v2.pdf](http://www.3gpp.org/IMG/pdf/lte_advanced_v2.pdf).
- Klaus Wehrle and James Gross. *Modeling and tools for network simulation*. Springer, 2010.
- Y. Yuan. *LTE-Advanced Relay Technology and Standardization*. Signals and Communication Technology. Springer, 2013.
- Hubert Zimmermann. OSI reference model—the ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4):425–432, 1980.