ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG
INSTITUT FÜR INFORMATIK

Lehrstuhl für Kommunikationssysteme
Prof. Dr. Gerhard Schneider
Betreuer: Dr. Dirk von Suchodoletz

Master-Arbeit zum Thema:

Open Source GSM BTS Setup und Analyse für Demo-Zwecke

Holger Bertsch
Holger.Bertsch@gmx.de

Diese Arbeit wurde eingereicht als Teilleistung zur Erlangung des Master-Grades an der Technischen Fakultät Albert-Ludwig-Universität Freiburg, 2009

Erklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Diese Arbeit entstand parallel zur Masterarbeit „Open Source IMSI-Catcher“ von Dennis Wehrle [30]. Aufgrund der Komplexität und der aufwendigen Einarbeitung in die Thematik war eine enge Zusammenarbeit unerlässlich. Die Beschaffung der Hardware, notwendige Modifikationen von Soft- und Hardware und die initiale Inbetriebnahme sowie die Lösung etlicher sich daraus ergebende Probleme, konnte nur in Teamarbeit bewerkstelligt werden. Infolgedessen ergaben sich inhaltliche Überschneidungen, in erster Linie in den einleitenden Kapiteln GSM, Software und Hardware. Die Ausformulierung und Erstellung der Grafiken erfolgte eigenständig, jedoch in gemeinsamer Abstimmung.

Freiburg, den 1. November 2009

Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mich während meiner Masterarbeit unterstützt haben. Dabei gilt ein besonderer Dank meinem Betreuer Dr. Dirk von Suchodoletz für die Unterstützung, konstruktive Kritik und die Anregungen zur Ausarbeitung sowie Herrn Prof. Dr. Gerhard Schneider für die Bereitstellung des Themas. An dieser Stelle ist auch die stets angenehme Arbeitsatmosphäre am Lehrstuhl für Kommunikationssysteme hervorzuheben. Darüber hinaus möchte ich mich noch bei meinen Studienkollegen Konrad Meier für die zahlreiche Tipps und die Unterstützung gerade zu Beginn dieser Masterarbeit und Dennis Wehrle für die Begleitung während des Studiums und bei der Erstellung dieser Masterarbeit bedanken. Zudem gilt mein Dank allen weiteren Personen in meinem Umfeld für ihre Ermutigungen und moralische Unterstützung. Zu guter Letzt möchte ich mich sehr herzlich bei meinen Eltern für den Rückhalt und ihre zahlreichen Formen der Unterstützung bedanken, die mir während meines gesamten Studiums zuteil wurden.

Kurzzusammenfassung

Für Demonstrationszwecke in Vorlesungen und für Sicherheitsuntersuchungen ist der Aufbau einer prototypischen GSM Base Tranceiver Station (mit notwendigem Backend) von Interesse. Ähnlich wie für Vorlesungen über Netzwerke, die typischerweise vielseitige praktische Demonstrationen bieten, sollen auch im Bereich Mobilfunk praktische Szenarien und Abläufe innerhalb eines GSM-Netzes beleuchtet werden. Hierzu soll für Demonstrationszwecke basierend auf einem Software-Defined Radio (USRP) eine prototypische GSM Base Tranceiver Station aufgebaut werden. Darüber hinaus werden verschiedene Methoden zur Analyse von GSM vorgestellt und Sicherheitsimplikationen genauer betrachtet, die gerade im Bereich der Lehre helfen sollen, GSM anschaulich besser verstehen zu können. Abläufe auf den verschiedenen Netzwerk-Layern können durch den Betrieb einer eigenen GSM Base Transceiver Station analysiert und nachvollziehbar gemacht werden. Dazu gehören vor allem die Registrierung eines Mobiltelefons, Gesprächsabwicklung und Handover von Gesprächen.

Inhaltsverzeichnis

1 Einleitung und Motivation **1**

2 GSM **3**
2.1 Historischer Überblick . . . 3
2.2 Frequenzen . . . 4
2.3 GSM-Netzarchitektur . . . 5
2.3.1 Mobile Station . . . 7
2.3.2 Network Subsystem (NSS) . . . 8
2.3.3 Base Station Subsystem . . . 11
2.4 Schnittstellen . . . 13
2.4.1 A-Schnittstelle . . . 14
2.4.2 A_{bis}-Schnittstelle . . . 14
2.4.3 U_m-Schnittstelle . . . 14
2.5 Das OSI-Referenzmodell und seine Bedeutung in GSM . . . 21
2.6 Systemdienste . . . 23
2.6.1 Einschalten und Einbuchen des Mobiltelefons . . . 23
2.6.2 Lokalisierung und Aktualisierung . . . 25
2.6.3 Eingehende Anrufe . . . 25
2.6.4 Ausgehende Anrufe . . . 27
2.6.5 Handover . . . 29

3 Software **31**
3.1 GNU Radio . . . 32
3.2 OpenBTS . . . 33
3.2.1 Vorrausetzungen . . . 33
3.2.2 Installation . . . 33
3.2.3 Anwendung . . . 33
3.2.4 Konfiguration OpenBTS . . . 34
3.3 Asterisk . . . 34

4 Hardware **36**
4.1 USRP2 . . . 38
4.2 Daughterboards . . . 38
4.3 Modifikationen . . . 39
4.3.1 USRP Taktgeber deaktivieren . . . 39
4.3.2 Externer Taktgeber . . . 40
4.3.3 Filter entfernen 900er Board . . . 40
4.4 Mobiltelefone und SIM-Karten . . . 41

5 OpenBTS-Versuchsaufbau **43**
5.1 Praktisches Setup . . . 43
5.1.1 Aufbau der Hardware-Komponenten . . . 43
5.1.2 Software-Inbetriebnahme . . . 44

5.2 Systemdienste in OpenBTS . . . 46
5.2.1 Registrierung und Authentifizierung eines Mobilfunkteilnehmers . . . 46
5.2.2 Gesprächsauf- und -abbau . . . 48
5.3 Unterschiede zwischen kommerziellem GSM und OpenBTS . . . 49

6 GSM-Analyse 51
6.1 Netzmonitor . . . 51
6.2 USRP und GNU Radio – Spektrumsanalyse . . . 54
6.2.1 OpenBTS Spektrumsanalyse . . . 56
6.3 USRP – AirProbe und GSSM . . . 57
6.3.1 GSSM . . . 58
6.4 GSM Decodierung mit Nokia 3310 und Wireshark . . . 62
6.4.1 Voraussetzungen und Installation . . . 63
6.4.2 Echte Netze analysieren . . . 63
6.4.3 OpenBTS-Netz analysieren . . . 63
6.4.4 Handover-Szenario . . . 66
6.4.5 Erzwungener Zellwechsel . . . 67
6.4.6 Ergebnis und Vergleich . . . 68

7 Sicherheitsimplikationen 69
7.1 OpenBTS als IMSI-Catcher . . . 69
7.2 Ortung . . . 70
7.3 Verschlüsselte Gespräche entschlüsseln . . . 71
7.4 Unverschlüsselte GSM-Backend-Struktur . . . 71
7.5 SMS-Spam . . . 72
7.6 Konfiguration „Over the Air“ . . . 72

8 Ausblick 73

Literaturverzeichnis 75

Abkürzungsverzeichnis 78

Abbildungsverzeichnis 80

Tabellenverzeichnis 82

A Installationsanleitung GNU Radio 83

B OpenBTS-Changelog 86

C Übersicht der CD 87

1 Einleitung und Motivation

Das Mobilfunknetz ist die weltweit meist verbreitetste elektronische Kommunikationsmöglichkeit. Mitte des Jahres 2009 verfügte es bereits über vier Milliarden Mobilfunkteilnehmer, die weltweit telefonieren oder Datendienste nutzen.[1] Zum Vergleich hatte das Internet zur gleichen Zeit etwa eine Milliarde Nutzer.[2] Im Gegensatz zum Internet sind wenige relevante Punkte im GSM-Netz, gerade im Bereich Sicherheit, genauer betrachtet worden. Das liegt einerseits an der bis vor kurzem finanziell nur schwer erschwinglichen Hardware zum Beispiel für den Betrieb eines IMSI-Catchers (200.000 - 300.000 Euro[3]) und zum anderen daran, dass die Mobilfunkanbieter, trotz offen einsichtlicher Spezifikationen, vieles unter Verschluss gehalten hatten.
Die Netzabdeckung und Verfügbarkeit der einzelnen Mobilfunkdienste lässt selten noch zu wünschen übrig und wird fast als selbstverständlich angesehen. In etlichen Entwicklungsländern in Asien, Afrika oder auch in Südamerika, in Luft- und Raumfahrttechnik oder auch in spärlich besiedelten Gebieten wie zum Beispiel in weiten Teilen Russlands sieht die Lage allerdings ganz anders aus. Rein technisch gesehen ließen sich diese Gebiete problemlos mit GSM ausstatten und versorgen. Die Kosten würden allerdings den Nutzen um ein Vielfaches übersteigen. An dieser Stelle setzt das OpenBTS Projekt an. Die Entwickler haben sich zum Ziel gesetzt, diese nicht abgedeckten Bereiche kostengünstig mit GSM versorgen zu können. In Kombination mit dem GNU Radio Projekt, der Hardware USRP der Firma Ettus und der Open Source Telefonanlagen Software Asterisk kann eine GSM-Zelle samt benötigter Hintergrundinfrastruktur aufgebaut werden. Jedes Mobilfunkgerät ist letztendlich als ein normales Endgerät in einem VoIP-Netz anzusehen. Diese eigene Mobilfunkzelle kann, wie diese Arbeit zeigen soll, auch dazu benutzt werden, die verschiedensten Abläufe in GSM sichtbar zu machen und für Lehr- und Forschungszwecke auf praktische Art und Weise zu demonstrieren.

Das Ziel dieser Arbeit ist der Aufbau und Betrieb eines eigenen GSM-Mobilfunknetzes unter der Benutzung des OpenBTS Projektes, der dazu notwendigen Hardware Universal Software Radio Peripheral (USRP) und softwareseitigen Asterisk Voice over IP Telefonanlage. Darüber hinaus sollen verschiedenste Analysemethoden vorgestellt werden, die gerade im Bereich der Lehre die Funktionsweise von GSM näher bringen können. Besonders Vorlesungen über Netzwerke bieten typischerweise auch praktische Demonstrationen. Diese sollen auch im Bereich GSM zur Verfügung stehen und verschiedenste Abläufe für Demonstationszwecke besser zugänglich machen. Hierzu gehören unter anderem das Registrieren eines Mobilfunkteilnehmers, Rufauf- und -abbau, Handover, aber auch Dienste wie SMS und die Analyse von Daten und vor allem Signalisierungsabläufen im experimentellen und realen GSM-Netz. Diese verschiedenen Abläufe und sich daraus ergebende Probleme im Bereich Sicherheit können so beispielsweise Studenten/Schülern durch konkrete Experimente und Versuche näher erläutert werden. Nur wenn konkrete Abläufe im Bereich GSM und sich daraus ergebende Risiken praktisch demonstriert werden, hilft es, diese Vorgänge und vor allem sich daraus ergebende Sicherheitsprobleme in GSM sichtbar zu machen,

[1] Roamingpartner – Telefonieren im Ausland mit den deutschen Netzbetreibern, *http://www.roaminginfo.de/Roamingpartner.pdf* [Online; letzter Aufruf 25.10.2009]
[2] Zahl der Internetnutzer weltweit übersteigt Milliardengrenze, *http://www.zdnet.de/news/wirtschaft_telekommunikation_zahl_der_internetnutzer_weltweit_uebersteigt_milliardengrenze_story-39001023-39201613-1.htm* [Online; letzter Aufruf 25.10.2009]
[3] IMSI-Catcher, *http://de.wikipedia.org/wiki/IMSI-Catcher* [Online; letzter Aufruf 25.10.2009]

zu verstehen und sie im Idealfall in zukünftigen Entwicklungen zu vermeiden.

In Kapitel 2 dieser Arbeit werden zunächst auf die GSM-Infrastruktur und ihre Komponenten genauer eingegangen und vor allem die Abläufe auf der Luftschnittstelle beschrieben. In Kapitel 3 folgt ein Überblick über die Software OpenBTS und die dafür nötigen Voraussetzungen wie GNU Radio und Asterisk. Die notwendige Hardware zum Aufbau eines eigenen GSM-Netzes wird in Kapitel 4 vorgestellt. Es wird auch auf notwendige Modifikationen und Erweiterungen der Hardware eingegangen und ein Überblick über die verwendeten Mobiltelefone gegeben. In Kapitel 5 werden der Versuchsaufbau von OpenBTS im Detail beschrieben und verschiedene Einstellungen und Parameter, die für den Betrieb nötig sind, erläutert. Außerdem erfolgt ein Vergleich der in OpenBTS bereits implementierten Systemdienste wie Einbuchen des Mobiltelefons, Rufauf- bzw. Rufabbau und Dienste wie SMS mit dem „realen“ GSM-Netz, wie in Kapitel 2 beschrieben, um Unterschiede bzw. noch nicht vorhandene Funktionen deutlich zu machen. Kapitel 6 gibt einen Überblick verschiedenster Analysemethoden. Mittels dieser Methoden können gerade im Bereich Lehre die Funktionsweise von GSM näher gebracht und konkrete Abläufe praktisch aufbereitet werden. Auf verschiedenste Sicherheitsimplikationen wird in Kapitel 7 näher eingegangen, und es werden mögliche Angriffsszenarien und Schwachstellen aufgezeigt. Das letzte Kapitel gibt einen Ausblick über zukünftige Projekte und mögliche weitere interessante Fragestellungen. Im Anhang befinden sich die Installationsanleitung von Gnu Radio, der Changelog von OpenBTS sowie eine Übersicht des CD-Inhaltes. Im entsprechenden Kapitel wurde darauf verwiesen.

Alle Befehle, Datei- und Ordner-Namen wurden kursiv hervorgehoben. Der Hauptteil der Grafiken wurde mit Microsoft Visio erstellt und ist im Ordner */Bilder* der CD für die Verwendung in Lehrveranstaltungen und Vorträge zu finden.

2 GSM

GSM steht für „Global System for Mobile Communication“ (ursprünglich für Groupe Spéciale Mobile). Es handelt sich dabei im Wesentlichen um eine standardisierte Spezifikation eines kompletten Mobilfunksystems (das Logo der „GSM Association“, einer Vereinigung der GSM-Mobilfunkanbieter -gegründet 1987-, ist in Abbildung 2.1 zu sehen). Zu den Zielen von GSM gehörten unter anderem die Schaffung eines europaweit einheitlichen Systems (inzwischen auf allen Kontinenten eingesetzt, vor allem in Europa und weiten Teilen von Asien) zur drahtlosen digitalen Sprach- und Datenübertragung. Das System soll die Mobilität aller Teilnehmer gewährleisten und komplett in das bereits vorhandene ISDN-Netz integriert werden. Darüber hinaus soll es unabhängig von der Anzahl der Teilnehmer eine gute Sprachqualität beziehungsweise später auch eine schnelle und stabile Datenübertragung bei gewährleisteter Abhörsicherheit bieten.

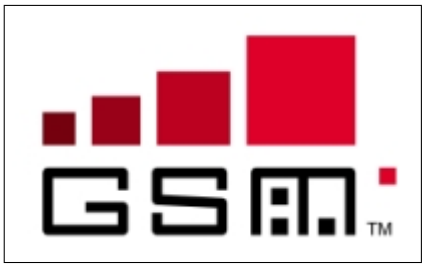

Abbildung 2.1: Logo der „GSM Association“

2.1 Historischer Überblick

Im Jahre 1992 begannen viele europäische Netzbetreiber mit einem kommerziellen Netzstart [10]. Die Nutzung zur Datenübertragung stand dabei zunächst nicht im Mittelpunkt, wurde aber bis heute durch Zusatzspezifikationen hinsichtlich der Datenrate stetig verbessert. Im Jahre 1982 wurde auf der CEPT (Europäische Konferenz der Post- und Fernmeldeverwaltungen) die „Groupe Spécial Mobile“ (etwa Arbeitsgruppe für Mobilfunk) eingerichtet. Das Ziel dieser Arbeitsgruppe sollte die Entwicklung eines einheitlichen pan-europäischen Mobilfunkstandards sein. Am 7. September 1987 unterzeichneten 17 GSM-Netzbetreiber aus 15 verschiedenen europäischen Ländern in Kopenhagen das GSM MoU (Memorandum of Understanding). Bis zum Juni 2008 gab es weltweit 2,9 Milliarden GSM-Nutzer [17]. Die tägliche Zuwachsrate liegt bei ca. 700.000 neuer Kunden. Diese stammen hauptsächlich aus Ländern in Asien, Afrika und Südamerika. Der jährliche Umsatz mit GSM-Technik liegt laut einer aktuellen Schätzung bei geschätzten 277 Milliarden Dollar. Bis Ende des Jahres 2008 hatte GSM einen globalen Marktanteil von 89,5% [12].
Tabelle 2.1 soll einen genaueren Überblick über die Historie wichtiger Netzwerktechnologien im Bereich der Mobilkommunikation geben.

Jahr	Ereignis
1982	Gründung der Groupe Spéciale Mobile (GSM).
1987	Entscheidung für TDMA/FDMA (aus 9 verschiedenen Vorschlägen).
1990	Phase 1 der **GSM 900-Spezifikationen** (für 900 MHz) wird beendet, d.h. sie werden nicht mehr verändert. Die Entwicklung für das System DCS 1800 (Digital Cellular System) (1800 MHz) beginnt.
1991	Spezifikationen für **DCS 1800** werden eingefroren. Die ersten lauffähigen Systeme werden auf Messen vorgeführt.
1992	Die meisten GSM-Netzbetreiber beginnen mit Sprachdiensten den kommerziellen Netzstart. Ende 1992 sind bereits 13 Netze in 7 Ländern verfügbar.
1995	Phase 2 der GSM-Standardisierung beginnt. Dazu gehören Dienste wie **FAX**, **Daten** und **SMS-Roaming**.
1999	Einführung von **WAP** (Wireless Application Protocol). Damit können erstmalige Inhalte des Internets auf mobile Endgeräte übertragen werden.
2000	Die GSM-Standardisierungsaktivitäten werden nach 3GPP überführt. Die Arbeitsgruppe dort trägt die Bezeichnung TSG GERAN (Technical Specification Group GSM EDGE Radio Access Network).
2000	Einführung von **GPRS** (General Paket Radio Service) als Erweiterung zu GSM, um einen drahtlosen Zugang zu paketorientierten Datendiensten mit bis zu 171,2 kbit/s zu ermöglichen.
2000	Einführung eines neuen Mobilfunkstandards der dritten Generation (3G): **UMTS** (Universal Mobile Telecommunication System).
2006	Die Zahl der Netzbetreiber liegt bei 147 in 213 verschiedenen Ländern. In Deutschland gibt es über 70 Millionen Mobiltelefone.

Tabelle 2.1: Historischer Überblick wichtiger Netzwerktechnologien im Bereich der Mobilkommunikation [[19], [31], [21]]

2.2 Frequenzen

Zu den momentan eingesetzten GSM-Standards gehören:

- GSM 900: Frequenzbereich um 900 MHz (D-Netze, in Deutschland T-Mobile und Vodafone)
- GSM/DCS[1] 1800: Frequenzbereich um 1800 MHz (E-Netze, in Deutschland alle Betreiber)
- GSM/PCS[2] 1900: Frequenzbereich um 1900 MHz (hauptsächlich in den USA eingesetzt)

Wie Tabelle 2.2 entnommen werden kann, sind in Deutschland zwei Varianten des GSM-Standards im Einsatz: Das etwas ältere D-Netz (GSM900) und das später eingeführte E-Netz (GSM1800). GSM wurde zu Beginn nur im 900 MHz-Band betrieben. Für den Uplink steht der Frequenzbereich 890,2-915 MHz und für den Downlink 935,2-960 MHz zur Verfügung. Das Band wurde im Laufe der Zeit um jeweils 9,8 MHz auf 880,4-915 beziehungsweise 925,4-960 MHz erweitert (Extended GSM oder E-GSM). Mit dem 1800 MHz-Band wurde aufgrund der immer stärker werdenden Nutzung ein weiteres Band eingeführt (GSM 1800 oder DCS 1800). Der Uplink wird im Bereich 1710,2 bis 1785 MHz betrieben, während die Frequenzen für den Downlink im 1805,2 bis 1880 MHz liegen. Die Bandbreite von GSM 1800 beträgt somit 74,8 MHz, was 374 Kanälen mit je 200 KHz entspricht.

[1] Digital Cellular System
[2] Personal Communications Service

Um einem entsprechenden Mobilfunkanbieter einen Frequenzbereich des Gesamtspektrums zuweisen zu können, werden die Frequenzbänder in nummerierte Kanäle eingeteilt. Mit dieser „Absolute Radio Frequency Channel Number“ (ARFCN) kann direkt sowohl die Uplink- als auch die Downlink-Frequenz bestimmt werden. GSM 900 benutzt die Kanäle mit den Nummern 1-124 (also 124 Kanäle), GSM 1800 die Kanalnummern 512-885 (374 Kanäle) und E-GSM zusätzlich die Kanalnummern 975-1023 (49 Kanäle). Eine genauere Übersicht über die verwendeten Frequenzbänder und deren dazugehörige Kanalnummern kann der Tabelle 2.2 entnommen werden [1].

2.3 GSM-Netzarchitektur

Beim GSM-Netz handelt es sich um eine zellular aufgebaute Struktur. Diese Zellen sind gedachte Hexagone. Abbildung 2.2 kann man eine theoretische Anordnung dieser Hexagone im Vergleich zu einer in der Praxis realistischen Anordnung der Funkzellen entnehmen. Durch unregelmäßige Parameter wie Bevölkerungsdichte, Bebauung und Landschaftsbeschaffenheit ergeben sich in der Praxis eine unregelmäßigere Form, die deutlich von der im Idealfall hexagonalen Anordnung der Funkzellen abweicht. Eine Basisstation befindet sich zumindest meistens genau in der Mitte einer solchen Zelle. Je größer die Reichweite einer solchen Basisstation ist, desto größer ist die Zelle. Je größer die Trägerfrequenz allerdings ist, desto geringer fällt die Größe einer Zelle aus. Beim D-Netz (900 MHz) ist die maximal mögliche Zellengröße noch ca. 35 km, wobei das E-Netz (1800 MHz) auf Grund der höheren Frequenz nur noch eine maximale Zellgröße von 8 km ermöglicht. Jede Zelle bekommt eine bestimmte Anzahl von Frequenzen zugeordnet, wobei benachbarte Zellen andere Frequenzen erhalten, um Interferenzen zu vermeiden. Die räumliche Wiederholung von Frequenzen führt zu einer Clusterbildung. Hierbei bilden eine Gruppe von Zellen einen Cluster, die den gesamten Frequenzbereich abdecken. Bei GSM-Netzen bilden in der Praxis meist sieben Zellen einen Cluster. Zwei Zellen mit derselben Frequenz müssen über einen Mindestabstand „D“ verfügen (siehe Abbildung 2.2). Dieser Mindestabstand wird als „frequency reuse distance“ bezeichnet. Bewegt sich nun ein mobiler Teilnehmer von einer zur anderen Zelle, muss gewährleistet werden, dass seine bestehenden Verbindungen von einer zur anderen Basisstation weitergegeben werden. Man spricht hierbei von einem sogenannten „Handover“ (siehe Kapitel 2.6.5).

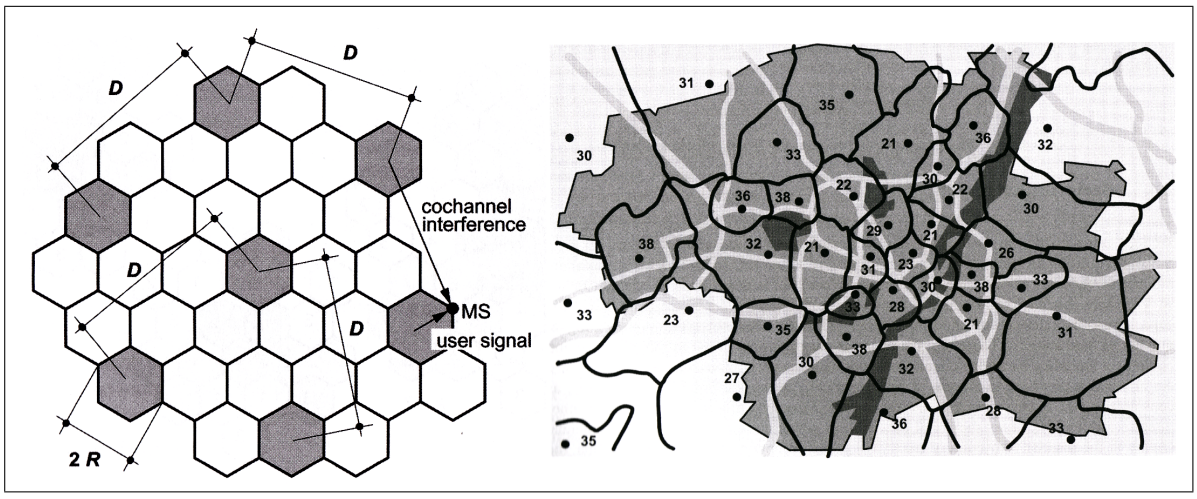

Abbildung 2.2: Struktur eines GSM-Netzes in der Theorie und Praxis [17]

Band-Bezeichnung	Bereich	Uplink (MHz)	Downlink (MHz)	ARFCN	Anmerkungen
T-GSM 380	GSM 400	380,2-389,8	390,2-399,8	dynamisch	T-GSM = TETRA GSM
T-GSM 410	GSM 400	410,2-419,8	420,2-429,8	dynamisch	
GSM 450	GSM 400	450,4-457,6	460,4-467,6	259-293	Das Frequenzband wird für GSM bisher nur von Celtel in Tansania eingesetzt
GSM 480	GSM 400	478,8-486,0	488,8-496,0	306-340	Das Frequenzband wird für GSM bisher nur von Celtel in Tansania eingesetzt
GSM 710	GSM 700	698,0-716,0	728,0-746,0	dynamisch	
GSM 750	GSM 700	747,0-762,0	777,0-792,0	438-511	Das Frequenzband wird für GSM bisher nicht eingesetzt
T-GSM 810	806,0-821,0	851,0-866,0	dynamisch		
GSM 850	GSM 850	824,0-849,0	869,0-894,0	128-251	Amerika
P-GSM	GSM 900	890,0-915,0	935,0-960,0	1-124	Afrika, Amerika, Asien, Australien, Europa
E-GSM	GSM 900	880,0-915,0	925,0-960,0	0,1-124,975-1023	Europa
R-GSM	GSM 900	876,0-915,0	921,0-960,0	0,1-124,955-1023	Asien, Europa
T-GSM 900	GSM 900	870,4-876,0	915,4-921,0	dynamisch	
DCS 1800	GSM 1800	1710,0-1785,0	1805,0-1880,0	512-885	Afrika, Amerika, Asien, Australien, Europa
PCS 1900	GSM 1900	1850,0-1910,0	1930,0-1990,0	512-810	Amerika

Tabelle 2.2: Übersicht über die verwendeten Frequenzbänder [21], [31]

GSM ist ein hierarchisch gegliedertes System verschiedenster Netzwerkkomponenten. Es besteht im Wesentlichen aus den Mobilstationen (MS, Mobilteilnehmer) und dem fest installierten GSM-Netz, zusammengesetzt aus Networksubsystem und Base Station Subsystem. Von einem BSC (Base Station Controller) werden mehrere, meist einige 10 bis einige 100, BTS (Base Transceiver Station) verwaltet. An jede BTS sind meist mehrere Antennen angeschlossen. Es wird zwischen omnidirektionalen Antennen und Sektorantennen unterschieden. Omnidirektionale Antennen strahlen rundum gleichmäßig in eine Zelle. Üblicherweise werden momentan Sektorantennen verwendet (meist zwei oder drei Stück). Bei drei Sektorantennen versorgt jede Antenne einen 120° Sektor. Die verschiedenen BSCs sind an das MSC (Mobile Switching Center) angeschlossen, welches über eine Datenbank VLR (Visitor Location Register) zur Benutzerauthentifizierung verfügt. Die GSM-Systemarchitektur kann Abbildung 2.3 entnommen werden und wird in den folgenden Abschnitten genauer erläutert.

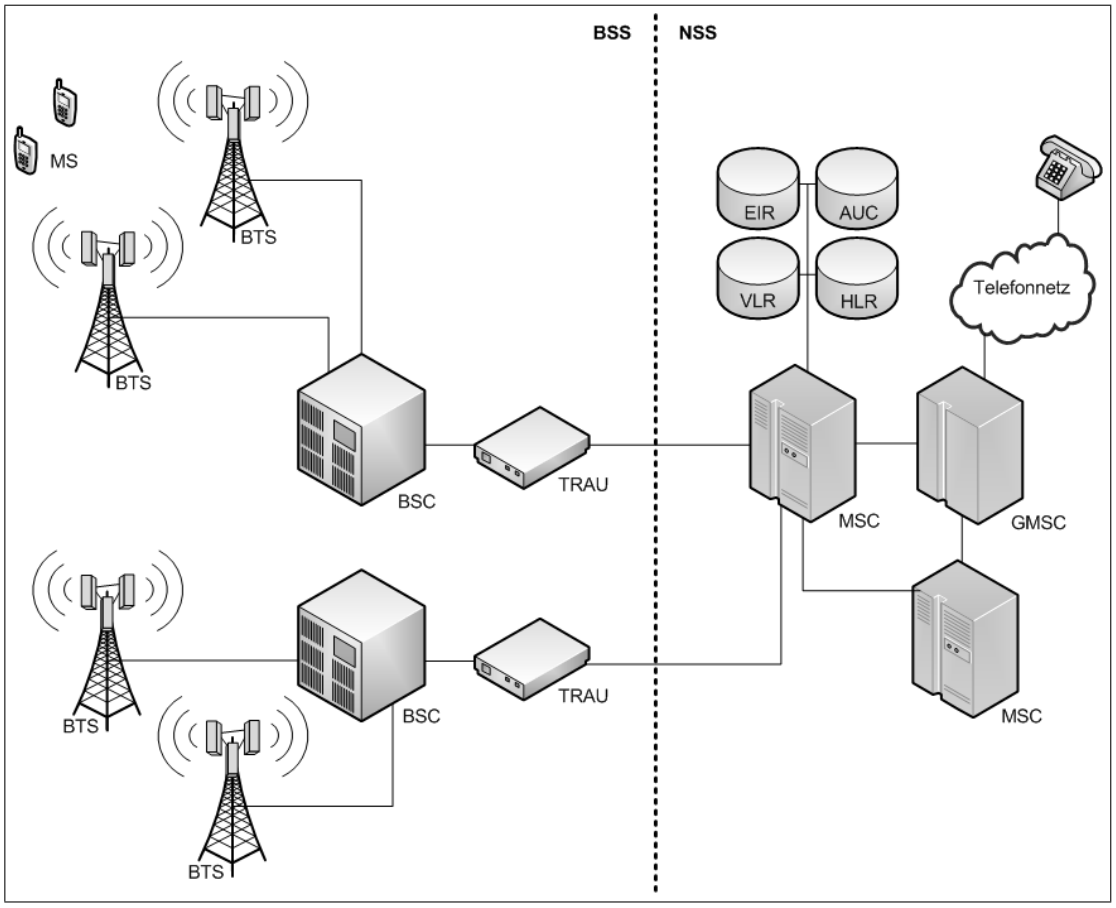

Abbildung 2.3: Struktur eines-GSM Netzes

2.3.1 Mobile Station

Die Mobile Station (MS) ist in der GSM-Mobilfunkarchitektur das eigentliche Mobiltelefon mit eingesetzter SIM-Karte (siehe Abschnitt 2.3.1). Die zentrale Idee dabei ist die Trennung von Funkhardware und Sicherheitsmodul. Die SIM-Karte ist stets Vertragsgegenstand des dazugehörigen Mobilfunkanbieters und übernimmt die sicherheitsrelevanten Dinge wie Authentifizierung und Verschlüsselung. Bei der Funkhardware handelt es sich zumeist um ein Mobiltelefon. Es

kann aber auch eine Datenkarte (PCIe) oder ein GSM-Modem für die Datenübertragung sein.[3] Die Mobile Station ermöglicht dem Nutzer den Zugang zum Mobilfunknetz. Das Empfangen und Senden von Nutz- und Steuerdaten gehört zu ihren wichtigsten Aufgaben.

SIM-Karte

Bei der SIM-Karte (Subscriber Identity Module) handelt es sich um eine Chipkarte aus der „Familie“ der Smart-Cards (Abbildung 2.4). Diese ist in das Mobiltelefon oder sonstiges Endgerät eingesteckt und dient zur Identifikation des Nutzers im Netz. Zu den Bestandteilen einer SIM-Karte zählen: CPU, Bussystem, Arbeitsspeicher, EEPROM (löschbarer Festspeicher) und Schnittstellen auf einem Kontaktfeld. Elektronische Komponenten sind in einer Schaltung integriert und durch dünne Metallfilme als Leiterbahnen miteinander verknüpft. Die SIM-Karte ist ein kleiner Prozessor mit Speicher (üblicherweise im ID-000-Format) mit Daten über den Nutzer, den Mobilfunkbetreiber und die Zugriffsberechtigung (Telefonnummer, Teilnehmerschlüssel, Rufnummern) zum Netz. Diese ist durch eine veränderbare PIN vor unbefugter Benutzung geschützt. Auf dem Mobiltelefon sind die IMEI (International Mobile Equipment Identity), der

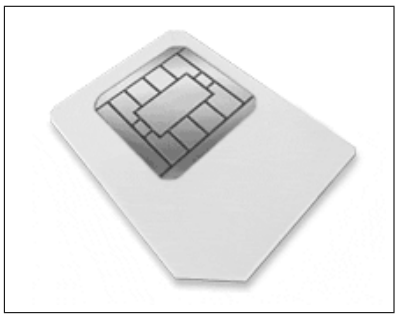

Abbildung 2.4: SIM-Karte

A5/x Verschlüsselungsalgorithmus[4] und zusätzliche teilnehmerspezifische Daten wie Telefonbuch und Geräteeinstellungen dauerhaft gespeichert. Auf der SIM-Karte sind permanent die IMSI, TMSI (Temporary Mobile Subscriber Identity), MSISDN (Mobile Station ISDN-Number), LAI (Location Area Identifier)[5], Kc (Schlüssel für Chiphering Kc = (Ki, RAND)), Ki (individueller Schlüssel für Authentifizierung), A8 Algorithmus zur Berechnung von Kc, A3 Algorithmus zur Berechnung von SRES, CKSN (Ciphering Key Sequence Number), ACC (Access Control Class), NCC (Network Colour Code des Heimnetzes), ARFCNs (Trägerfrequenzen des Heimnetzes) und zusätzliche teilnehmerspezifische Daten wie Telefonbuch, Gebührenzähler, Benutzerprofile, SMS, Anruflisten gespeichert.

2.3.2 Network Subsystem (NSS)

In einem GSM-Netz übernimmt das Netzwerk-Subsystem (NSS), auch Switching Subsystem (SSS) genannt, die vermittlungstechnischen Aufgaben, zu denen das Durchschalten der Funkkanäle auf Festnetze wie ISDN oder Datex-P[6] gehört.

[3] GSM-Systemarchitektur, *http://www.elektronik-kompendium.de/sites/kom/0910191.htm* [Online; letzter Aufruf 28.09.2009]

[4] Es gibt ihn in zwei Varianten: A5/1: die strengere Variante mit ca. 130 Mio. Benutzern innerhalb Europas und A5/2: die schwächere Variante mit ca. 100 Mio. Benutzern in anderen Regionen.

[5] Wird von der Basisstation zur Gebietskennung benutzt, Kombination aus Landes-, Netz- und Gebietskennzahl

[6] Datex-P (**Da**ta **Ex**change, **p**aketorientiert) ist die Produktbezeichnung der Deutschen Telekom für ein Kommunikationsnetz zur Datenübertragung

Mobile Switching Center

Das Mobile Switching Center (MSC) ist die zentrale Einheit, die diese Funktionen durchführt. Die MSC unterstützt diverse Dienste für die Datenübertragung und den Kurznachrichtendienst SMS sowie das Routing von Fremdnetzen hin zur Mobilstation. MSC und GMSC haben Zugriff auf verschiedene Datenbanken, deren Aufgabe in den folgenden Abschnitten genauer beschrieben wird.

Gateway Mobile Switching Center

Das Gateway Mobile Switching Center (GMSC) operiert als MSC. Zusätzlich übernimmt es die Verwaltung, die nötig ist, um mit einem fremden ISDN- bzw. PSTN-Netz (Public Switched Telefon Network = Festnetz) oder auch PLMN (Public Land Mobile Network = Mobilfunknetz) zu kommunizieren. Das GMSC dient also als Vermittlungsstelle zwischen dem eigenen GSM-Netz und allen anderen leitungsorientierten Telefonnetzen wie Festnetz oder einem anderen Mobilfunknetz. Es übernimmt Routingaufgaben für ein- bzw. ausgehende Anrufe.

Home Location Register

Das Home Location Register (HLR) entspricht der Teilnehmerdatenbank eines GSM-Netzes pro Mobilfunkanbieter. Es beinhaltet permanent gespeicherte Daten eines Mobilfunkteilnehmers. Im HLR ist zu jedem Mobilfunkteilnehmer gespeichert, für welche zusätzlichen Dienste des Mobilfunknetzwerkes er autorisiert ist. Dazu können unter anderem Dienste wie Gesprächsweiterleitung, Anklopfen, Konferenzschaltung und Rufumleitung gehören. Der wohl wichtigste Datensatz, der im HLR gespeichert ist und auf der SIM-Karte, ist die weltweit eindeutige International Mobile Subscriber Identity (IMSI). Sie wird bei fast allen teilnehmerbezogenen Signalisierungsvorgängen verwendet. Diese Nummer wird dezimal gespeichert und setzt sich zusammen aus:

- dreistelligem Mobile Country Code (MCC) des entsprechenden Landes (z.B. Deutschland 262, Österreich 232),
- dem zweistelligen Mobile Network Code (MNC) des Netzwerkes eines Teilnehmers (in Deutschland z.B. 01 für T-Mobile, 02 für Vodafone),
- und der zehnstelligen Mobile Subscriber Identification Number (MSIN), die im nationalen Netzwerk eindeutig ist.

Visitor Location Register

Die Aufgabe eines Visitor Location Register (VLR) besteht darin, temporär eine Kopie des entsprechenden Datensatzes eines Mobilfunkteilnehmers aus dem HLR zu laden, der sich gerade im Empfangsbereich aufhält. Der Sinn liegt darin, die Kommunikation zwischen MSCs und HLRs zu entlasten und die Zugriffszeit auf die Daten zu verringern. Gespeichert werden unter anderem die Mobile Station Roaming Number (MSRN), die Temporary Mobile Subscriber Identity (TMSI) und die Location Area (LA). Diese temporären Daten dienen einerseits zur Mobilitätsverwaltung, andererseits aber auch für Sicherheitsfunktionen. Die Schnittstelle zwischen HLR und VLR bezeichnet man als Mobile Application Part (MAP).

Authentification Center

Das Authentification Center (AC) hat die Aufgabe, jede SIM-Karte, die sich versucht in das GMS-Netzwerk zu verbinden, zu authentifizieren. Dies geschieht üblicherweise nach dem Einschalten des Mobiltelefons.

Bei dem Authentifizierungsprozess im GSM-Netzwerk handelt es sich um ein klassisches Challenge-Response-Verfahren. Durch dieses Verfahren wird auch die Verschlüsselung ausgehandelt. Verwendet werden dabei die Algorithmen A3 und A8. Detailliert funktioniert dieses Challenge-Response-Verfahren wie folgt:

1. Ein Mobiltelefon loggt sich in das GSM-Netz ein.
2. Das Mobile Services Switching Center (MSC) fordert fünf „Triples“ vom Home Location Register (HLR) an.
3. Mit Hilfe des A8 Algorithmus werden durch das Home Location Register fünf „Triples“ generiert. Jedes dieser fünf Tripels enthält:
   - eine 128-Bit zufällige Challenge-Zahl (RAND)
   - einen 32-Bit matching Signed Response (SRES)
   - einen 64-Bit Ciphering Key, der als Session Key (Kc) verwendet wird
4. Diese fünf Triples werden durch das Home Location Register dem Mobile Services Switching Center gesendet.
5. Das Mobile Services Switching Center sendet aus dem ersten Triple die zufällige 128-Bit Challenge-Zahl an die Base Transceiver Station (BTS).
6. Die Base Transceiver Station überträgt diese Zahl an das entsprechende Mobiltelefon.
7. Das Mobiltelefon empfängt die zufällige 128-Bit Challenge-Zahl und verschlüsselt diese mittels des A3 Algorithmus. Hierzu wird diese mit dem individuellen „Subscriber Authentication Schlüssel“ des entsprechenden Mobilfunkteilnehmers kombiniert (dieser ist auf der SIM-Karte gespeichert).
8. Das Mobiltelefon sendet das Ergebnis („Signed Response“) zurück zur Base Transceiver Station. Diese wiederum leitet diese an das Mobile Services Switching Center weiter.
9. Das Mobile Services Switching Center überprüft dieses Signed Response (SRES). Falls diese mit der durch das Mobile Services Switching Center berechnete Signed Response (SRES) übereinstimmt, ist die Authentifizierung erfolgreich abgeschlossen.
10. Mittels des A8 Algorithmus, des individuellen Subscriber Authentication Schlüssels und der zufälligen 128-Bit Challenge-Zahl generiert das Mobiltelefon einen Session Key (Kc).
11. Mittels des Session Keys (Kc) und des A5 Algorithmus kann nun eine verschlüsselte Kommunikation erfolgen und die entsprechenden Frames können ver- bzw. entschlüsselt werden.

Somit authentifiziert sich das Mobiltelefon gegenüber dem entsprechenden GSM-Netzwerk. Es handelt sich hierbei allerdings nicht um eine bidirektionale Authentifizierung. Das GSM-Netzwerk authentifiziert sich umgekehrt nie bei einem Mobiltelefon, was eine Man-in-the-middle Attacke problemlos ermöglicht. Dadurch ergeben sich mehrere Sicherheitsimplikationen, auf die in Kapitel 7.1, 7.2 und 7.6 genauer eingegangen wird. Es findet auch eine Verschlüsselung lediglich zwischen Mobile Station und BTS auf der U_m-Schnittstelle statt. Mögliche Risiken der unverschlüsselten Übertragung in der restlichen GSM-Infrastruktur werden in Kapitel 7.4 genauer betrachtet. Der gesamte Authentifizierungsprozess kann noch einmal zusammengefasst der Abbildung 2.5 entnommen werden.

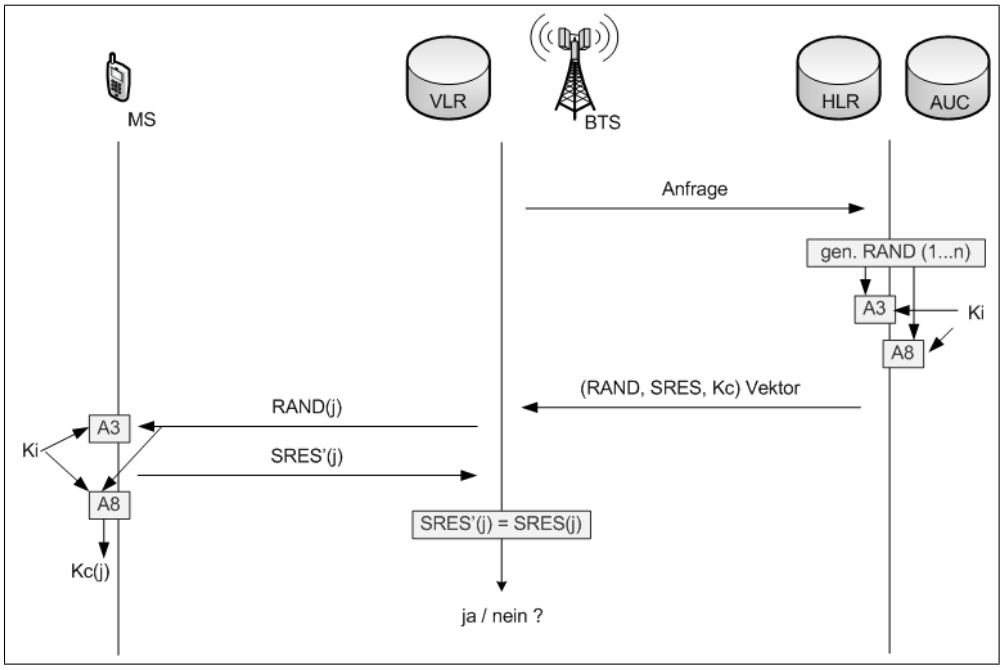

Abbildung 2.5: Authentifizierungsprozess im GSM-Netzwerk unter Verwendung der Algorithmen A3 und A8 (Vorlage nach [6])

Equipment Identity Register

In dem Equipment Identity Register (EIR) sind sämtliche Seriennummern (International Mobile Station Euquipment Idendity, IMSI) aller Endgeräte gespeichert. Es kann eine Art schwarze Liste geführt werden, welche Mobilfunkgeräte wegen Diebstahls oder aus anderen technischen Gründen gesperrt werden müssen. Diese Daten befinden sich für alle Mobilfunkbetreiber zugänglich im Central EIR (CEIR) in Dublin. Die Identifikation des jeweiligen Gerätes erfolgt dabei über die International Mobile Equipment Identity (IMEI). Diese besteht aus vier Teilen: einem Type Approval Code (24 Bit), einem Final Assembly Code (8 Bit), einer Seriennummer (24 Bit) und einer Check Digit (4 Bit). In Deutschland wird diese Speicherung der IMEI lediglich von Vodafone praktiziert.

2.3.3 Base Station Subsystem

Ein Base Station Subsystem (BSS) sorgt in einem GSM-Netz für die Verbindung zwischen den Mobilstationen und dem vermittlungstechnischen Teilsystem. Das BSS-System verwaltet die Mobilfunkfrequenzen und besteht aus den Komponenten Base Station Controller (BSC), Base Transceiver Station (BTS) und der Transcoding und Rate Adaption Unit (TRAU)[7]. Diese TRAU kann Bestandteil des Base Station Subsystems sein. Sie dient der Anpassung von unterschiedlichen Sprachcodierungen zwischen GSM- und Festnetz. In einem GSM-Zeitschlitz als Funkkanal können maximal 22,8 kbit/s übertragen werden (siehe Kapitel 2.4.3). Sämtliche Vermittlungsnetze verwenden allerdings eine Datenrate von 64 kbit/s. So hat ein Gespräch vom Festnetz in das Mobilfunknetz eine Datenrate von 64 kbit/s und muss entsprechend konvertiert und komprimiert werden. Diese Sprachkompression wird von der TRAU übernommen.

[7] BSS :: base station subsystem :: ITwissen.info, *http://www.itwissen.info/definition/lexikon/base-station-subsystem-BSS.html* [Online; letzter Aufruf 28.09.2009]

Base Station Controller

Der Base Station Controller (BSC) ist die Schnittstelle zwischen einer oder mehreren BTS (siehe Abschnitt 2.3.3) und einem MSC. Die Aufgabe des Base Station Controller besteht im Auf- und Abbau, in Überwachung und der Aufrechterhaltung sämtlicher Verbindungen (Steuerungsaufgaben). Der BSC muss also die Frequenzkanäle und Zeitschlitze all seiner Zellen verwalten, um zu wissen, welchen noch nicht besetzten Kanal er bei neuen Gesprächen vergeben kann. Hierzu fragt das MSC einen Funkkanal, um z.B. ein Gespräch vom Festnetz zu einem Mobilteilnehmer durchzustellen, beim zuständigen BSC an. Dieser wird anschließend zugeteilt, sofern ein Funkkanal für diesen Dienst verfügbar ist. Eine weitere wichtige Aufgabe des BSC ist die Sendeleistungskontrolle der kommunizierenden Funkstationen – also MS und BTS (Qualitäts- und Feldstärkekontrolle). Weitere Aufgaben sind die Überwachung der IDLE-Kanäle, BTS- und MS-Power-Control [2]. Das Handover wird ebenfalls durch den BSC durchgeführt, solange es sich beim Zellenwechsel um seine verwalteten Zellen handelt.

Base Transciever Station

Für die eigentliche Funkübertragung ist die Base Transciever Station (BTS) zuständig. Sie dient als Sende- und Empfangseinheit einer Zelle. Üblicherweise sendet und empfängt jede BTS auf genau einem Frequenzpaar. Der entsprechende BSC übernimmt dabei die notwendigen Steueroperationen.

Weitere Aufgaben einer BTS sind unter anderem [2]:

- Meldung der Qualität der IDLE-Kanäle an den BSC
- Timing der BCCHs (Broadcast Control Channel) und CCCHs (Common Control Channel) (siehe Kapitel 2.4.3)
- Kanalcodierung und -decodierung auf der Luftschnittstelle
- Ausführung der Rate Adapation und des Transcoders
- Verschlüsselung auf der Luftschnittstelle
- Weiterleitung der MS- und BTS-Meldungen an dden BSC
- Aufrechterhaltung der Synchronisation zwischen MS und BTS
- Entdeckung eines Anrufs einer MS (RACH)

Eine in der Praxis übliche Konfiguration ist eine BTS mit drei sektorisierten Zellen, die über jeweils zwei Frequenzen (á 200 KHz) verfügen [28]. Somit stehen in jedem Sektor 2 * 8 = 16 Zeitschlitze (wird in Abschnitt 2.4.3 genauer erklärt) zur Verfügung, was bei drei Sektoren 48 Zeitschlitzen entspricht. In jedem Sektor müssen zwei Zeitschlitze für Signalisierungsaufgaben abgezogen werden. Vier oder mehr Zeitschlitze sind Paketdiensten wie GPRS vorbehalten. Somit stehen effektiv pro Sektor zehn Zeitschlitze (insgesammt also 30) für Sprachübertragung zur Verfügung. Damit ist eine Kommunikation von 30 Mobilfunkteilnehmern gleichzeitg möglich. Die Mobilfunknetzbetreiber rechnen im Durchschnitt, dass ein Teilnehmer pro Stunde eine Minute telefoniert. Somit können von einer BTS etwa 60 mal mehr passive als aktive Teilnehmer versorgt werden, was einer Gesamtanzahl von ungefähr 1800 Teilnehmern entspricht, da nicht alle Teilnehmer gleichzeitig telefonieren. Der Abbildung 2.6 können zusammengefasst die Größenordnungen der einzelnen Netzkomponenten entnommen werden.

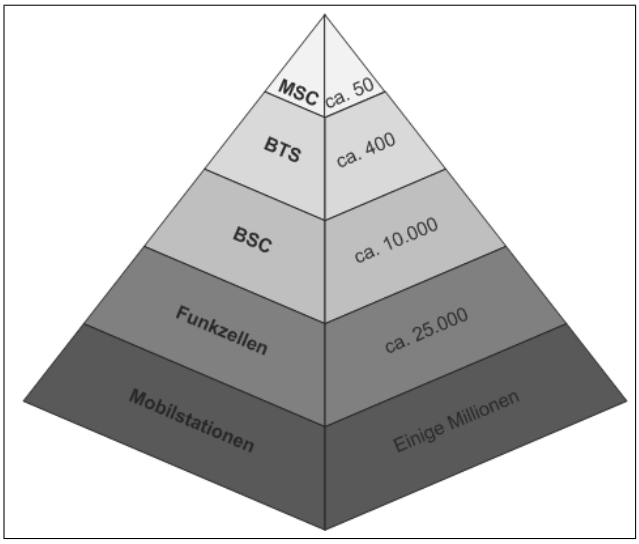

Abbildung 2.6: Größenordnungen der verschiedenen GSM-Netzkomponenten

2.4 Schnittstellen

Zu besonders wichtigen Schnittstellen gehören die A_{ter}-Schnittstelle zwischen Festnetz und Basisstation, die A-Schnittstelle im Festnetz, die A_{bis}-Schnittstelle zwischen Base Station Controller und Base Transciever Station und die Luftschnittstelle (Air-Interface) U_m zwischen Mobilstationen und Base Transciever Station. Alle Schnittstellen zwischen den Komponenten des GSM-Netzes sind in Abbildung 2.7 grafisch dargestellt und noch einmal tabellarisch in Tabelle 2.3 zusammengefasst. Die Schnittstellen sind standardisiert, so dass die einzelnen Netzwerkkomponenten unterschiedlichster Hersteller miteinander kommunizieren können.

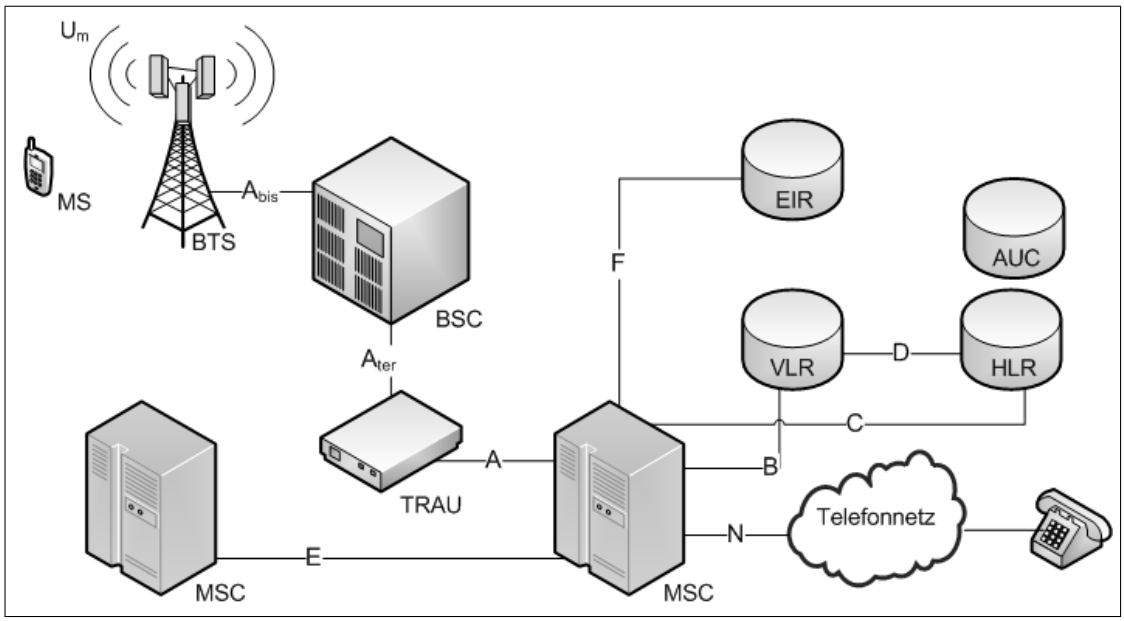

Abbildung 2.7: Übersicht der GSM-Netzschnittstellen

Bezeichnung	von	nach
A-Schnittstelle	MSC	TRAU
A_{ter}-Schnittstelle	TRAU	BSC
A_{bis}-Schnittstelle	BSC	BTS
B-Schnittstelle	MSC	VLR des MSC
C-Schnittstelle	MSC	HLR
D-Schnittstelle	VLR des MSC	HLR
E-Schnittstelle	MSC	MSC (GMSC)
F-Schnittstelle	MSC	EIR
G-Schnittstelle	VLR des MSC	VLR eines anderen MSC
N-Schnittstelle	GMSC	PSTN
U_m-Schnittstelle	BTS	MS

Tabelle 2.3: Übersicht der GSM-Schnittstellen [25]

2.4.1 *A*-Schnittstelle

Das MSC und BSS sind mittels mehrerer 2 MBit/s Leitungen miteinander verbunden. Diese Verbindung bezeichnet man als *A*-Schnittstelle oder *A*-Interface. An der A-Schnittstelle wird das PCM30-Verfahren verwendet. Hierbei handelt es sich um eine Pulse-Code-Modulation mit 30 Verkehrskanälen. Ein PCM30-Rahmen ist folgendermaßen aufgebaut: (0) Synchronisation, (1-15) 15 Nutzkanäle, (16) Signalisierung und (17-31) 14 Nutzkanäle [2].
In Europa stehen 32 * 64 kbit/s Kanäle zur Verfügung. Diese 64 kbit/s ergeben sich wie folgt: Für die Analog-Digital-Umwandlung wird eine Abtastfrequenz von 8 KHz (also alle 125 µs) verwendet. Es wird die Amplitude gemessen und in eines der 256 Intervalle (in 8 Bit) codiert. Dieser Wert wird im nächsten Zwischenabtastintervall übertragen. Somit werden 8 Bit in 125 µs übertragen (=15,625 µs pro Bit), was einer Datenrate von 64 kbit/s entspricht [2].

2.4.2 A_{bis}-Schnittstelle

Von der BTS werden sämtliche Daten der logischen Kanäle über die A_{bis}-Schnittstelle an den BSC weitergeleitet. Der größte Teil der Bandbreite wird von den Traffic Channels (13 kbit/s Daten) verwendet. Sämtliche Signalisierungsdaten (Common Channels, SDCCH, SACCH) sind nicht zeitkritisch und werden gebündelt übertragen. Als Übertragungsprotokoll für diese Signalisierungen wird ein leicht modifiziertes Link Access Protokoll D-Cannel (LAPD) Protokoll verwendet, das aus der ISDN-Welt bekannt ist. Die nicht benötigte Bandbreite zwischen BTS und BSC dient der Kommunikation zwischen BSC und einer oder mehreren weiteren Basisstationen. Über die A_{bis}-Schnittstelle werden die Funkkanäle verwaltet, die Reihenfolge des Frequenzwechsels bestimmt und Handover-Funktionen ausgeführt.

2.4.3 U_m-Schnittstelle

Als Luftschnittstelle, Air-Interface oder U_m-Schnittstelle bezeichnet man bei GSM den physikalischen Übertragungsweg zwischen einer BTS und einem Mobilfunkteilnehmer. Dazu gehören Aufgaben wie Verwaltung der Verkehrs- und Signalisierungskanäle, Handover, Frequenzsprünge, Vielfachzugriff, Multiplexing, Zeitschlitz-/Burstdefinition, Kanalcodierung und Synchronisation. Die BTS verwendet dabei zwei verschiedene Verfahren, um zeitgleich bei gleichzeitiger Nutzung mehrerer Frequenzen mit mehreren Mobilfunkteilnehmern kommunizieren zu können. Die Verfahren werden als Frequency Division Multiple Access Verfahren (**FDMA** = Nutzung mehrerer Frequenzen pro geografischem Ort) und Time Division Multiple Access Verfahren (**TDMA**)

bezeichnet. Es können jeweils acht Mobilfunkteilnehmer pro Trägerfrequenz mit 200 kHz Bandbreite miteinander kommunizieren. Jeder Träger wird in Frames mit einer Länge von 4,615 ms eingeteilt, die wiederum acht voneinander unabhänige Zeitschlitze (Timeslots) mit einer Länge von 577 µs sogenannten Bursts beinhalten (siehe Abbildung 2.8[8]). Jeder Mobilfunkteilnehmer sendet bzw. empfängt in einem entsprechenden Timeslot. Wird einem Mobilfunkteilnehmer beispielsweise der Zeitschlitz drei zugeteilt, darf er in jedem Frame in diesem Zetschlitz senden und empfangen. Up- und down-Link sind dabei allerdings um drei Zeitschlitze versetzt. Der Up-Link liegt aufgrund der geringeren Dämpfung auf dem tieferen Frequenzband.

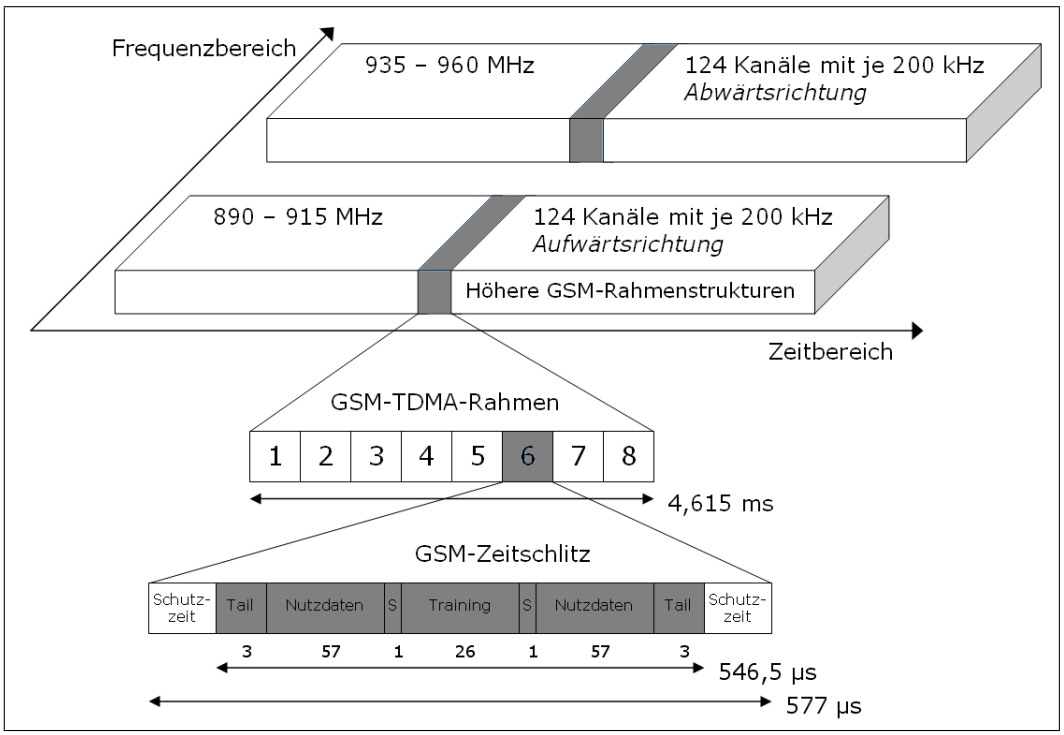

Abbildung 2.8: GSM-Rahmenstruktur

Bursts und logische Kanäle

Sprache und Signalisierungen werden in einer speziellen Form (Bursts) übertragen. Es gibt fünf verschiedene Typen von Bursts: Normal Burst, Frequency Correction Burst, Synchronisation Burst, Dummy Burst und Access Burst [33]. Der Aufbau eines Normal Burst kann Abbildung 2.9 entnommen werden. Am Anfang und Ende eines Burst gibt es einen Bereich, in dem keine Daten gesendet oder empfangen werden (guard time), womit Überschneidungen mit vorherigen oder nachfolgenden Bursts verhindert werden können. Diese Überschneidung könnten durch Verzögerungen, Echos oder Reflektionen entstehen. Ein Teilnehmer bewegt sich während eines Gespräches, so dass sich der Abstand zur BTS stetig verändert. Funkwellen bewegen sich mit Lichtgeschwindigkeit, so dass die Daten eines weiter entfernten Mobilfunkteilnehmers erst später eintreffen als die Daten eines Teilnehmers, der sich näher an einer Basisstation befindet. Um so Überschneidungen zu vermeiden, sind diese guard times notwendig. Für die Nutzdatenübertragung stehen zwei Felder à 57 Bits = 114 Bits zur Verfügung. Die Trainings-Sequence und die

[8] GSM-Rahmenstruktur, *http://de.wikipedia.org/wiki/Datei:Gsm-rahmenstruktur.png* [Online; letzter Aufruf 28.09.2009]

Tail-Bits gewährleisten die Korrektheit der Signalübertragung durch bitgenaue Synchronisation der Bursts. Die zwei Flag-Bits vor und hinter der Training-Sequence geben an, ob es sich um Nutzdaten- oder Signalisierungsbits handelt. Eine Übersicht der verschiedenen Bursts ist in Tabelle 2.4 und Abbildung 2.10 zusammengestellt. Der physikalische Kanal setzt sich aus einem

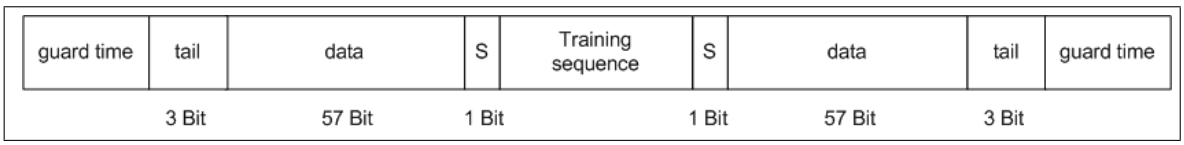

Abbildung 2.9: Aufbau eines Normal Burst

Name	Aufgabe
Normal Burst	Übertragung von Nutzinformationen wie Daten oder Sprache
Frequenzkorrekturburst	Korrektur und Anpassung (nur Downlink), Übertragung in TDMA-Zeitschlitz 0 (dadurch Slotnummerierung erkennbar), besteht nur aus 0-Bits, ermöglicht dem Telefon einen Synchronisationburst zu finden und demodulieren
Synchrnonisationsburst	Zeitsynchronisation der Mobilstation (nur Downlink), Übertragung in TDMA-Zeitschlitz 0 im SCH
Dummy Burst	wird auf jeder Frequenz gesendet, auf der gerade kein anderer Burst gesendet wird (nur Downlink), füllt nicht benutzte Timeslots (z.B. muss im BCCH laut Spezifikation in jedem Timeslot mit der gleichen Leistung gesendet werden)
Access Burst	nur von Mobilstation an BTS gesendet (nur im Uplink im RACH), enthält eine Zugriffs-Burst, lange Trainingssequenz und eine lange Schutzzeit, um der Empfangsstation die erfolgreiche Demodulation zu erleichtern (erhöht Erfolgswahrscheinlichkeit)

Tabelle 2.4: Übersicht über die verschiedenen Bursts

kontinuierlichen Datenstrom zusammen. Dieser besteht aus den Datenpaketen zusammengehöriger Zeitschlitze. Je nach gewünschter Funktion werden den Daten des physikalischen Kanals logische Kanäle zugeordnet. Es findet eine Unterscheidung zwischen Verkehrskanälen, sogenannten Traffic Channels (TCH), und Signalisierungskanälen, sogenannten Control Channels (CCH), statt. Die Control Channels sind noch aufgeteilt in Broadcast Control Channel (BCCH), Dedicated Control Channel (DCCH) und Common Control Channel (CCCH). Diese Kanäle und deren jeweilige Funktion werden in den folgenden Abschnitten genauer erläutert. Eine hierarchische Übersicht gibt Abbildung 2.11.

Übersicht und Funktion der logischen Kanäle

TCH (Traffic Channel): Nutz- bzw. Verkehrskanal, bei voller Auslastung in allgemeinen GSM-Systemen 992 (124 * 8) Fullrate Kanäle, zwei unterschiedlichen Kanalarten [2]:

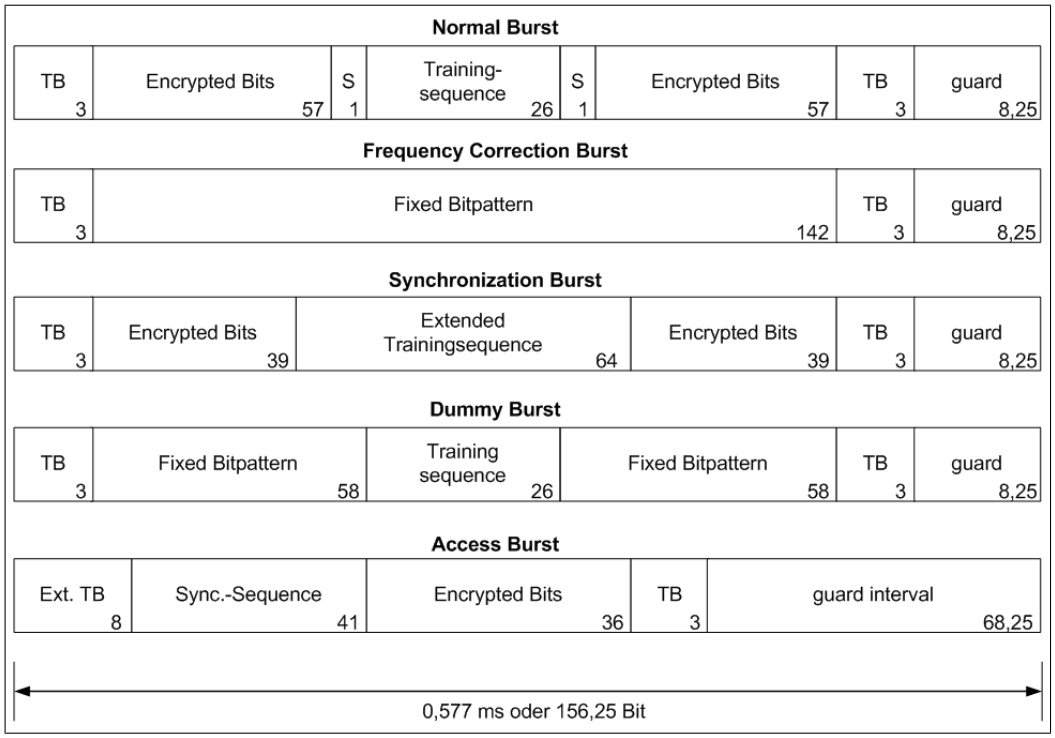

Abbildung 2.10: Übersicht über die verschiedenen Bursts (Vorlage nach [25])

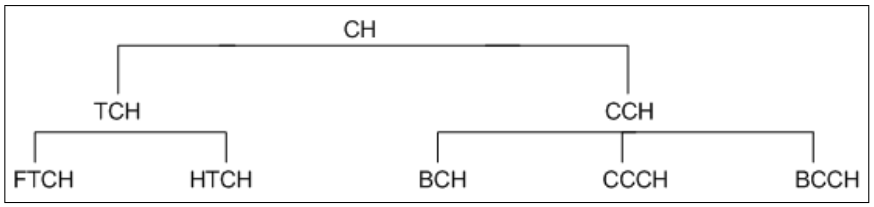

Abbildung 2.11: Hierarchische Übersicht der logischen Kanäle

- Fullrate: Sprachübertragung mit 22,8 kbits/s (theoretisch) bzw. 13,0 kbits/s (in der Praxis durch Fehlerkorrekturmechanismen); Datenübertragung mit 9,6 , 4,8 , 2,4 kbits/s
- Halfrate: Sprachübertragung mit 11,4 kbits/s (theoretisch) bzw. 5,6 kbits/s (in der Praxis); Datenübertragung mit 4,8 , 2,4 kbits/s

CCH (Control Channel): Steuer- bzw. Signalisierungskanal, Signalisierung und Transport von Steuerinformationen, zwei Standard-Signalisierungsarten:

- Digital Subscriber Signaling System No.1 (DSS1)
- Common Channel Signalling No. 7(CCS7)

BCH (Broadcast Channel): undirektional, 1:n, Downlink, von allen Teilnehmern abgehört, Systeminformationen über die Zelle (Trägerfrequenz, Frequency Hopping (FH), VAD-Fähigkeit[9])

SCH (Synchronization Channel): Synchronisationsburst zur Bit- und Rahmensynchronisation, BTS-Identifizierung.

[9] Möglichkeit zur Feststellung, ob gesprochen wird oder gerade eine Gesprächspause ist

FCCH (Frequency Correction Channel): undirektional, 1:n, Downlink; BTS sendet Frequenzburst mit voller Leistung und ohne feste Frequenz. Diese Information wird von MS zur Frequenzkorrektur und zur Auswahl der Zelle des BTS mit der „besten“ Empfangsfrequenz benötigt. Beim Einschalten der MS wird dieser Kanal zuerst gesucht (Kalibrierung). Der Kanal dient außerdem dazu, den Anfang eines 51-Multiframes (siehe Kapitel 2.4.3) zu finden.

BCCH (Broadcast Control Channel): undirektional, 1:n, Downlink, ständig gesendet, Systeminformationen (Location Area Code (LAC), Zelle (Cell-ID), Mobile Country Code (MCC), Mobile Network Code (MNC), Frequenzen, Optionen) zur Orientierung eines mobilen Endgerätes innerhalb einer Zelle

CCCH (Common Control Channel): zur Verbindungsaufnahme

PCH (Paging Channel): undirektional, Downlink, Ziel: Rufkanal (wird benötigt, falls ein Anruf für eine Mobilstation in einer Zelle vorliegt)

RACH (Random Access Control Channel): undirektional, Uplink, gemeinsamer Kanal (Aloha-Zugriff), Ziel: Allokation eines Signalisierungskanals für Gesprächsaufbau oder Lokation-Aktualisierung (z.B. SDCCH); gesendet wird ein Access Burst stets in Zeitschlitz 0 (kann auch Antwort auf eine PCH-Nachricht sein); um Blockierungen zu vermeiden, werden Kontrollinformationen auf BCCH (hier PCH) gesendet.

AGCH (Access Grant Channel): undirektional, Downlink, Ziel: SDCCH-Zuweisung, um Antwort auf ein RACH zu geben

DCCH (Dedicated Channel): 1:1 Verbindung, bidirektional, eigentlicher Steuerkanal (z.B. Synchronisation der Verschlüsselung)

SDCCH (Stand-Alone DCCH): dedizierter Einzelsteuerkanal, bidirektionaler Kanal mit 9,2 kbits/s, Zuweisung eines exklusiven TCH; Signalisierungsdaten: SMS senden/empfangen, Location Update; Infos über: Zielnetz (Teilnehmernummer), Verbindungsaufbau, verwendeten TCH, Anrufe in Abwesenheit, SMS-Dienste

SACCH (Slow Associated Control Channel): Steuerkanal zur Begleitung eines TCH (oder SDCCH), Informationen über Sendeleistungsanpassungen, Hintergrundgeräusche, Rahmen-Ausrichtung, Timing Advance, Kontrolldaten für Handover, SACCH/T (traffic) 184 bit alle 480 ms, SACCH/C (control) 184 Bit alle 480,77 ms

FACCH (Fast Associated Control Channel): ACCH mit „frame stealing“ vom zugeordneten TCH, diese „gestohlenen“ Rahmen werden unter anderem für dringende Signalisierungsnachrichten und beim Handover benutzt (Handover Kommando); Sprachübertragung in einzelnen Blöcken, Datenübertragung in mehreren Blöcken möglich

Kombination der logischen Kanäle

GSM spezifiziert ein hochentwickeltes Multiplexing-Schema, welches mehrere Hierarchien von Rahmen definiert. Die logischen Kanäle können somit nicht beliebig, sondern nur in bestimmter

Weise miteinander kombiniert werden. Diese Kombinationen werden dann auf den physikalischen Kanal abgebildet. Es gibt folgende Kombinationsmöglichkeiten:

- TCH + SACCH + FACCH (Half- oder Full-Duplex)
- 1x BCCH + mehrere CCCHs
- 8x SDCCHs
- FCCH + SCH + CCCH + BCCH
- 4x SDCCH + 1x BCCH + CCCH + SCH + FCCH

Es werden nicht beliebig lange Folgen von Rahmen übertragen, sondern es erfolgt eine Gruppierung. Die ersten beiden Timeslots einer Trägerfrequenz werden üblicherweise für allgemeine Signalisierungskanäle genuzt, die restlichen sechs Timeslots können für unabhänige Nutzkanäle oder GPRS verwendet werden.

Die Verkehrskanäle für Sprach- und Datenübertragung und ihre dazugehörigen FACCH und SACCH werden zu 26 aufeinanderfolgenden TDMA Rahmen zu 26er-Multiframes zusammengefasst. Ein Multiframe hat folglich die Länge von 120 ms (8 Zeitschlitze x 576,9 µs x 26). Signalisierungskanäle werden zu einem 51er-Multiframe der Länge 235 ms (8 Zeitschlitze x 576,9 µs x 51)) zusammengefasst. In derartig sich ständig wiederholenden Multiframes ist genau festgelegt, in welchem Burst von Timeslot 0 und 1 welche logischen Kanäle übertragen werden.
Laut GSM-Rahmenspezifikation können beide Multiframes zu einem Superframe der Länge 6,12 s zusammengefasst werden [24]. 2048 Superframes werden wiederum zu einem Hyperframe zusammengefasst, für dessen Übertragung ungefähr 3,5 Stunden benötigt werden. Ein Überblick über die Hierarchie der Rahmenstruktur ist in Abbildung 2.12 zu sehen. Abbildung 2.13 zeigt die eindeutige Zuordnung der logischen Kanäle zu den physikalischen Kanälen.

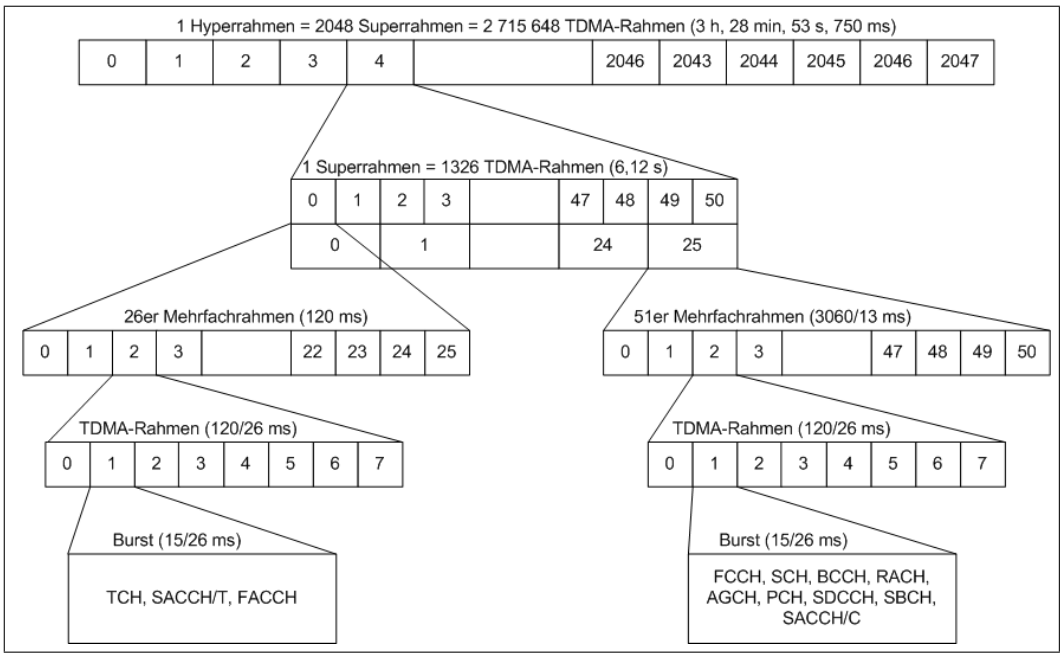

Abbildung 2.12: Hierarchie der Rahmenstruktur

FN	TS-0	TS-1
0	FCCH	SDCCH/0
1	SCH	SDCCH/0
2	BCCH	SDCCH/0
3	BCCH	SDCCH/0
4	BCCH	SDCCH/1
5	BCCH	SDCCH/1
6	AGCH/PCH	SDCCH/1
7	AGCH/PCH	SDCCH/1
8	AGCH/PCH	SDCCH/2
9	AGCH/PCH	SDCCH/2
10	FCCH	SDCCH/2
11	SCH	SDCCH/2
12	AGCH/PCH	SDCCH/3
13	AGCH/PCH	SDCCH/3
14	AGCH/PCH	SDCCH/3
15	AGCH/PCH	SDCCH/3
16	AGCH/PCH	SDCCH/4
17	AGCH/PCH	SDCCH/4
18	AGCH/PCH	SDCCH/4
19	AGCH/PCH	SDCCH/4
20	FCCH	SDCCH/5
21	SCH	SDCCH/5
22	SDCCH/0	SDCCH/5
23	SDCCH/0	SDCCH/5
24	SDCCH/0	SDCCH/6
25	SDCCH/0	SDCCH/6
26	SDCCH/1	SDCCH/6
27	SDCCH/1	SDCCH/6
28	SDCCH/1	SDCCH/7
29	SDCCH/1	SDCCH/7
30	FCCH	SDCCH/7
31	SCH	SDCCH/7
32	SDCCH/2	SDCCH/0
33	SDCCH/2	SDCCH/0
34	SDCCH/2	SDCCH/0
35	SDCCH/2	SDCCH/0
36	SDCCH/3	SDCCH/1
37	SDCCH/3	SDCCH/1
38	SDCCH/3	SDCCH/1
39	SDCCH/3	SDCCH/1
40	FCCH	SDCCH/2
41	SCH	SDCCH/2
42	SDCCH/0	SDCCH/2
43	SDCCH/0	SDCCH/2
44	SDCCH/0	SDCCH/3
45	SDCCH/0	SDCCH/3
46	SDCCH/1	SDCCH/3
47	SDCCH/1	SDCCH/3
48	SDCCH/1	free
49	SDCCH/1	free
50	free	free

FN	TS-2	…	TS-7
0	TCH		TCH
1	TCH		TCH
2	TCH		TCH
3	TCH		TCH
4	TCH		TCH
5	TCH		TCH
6	TCH		TCH
7	TCH		TCH
8	TCH		TCH
9	TCH		TCH
10	TCH		TCH
11	TCH		TCH
12	SACCH		SACCH
13	TCH		TCH
14	TCH		TCH
15	TCH		TCH
16	TCH		TCH
17	TCH		TCH
18	TCH		TCH
19	TCH		TCH
20	TCH		TCH
21	TCH		TCH
22	TCH		TCH
23	TCH		TCH
24	TCH		TCH
25	free		free
0	TCH		TCH
1	TCH		TCH
2	TCH		TCH
3	TCH		TCH
4	TCH		TCH
5	TCH		TCH
6	TCH		TCH
7	TCH		TCH
8	TCH		TCH
9	TCH		TCH
10	TCH		TCH
11	TCH		TCH
12	SACCH		SACCH
13	TCH		TCH
14	TCH		TCH
15	TCH		TCH
16	TCH		TCH
17	TCH		TCH
18	TCH		TCH
19	TCH		TCH
20	TCH		TCH
21	TCH		TCH
22	TCH		TCH
23	TCH		TCH
24	TCH		TCH
25	free		free

Abbildung 2.13: Nutzung der Timeslots im Downlink [28]

2.5 Das OSI-Referenzmodell und seine Bedeutung in GSM

Das OSI-Referenzmodell definiert sieben Schichten: Physical Layer, Data Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer und Application Layer. Tabelle 2.5 veranschaulicht die Unterschiede zwischen OSI- und GSM-Schichtenmodell. Diese beschränken sich auf die untersten drei Schichten.

OSI-Schicht	**GSM-Funktionen**
1 Pysical	Übertragung von Verkehrs- und Signalisierungsdaten, Modulation, Codierung der Kanäle (Kapitel 2.4.3)
2 Data Link	Verwaltung der Signalisierungsverbindungen
3 Network	Radio-Management, Mobilitäts-Management (Kapitel 2.6), Call-Management

Tabelle 2.5: Übersicht OSI- und GSM-Schichtenmodell

Abbildung 2.14 kann die Zuordnung der einzelnen Bestandteile der GSM-Architektur zu dem entsprechenden Layer des OSI-Schichtenmodells entnommen werden (**CM** Call Management, **MM** Mobility Management, **RR** Radio Resource Management, **BTSM** BTS Management, **LAPD** Link Access Procedure D-Channel, **BSSAP** Base Station Subsystem Application Part, **SCCP** Signalling Connection Control Part, **MTP** Message Transfer Part).[10]

Physical Layer (Schicht 1): Mittels eines Übertragungsmediums wie Kabel, Richtfunk oder Satellitenverbindung werden auf der Schicht 1 pysikalisch bitweise Informationen übertragen. Die Schicht 1 ist hardwarebezogen. Auf der Schicht 1 gibt es keine Unterscheidung zwischen Nutz- oder Kontrolldaten. Bei einer BTS beziehungsweise einem Mobiltelefon zählt auf der Air-Interface-Seite die gesamte Modulation und die HF-Einrichtung.

Data Link Layer (Schicht 2): Die Daten werden paketorientiert verarbeitet. Dies kann synchron oder asynchron vom Physical Layer geschehen. Es werden Methoden der Fehlererkennung und Korrektur eingesetzt. Der von der Schicht 2 vor dem Senden erzeugte Datenrahmen (Start- bzw. Stopmarkierung, Checksumme) kann nach dem Empfang auf Schicht 2 auf Fehler hin überprüft werden. So können festgestellte Übertragungsfehler eigenständig korrigiert werden. Sollte die Korrektur fehlschlagen, wird der Frame beim Sender neu angefordert. Auf dem Air-Interface bildet das modifizierte Link Access Protocol für den D-Kanal (LAPDm) gemeinsam mit Channel Coding und Burstformatierung die Schicht 2. Hier wird keine Frame Check Sequence benötigt, da die Datensicherung und Fehlererkennung über das Channel Coding erfolgt. Das A_{bis}-Interface nutzt ebenfalls das LAPD Protokoll. Die Schicht 2-Daten haben nur abschnittsweise Gültigkeit, da von Knoten zu Knoten das Data Link Layer Protokoll ein anderes sein kann.

Network Layer (Schicht 3): Diese Schicht übernimmt die Vermittlung der Daten. Dazu werden die Zieladresse und der Weg der Daten verarbeitet. Die Schicht-3-Informationen sind ebenso wie die Schicht-2-Informationen nur abschnittsweise gültig und müssen beim Wechsel des Knotens neu bewertet werden (geändertes Layer Protokoll). Zu den typischen Schicht-3-Einheiten gehören das Radio Resource Management (RR) zwischen Mobiltelefon, BTS, BSC und MSC. Bei Call Control (CC) und Mobility Management (MM) handelt es sich nicht um eine Layer 3 Funktionalität. Für CC und MM stellt RR nur den Transport-Container bereit. Die Daten für

[10] Vorlage nach: Überblick über die GSM Subsysteme, *http://www.embsys.de/gsm/Overview.html* [Online; letzter Aufruf 28.09.2009]

CC und MM werden von der BTS transparent übertragen.

Transport Layer (Schicht 4): Bei Schicht 4 wird eine End-to-End- bzw. eine Peer-to-Peer-Verbindung aufgebaut. Diese bietet Mechanismen, um die korrekte Reihenfolge der Daten zu garantieren (Sequenzierung). Liegen die Daten in der richtigen Reihenfolge vor, können die Daten an die übergeordnete Schicht weitergeleitet werden. Solche Maßnahmen sind notwendig, wenn sich die Gesamtdaten aus verschiedenen Teildaten zusammensetzen (Segmentierung). Die Schicht-4-Prozeduren laufen zwischen den Endpunkten einer Verbindung ab.

Session Layer (Schicht 5): In dieser Schicht werden die Kommunikationsprozesse synchronisiert und zueinander abgestimmt. Bei GSM gibt es in dieser Schicht die Unterscheidung zwischen Mobile Terminated Call (MTC), Mobile Originated Call (MOC) und Location Update (LUP). Auf diese Prozesse wird in den folgenden Abschnitten genauer eingegangen.

Presentation Layer (Schicht 6): Schicht 6 bestimmt das Datenformat bzw. den Datentyp sowie die Aufbereitung der Daten vor der Übergabe an die Schicht 7. Zur Aufbereitung der Daten zählen das Komprimieren und das Dekomprimieren. Die ASN.1 Notation gehört zur Schicht 6, es ist eine komprimierte Darstellung von Nachrichten. Wird eine ASN.1 Message aufbereitet, kann sie durch einen Protokollmonitor angezeigt werden (Application Layer).

Application Layer (Schicht 7): Der Application Layer ist die Schnittstelle zur Anwendung. Im Beispiel des Protokollmonitors werden die Daten in einer grafischen Oberfläche dargestellt [16].

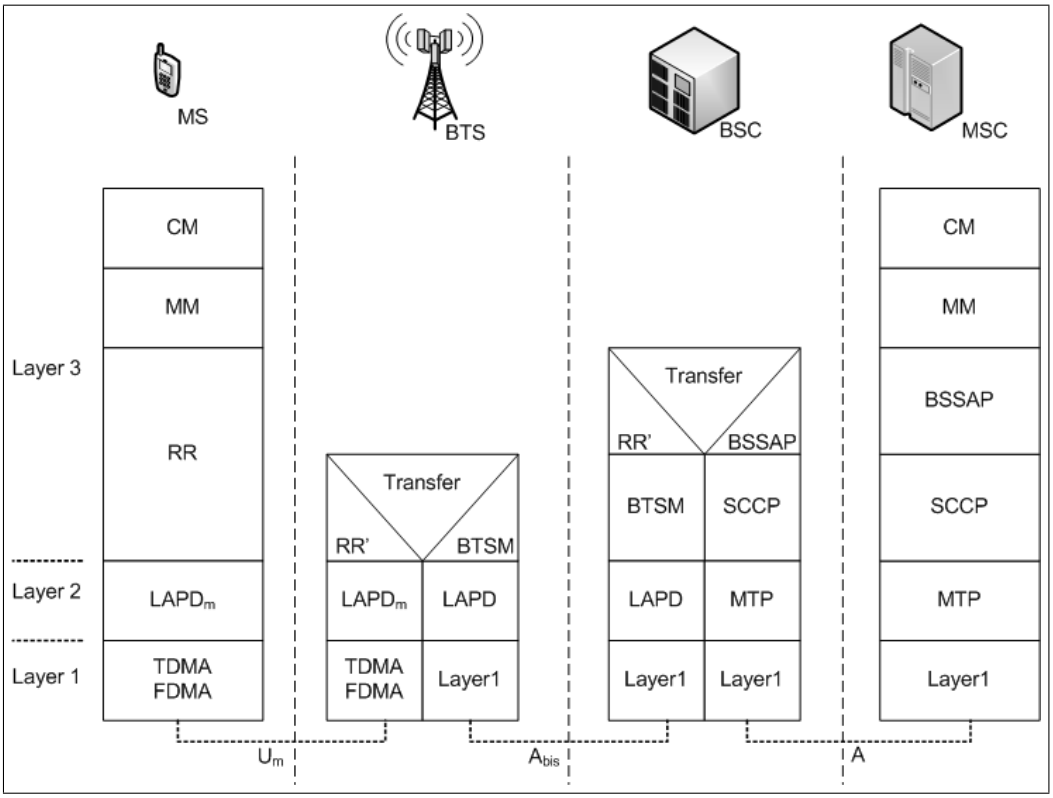

Abbildung 2.14: GSM Protokoll Architektur

2.6 Systemdienste

Zu den notwendigen Vorgängen, die die Mobilität der Mobilfunkteilnehmer gewährleisten gehören Einschalten und Einbuchen des Mobiltelefons, eingehende bzw. ausgehende Anrufe, Lokalisierung und dessen Aktualisierung und Handoverprozesse. Diese sollen in Kapitel 5.2 mit der Implementierung in OpenBTS verglichen werden und anhand der in Abschnitt 2.4.3 erklärten logischen Kanäle Unterschiede bzw. noch nicht implementierte Funktionen herausgearbeitet werden. Diese Vorgänge spielen sich alle auf Layer 5 des GSM Stacks ab.

Zunächst einmal ist es sinnvoll, zwischen verschiedenen Betriebszuständen des Mobiltelefons zu unterscheiden. Diese sind:

- **Detatched**: Mobiltelefon ist ausgeschaltet und abgemeldet.
- **Idle**: Mobiltelefon ist eingeschaltet, es ist kein Verkehskanal zugeordnet, Mobiltelefon nimmt Messungen vor und wartet auf Paging-Signale, regelmäßiges Location Update und SMS empfangen und versenden.
- **Dedicated**: Mobiltelefon hat Kanal zugewiesen bekommen, Handover wird gegebenenfalls durchgeführt.

2.6.1 Einschalten und Einbuchen des Mobiltelefons

Abbildung 2.15 beschreibt den Registrierungsprozess eines Mobiltelefons bei GSM. Hierzu werden folgende Schritte durchgeführt:

1. Auf der SIM-Karte ist die letzte verwendete Zelle (ARFCN) gespeichert. Das Mobiltelefon versucht sich mit dieser zu verbinden.
2. Falls dies scheitert, wird eine neue Suche ausgeführt und der stärkste Sender gesucht und gespeichert. Das Mobiltelefon synchronisiert sich mit der BTS.
3. Das Mobiltelefon liest die Systeminformationen (*sys_info*), die auf dem BCCH der aktuellen Funkzelle übertragen werden, um Voreinstellungen vornehmen zu können:
   - *sys_info1*: Frequenzkanäle und Zugriffsrechte der Zelle
   - *sys_info2*: Frequenzen und Zugriffsrechte der Nachbarzellen und deren NCC
   - *sys_info3*: Zellidendität, LAC und Konfiguration des CCCH
   - *sys_info4*: Wiederholung von *sys_info1-3*
4. Auf stärkster Frequenz FCCH suchen.
5. Falls ein FCCH gefunden wird, folgt sofort ein SCH zur Ermittlung der weiteren Parameter.
6. Das Mobiltelefon kann sich für eine Zelle entscheiden. Eine Liste, welche PLMNs erlaubt bzw. verboten (ohne Roaming Abkommen) sind, ist auf der SIM-Karte gespeichert.
7. Das Mobiltelefon möchte seinen Aufenthaltsort im VLR aktualisieren und sendet dazu ein Location Update Request über den SDCCH Kanal.
8. Es folgt eine vollständige Authentifizierung und das VLR generiert eine neue TMSI (der vollständige Authentfizierungsprozess kann Kapitel 2.3.2 entnommen werden).
9. Die generierte TMSI wird verschlüsselt über die BTS an das Mobiltelefon übertragen.

10. Die BTS sendet ein Location Update Accept an das Mobiltelefon und der verwendete SDCCH Kanal kann geschlossen werden.

11. Der Erhalt der TMSI wird durch das Mobiltelefon mittels Acknowlege Nachricht bestätigt.

12. Das Mobiltelefon ist vollständig eingebucht und wartet auf dem CCCH Kanal auf Paging Requests (eingehende Anrufe, oder SMS-Nachrichten). Alternativ kann ein Mobilfunkteilnehmer natürlich auch einen ausgehenden Anruf durchführen.

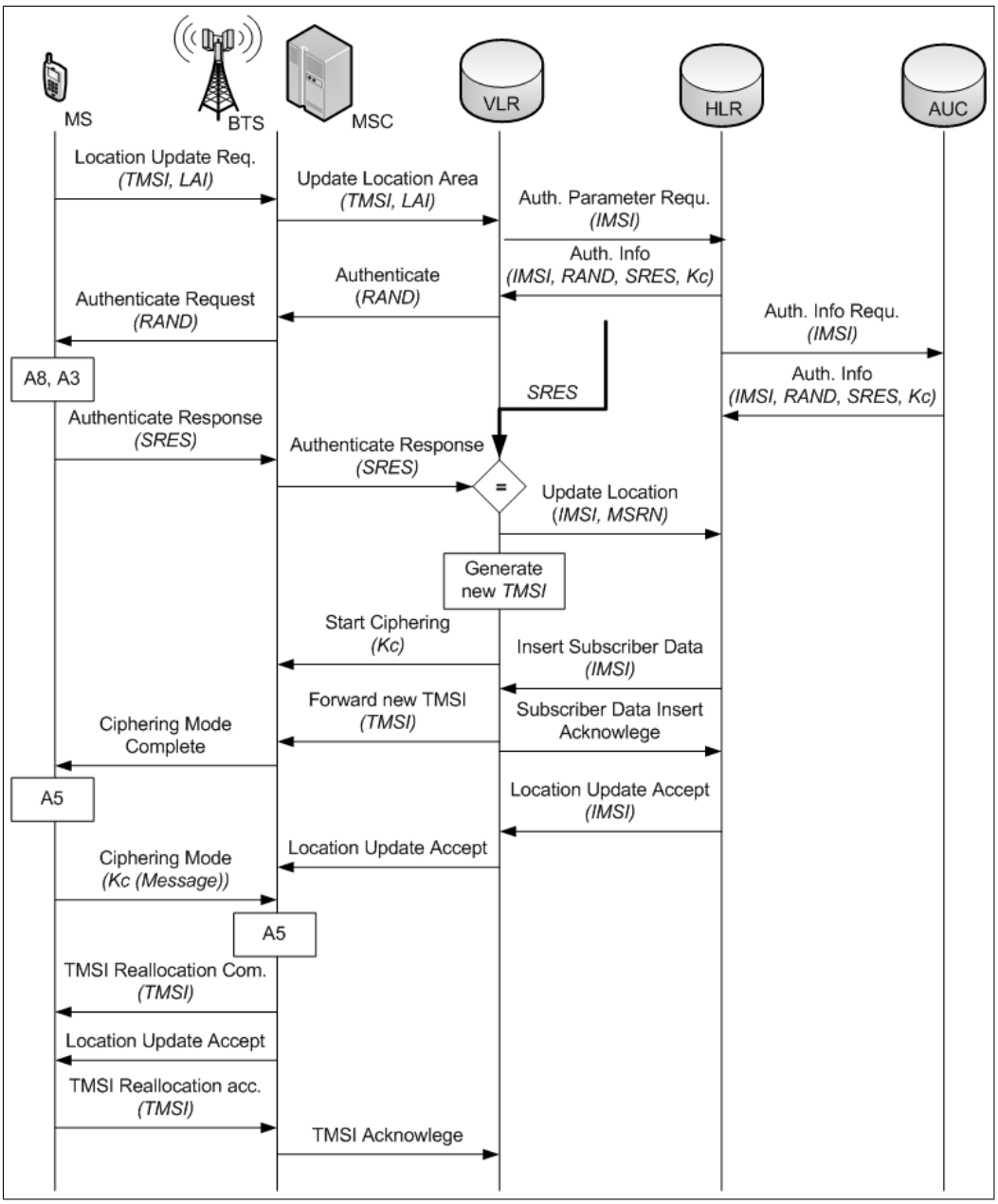

Abbildung 2.15: Registrierungsprozess beim Einschalten des Mobiltelefons (Vorlage nach [25])

Zum Abmelden eines Mobiltelefons wird eine kurze Nachricht gesendet.

2.6.2 Lokalisierung und Aktualisierung

Im Gegensatz zum Festnetz wechselt im Mobilfunk der Aufenthaltsort eines Teilnehmers möglicherweise ständig. Um nun ein Gespräch an einen Mobilfunkteilnehmer vermitteln zu können, muss dessen momentaner Aufenthaltsort bekannt sein. Zur regelmäßigen Aktualisierung fragt die entsprechende Base Station in regelmäßigen Zeitintervallen zwischen 6 Minuten und 25 Stunden in ihrer Zelle nach den Mobilstationen. Nach mehreren Fehlversuchen wird eine nicht erreichbare Mobilstation mit dem Status detatched im VLR markiert.

Es gibt im wesentlichen drei Fälle, die ein Location Update notwendig machen. Diese sind:

1. Einschalten eines Mobiltelefons
2. Regelmäßige Aktualisierung (Timer abgelaufen)
3. Normale Aktualisierung: LAI weicht von der der Mobilstation zugeteilten LAI ab

In Abbildung 2.16 ist der genaue Prozess des Location Updates zu sehen. Das initiale Location Update beim Einschalten des Mobiltelefons ist in Abbildung 2.15 dargestellt.
Mehrere Zellen sind zu einer Location Area zusammengefasst (meist ca. 20 Stück), so dass ein Location Update nicht bei jedem Zellwechsel von der Mobilstation durchgeführt werden muss [28]. Dies sorgt auf der einen Seite für eine geringere Signalisierungslast und einen geringeren Energieverbrauch des Mobiltelefons, auf der anderen Seite ist damit aber nur die aktuelle Location Area eines Mobilfunkteilnehmers bekannt und nicht die aktuelle Zelle. Mit Hilfe von Paging wird bei einem ankommenden Anruf oder einer ankommenden SMS der Teilnehmer in allen Zellen einer Location Area ausfindig gemacht. Mittels BCCH (siehe Abschnitt 2.4.3) informiert das Netzwerk alle Teilnehmer, zu welcher Location Area die aktuelle Zelle gehört. Es werden die Cell-ID und Location Area ID übertragen.

2.6.3 Eingehende Anrufe

Ein eingehendes Gespräch wird an die betreffende Basisstation durchgestellt. Diese Base Station versucht das Mobiltelefon zu erreichen. Im Idealfall antwortet das Mobiltelefon, und das Gespräch kann durchgestellt werden. Es kann aber auch passieren, dass das Mobiltelefon auf den Ruf der Basisstation nicht reagiert. Die Basisstation gibt dann nach einer bestimmten Wartezeit auf und meldet dem Anrufer, dass momentan kein Gespräch aufgebaut werden kann (z.B. wenn sich ein Mobilfunkteilnehmer in einem Funkloch befindet). Sollte das Mobiltelefon komplett abgeschaltet sein, kann dies im HLR vermerkt werden und somit dem Anrufer sofort mitgeteilt werden, dass der Gesprächsteilnehmer nicht erreichbar ist.

Ein eingehender Anruf bei einem Mobiltelefon wird als Mobile Terminated Call (MTC) bezeichnet. Folgende Schritte sind dafür notwendig [2]:

1. Es wird von sämtlichen BTS in der entsprechenden Location Area ein Paging-Signal geschickt.
2. Das Mobiltelefon empfängt dieses Paging-Signal, wenn
   - es eingeschaltet und erreichbar ist,
   - eine Ortsaktualisierung durchgeführt wurde (siehe 2.6.2),
   - im Zustand *idle* der PCH empfangen wird.
3. Das Mobiltelefon antwortet mit RACH.

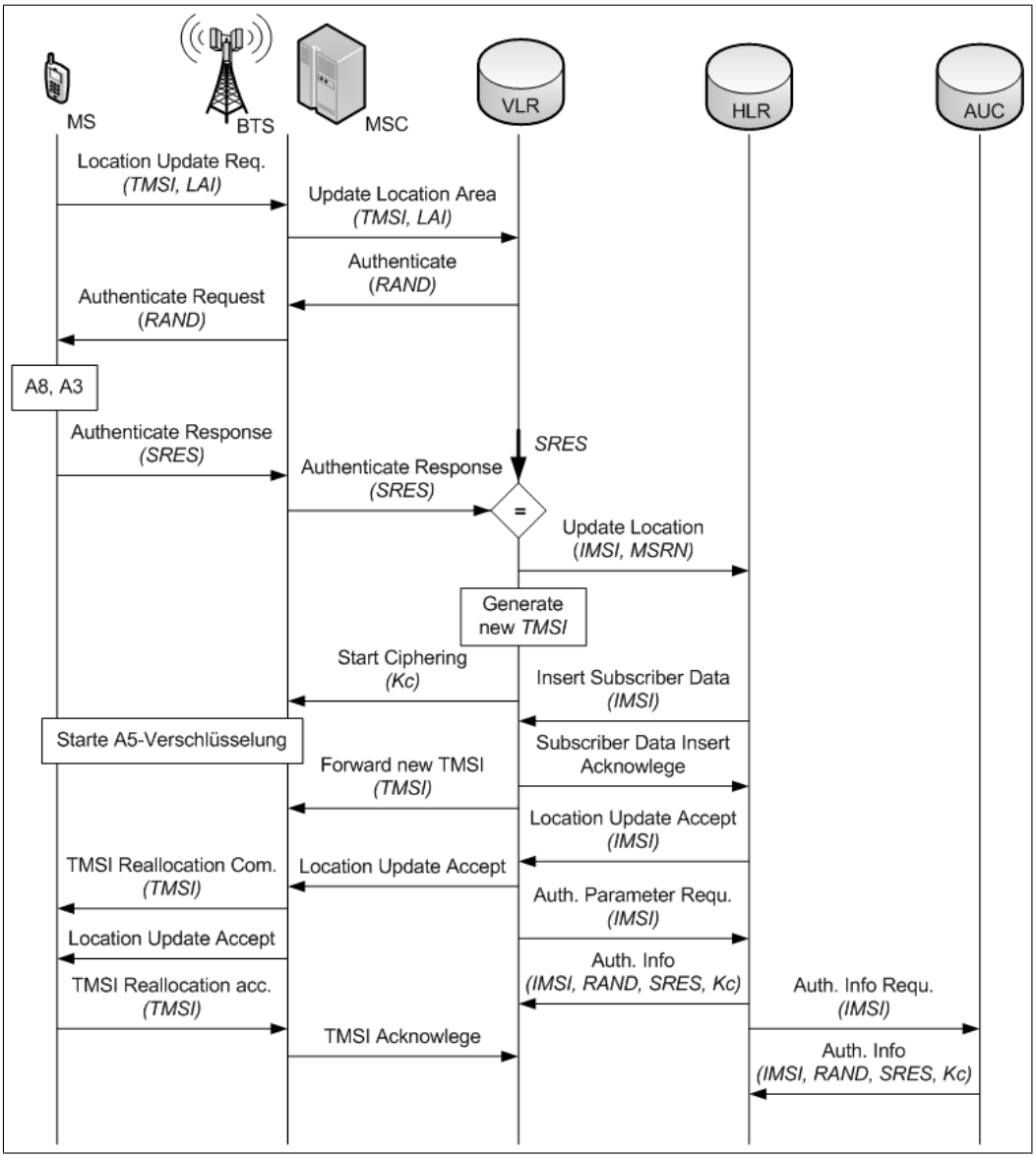

Abbildung 2.16: Location Update (Vorlage nach [25])

4. Das Mobiltelefon geht in den Zustand *dedicated*.

Grafisch ist der gesamte Vorgang in Abbildung 2.18 veranschaulicht und in Abbildung 2.17 chronologisch dargestellt. Im Detail werden bei einem eingehenden Anruf folgende Schritte durchgeführt, die Abbildung 2.17 zugeordnet werden können (siehe entsprechende Klammer):

1. Anruf eines GSM-Teilnehmers von einem Festnetztelefon (1)
2. Weiterleitung (IAM-Nachricht) zum GMSC (2)
3. Nachricht (Send Routing Information (SRI) Nachricht) über den Verbindungsaufbau an das zugehörige HLR (3)
4. Prüfung, ob der gewünschte Dienst abonniert ist und Anfrage der MSRN[11] vom VLR (4)

[11] Die Mobile Station Roaming Number oder auch Mobile Subscriber Roaming Number (MSRN) über-

5. Nach Erhalt der MSRN ermittelt das VLR anhand der IMSI (mittels MSISDN Nummer) das zuständige MSC (5)
6. HLR gibt MSRN an GMSC zurück (6)
7. Anrufweiterleitung zum zuständigen MSC (7)
8. Statusabfrage der MS (ab jetzt nur noch das MSC für alle weiteren Schritte zuständig) (8,9)
9. Wenn die MS verfügbar ist, Ruf der MS mittels Paging (Paging Nachricht an BSC, und dieser wiederum leitet diese an alle Zellen, die er verwaltet (über PCH) weiter) (10,11,12)
10. MS antwortet (13,14)
11. Sicherheitsüberprüfung: Authentifizierung und Verschlüsselung (15,16)
12. Verbindungsaufbau: MS bestätigt mit einer Call Confirmed Nachricht, MSC beantragt bei dem BSC einen TCH (17,18)

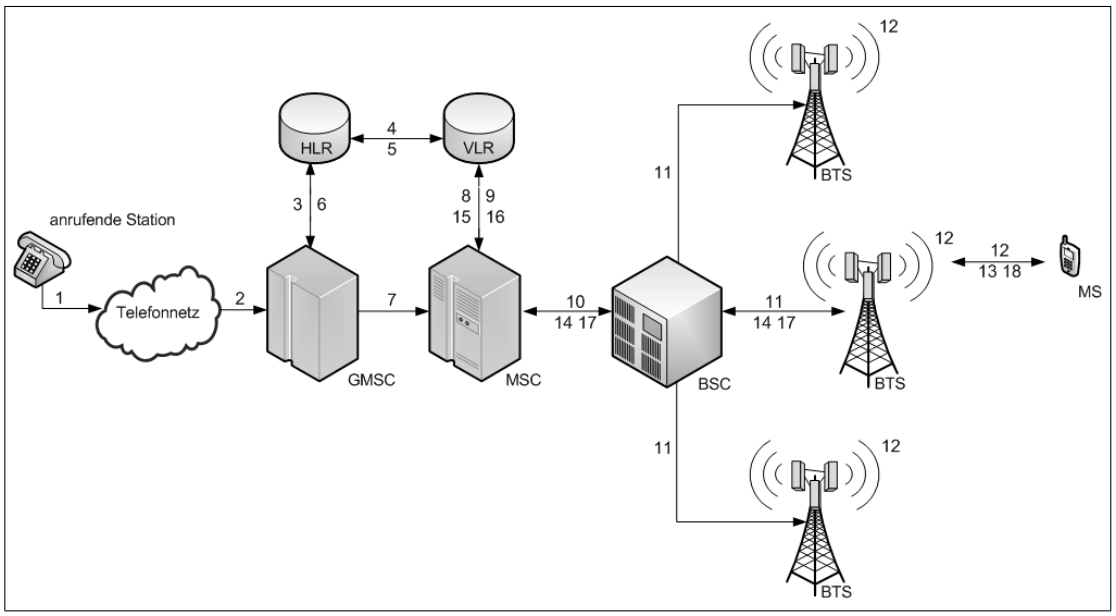

Abbildung 2.17: Ablauf eines Mobile Terminated Calls

Zum Beenden des Gesprächs schickt einer der zwei Gesprächspartner eine spezielle Disconnect-Nachricht. Der bestehende Sprachkanal zum jeweiligen Endgerät wird abgebaut und eine ISUP Release Complete Nachricht gesendet. Die Verbindung ist anschließend vollständig beendet.

2.6.4 Ausgehende Anrufe

Für ein ausgehendes Gespräch (Mobil Originated Call) kontaktiert ein Mobiltelefon die Basisstation. Sofern diese einen Gesprächsaufbau genehmigt, wird das Gespräch über die Hintergrundinfrastruktur an andere Zellen oder das Festnetz weitervermittelt. Die Basisstation kann aber auch

mittelt in Mobilfunknetzen die MSISDN (Mobile Subscriber ISDN Number) über das Home Location Register (HLR) zum Visitor Location Register (VLR). Dadurch wird beim Roaming das Auffinden des Teilnehmers in fremden Netzen ermöglicht. Quelle: Mobile Station Roaming Number, *http://de.wikipedia.org/wiki/Mobile_Station_Roaming_Number* [Online; letzter Aufruf 28.09.2009]

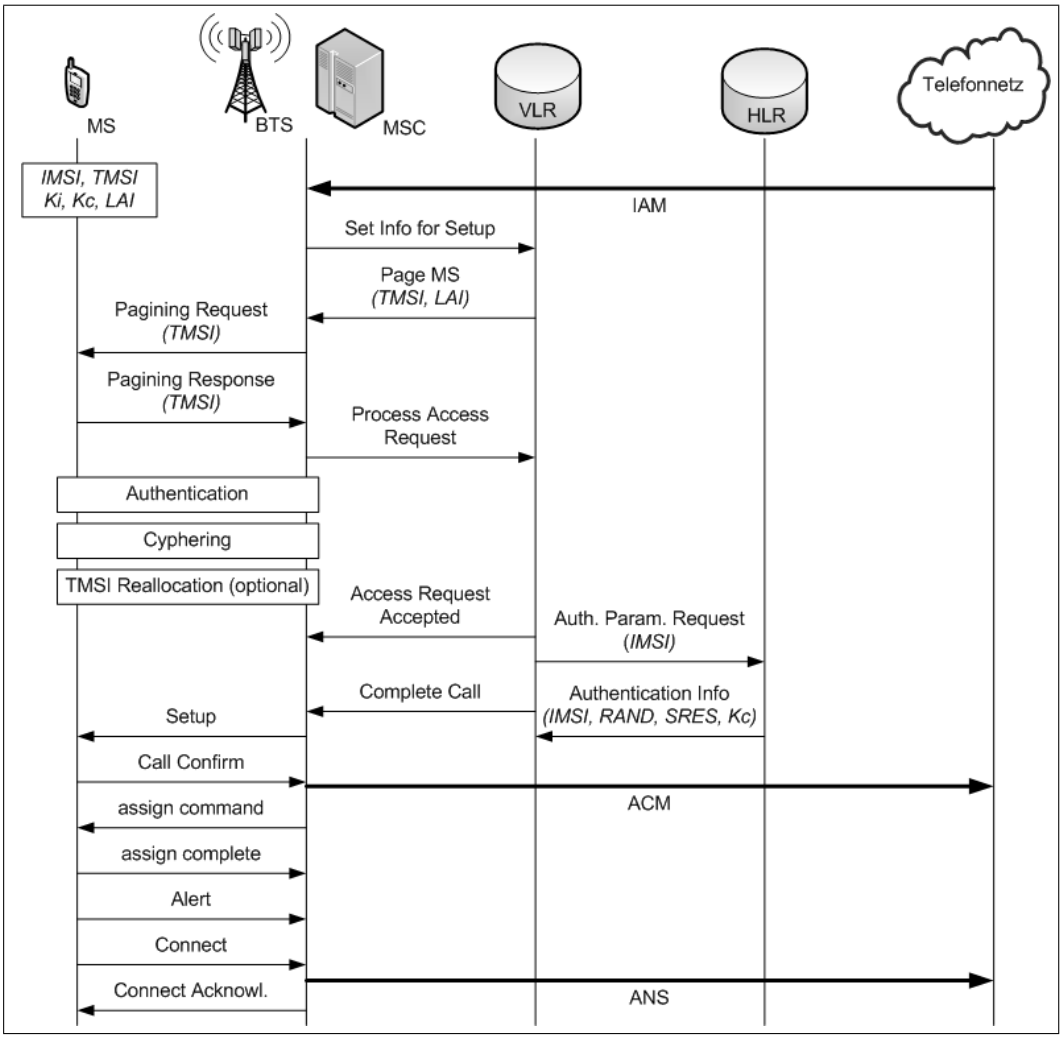

Abbildung 2.18: Eingehender Anruf in GSM (Vorlage nach [25])

einen Gesprächsaufbau verweigern, sofern z.B. alle zur Verfügung stehenden Kapazitäten erschöpft sind oder eine Störung vorliegt.

Der Vorgang eines Gesprächaufbaus ist in Abbildung 2.19 dargestellt und einem eingehenden Anruf recht ähnlich (siehe Abschnitt 2.6.3). Abbildung 2.20 zeigt die Zuweisung eines Sprachkanals. Dies geschieht wie folgt:

1. MSC fordert bei der BSC einen Sprachkanal an.
2. MSC und Mobilstation verständigen sich über SDCCH Kanal für den Aufbau einer Sprachverbindung (Signalisierung).
3. MSC schickt Assignment Request an BSC.
4. BSC überprüft, ob ein freier TCH Slot verfügbar ist und aktiviert diesen in der BTS.
5. Mobilstation wird über SCCH informiert, dass ein TCH verfügbar ist.
6. Mobilstation wechselt auf entsprechendne TCH und FCCH und sendet SABM Frame zur BTS.

7. BTS bestätigt mit einem UA Frame.
8. Mobilstation sendet ein Assignment Complete an BSC.
9. BSC leitet diese Assignment Complete Nachricht an MSC weiter.

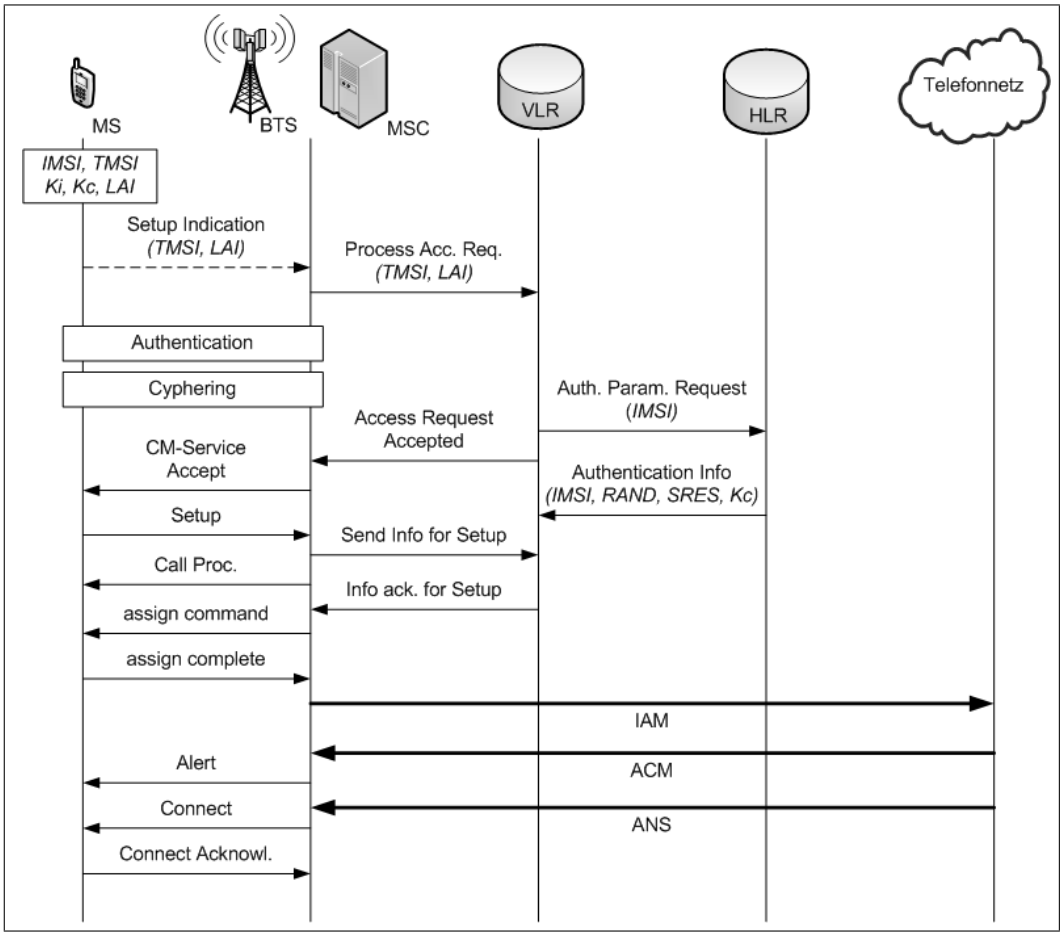

Abbildung 2.19: Ausgehender Anruf in GSM

2.6.5 Handover

Beim Handover handelt es sich um einen vom Netz angestoßenen Zellwechsel während eines laufenden Gesprächs oder einer bestehenden Datenverbindung. Mögliche Gründe hierfür sind die Qualität der Funkverbindung, die Verkehrslast der Zelle oder auch das Fortbewegen eines Mobilfunkteilnehmers aus dem Empfangsbereich einer bis dato genutzten BTS. Um eine Überlastung zu vermeiden, kann so ein Gespräch an eine weiter entfernte Zelle übergeben werden. Man spricht hierbei von einem Inter-Cell-Handover. Im Gegensatz dazu findet bei dem sogenannten Intra-Cell-Handover lediglich ein Kanalwechsel innerhalb einer Zelle statt. Dies kann beispielsweise durch eine sich verändernde Kanalqualität notwendig sein. Hierbei handelt es sich um „load balancing“. Die Mobilstation und das BTS führen regelmäßige Messungen durch, um so rechtzeitig ein Handover veranlassen zu können. Es werden zudem von der Mobilstation bestimmte Kanäle der Nachbarzellen mit einbezogen. Zu unterscheiden sind vier verschiedene Handover-Arten:

1. intra-cell Handover: der BSC führt einen Frequenzwechsel durch, beispielsweise aufgrund einer Frequenzstörung

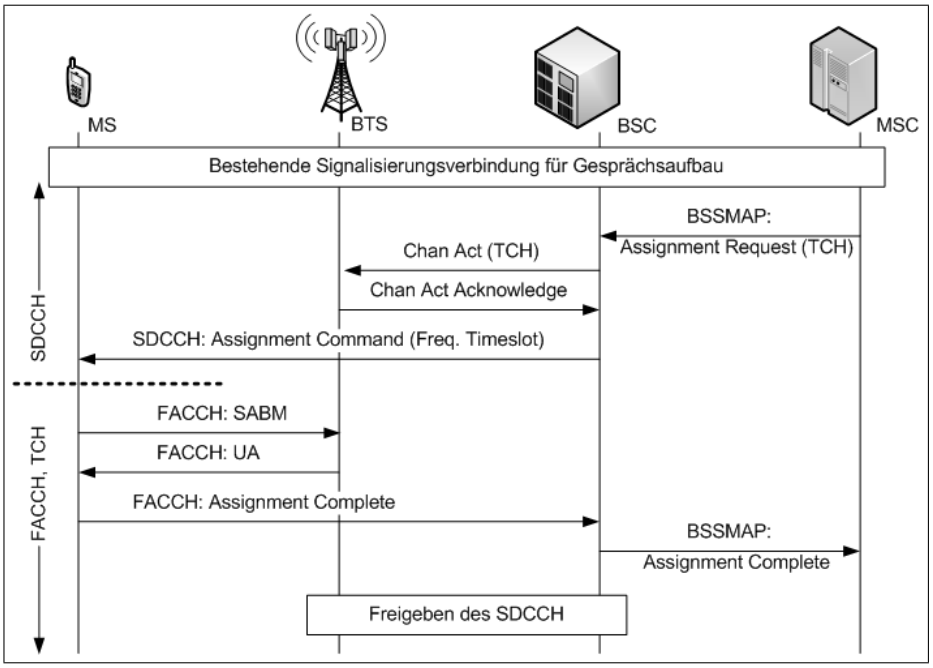

Abbildung 2.20: Ausgehender Anruf bzw. Aufbau eines Sprachkanals (TCH) (Vorlage nach [28])

2. inter-cell Handover: das Mobiltelefon wechselt die Zelle, bleibt aber im Zuständigkeitsbereich des bis dato verwendeten BSCs
3. inter-BSC Handover: das Mobiltelefon wechselt die Zelle und den Zuständigkeitsbereich des bis dato zuständigen BSCs
4. inter-MSC handover: das Mobiltelefon wechselt die Zelle und den Zuständigkeitsbereich des bis dato zuständigen MSCs

Abbildung 2.21 illustriert das typische Verhalten des empfangenen Signallevels, wenn das Mobiltelefon sich von einer BTS zu einer benachbarten BTS bewegt. Diese Handover-Entscheidungen beruhen auf Durchschnittswerten.

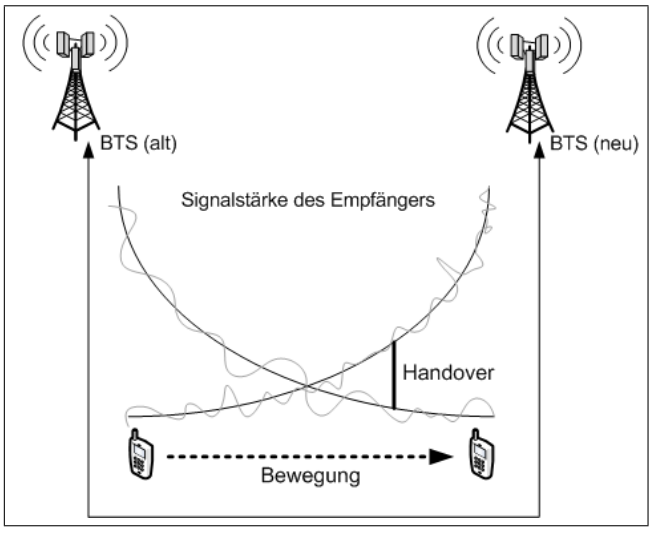

Abbildung 2.21: Handover-Entscheidung

3 Software

Zum Betrieb einer GSM-Basisstation im hier gezeigten Setup ist neben der entsprechenden Hardware auch Software notwendig. Folgende frei verfügbaren Softwarepakete werden in den weiteren Abschnitten dieses Kapitels näher betrachtet:

- GNU Radio (in Abschnitt 3.1)
- OpenBTS (in Abschnitt 3.2)
- Asterisk (in Abschnitt 3.3)

Mit Hilfe des OpenBTS-Projektes wird eine Open-Source Software zur Verfügung gestellt, mit der es gelingt, eine GSM-Basisstation zu betreiben. Die Asterisk PBX-Software ermöglicht mit GSM-fähigen Mobiltelefonen oder auch SIP-Telefonen Telefongespräche zu führen. Hardwareseitig basiert das ganze Projekt ausschließlich auf einem von der Firma Ettus entwickeltem Universal Software Radio Peripheral (USRP). Um auf den entsprechenden GSM-Frequenzbändern senden und empfangen zu können, werden zwei Boards, RFX900 für die GSM Frequenzbänder 850 und 900 MHz und RFX1800 für die GSM-Frequenzbänder 1800 und 1900 MHz, benötigt. Eine genauere Beschreibung der verschiedenen Hardwarekomponenten erfolgt in Kapitel 4. Abbildung 3.1 illustriert das Zusammenspiel zwischen Hardware und den in den folgenden Abschnitten genauer vorgestellten Softwarekomponenten GNU Radio, OpenBTS und Asterisk. Die Hardware und diese drei Softwarekomponenten bilden eine BTS (vgl. Abbildung 2.3 in Kapitel 2.3). Ein BSC ist noch nicht notwendig, da bisher mehrere OpenBTS Zellen noch nicht miteinander interagieren können (z.B. Handover). Ein MSC ist ebenfalls noch nicht vorhanden. Lediglich die Datenbanken VLR, HLR und AUC sind teilweise umgesetzt. OpenBTS kann ab Version 2.3 jedem Benutzer eine TMSI zuweisen (VLR). Dies ist der erste Schritt für ein zukünftiges Mobilitätsmanagement. Asterisk kann dabei eine Benutzer-Authentifizierung (AUC) übernehmen und hat darüber hinaus erste benutzerspezifische Daten wie IMSI und eine mögliche Rufnummer (HLR) gespeichert (siehe Abschnitt 3.3). Auch die nötige Konvertierung verschiedener Sprachcodecs (TRAU) und die Vermittlung in das Telefonnetz (GMSC) sind aktuell durch Asterisk realisiert. Auf den detaillierten Versuchsaufbau und die genauere Inbetriebnahme der Hard- und Software wird in Kapitel 5.1 eingegangen.

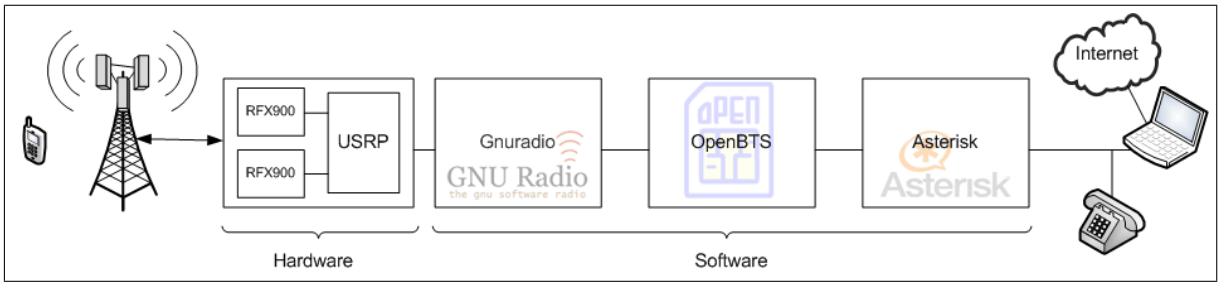

Abbildung 3.1: System-Überblick: Hard- und Software [20]

3.1 GNU Radio

Bei der frei verfügbaren Software GNU Radio (Logo siehe Abbildung 3.2) handelt es sich um eine Laufzeitumgebung, mit der in Kombination mit entsprechender Hardware Signale analysiert und verarbeitet werden können [13]. Ursprünglich war die Idee der Entwickler, ein „Software-Defined Radio“ (SDR) zu realisieren. Ziel eines solchen SDR ist es, möglichst die gesamte Signalverarbeitung eines Hochfrequenz-Senders oder -Empfängers mit Hilfe anpassbarer Hardware in Software abzubilden. Im engeren Sinn handelt es sich um ein Funktelekommunikationssystem, das eine software-konfigurierbare Plattform zur Modulation/Demodulation und Aufwärts- bzw. Abwärtsmischung eines Datensignals benutzt.[1] Es werden für Vorgänge wie Modulation bzw. Demodulation, Filtern oder auch Transformationen keine teure Spezialhardware benötigt, da diese Vorgänge somit alle softwareseitig umgesetzt werden. Das USRP (siehe Kapitel 4) ist eine der meist genutzten physikalischen Schnittstellen für GNU Radio und softwareseitig vollständig implementiert und ansteuerbar. Um eine direkte Signalverarbeitung durchführen zu können, wird entsprechende Hardware benötigt. Hierzu bietet sich vornehmlich das „Universal Software Radio Peripheral“ (USRP, genaueres siehe Kapitel 4) der Firma Ettus Research an. GNU Radio verarbeitet die vom USRP generierten Daten und bereitet diese auf.

Abbildung 3.2: GNU Radio, OpenBTS und Asterisk Logo

Zum Installationsumfang der GNU Radio Software gehören einige umfangreiche Beispiele, meist in der Programmiersprache Python geschrieben (Empfang von FM-Signalen, Erzeugung von USB-Signalen, verschiedene Bandfilter). Somit wird der Einstieg in die Entwicklung von eigenen Skripten mit GNU Radio deutlich vereinfacht. Besonders hervorzuheben ist hierbei das Skript *usrp_fft.py*. Hierbei handelt es sich um einen Spektrumsanalysator, der für verschiedene Versuche zur GSM-Analyse verwendet wird (hierzu siehe Kapitel 6.2). Die eigentliche Signalverarbeitung wurde von den Entwicklern in C++ implementiert, da ein möglichst genaues Timing sowie eine entsprechende Programmperformance gewährleistet sein müssen. Darüber hinaus kann die GNU Radio Software auch als Simulationssoftware benutzt werden. Mittels gespeicherter bzw. aufgezeichneter Daten kann eine nachträgliche Signalverarbeitung durchgeführt werden. Eine Installationsanleitung, die an ein paar Stellen von der auf der GNU Radio Webseite verfügbaren Anleitung abweicht, kann Anhang A entnommen werden.[2]

[1] Software Defined Radio, *http://de.wikipedia.org/wiki/Software_Defined_Radio* [Online; letzter Aufruf 29.09.2009]

[2] UbuntuInstall - GNU Radio, *http://gnuradio.org/trac/wiki/UbuntuInstall* [13] [Online; letzter Aufruf 29.09.2009]

3.2 OpenBTS

OpenBTS besteht im Wesentlichen aus drei Bestandteilen [14]:

1. OpenBTS GSM Stack: Es gibt eine spezielle Konfigurationsdatei, in der verschiedenste Parameter wie Frequenz, Ports, Identifikationsnummern genauer eingestellt werden können (Abschnitt 3.2.4)
2. Software „transceiver“: Dieser kommuniziert mit der USRP Hardware
3. Software Asterisk PBX

3.2.1 Vorrausetzungen

Um OpenBTS erfolgreich kompilieren und installieren zu können, müssen einige Voraussetzungen erfüllt sein. Es wird ein Treiber für das USRP benötigt. Dieser nennt sich „libusrp“ und kann mittels GNU Radio installiert werden (siehe Anhang A). Um lediglich den USRP Treiber zu kompilieren, kann folgender Befehl verwendet werden: *./configure –disable-all-components –enable-usrp –enable-omnithread –enable-mblock –enable-pmt*
Darüber hinaus werden noch zwei weitere Bibliotheken benötigt: Zum einen die SIP API (Application Programming Interface) *osip2* und zum anderen die oRTP API *ortp*.[3]

3.2.2 Installation

Nach erfolgreicher Einrichtung der *osip2* und *ortp* API kann die aktuelle OpenBTS Version aus dem GNU Radio SVN heruntergeladen und anschließend kompiliert und installiert werden:[4]

```
svn co http://gnuradio.org/svn/openbts/trunk/ openbts
cd openbts
./bootstrap
./configure
make
make install
```

Es stand darüber hinaus noch die bis dato aktuellste Version 2.4 zu Testzwecken zur Verfügung. Die Installation dieser Versionen erfolgt auf dieselbe Art und Weise.

3.2.3 Anwendung

Nach erfolgreicher Installation besteht die OpenBTS Software aus folgenden Komponenten:

- der eigenständigen Anwendung „transceiver“, die sich im Ordner */Transceiver/* befindet. Diese kommuniziert mittels eines UDP Interfaces über einen einstellbaren Port mit anderen OpenBTS Anwendungen.
- einem Set von Objekten, die die einzelnen Komponenten von OpenBTS bereitstellen:

[3] Erstere ist über den GNU FTP Server verfügbar: *http://ftp.gnu.org/gnu/osip/*. Es sollte eine Version >3.0 verwendet werden. Die oRTP API kann unter folgender URL: *http://savannah.inetbridge.net/linphone/ortp/sources/* heruntergeladen werden. Es kam die Version 1.4 zum Einsatz. Hierbei handelt es sich um eine Implementierung des Real-Time Transport Protocols.

[4] OpenBTS neuste Version (aktuell 2.4.1), *http://gnuradio.org/svn/openbts/trunk/* [13] [Online; letzter Aufruf 29.09.2009]

  - GSM spezifische Komponenten in */GSM/*
  - SIP spezifische Komponenten in */SIP/*
  - ein Kontrollinterface des Transceivers in */TRXManager/*
  - GSM/SIP Kontrollmechanismen in */Control/*
  - allgemeine Komponenten und Fehlerkorrektur-Komponenten in */CommonLibs/*

- einem OpenBTS Programm in */apps/* mit eigenständigem Konfigurationsfile: *OpenBTS.conf*, in dem sich verschiedene Parameter einstellen lassen (für eine genauere Beschreibung einzelner wichtiger Parameter siehe 3.2.4)

OpenBTS kann in der verwendeten Version 2.4 direkt mittels *./OpenBTS* im Ordner */apps/* gestartet werden.

3.2.4 Konfiguration OpenBTS

Die vollständige Konfigurationsdatei *OpenBTS.conf* kann auf der beiliegenden CD im Ordner */OpenBTS/OpenBTS-Konfigurationen* eingesehen werden. Die wohl wichtigsten Konfigurationsparameter sind im Abschnitt „GSM“ der Datei zu finden.

Durch die Variable *GSM.Band 900* kann zwischen GSM 900 und 1800 gewechselt werden. Mittels *GSM.ARFCN 29* wird die entsprechende Frequenz eingestellt (für mögliche GSM-Frequenzen siehe Tabelle 2.2 in Kapitel 2.2). ARFCAN 29 entspricht dabei 940.8 MHz (eine genauere Erklärung ist in Kapitel 6.2.1 beschrieben). Durch die Variablen *GSM.MCC 922*, *GSM.LAC 667*, *GSM.CI 10* und *GSM.ShortName „OpenBTS“* wird eine GSM-Zelle mit dem Namen OpenBTS gestartet. Ältere Mobiltelefone zeigen diese in der Netzliste als „901 55“ oder „Nor 55“ an. Die 55 entspricht dem festgelegten Mobile Network Code (*GSM.MNC 55*). Inwieweit diese Einstellungen genutzt werden können, um „Orginal“-Zellen komplett zu simulieren, wird in Kapitel 7.1 genauer betrachtet. Sämtliche sich daraus ergebenden Sicherheitsprobleme sind dort aufgelistet und genauer beschrieben. Vorgefertigte *OpenBTS.config* Dateien der einzelnen Netze wie Vodafone, T-Mobile, O2 und E-Plus sind ebenfalls im Ordner */OpenBTS/OpenBTS-Konfigurationen* der CD zu finden.
Die Parameter *Asterisk.IP 127.0.0.1* und *Asterisk.Port 5060* sind Einstellungen der IP-Adresse und Port-Nummer des Asterisk Servers. In den Versuchen wurde ein lokaler Asterisk Server verwendet, der allerdings über eine Anbindung an den Asterisk Server der Universität verfügte, so dass Gespräche ins deutsche Festnetz durchgeführt werden konnten.

3.3 Asterisk

Bei Asterisk handelt es sich um eine Open Source Softwarelösung, mit der eine VoIP-Telefonanlage realisiert werden kann. Die Software kann kostenlos von der Webseite der Entwickler heruntergeladen werden.[5] Abbildung 3.2 zeigt das dazugehörige Software Logo.

Asterisk wird verwendet, um Gespräche zwischen den einzelnen Teilnehmern zu vermitteln. Neben der Vermittlung von Mobiltelelfon-zu-Mobiltelefon Gesprächen können auch Gespräche mit beliebigen, zu dem entsprechenden Asterisk Server verbundenen Endgeräten erfolgen. So ist beispielsweise auch ein Gespräch zwischen einem Mobiltelefon und einem Voice-over-IP Telefon oder Voice-over-IP Client problemlos möglich. Jedes Mobiltelefon mit eingesetzter SIM-Karte muss

[5] Asterisk - The Open Source Telephony Project, *http://www.asterisk.org/* [Online; letzter Aufruf 15.10.2009]

vorher als Benutzer registriert werden. Der Benutzername ist hierbei die entsprechende IMSI. Diese ist weltweit einmalig und an die entsprechende SIM-Karte gekoppelt und darauf gespeichert und somit ideal für eine eindeutige Identifizierung geeignet.
Sämtliche Asterisk-Konfigurationsdateien befinden sich unter */etc/asterisk*. Um OpenBTS betreiben zu können, sind Änderungen in den Dateien *sip.conf* und *extensions.conf* notwendig. Um durchgeführte Änderungen im laufenden Betrieb von Asterisk zu übernehmen, muss der Befehl *sip reload* bzw. *extensions reload* ausgeführt werden. Eine Asterisk-Konsole zur Verwaltung kann mit folgendem Befehl gestartet werden: *asterisk -rcvvvv*

Folgende Konfiguration zeigt einen Benutzer, der in die entsprechende Konfigurationsdatei (*sip.conf*) hinzugefügt wurde. Die Authentifizierung erfolgt mittels der entsprechenden IMSI der verwendeten SIM-Karte (siehe hierzu Kapitel 5.2.1). Als Audio-Codec wird „gsm“ benutzt, da nur dieser Codec von OpenBTS unterstützt wird:

```
;Samsung Mobiltelefon mit Vodafone Prepaidkarte
[262027033983293]
canreinvite=no
type=friend
context=sip-external
allow=gsm
host=dynamic
```

Darüber hinaus fungiert der Asterisk Server als Trau. Die Software ist in der Lage zwischen beliebigen Audio-Codecs und Bitraten zu konvertieren. Anschließend muss nur noch eine entsprechende Wahlregel in der Datei *extensions.conf* erstellt werden. Hierbei kann für jeden Mobilfunkteilnehmer eine eigene Rufnummer vergeben werden. Für ein Mobiltelefon wurde beispielsweise die Nummer 2002 vergeben, unter der dieses erreichbar ist. Ein ausgehendes Gespräch wird über den Macro „dialSIP“ abgewickelt. Hierbei gibt es noch drei zusätzliche Möglichkeiten, und zwar, dass der Teilnehmer bereits telefoniert und somit die Leitung besetzt ist: *BUSY*, oder nicht antwortert: *NOANSWER* oder nicht erreichbar ist.
Auszug aus der Datei *extensions.conf*:

```
[macro - dialGSM]
exten => s ,1, Dial (SIP/${ ARG1 })
exten => s ,2, Goto (s-${ DIALSTATUS } ,1)
exten => s-CANCEL ,1, Hangup
exten => s-NOANSWER ,1, Hangup
exten => s-BUSY ,1, Busy (30)
exten => s-CONGESTION ,1, Congestion (30)
exten => s-CHANUNAVAIL ,1, playback (ss - noservice )


...
[sip-local]
; local extensions
exten => 2000,1,Macro(dialSIP,wiredPhone)
exten => 2001,1,Macro(dialSIP,softPhone)
; This is a simple mapping between extensions and IMSIs.
exten => 2002,1,Macro(dialGSM,262027033983293)
...
```

4 Hardware

Das von der Firma Ettus Research LLC hergestellte Universal Software Radio Peripheral (USRP) ermöglicht, die gesamte Signalverarbeitung mittels Software am Computer zu realisieren.[1] Hierzu werden verschiedene Hardwarekomponenten angeboten, mit deren Hilfe Frequenzen von 1 MHz bis zu 5.9 GHz empfangen und anschließend am Computer verarbeitet werden können.[2] Momentan existieren zwei verschiedene Modelle: USRP1 und USRP2. Für das OpenBTS-Projekt kann ausschließlich das USRP1 verwendet werden. Der gesamte Code für OpenBTS ist nur für den USRP1 programmiert worden und nicht zu dem USRP2 kompatibel. OpenBTS setzt zwei Empfangskarten voraus, die der USRP2 allerdings nicht aufnehmen kann, da er nur über einen Kartenslot verfügt. Diese zwei Empfangskarten werden von OpenBTS verwendet, um den Up- bzw. Downlink voneinander trennen zu können und mögliche Interferenz von vorneherein zu vermeiden. In Abbildung 4.1 ist das USRP1 samt Gehäuse zu sehen. Ist in folgenden Abschnitten oder Kapiteln von dem USRP die Rede, so ist stets das USRP1 gemeint. Mit verschiedenen Modulen können unterschiedliche Frequenzen verarbeitet werden. So kann damit beispielsweise Hörfunk empfangen und gesendet werden. Auch ist es möglich, in der WLAN-Frequenz zu arbeiten. In dieser Arbeit sind vor allem die GSM-Frequenzen von Bedeutung. Das USRP ist

Abbildung 4.1: USRP1 mit Gehäuse

eine modular aufgebaute Platine und kann mit Hilfe von zwei aufgesteckten Daughterboards (siehe Abschnitt 4.2) gleichzeitig senden und empfangen. Die Platine samt aufgesteckter Daughterboards ist in Abbildung 4.2 dargestellt. Auf ihr befindet sich für die Signalverarbeitung ein Altera Cyclone EP1C12Q240C8 FPGA (field programmable gate array, siehe Tabelle 4.1) und für die Weiterleitung der Daten an den Computer ein USB 2.0 Controller [15]. Das FPGA wird mit einem 64 MHz-Taktgeber betrieben und synchronisiert alle internen Komponenten. Das „high sample-rate processing“ (hohe Abtastrate) findet im FPGA und die „low sample-rate processing“

[1] Ettus Research LLC, *http://www.ettus.com/* [Online; letzter Aufruf 29.09.2009]

[2] Building Software Radio Systems - The USRP Product Family, *http://www.ettus.com/downloads/er_broch_trifold_v5b.pdf* [Online; letzter Aufruf 29.09.2009]

(niedrige Abtastrate) findet auf dem Computer statt. Das FPGA ermittelt diskrete Messwerte, die sofwareseitig interpretiert und verarbeitet werden (Signalverarbeitung).

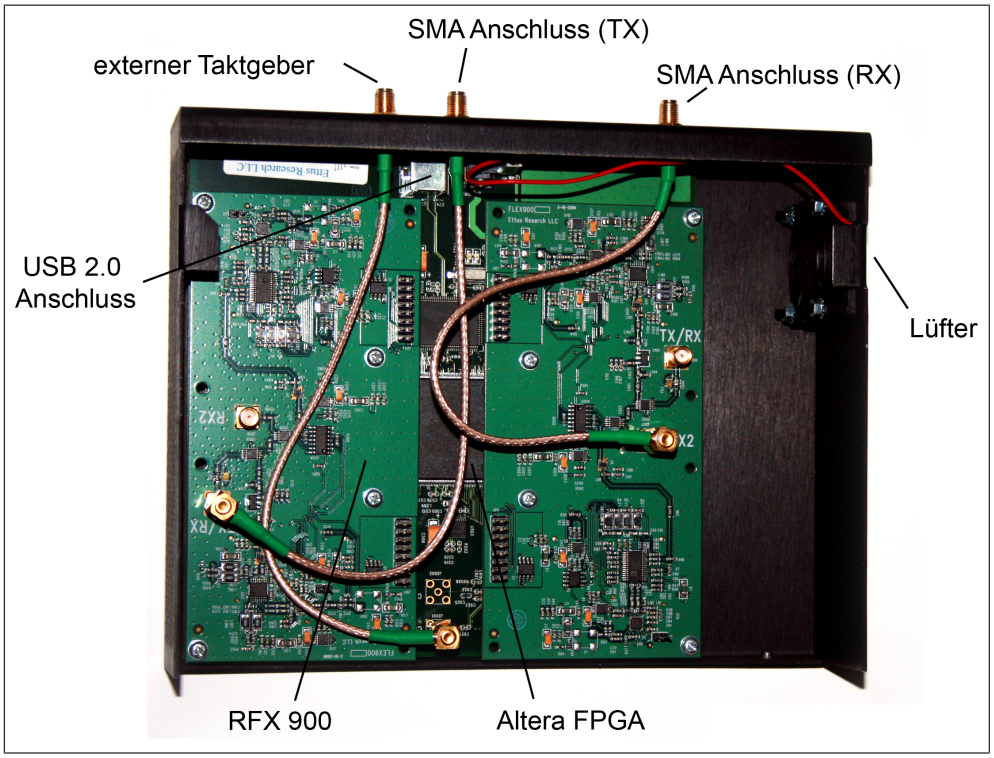

Abbildung 4.2: USRP Motherboard mit 2 RFX900 Daughterboards

LEs	12,060
M4k RAM blocks (128 x 36 Bits)	52
Total RAM Bits	239,616
PLLs	2
Maximum user I/O pins	173

Tabelle 4.1: Features des EP1C12 FPGAs

Eine detailliertere Auflistung der einzelnen Bestandteile kann den Plänen des Herstellers Ettus aus dem entsprechenden SVN-Repository entnommen werden oder auch auf der Radioware Webseite[3] [26]. Zusammenfassend noch eine Übersicht über die wichtigsten Komponenten des USRP:

- 4 64 MS/s 12-Bit analog-digital Konverter
- 4 128 MS/s 14-Bit digital-analog Konverter
- 1 Quarzoszillator
- 4 Erweiterungssteckplätze für 2-4 Daughterboards (2x TX, 2x RX)
- 1 High-speed USB 2.0 interface (480 Mbit/s)
- 1 Altera Cyclone FPGA

[3] GNU Radio, *http://gnuradio.org/trac/browser/usrp-hw/trunk* [13] [Online; letzter Aufruf 29.09.2009]

4.1 USRP2

Das USRP2 ist eine Erweiterung des USRP1 und bietet eine höhere Signalbandbreite. Das USRP2 wird – im Gegensatz zum USRP1, das mit einem USB-Anschluss angeschlossen wird – mittels Gigabit Ethernet mit dem PC verbunden. Dadurch können auch bestimmte Anwendungen, die eine höhere Signalbandbreite benötigen, realisiert werden. Dazu gehören unter anderem Anwendungen wie DVB-T, UMTS oder WLAN 802.11. Die Entwicklung der USRP2 Firmware befindet sich allerdings momentan noch in einer Testphase. Diese soll bis Ende 2009 abgeschlossen sein.

Wichtige Komponenten sind:

- 1 Xilinx Spartan 3-2000 FPGA (anstelle des Altera FPGA)
- 1 Gigabit Ethernet Interface (ersetzt USB 2.0 Schnittstelle)
- 2 100 Mega-Samples/s, 14 Bit, AD-Konverter
- 2 400 Mega-Samples/s, 16 Bit, DC-Konverter
- 1 SD-Karten Leser

4.2 Daughterboards

Es gibt auf dem Motherboard des USRP1 vier Steckplätze, in die zwei RX Basis Daughterboards und zwei TX Daughterboards oder zwei RFX Boards aufgesteckt werden können. Jeder Steckplatz hat Zugriff auf zwei der vier High-Speed AD- / DA-Konverter [15]. Es gibt verschiedene Daughterboards von 1 MHz bis 5,9 GHz. Abbildung 4.3 zeigt ein RFX900 und ein DBSRX-Board.[4]

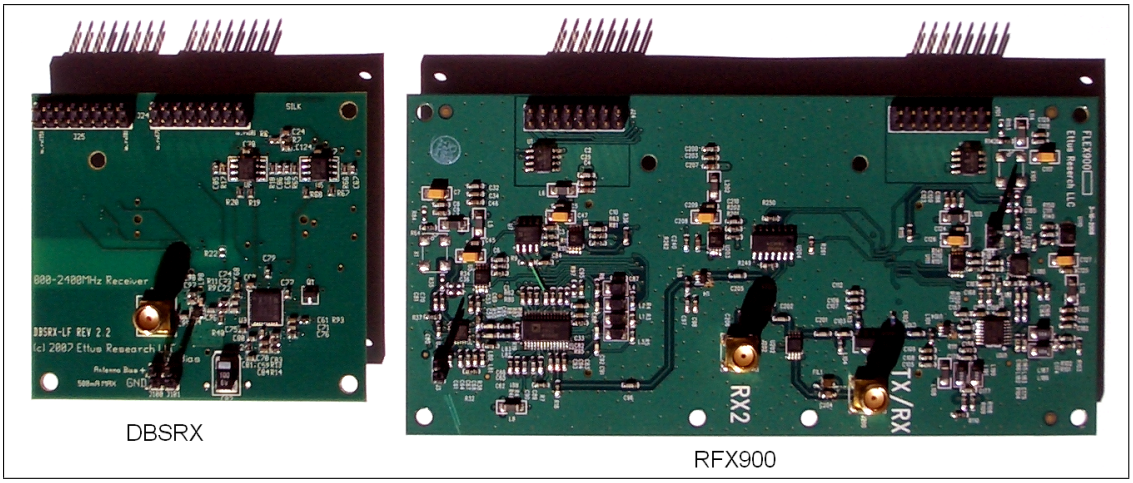

Abbildung 4.3: DBSRX und RFX900 Daughterboard

Durch die verschiedenen USRP-Daughterboards steht eine Vielzahl nutzbarer Frequenzen zur Verfügung. Diese können meist in Kombination mit der GNU Radio Software für die unterschiedlichsten Anwendungen verwendet werden. Für den Betrieb von OpenBTS werden zwei

[4] Eine genauere Übersicht und eine Bestellmöglichkeit sind auf der Webseite der Firma Ettus zu finden (*http://www.ettus.com/order*)

RFX900 (für den Betrieb im GSM 900 Band) beziehungsweise zwei RFX1800 (für den Betrieb im GSM 1800 Band)-Karten benötigt. Zur Analyse des GSM-Netzes kann auch eine DBSRX-Empfangskarte[5] verwendet werden. Tabelle 4.2 gibt eine Übersicht über den möglichen Frequenzbereich der jeweiligen Karte. Die entsprechende Karte, die für den Empfang benutzt wird, ist für den Betrieb von OpenBTS an den RX2 Port des USRP Mainboards anzuschließen.
Neben dem Betrieb von OpenBTS gibt es eine Vielzahl weiterer Projekte, die entsprechende Sende- bzw. Empfangskarten verwenden. Es gibt Projekte wie zum Beispiel ein RFID-Lesegerät,[6] einen GPS-Empfänger oder auch ein FM Rundfunk-Projekt.[7]

Daughterboard	**RFX900**	**RFX 1800**	**DBSRX**
Frequency	800 bis 1000 MHz	1.5 bis 2.1 GHz	800 MHz bis 2,4 GHz
Transmit	Power 200 mW (23 dBm)	100 mW (20 dBm)	100 mW (20 dBm)

Tabelle 4.2: Übersicht der Daughterboards RFX900, RFX1800 und DBSRX

4.3 Modifikationen

Für einen reibungslosen Betrieb einer OpenBTS waren drei Modifikationen an der Hardware notwendig. Dazu gehören die Deaktivierung des internen Taktgebers des USRPs (Abschnitt 4.3.1) und die Konfiguration und Verwendung eines externen Taktgebers (Abschnitt 4.3.2) sowie das Entfernen eines Filters auf den 900 MHz Empfangskarten (Abschnitt 4.3.3).

4.3.1 USRP Taktgeber deaktivieren

Bei dem im USRP verbauten Taktgeber handelt es sich um das Modell EC2620ETTS-64.000M der Firma Ecliptek. Diese hat eine Nominalfrequenz von 64 MHz mit einer Abweichung von ± 20 ppm.[8] Es stellte sich nach einigen Versuchen allerdings recht schnell heraus, dass ein externer Signalgeber unverzichtbar sein würde, da der im USRP verbaute Oszillator eine sehr hohe Frequenzinstabilität aufwies. Somit sind die Frequenzfehler des abgestrahlten Signals größer als die Toleranz eines Mobiltelefons erlaubt [9]. Normalerweise toleriert ein Mobiltelefon eine Abweichung von bis zu ± 10 kHz von der in GSM spezifizierten Referenzfrequenz. Laut Spezifikation ist eigentlich eine Abweichung von lediglich 0.05 ppm erlaubt. Das entspricht bei einem Taktgeber mit 64 MHz einer Abweichung von 3 Hz. Sollte diese Toleranz über- bzw. unterschritten werden, ist es für das Mobiltelefon eventuell nicht mehr möglich, die Basisstation zu erkennen. Im GSM-Band bei 900 MHz ist somit eine maximale Abweichung von 45 Hz und im GSM Band bei 1800 MHZ eine Abweichung von 90 Hz möglich, um einen einwandfreien Betrieb zu gewährleisten.
Um einen externen Taktgeber an das USRP anschließen zu können, muss der interne Taktgeber des USRPs deaktiviert werden. Die Deaktivierung des internen Taktgebers kann dem GNURadio Wiki unter dem Stichwort „USRP Clocking Notes“ entnommen werden.[9] Je nach Revision des USRP unterscheidet sich die Vorgehensweise leicht.

[5] 800 MHz bis 2.4 GHz Breitband-Empfangskarte
[6] RFID Hacking/usrp/, *https://www.noisebridge.net/wiki/RFID_Hacking/usrp/* [Online; letzter Aufruf 29.09.2009]
[7] Exploring GPS with Software Defined Radio, *http://www.gps-sdr.com/* [Online; letzter Aufruf 29.09.2009]
[8] Ecliptek Corporation, *http://www.ecliptek.com/SpecSheetGenerator/specific.aspx?PartNumber=EC2620ETTS-64.000M* [Online; Rev. T vom 29.09.2009]
[9] USRP Clocking Notes - GNU Radio, *http://gnuradio.org/trac/wiki/USRPClockingNotes* [13] [Online; letzter Aufruf 29.09.2009]

4.3.2 Externer Taktgeber

Durch die Verwendung eines externen Taktgebers kann das aufgetretene Problem wesentlich besser gelöst werden. Als möglichst kostengünstiger externer Taktgeber kam der Oszillator „FA-SY 1“ zum Einsatz.[10] Der zusammengebaute Bausatz samt Gehäuse ist in Abbildung 4.4 dargestellt. Prinzipell besteht auch in der Datei *Transceiver.cpp* (diese befindet sich im Ordner */OpenBTS/transceiver*) die Möglichkeit, durch eine Korrektur den Offset auszugleichen. Die Variable *FREQOFFSETT* (in Hz) kann das GMSK-Signal im Gesamten verschieben [27]. Dies ist allerdings nicht unbedingt zu empfehlen, da vor allem nach einer längeren Laufzeit des USRPs durch die entstehende Wärmeentwicklung das Signal weiter driftet. Darüber hinaus wird die Symbolrate des Taktes so verändert, dass Mobiltelefon und die BTS einen asynchronen Takt verwenden.

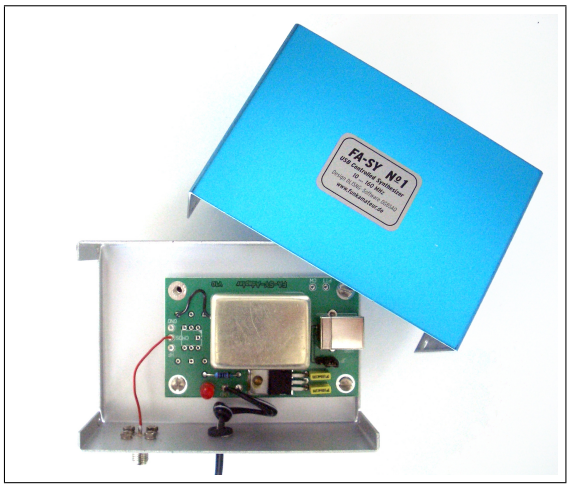

Abbildung 4.4: Signalgenerator FA-SY 1 zusammengebaut samt Gehäuse

Mittels eines angelöteten USB-Anschlusses lässt sich der Taktgeber mit der mitgelieferten Software *USB_Synth.exe* kalibrieren. Die Frequenz wurde dabei auf 64 MHz eingestellt. Die genauen Konfigurationsparameter können Abbildung 4.5 entnommen werden. Die eingestellte Frequenz wird dauerhaft gespeichert, bis sie mittels Software auf eine andere Frequenz umgestellt wird. Betrieben wird das Gerät mit 12 Volt Gleichspannung. Ein verbautes Heizelement, das für konstante Temperaturverhältnisse und somit für eine möglichst gleich bleibende Frequenz sorgt, benötigt 12 Volt. Der Signalgenerator selber wird über einen Spannungswandler mit 5 Volt versorgt. Um den Signalgenerator mit dem SMA-Clock-Anschluss des USRPs verbinden zu können, wurde eine SMA-Buchse auf der Adapterplatine angelötet.
Als Alternative könnte auch ein anderer externer Signalgenerator verwendet werden, der allerdings meistens einen wesentlich höheren Kostenaufwand mit sich bringt. Der kostengünstige Signalgenarator FA-SY 1 hat sich in einem Langzeittest über fast ein halbes Jahr als äußerst präzise und völlig ausreichend erwiesen.

4.3.3 Filter entfernen 900er Board

Das RFX900-Board verfügt bei der Auslieferung über einen ISM-Band-Filter. Dieser lässt kein RF-Signal außerhalb von 902-928 MHz zu, da in den USA ausschließlich auf diesem Band frei

[10] Hierbei handelt es sich um einen Selbstbausatz, der für ca. 50 Euro über den Onlineshop der Fachzeitschrift Funkamateur bezogen werden kann (*http://www.box73.de/catalog/*). Entsprechende Bauanleitungen können der Herstellerwebseite entnommen werden (*http://www.box73.de/catalog/pdf/BX-029.pdf?osCsid=983aae56f383b* und *http://www.box73.de/catalog/pdf/BX-026.pdf?osCsid=983aae56f383b*)

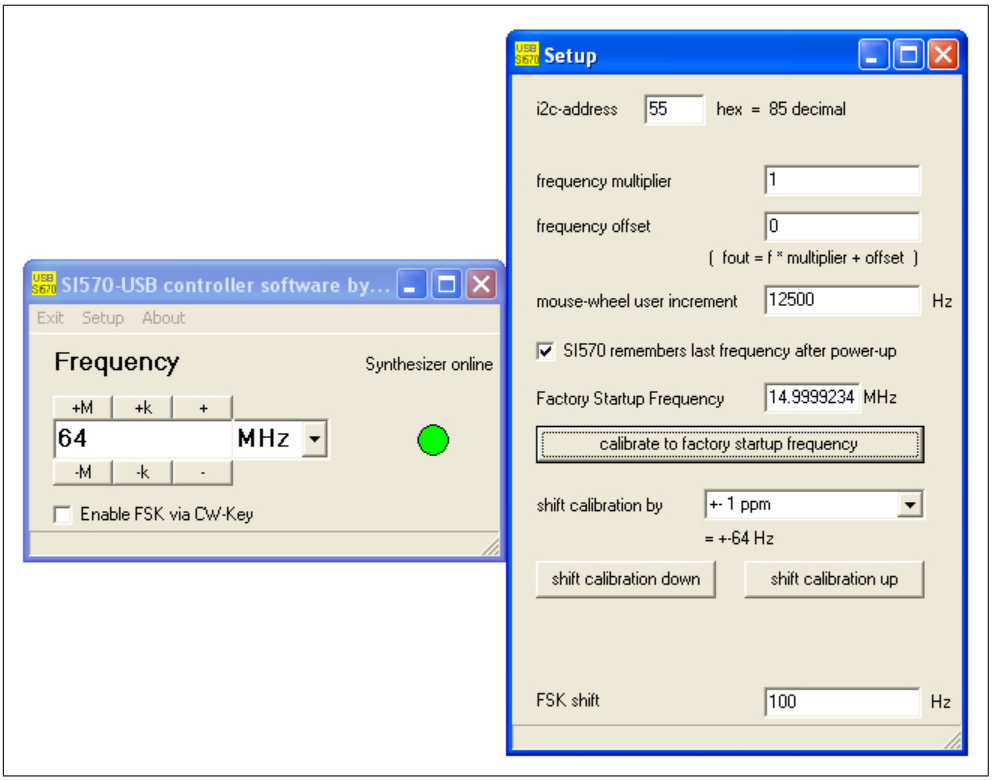

Abbildung 4.5: Konfigurationsparameter für FA-SY 1 mittels *USB_Sync.exe*

gesendet und empfangen werden kann. Wie in Kapitel 2.2 und Tabelle 2.2 beschrieben wurde, wird im GSM 900 MHz Band in Europa allerdings ein Frequenzbereich für den Downlink von 925,0-960,0 MHz und für den Uplink von 880,0-915,0 MHz benötigt. Um das Board nun außerhalb dieses amerikanischen ISM-Bandes nutzen zu können, ist es notwendig, den Filter auf dem Board zu entfernen. Dies kann relativ leicht durchgeführt werden, indem die Leitung FIL.1 zum Filter getrennt wird und ein 100 pF Kondensator parallel aufgelötet wird.[11]

4.4 Mobiltelefone und SIM-Karten

Eine Übersicht über die verwendeten Mobiltelefone ist Abbildung 4.6 und Anmerkungen zu den Mobiltelefonen sind Tabelle 4.3 zu entnehmen. Als besonders gut geeignet haben sich vor allem Mobiltelefone der Firma Nokia herausgestellt. Lediglich neuere Mobiltelefone sind in der Lage im Display einen „eigenen Netznamen“ wie z.B. „OpenNetz“ anzuzeigen. Als SIM-Karten wurden hauptsächlich Prepaid-Karten der Firmen Vodafone und O2 eingesetzt. Diese funktionierten alle ohne Beanstandung und die Mobiltelefone konnten sich ohne Probleme in das OpenBTS-Netz einbuchen. In Kapitel 5.1 werden die verwendete Hardware aufgelistet und genauere Softwarekonfiguration erläutert.

[11] GNU Radio, *http://gnuradio.org/trac/wiki/* [13] [Online; letzter Aufruf 30.09.2009]

Abbildung 4.6: Übersicht der verwendeten Mobiltelefone

Mobiltelefon	**Verhalten mit OpenBTS**	**Anmerkungen / Besonderheiten**
Nokia 3210	Probleme mit OpenBTS 2.3, Netzname nur als ID sichtbar	Vodafone Prepaidsimkarte, manuelle Netzsuche hängt sich ab und zu auf, Netzmonitor aktiviert, Datenkabel
Nokia 3310	Probleme mit OpenBTS 2.3, Netzname nur als ID sichtbar	Netzmonitor aktiviert, Datenkabel, zur GSM-Analyse verwendet (siehe Kapitel 6.4)
Sagem 226	optimal, Netzname nur als ID, Probleme mit SMS	Vodafone Prepaidsimkarte
Nokia 6230i	Netz nicht immer gefunden	T-Mobile Prepaidsimkarte
Nokia E71	optimal, Netzname wird angezeigt	E-Plus Vertragssimkarte
iPhone 2G	optimal, bucht sich teilweise schon automatisch ein, Netzname nur als ID	O2-Vertragssimkarte
Nokia N80	optimal, Netzname wird angezeigt	Vodafone Prepaidsimkarte
Siemens S55	optimal, Probleme mit OpenBTS Versionen <2.3	schnell das Netz gefunden und eingebucht
Samsung SGH-E330	muss nach Netzwahl neu gestartet werden („Zugriff nicht möglich“)	O2 Prepaidsimkarte
Motorola W156	optimal	Vodafone Prepaidsimkarte

Tabelle 4.3: Übersicht über die mit OpenBTS verwendeten Mobiltelefone

5 OpenBTS-Versuchsaufbau

In den folgenden Abschnitten wird auf den Aufbau der Hardware und das Zusammenspiel der einzelnen Software Komponenten genauer eingegangen. Es wird der Funktionsumfang einer OpenBTS-Zelle erklärt, und einzelne Systemdienste werden in Abschnitt 5.2 mit der Implementierung der Dienste aus der „realen“ GSM-Welt aus Kapitel 2 verglichen. So können bereits implementierte Funktionen wie das initiale Registrieren eines Mobiltelefons oder auch ein Gesprächsauf- bzw. -abbau für Lehrzwecke praktisch durchgeführt und die dafür notwendigen Schritte nachvollzogen werden. Der Betrieb einer eigenen GSM-Funkzelle durch OpenBTS bildet die Grundlage, Voraussetzungen, Abläufe und notwendige Sicherheitsmechanismen im Bereich GSM verstehen und analysieren zu können.

5.1 Praktisches Setup

Abbildung 5.1 gibt eine Übersicht der einzelnen Hardware-Komponenten. Abbildung 5.3 fasst die einzelnen Softwarekomponenten zusammen. Deren Zusammenspiel und Funktion wird in Abschnitt 5.1.2 detailliert beschrieben. Es kamen folgende Soft- und Hardware zum Einsatz:

Software

- GNU/Linux - Ubuntu 8.10 - 32 Bit mit Kernelversion 2.6.23
- OpenBTS 2.4
- GNURadio 3.1.3
- C++ Boost 1.37

Hardware

- Dell Notebook XPS 1330 (Core 2 Duo 2.5 GHz, 3 GByte RAM, USB 2.0 Port)
- SRP-PKG (USRP Package, includes Motherboard, Enclosure, 2 RF Cables, USB Cable, Power Supply, and Hardware Package)
- RFX900 für GSM 850/900 (800-1000 MHz Transceiver, 200 mW output);
- 2x VERT900 (824-960 MHz, 1710-1990 MHz Quad-band Cellular/PCS and ISM Band Vertical Antenna, 3 dBi Gain, 9 Inches, geeignet für RFX900 und RFX1800).
- diverse Mobiltelefone: Nokia 3210, 3310, N80, E71; Apple iPhone 2G; Sony-Ericson T610; Siemens C35; ... (Details siehe Kapitel 4.4 und Abbildung 4.6)

5.1.1 Aufbau der Hardware-Komponenten

Die einzelnen Hardware-Komponenten werden wie in Abbildung 5.1 miteinander verbunden. Abbildung 5.2 zeigt den USRP mit angeschlossenen Antennen und dem externen Signalgeber FA-SY 1. Als ausreichend leistungsfähiger Computer wurde ein Notebook mit 3 GByte Arbeitsspeicher

und einem Dualcore-Prozessor mit 2.5 GHz verwendet. Dadurch ist genügend Rechenleistung gegeben, um bis zu sieben Gespräche[1] gleichzeitig führen zu können. Der interne Taktgeber des USRP wurde deaktiviert und ein externer Signalgeber (mit einer Betriebsfrequenz von 64 MHz) angeschlossen (wie im vorangegangenen Kapitel 4.3 beschrieben). An den entsprechenden RX- beziehungsweise TX-Ausgang des USRP sind die zwei mitgelieferten 3 dBi Rundstrahlantennen angeschlossen, die über eine Reichweite von ungefähr zehn Metern verfügen. Diese Reichweite konnte in mehreren Versuchen in unterschiedlichen Räumen bestätigt werden. Mit entsprechendem Verstärker und leistungsstärkeren Antennen kann noch eine deutlich weitere Reichweite erzielt werden. Per Internetverbindung besteht eine dauerhafte Verbindung zu einem Asterisk-Server der Universität Freiburg. Dadurch können Gespräche von OpenBTS direkt ins Festnetz weitervermittelt werden.

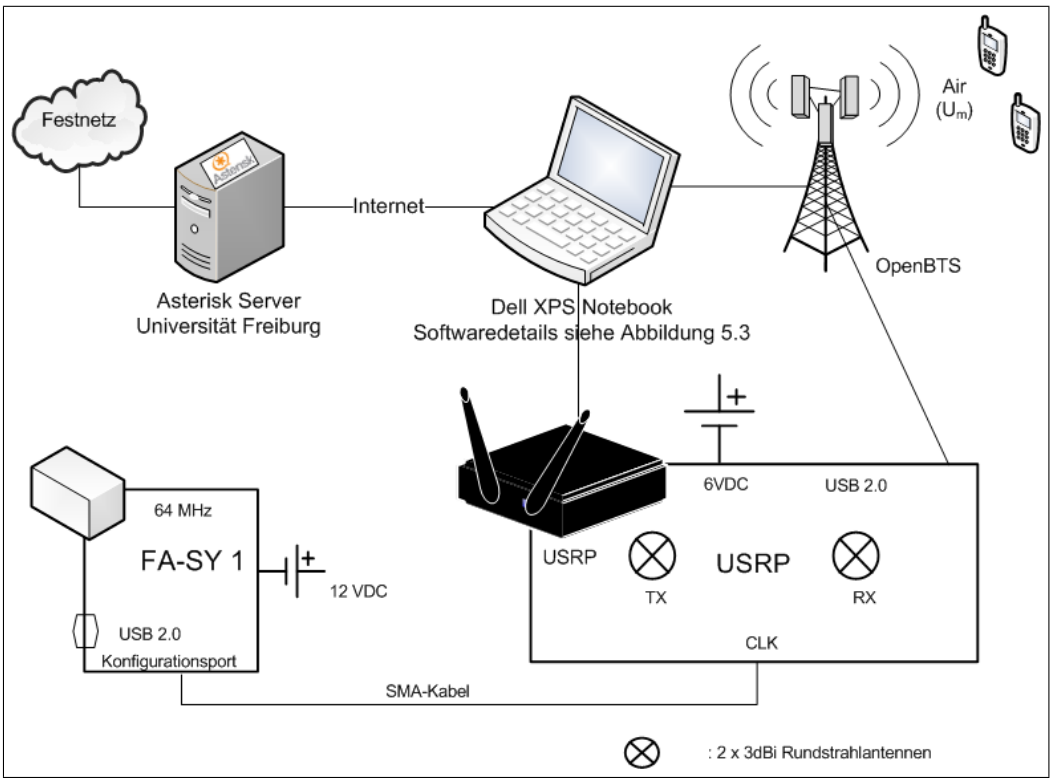

Abbildung 5.1: Übersicht über den Aufbau der Hardware-Komponenten

5.1.2 Software-Inbetriebnahme

Abbildung 5.3 verdeutlicht im Detail das Zusammenspiel der einzelnen Softwarekomponenten. Zu Testzwecken standen die Versionen 1.6, 2.3 und 2.4 von OpenBTS zur Verfügung. Diese sind auf der CD im Verzeichnis */OpenBTS/Versionen* zu finden. Die Änderungen von Version zu Version sind im Anhang B tabellarisch dargestellt.
Gestartet werden kann OpenBTS mit dem Befehl *./OpenBTS*. Wichtig dabei ist, dass die Konfigurationsdatei *OpenBTS.config* zuvor bearbeitet wurde und entsprechende Parameter eingestellt wurden (in Kapitel 3.2.4 genauer beschrieben). Es erscheint eine Konsole, mit der OpenBTS verwaltet und im laufenden Betrieb konfiguriert werden kann. Folgende Befehle stehen ab Version

[1] Durch OpenBTS wird eine GSM-Zelle betrieben, die über acht Zeitschlitze verfügt. Ein Zeitschlitz wird für Signalisierungsprozesse verwendet, mittels der sieben übrigen können sieben Mobilfunkteilnehmer Gespräche führen.

Abbildung 5.2: USRP mit externem Taktgeber und Antennen

2.4 zur Verfügung (diese können mit dem Befehl *help* angezeigt werden, siehe Abbildung 5.4):

- *calls*: Übersicht über die momentan geführten Gespräche
- *assignment*: (Very) Early Assignment
- *cellid*: MCC, MNC, Location Area und Cell-ID werden ausgegeben
- *config*: *OpenBTS.config* wird ausgegeben
- *configsave*: aktuelle Konfiguration kann in einer vom Benutzer frei wählbaren Datei abgespeichert werden
- *load*: Übersicht über die Auslastung der Kanäle
- *loglevel*: Einstellung des Loglevels
- *regperiod*: Einstellung der Sipregistration-Zeit und des t3212-Parameter (Zeit bis zum nächsten Location update)
- *sendsms*: Sendet SMS an eingebuchte IMSI (siehe Befehl *tmsis*)
- *setlogfile*: Datei, in der Log-Ereignisse gespeichert werden
- *shortname*: Spezifikation des angezeigten Netznamens (z.B. „OpenBTS“)
- *tmsis*: Übersicht über die eingebuchten Mobiltelefone samt dazugehöriger IMSI und TMSI
- *uptime*: aktuelle Laufzeit der OpenBTS
- *help*: Übersicht über die in Abbildung 5.4 dargestellten Befehle
- *exit*: Stoppt und beendet die OpenBTS

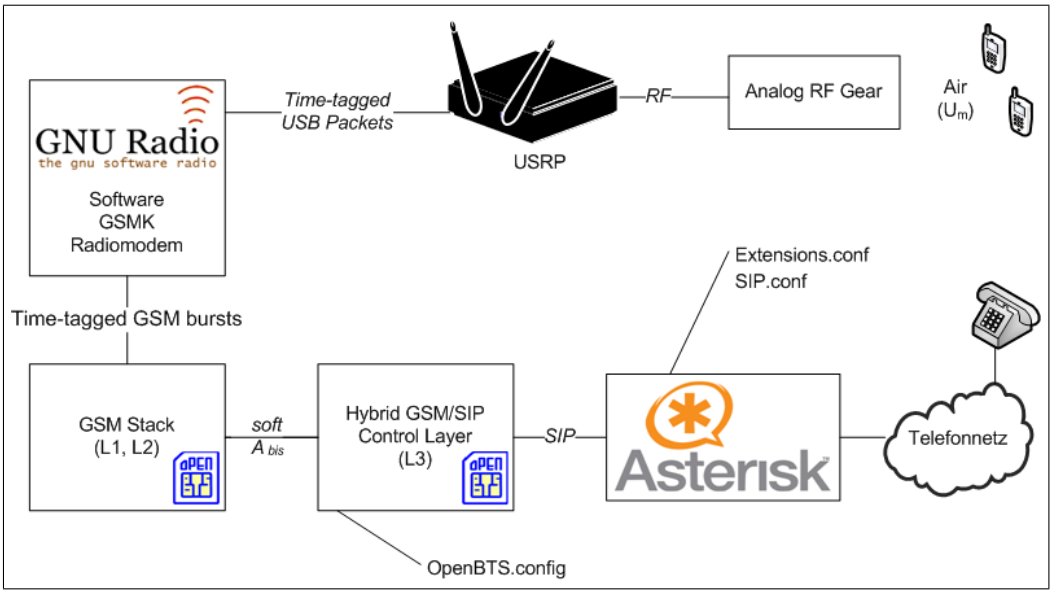

Abbildung 5.3: Übersicht über das Zusammenspiel der Softwarekomponenten

```
Welcome to OpenBTS.  Type "help" to see available commands.

OpenBTS> help
assignment      calls   cellid  config  configsave
exit    help    load    loglevel        regperiod
sendsms setlogfile      shortname       tmsis   uptime
```

Abbildung 5.4: OpenBTS-Konsole mit Übersicht über die möglichen Befehle

5.2 Systemdienste in OpenBTS

In Kapitel 2.5 wurde das OSI-Referenzmodell für GSM genauer erläutert. Da OpenBTS nur eine GSM-Zelle bereitstellt, sind gerade Schicht 3 Funktionen nur teilweise bzw. noch nicht vollständig umgesetzt. In dieser Schicht steckt die eigentliche Logik der Vermittlung. Zu deren Schwerpunkte gehört die Verwaltung und Zuweisung der logischen Kanäle durch den BSC (Radio Resource), die Authentifizierung und Administration der Mobilfunkteilnehmer über einzelne Zellen hinweg (Mobility Management) und die Regelung des logischen Ablaufes bei Anrufen und deren Terminierung (Call Control). In den folgenden Abschnitten sollen die Systemdienste Registrierung und Authentifizierung eines Mobilfunkteilnehmers (Abschnitt 5.2.1) und der Gesprächsauf- bzw. -abbau (Abschnitt 5.2.2) genauer beleuchtet und mit der realen GSM-Welt verglichen werden.

5.2.1 Registrierung und Authentifizierung eines Mobilfunkteilnehmers

Kapitel 2.6.1 beschreibt den Registrierungsprozess eines Mobilfunkteilnehmers in GSM. Abbildung 5.5 zeigt den initialen Registrierungsprozess eines Mobiltelefons (vgl. Abbildung 2.15). Dieser entspricht in großen Teilen exakt dem in GSM spezifizierten Registrierungsprozess. Der Hauptunterschied besteht allerdings darin, dass eine Authentifizierung, wie sie in GSM vorgesehen ist, nicht stattfindet. Wie der Abbildung entnommen werden kann, findet lediglich eine Authentifizierung mittels IMSI zwischen Mobiltelefon und der Asterisk Software statt (siehe Kapitel 3.3). Es ist möglich, eine offene Registrierung in OpenBTS zu verwenden (Parameter *Control.OpenRegistration* in der *OpenBTS.config*). Sollte diese offene Registrierung aktiviert sein,

ist es für jedes Mobiltelefon möglich, sich an der OpenBTS zu registrieren, auch wenn in der *sip.conf* kein gültiger Benutzer vorhanden ist. Es kann auch keine verschlüsselte Kommunikation erfolgen, da hierzu eine vollständige Authentifizierung, wie sie GSM verwendet, notwendig wäre. Somit wird die TMSI, die durch die OpenBTS Software generiert wurde (in GSM durch das HLR), auch unverschlüsselt an das Mobiltelefon übertragen und von diesem per Acknowlege bestätigt. Der gesamte Registrierungsprozess in OpenBTS kann mittels Logfiles nachvollzogen werden (Screenshot siehe Abbildung 5.6). Diese wurden mit der Einstellung *Loglevel = Debug* bzw. *Loglevel = Info* erstellt und befinden sich auf der CD im Ordner *OpenBTS/Registrieungsprozess/* unter dem Namen *Manuelle_ Registrierung_ Nokia_ 3310_ DEBUG.out* und *Manuelle_ Registrierung_ Nokia_ 3310_ INFO.out*. Dieser Registrierungsprozess eines Mobilte-

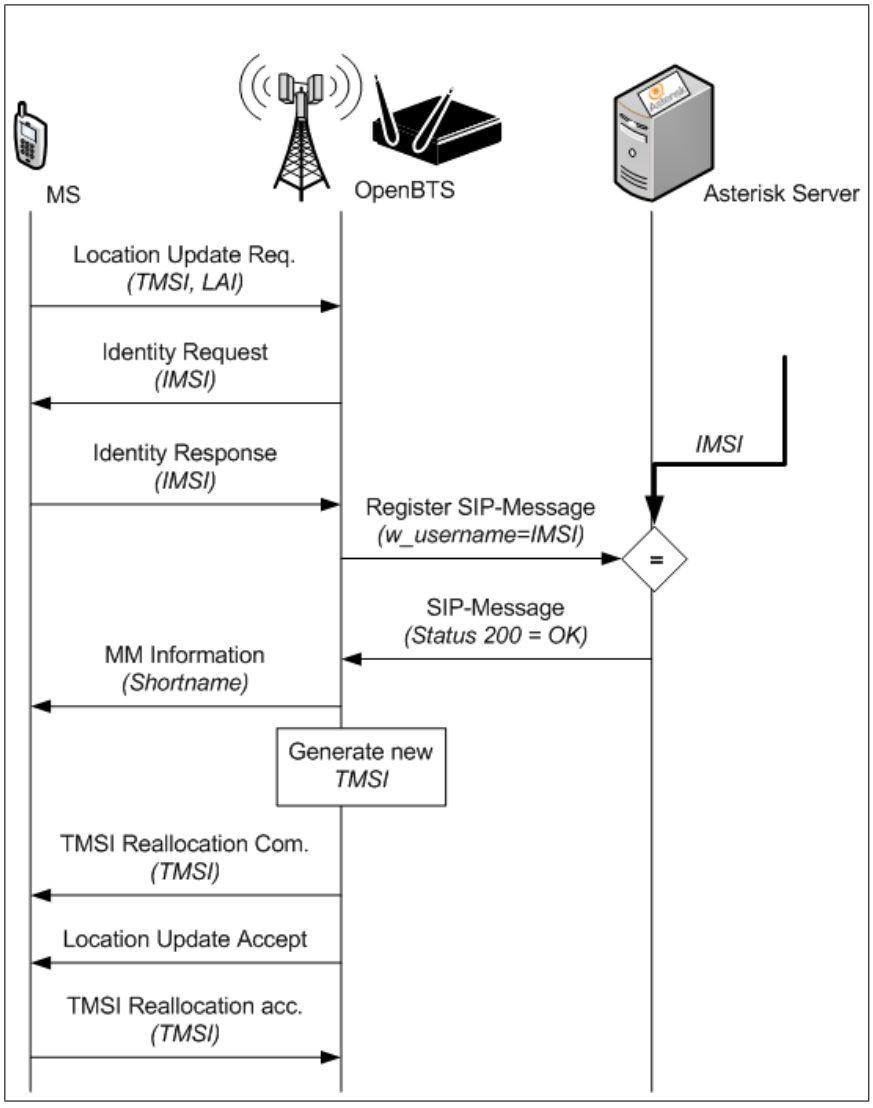

Abbildung 5.5: Registrierungsprozess beim Einschalten des Mobiltelefons in OpenBTS

lefons in OpenBTS wurde auch mit der in Kapitel 6.4 beschriebenen Analysemethode mittels eines Nokia 3310 Mobiltelefons und Wireshark verifiziert. Wireshark (ab Version 1.2.1) ist in der Lage, den Einbuchvorgang eines Mobiltelefons und die erzeugten Pakete zu einem großen Teil sichtbar zu machen. Hierbei wurde der Registrierungsprozess in OpenBTS und der in ein echtes Vodafone-Netz aufgezeichnet. Gerade die nicht vorhandene Authentifizierung (bzw. nur mittels IMSI und Asterisk) in OpenBTS und die vollständige Authentifizierung und Verschlüsselung im

```
1252923264.8314 INFO 3060931472 Logger.cpp:100: new log path test.out
1252923265.8593 WARNING 3060931472 TRXManager.cpp:269: retrying transceiver command after response timeout
1252923266.1031 INFO 3012529040 RadioResource.cpp:375: Pager::pageAll paging 0 mobile(s)
1252923266.1033 INFO 3060931472 main.cpp:654: system ready
1252923314.7339 INFO 3035220880 RadioResource.cpp:152: AccessGrantResponder RA=0x1a when=0:367744 age=23 TOA=
1.0000
1252923314.7342 INFO 3035220880 RadioResource.cpp:196: AccessGrantResponder sending PageMode=(0)
DedicatedModeOrTBF=(TMA=0 Downlink=0 DMOrTBF=0) ChannelDescription=(typeAndOffset=SDCCH/4-0 TN=0 TSC=0 ARFCN=
29) RequestReference=(RA=26 T1'=21 T2=0 T3=34) TimingAdvance=0
1252923315.2332 INFO 3033090960 MobilityManagement.cpp:140: LocationUpdatingController MM Location Updating
Request LAI=(MCC=262 MNC=02 LAC=0x319) MobileIdentity=(TMSI=0x3d0660c)
1252923316.1734 INFO 3033090960 SIPEngine.cpp:151: SIPEngine::Register mState=Null 0 callID 751408212
1252923316.1735 INFO 3033090960 SIPInterface.cpp:103: creating SIP message FIFO callID 751408212
1252923316.1781 INFO 3033090960 SIPInterface.cpp:110: removing SIP message FIFO callID 751408212
1252923316.1783 INFO 3033090960 MobilityManagement.cpp:217: LocationUpdatingController registration SUCCESS:
IMSI=262027033983293
1252923317.4072 INFO 3033090960 MobilityManagement.cpp:120: sending Control.NormalRegistrationWelcomeMessage
message to handset
1252923318.2925 INFO 3033090960 SMSControl.cpp:432: MTSMS: sending CP-DATA TI=(0,5) RPDU=( primitive=undefined
data=
(000000010110000100000011101000010000000000000000000000000100000100100000000010010100001000000000000000000000
000000000001001000010010000010000010010000101010001100000010000100000010011110101110011001001111011111111100011
011101001011101000001111101000011011111101000000010010010111110111011100001011101010001101001000010000000000)
)
1252923319.9402 INFO 3033090960 SMSControl.cpp:461: MTSMS: sending RPAck in CPData
```

Abbildung 5.6: OpenBTS Logfile mit *Loglevel = INFO*

Vodafone-Netz wird deutlich. Die erzeugten XML-Dateien, die mit Wireshark betrachtet werden können (siehe Abbildung 5.7), sind im Ordner *GSM-Analyse/Nokia3310/Wireshark* auf der CD unter dem Namen *Registrierungsprozess_bei_OpenBTS_mit_manueller_Netzsuche.xml* und *Registrierungsprozess_bei_Vodafone_mit_manueller_Netzsuche.xml* zu finden.

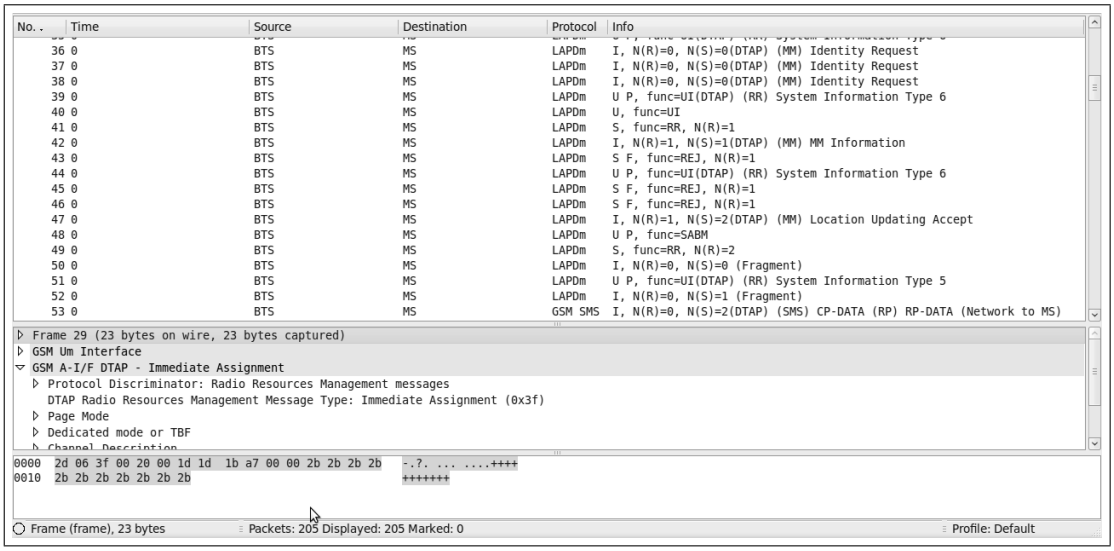

Abbildung 5.7: Wireshark-Mitschnitt: OpenBTS Registrierungsprozess

5.2.2 Gesprächsauf- und -abbau

Abbildung 5.8 zeigt den Gesprächsaufbau in OpenBTS. Der TCH und der FACCH werden zu Beginn der Prozedur zugewiesen. Diese frühe Zuweisung wird als „Very Early Assignment“ bezeichnet und kann mittels des Befehls *assignment* oder einer entsprechenden Änderung in der *OpenBTS.config*-Datei aktiviert bzw. deaktiviert werden. Die GSM-Spezifikationen setzen diese frühe Zuweisung nicht voraus. Die Zuweisung dieser zwei Kanäle kann auch unmittelbar vor dem eigentlichen Verbindungsaufbau erfolgen. Alle weiteren Schritte orientieren sich an den aus der ISDN-Welt bekannten Prozessen. Diese entsprechen auch exakt den Schritten aus der realen

GSM-Welt, mit dem wichtigen Unterschied, dass keine Verschlüsselung stattfindet. Dies liegt daran, dass die Authentifizierung eines Teilnehmers über Asterisk erfolgt und ein Challenge-Response-Verfahren zur Authentifizierung und Generierung des benötigten Schlüssels zur Verschlüsselung zwischen SIM-Karte und MSC, wie es in Kapitel 2.3.2 beschrieben wurde, nicht vorgesehen ist.

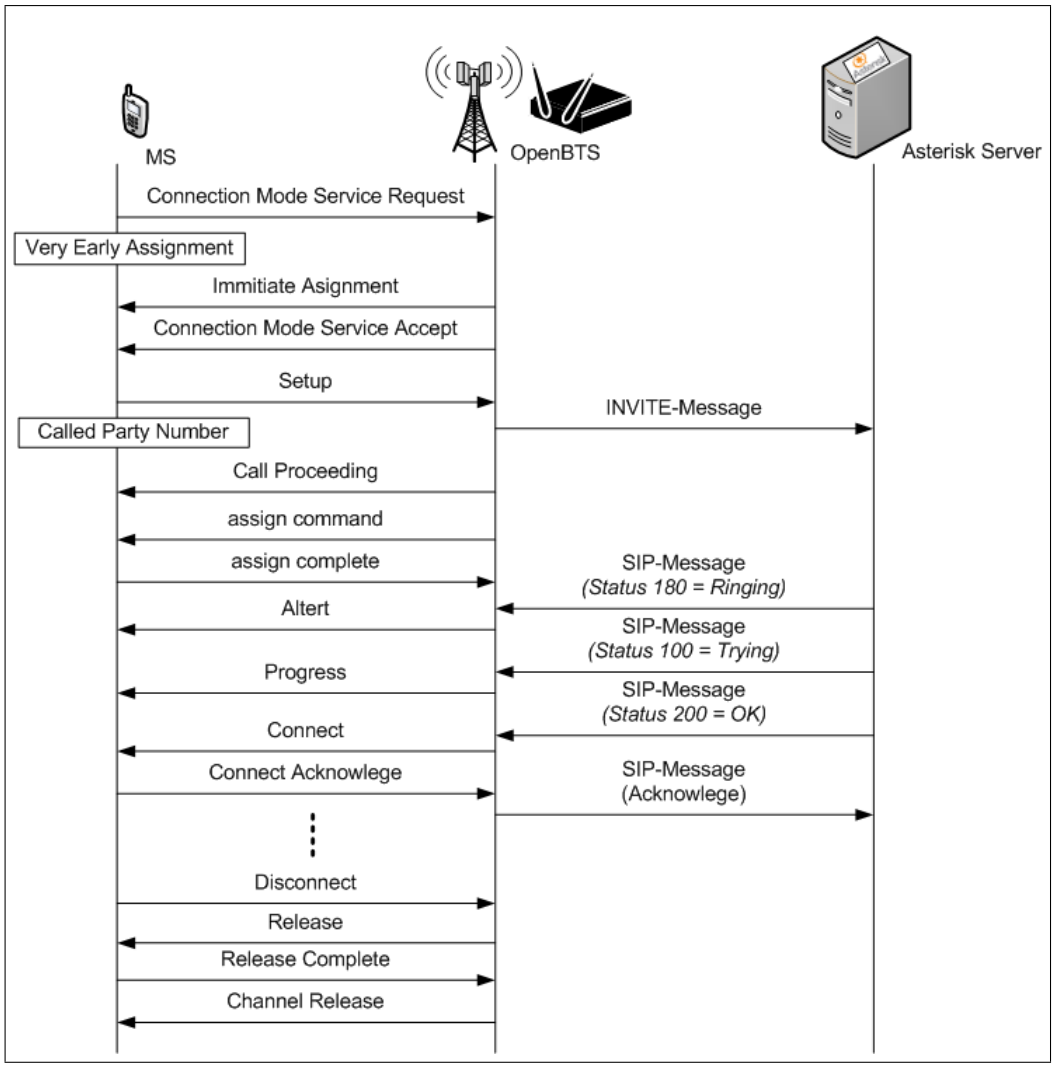

Abbildung 5.8: Mobile Originated Call und Call Clearing Prozedur (Vorlage nach [27])

5.3 Unterschiede zwischen kommerziellem GSM und OpenBTS

Beim betrachten der realen GSM-Infrastruktur aus Abbildung 2.3 und dem Vergleich mit der von OpenBTS in Abbildung 5.1, ist festzustellen, dass gerade in der Hintergrundinfrastruktur vieles noch nicht realisiert ist. Tabelle 5.1 soll gravierende Unterschiede, die nicht bzw. noch nicht oder auf andere Art und Weise implementiert wurden, zusammenfassen.

GSM-Netz	**OpenBTS**
MSC	softwareseitig umgesetzt; Benutzerverwaltung und Gesprächsvermittlung mittels Asterisk-Server
GMSC	Routing der Gespräche über Voice-over-IP ins Festnetz mittels Asterisk-Server
BSC	nur eine BTS, kein BSC; mehrere OpenBTS können noch nicht zusammengeschaltet und verwaltet werden
AUC (Authentifizierung siehe Kapitel 2.3.2)	Authentifizierung mittels IMSI am Asterisk-Server
Nutzdatenverschlüsselung durch A5-Algorithmus	keine Verschlüsselung der Nutzdaten
Frequenzen: mehrere AFCRNs pro Zelle	nur eine Frequenz/AFCRN
Datenbanken: HLR, VLR, EIR	Benutzerverwaltung nur mittels Asterisk-Server (zukünftig als MySQL-Server geplant, der mit Asterisk interagiert [22])
Reichweite: mehrere Kilometer	Reichweite: ca. 10 Meter (ohne zusätzlichen Verstärker und optimierte Antennen)
SMSC: SMS-Dienst	kein SMSC, SMS-Dienst nur teilweise implementiert (bis Version 2.4) und keine Speicherung von SMS-Nachrichten oder SMS-Nachrichten zwischen den Mobilfunkteilnehmern möglich
Handover	kein Handover möglich, auch nicht zwischen zwei OpenBTS bzw. Asterisk-Servern
MSISDN als Rufnummer eindeutig zur entsprechenden IMSI	nur IMSI bekannt, aber dazugehörige MSISDN unbekannt
ISDN-Funktionen wie Rufnummerübermittlung, Gespräch halten, Anklopfen	(noch) nicht vorhanden
Datendienste wie WAP, GPRS, EDGE	(noch) keine Datendienste
deutlich höhere Kosten für den Betrieb der kompletten GSM-Infrastruktur (ca. 5-6 $ pro Mobilfunkteilnehmer im Monat [4])	Kosten für den Betrieb von OpenBTS samt nötiger Infrastruktur (größtenteils softwareseitig realisiert) pro Mobilfunkteilnehmer im Monat: ca. 1 $ [4] (Anschaffungskosten der Hardware: 1500 $)
großer Verwaltungsaufwand der einzelnen GSM-Netzwerkkomponenten wie MSC, SMSC, BSC, AUC	zentrale Verwaltung aller nötigen Netzwerkkomponenten mit wesentlich geringerem Aufwand

Tabelle 5.1: Unterschiede zwischen GSM-Netz und OpenBTS

6 GSM-Analyse

Es werden in den folgenden Abschnitten verschiedene Methoden vorgestellt, mit deren Hilfe es möglich ist, GSM-Daten zu analysieren. Dazu gehören einerseits die Analyse der realen GSM-Infrastruktur und der übertragenen Daten als auch die Analyse der OpenBTS. Die daraus gewonnenen Erkenntnisse können für Lehr- und Forschungszwecke eingesetzt werden, um die Funktionsweise und eventuell sich daraus ergebende Probleme im Bereich Sicherheit praktisch zu demonstrieren. Ziel ist es, verschiedenste Abläufe wie Einbuchen eines Mobilfunkteilnehmers, Informationen über die verschiedenen Funkzellen, aber auch ganze Prozessabläufe wie den Aufbau eines ausgehenden Gespräches, Handover-Prozesse, einen Zellwechsel oder das Senden einer SMS nachvollziehbar zu machen. Folgende Analyseverfahren werden in den folgenden Abschnitten genauer betrachtet:

- Netzmonitor (in Abschnitt 6.1)
- Spektrumsanalyse (in Abschnitt 6.2)
- AirProbe und GSSM (in Abschnitt 6.3)
- GSM Decodierung mit Nokia 3310 und Wireshark (in Abschnitt 6.4)

6.1 Netzmonitor

Einige ältere Mobiltelefone verfügen über einen Netzwerkmonitor, der über das gängige Menü normalerweise nicht zugänglich ist und erst mittels spezieller Software freigeschaltet werden muss. Die Vorgehensweise unterscheidet sich von Mobiltelefon zu Mobiltelefon. Eine ausführliche Anleitung für verschiedene Nokia Modelle ist auf der Homepage *„nokiaport.de“* zu finden.[1] Mittels dieses Monitors lassen sich Parameter wie Kanalzuteilung, Leistungsregelung, Cell-ID, Informationen über Nachbarzellen, Handover und weitere ermitteln [28]. Diese sind in der Regel nicht ohne Weiteres einsehbar. Vier dieser Netzmonitor-Displays eines Nokia 3310 sind in Abbildung 6.1 dargestellt. Tabelle 6.1 zeigt eine Übersicht aller NetMonitor-Displays eines Nokia DCT-3 Geräts wie dem Nokia 3210 und 3310. Für neuere Nokia Mobiltelefone steht dieser Netzmonitor als optionale Software zur Verfügung. Für das Nokia E71 gibt es beispielsweise die frei verfügbare Software PhoNetInfo[2], mit der sich problemlos unter anderem Parameter wie IMSI, Cell-ID, Signalstärke auslesen lassen. Mittels dieser Software können aber längst nicht mehr so viele Informationen wie mit dem Nokia 3210/3310 festgestellt werden. Es wurden mit dem Nokia 3310, Nokia 3210 und dem Nokia E71 mit installierter PhoNetInfo-Software verschiedenste Parameter gemessen. Diese sind in Tabelle 6.2 zusammengefasst aufgelistet. Für das E-Plus Netz wurden die ermittelten Parameter mittels Nokia E71 und PhoNetInfo-Software noch einmal überprüft und konnten bestätigt werden (Tabelle 6.3).

[1] NokiaPort.de - Nokia Net-Monitor *http://nokiaport.de/index.php?pid=netmon&dct3mid=* [Online; letzter Aufruf 29.09.2009]

[2] Frei Software Development, *http://www.patrickfrei.ch/phonetinfo/index.html* [Online; letzter Aufruf 29.09.2009]

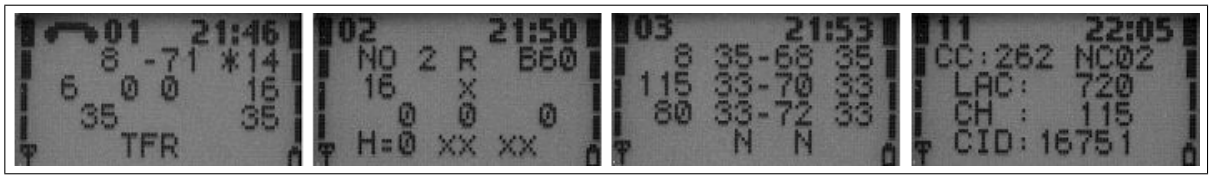

Abbildung 6.1: Nokia Netzmonitor

Display	Beschreibung
01	Information on the serving cell
02	More Information on the serving cell
03	Information on the serving cell, 1st and 2nd neigbour
04	Information on the 3rd, 4th and 5th neighbour
05	Information on the 6th, 7th and 8th neighbour
06	Network selection display
07	System information bits for the serving cell
10	Paging Repetition Period, TMSI, Location Update Timer, AFC and AGC
11	Network parameters
12	Ciphering, hopping, DTX status and IMSI
13	Uplink DTX switching display
14	Toggle screening indicator
17	Switch „BTS Test“ status
18	Lights status control
19	Toggle cell barred status

Tabelle 6.1: Nokia NetMonitor DCT-3-Übersicht

Vor allem der Monitormode „01“ ist zu Analysezwecken einer GSM-Zelle oder auch einer OpenBTS-Zelle geeignet.[3]

Übersicht über die möglichen Parameter (Mode: 01):

Mode: 01

ARF RSI PCC
S TT Q MMM
C1 C2
CHAN

- **ARF**: aktuelle ARFCN (damit die Frequenz)
- **RSI**: RSSI in dBm
- **PCC**: Power Control Command (bei einer Übertragung), kein dBm Wert. Je höher der Wert, desto geringer die Übertragungsstärke. (Im GSM 900 Band: „5“ = maximaler Stärke (meistens 1 Watt), „19“ = geringe Stärke (5 mW))
- **S**: aktueller Timeslot

[3] Nokia DCT-3 Test Phones and OpenBTS, *http://sourceforge.net/apps/trac/openbts/wiki/OpenBTS/NokiaDCT3* [Online; letzter Aufruf 29.09.2009]

- **TT**: Timing Advance, Wert zur Synchronisation zwischen Up- und Downlink (TT*550 Meter entspricht dem ungefähren Abstand des Mobiltelefons zur BTS)
- **Q**: Bewertung der Linkqualität: 0-7 (0=beste)
- **MMM**: Radio Link Timeout Zähler
- **C1**: Path Loss Kriterium
- **C2**: Cell Reselection Kriterium
- **CHAN**: aktueller Kanal Typ: z.B. AGCH (Service wird angefordert), BCCH (Informationen der BTS), CCCH, SDCCH (Steuerungskanäle), FA (Steuerungskanal), TFR (TCH + FACCH, full rate, Sprachanruf), TEFR (TCH + FACCH, enhanced full rate, Sprachanruf), F144 (Datenübertragung 14,4 kbit/s)

E-Plus Karte (Nokia 3210/3310 Netzmonitor)	
CC:	262
NC:	03
LAC:	588
CH:	14
CID:	11768

Vodafone Karte (Nokia 3210/3310 Netzmonitor)	
CC:	262
NC:	02
LAC:	793
CH:	64
CID:	6913

T-Mobile Karte (Nokia 3210/3310 Netzmonitor)	
CC:	262
NC:	01
LAC:	29191
CH:	102
CID:	29242

O2 Karte (Nokia 3210/3310 Netzmonitor)	
CC:	262
NC:	07
LAC:	50945
CH:	711
CID:	29790

Tabelle 6.2: Mit Netzmonitor gescannte Werte der verschiedenen Mobilfunkbetreiber (Ort: Rechenzentrum der Universität Freiburg)

E-Plus Karte (Nokia E71 PhoNetInfo Software)	
MCC:	262
Network ID (MNC/NID):	03
Location Area Code (LAC):	588(0x24C)
Cell ID (CID):	11768 (0x2DF8)

Tabelle 6.3: Nokia E71 mit PhonNetInfo-Software für das E-Plus Netz

Als weitere Alternative stand noch ein Siemens C35 Mobiltelefon und die unter anderem für Siemens Mobiltelefone konzipierte Software S25@once! v3.7.5 und der Netzmonitor Tapir-G v2.4.1.707 zur Verfügung [29], [23]. Die Abbildungen 6.2 und 6.3 zeigen die Software s25@once! bzw. den Netzmonitor Tapir-G und die für das T-Mobile Netz ermittelten Werte. In Kapitel 7.1 wird beschrieben, welche mittels dieser ermittelten Parameter wie IMSI, Netzwerk ID und LAI sicherheitsrelevanten Probleme entstehen können.

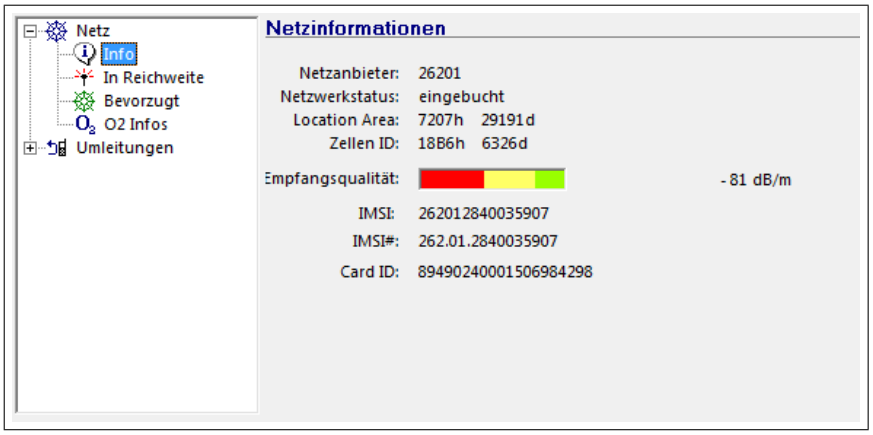

Abbildung 6.2: Siemens C35 und s25@once!-Software: T-Mobile Netzinformationen (Location Area, Cell ID, Empfangsqualität)

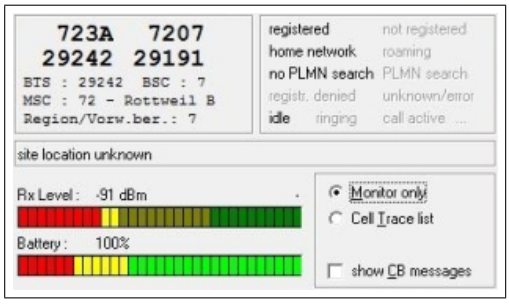

Abbildung 6.3: Siemens C35 und Tapir-G Netzmonitor - T-Mobile Netzinformationen

6.2 USRP und GNU Radio – Spektrumsanalyse

GNU Radio bringt eine Vielzahl von Programmbeispielen mit. Diese werden nach der Installation in den Verzeichnissen */usr/local/bin* und *gnuradio/gnuradioexamples/python/usrp/* abgelegt (z.B. Oszilloskop, Sepektrumsanalysator, FM Radio-Empfänger). Darin befindet sich auch ein Python-Skript *usrp_fft.py*. Hierbei handelt es sich um einen Software Spektrumsanalysator, der für die weiteren Untersuchungen und Experimente zum Thema Frequenzen genutzt wird. Mit Hilfe dieses Analysators soll das gesamte GSM-Frequenzband untersucht werden, um Base Transciever Stations am entsprechenden Aufenthaltsort zu erkennen. Das mögliche Frequenzspektrum des GSM-Bandes wurde bereits in 2.2 genauer beschrieben und kann Tabelle 2.2 entnommen werden. Die Experimente wurden im Rechenzentrum der Universität Freiburg durchgeführt.[4]

Wie in Abschnitt 5.1 beschrieben, wird ein USRP verwendet, ein RX900 Daughterboard zum Empfang der Analysedaten und softwareseitig GNU Radio, um das GSM Band scannen zu können. Das RX900 Board wird auf den RX-Slot des USRP-Mainboards aufgesteckt und mittels SMA-Kabel eine externe Rundstrahlantenne angeschlossen. Da das Board über einen empfangbaren Frequenzbereich von 800 bis 1000 MHz verfügt, können damit sämtliche GSM-Signale im Downlink-Bereich des 900er Bandes empfangen und analysiert werden (925.0 - 960,0 MHz). Weil lediglich die Signale, die von der BTS zu der entsprechenden Mobilstation gesendet werden, analysiert werden sollen, reicht die Betrachtung der Empfangsrichtung aus.

[4] 1.OG des Rechenzentrums in der Hermann-Herder-Str. 10 in Freiburg

Durch die Installation von GNU Radio stehen einige Beispielprogramme zur Verfügung. Hierbei handelt es sich meist um Python-Skripte. Mit folgendem Befehl kann ein Scanvorgang begonnen werden: *usrp_fft.py -R A -d 8 -g 47 -f 928M*. Dieser Befehl sorgt dafür, dass ein Spektrumsanalysator gestartet wird und um die Frequenz von 928 MHz (± 4 MHz) nach BTSs in diesem Frequenzbereich gescannt wird. Das Ergebnis und sonstige Einstellungen wie „Average“ und „Peek Hold“ können Abbildung 6.4 entnommen werden. Es ist deutlich zu erkennen, dass es mehrere Ausschläge im Bereich um 928 MHz gibt (rot markiert). Hierbei handelt es sich um 200 kHz breite Kanäle.

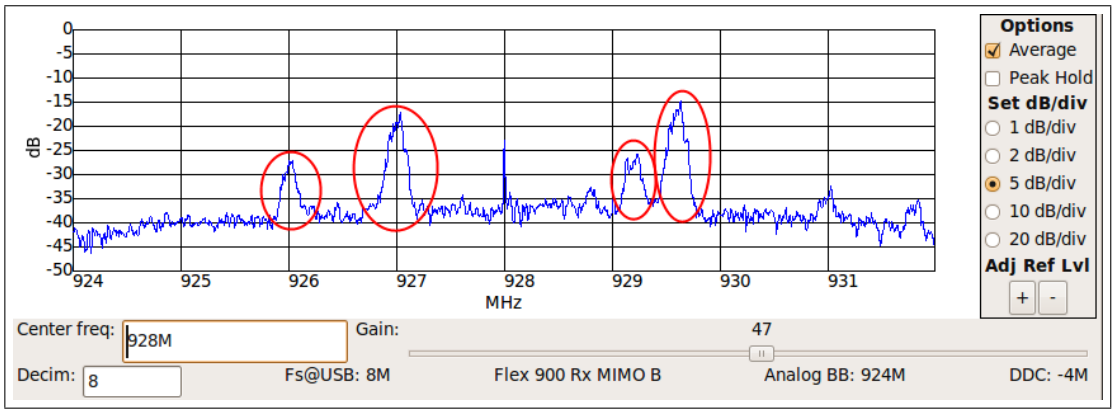

Abbildung 6.4: Spektrumsanalyse im Bereich 924-932 MHz mittels GNU Radio-Skript: *usrp_fft.py*

Es wird der größte Ausschlag bei 927.0 MHz gewählt und genauer betrachtet (BTS mit bester Sende- bzw. Empfangsstärke). Die Anzeige des Spektrums-Analysators wird auf 927.0 MHz fokussiert und eine feinere Auflösung der Frequenz (=Decim) gewählt: erst Decim=60 (Abbildung 6.5) und anschließend Decim=112 (Abbildung 6.6). Dies erfolgt mit dem Befehl: *usrp_fft.py*

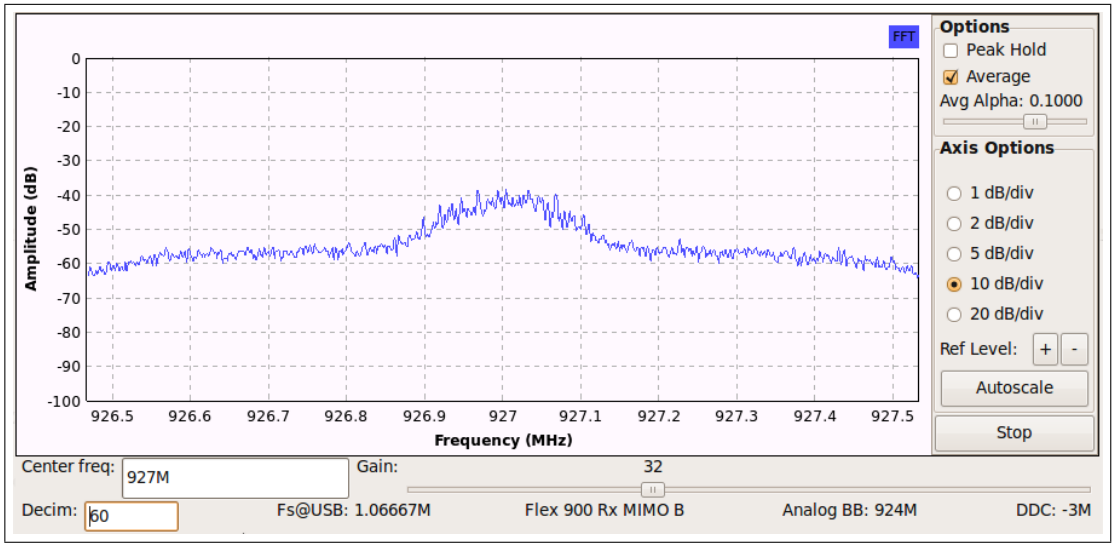

Abbildung 6.5: Frequenzspektrum eines BTS-Kanals bei 927.0 MHz (Decim=60)

-R A -d 112 -g 32 -f 927M. In Abbildung 6.6 wurde außerdem die Option „Peek Hold“ verwendet. Diese sorgt dafür, dass der größte Ausschlag gespeichert wird (grüne Linie). Es ist deutlich bei +67,7 kHz von dem Zentrum des Kanals entfernt ein Ausschlag (Peek) zu erkennen. Hierbei handelt es sich um ein FCCH-Paket, das periodisch alle zehn Pakete im Zeitschlitz 0 übertragen

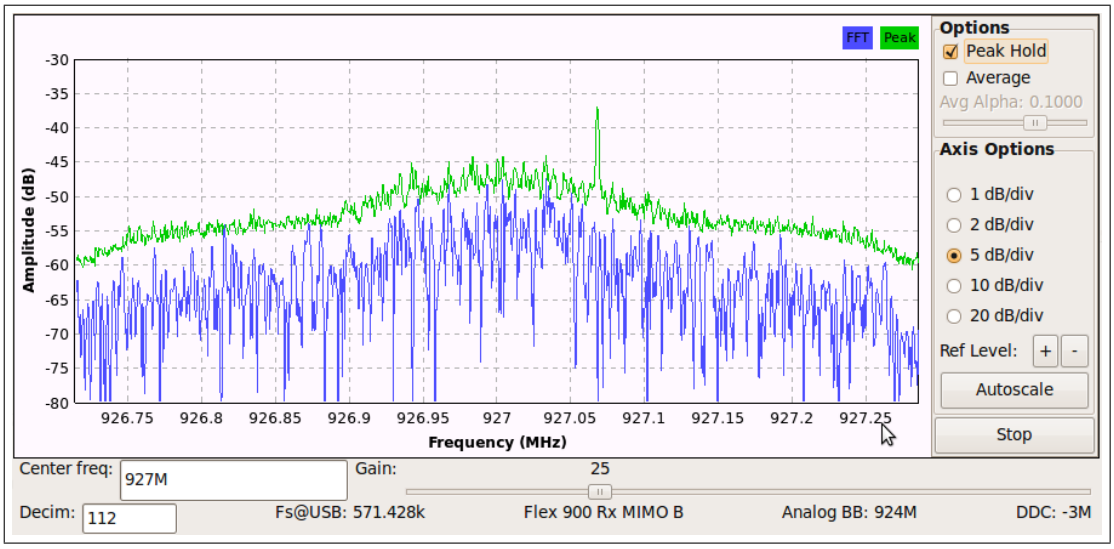

Abbildung 6.6: Frequenzspektrum eines BTS-Kanals bei 927.0 MHz (Decim=112)

wird. Die genauere Aufgabe dieses Paketes kann Kapitel 2.4.3 entnommen werden. Es dient der Frequenzkorrektur und zur Auswahl der Zelle des BTS mit der besten Empfangsfrequenz. Die Messergebnisse wurden mittels USRP2 noch einmal verifiziert (Abbildung 6.7). Der Kanal und der Peek bei 927.071 MHz (theoretischer Wert liegt bei 927.068 MHz) sind ebenfalls deutlich zu erkennen.

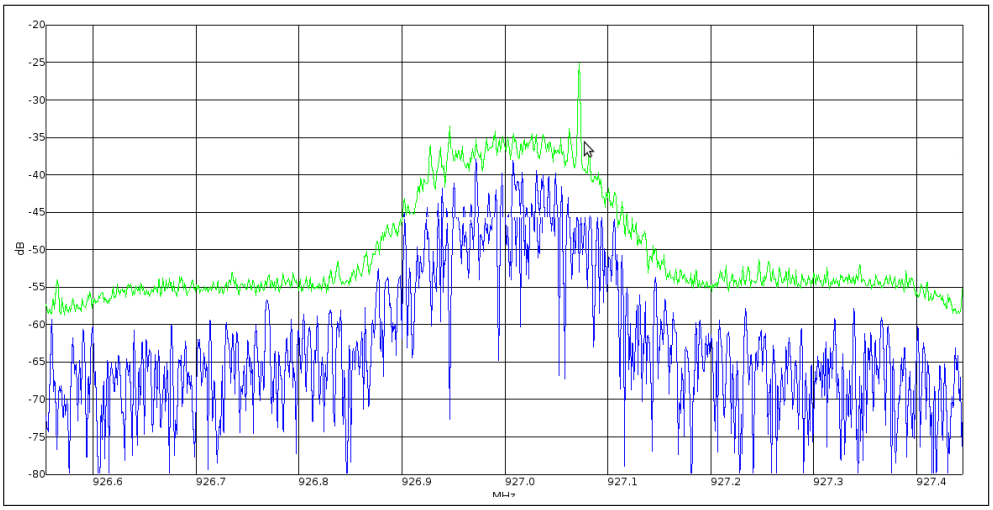

Abbildung 6.7: Frequenzspektrum eines BTS-Kanals bei 927.0 MHz mittels USRP2

6.2.1 OpenBTS Spektrumsanalyse

Mittels Konfigurationsdatei wurde für OpenBTS ARFCN 29 gewählt. Somit ergibt sich eine Betriebsfrequenz von 940,8 MHz. Dies lässt sich mittels eines Onlinerechners[5] (siehe Abbildung 6.8) errechnen. Durch eine Spektrumsanalyse mittels USRP2 konnte diese Frequenz bestätigt werden. Die OpenBTS lässt sich deutlich bei 940,8 MHz erkennen (Abbildung 6.9).

[5] ARFCN Calculator: ARFCN to Frequency Converter, *http://www.aubraux.com/design/arfcn-calculator.php* [Online; letzter Aufruf 29.09.2009]

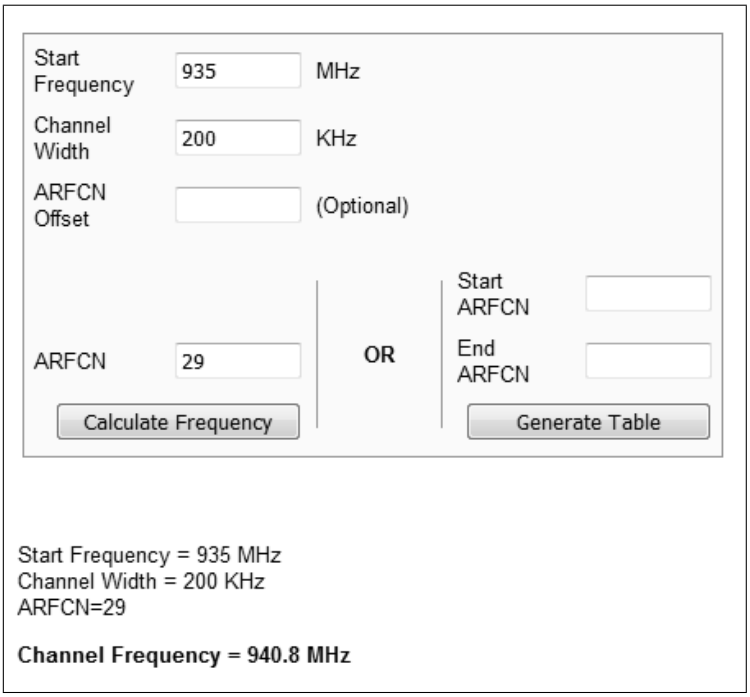

Abbildung 6.8: ARFCAN - Frequenz Umrechner

6.3 USRP – AirProbe und GSSM

Bei der Softwaresammlung AirProbe handelt sich es um ein GSM-Sniffer-Projekt des Chaos Computer Clubs. Ziel dieses Projektes ist es, ein Analyse-Tool zu schaffen, das in der Lage ist, GSM-Daten (später vielleicht auch UMTS-Daten) auf dem Air-Interface zu analysieren. Dadurch ist es möglich, die GSM-Technologie besser verstehen zu können und auch konkret sichtbar zu machen. Außerdem können Projekte wie OpenBTS von diesen Informationen profitieren und weiterentwickelt werden. Der dritte und vielleicht wichtigste Aspekt des Projektes ist es, die Sicherheitslücken des GSM-Standards zu demonstrieren. Das AirProbe-Projekt gliedert sich in drei Hauptprojekte: Erfassung von Daten, Demodulation und Analyse. Genauere Informationen zu den einzelnen Projekten sind auf der AirProbe-Webseite zu finden.

Analyse

Für sämtliche Tests wurde die Anfang August 2009 aktuelle AirProbe-Version aus dem Git-Repository[6] des Chaos Computer Clubs verwendet [5]. Eine genaue Installationsanleitung kann dem AirProbe-Wiki entnommen werden.[7] Der GSM-Receiver wurde allerdings durch eine von Piot Krysik weiterentwickelte Version vom 28.06.2009 ersetzt.[8] Diese befindet sich auch auf der CD im Ordner */GSM-Analyse/AirProbe/* unter dem Dateinamen *gsm-receiver.tar.gz*.
Der Ordner *gsmdecode/src/python* des AirProbe Verzeichnisses enthält zwei Skripte: *capture.sh* und *go.sh*. Ersteres dient dazu, mittels USRP und GNU Radio Daten aufzuzeichnen, und das zweite, diese zu dekodieren. Als Vorbedingung ist es nötig, wie in Kapitel 6.2 beschrieben, eine Frequenz zu ermitteln, auf der eine Funkzelle sendet. Als geeignet stellte sich die Frequenz 927.0 MHz heraus. Der Aufzeichnungsvorgang konnte mit folgendem Befehl gestartet werden:

[6] AirProbe Git-Repository, *git://svn.berlin.ccc.de/* [Online; letzter Aufruf 25.10.2009]
[7] WorkingWithTheUsrp - airprobe, *https://svn.berlin.ccc.de/projects/airprobe/wiki/WorkingWithTheUSRP* [5] [Online; letzter Aufruf 29.09.2009]
[8] Index of /∼pkrysik/GSM, *http://home.elka.pw.edu.pl/∼pkrysik/GSM/* [Online; letzter Aufruf 29.09.2009]

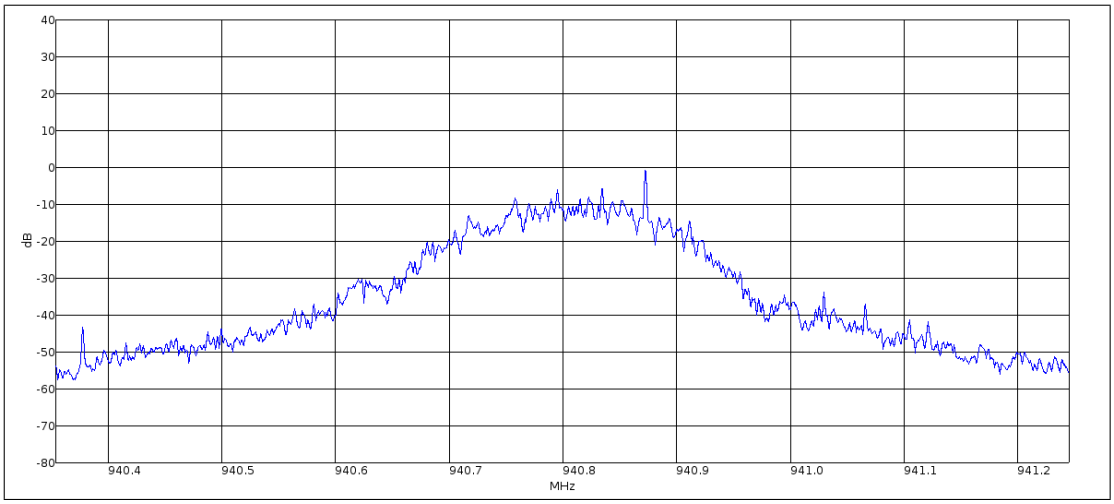

Abbildung 6.9: OpenBTS bei 940.8 MHz mittels USRP2

./capure.sh 927.0M. Die damit erzeugte, 10 Sekunden lange Capture-Datei *capture_ 927.0M_ 112 .cfile* befindet sich im Ordner */GSM-Analyse/AirProbe/* der CD. Diese wurde mittels des Skripts *go.sh* analysiert und dekodiert. Abbildung 6.10 und 6.11 zeigen jeweils einen Ausschnitt. Es handelt sich, wie zu erkennen ist, um eine E-Plus Funkzelle. Diverse Parameter wie Mobile Country und Network Code, Ordinary Subscribers und Emergency Call konnten ebenfalls sichtbar gemacht werden. Der in Kapitel 6.2 beschriebene Vorgang zur Ermittlung der einzelnen Funkzellen kann somit mit dem AirProbe-Projekt kombiniert werden, um die ermittelten Funkzellen genauer zu analysieren und um detaillierter bestimmen zu können, um welche Funkzelle es sich handelt.

6.3.1 GSSM

GSSM ist ein Softwarepaket, das in der Lage ist, die GSM Base Station Control Channels zu überwachen. Hierzu werden ein USRP und diverse Empfangskarten benötigt. Die Analyse der gesammelten Pakete erfolgt in einem gepatchten Wireshark mittels eines virtuellen TUN-Interfaces. GNU Radio übernimmt dabei die Demodulation und Dekodierung der einzelnen GSM-Pakete. Folgende Kontrollkanäle (zwischen BTS und MS) können mittels GSSM v.0.1.1.1a decodiert werden:

- FCCH
- SCH
- BCCH
- PCH (nur Downlink)
- AGCH (nur Downlink)
- SACCH
- SDCCH

```
HEX l2_data_out_Bbis:462 Format Bbis DATA
000: 49 06 1b d7 c6 62 f2 30  - 02 4c c9 05 78 46 65 03
001: 94 00 00 89 1f 40 4b
    0: 49 010010-- Pseudo Length: 18
    1: 06 0------- Direction: From originating site
    1: 06 -000---- 0 TransactionID
    1: 06 ----0110 Radio Resouce Management
    2: 1b 00011011 RRsystemInfo3C
    3: d7 55238    [0xd7c6] Cell identity
    5: 62 262      Mobile Country Code (Germany)
    6: f2 03f      Mobile Network Code (E-Plus Mobilfunk GmbH & Co. KG)
    8: 02 588      [0x024c] Local Area Code
   10: c9 1------- Spare bit (should be 0)
   10: c9 -1------ MSs in the cell shall apply IMSI attach/detach procedure
   10: c9 --001--- Number of blocks: 1
   10: c9 -----001 1 basic physical channel for CCCH,     combined with SDCCHs
   11: 05 00000--- spare bits (should be 0)
   11: 05 -----101 7 multi frames period for paging request
   12: 78 01111000 T3212 TimeOut value: 120
   13: 46 0------- spare bit (should be 0)
   13: 46 -1------ Power control indicator is set
   13: 46 --00---- MSs may use uplink DTX
   13: 46 ----0110 Radio Link Timeout: 28
   14: 65 011----- Cell Reselect Hyst. :  6 db RXLEV
   14: 65 ---xxxxx Max Tx power level: 5
   15: 03 0------- No additional cells in SysInfo 7-8
   15: 03 -0------ New establishm cause: not supported
   15: 03 --xxxxxx RXLEV Access Min permitted = -110 + 3dB
   16: 94 10------ Max. of retransmiss : 4
   16: 94 --0101-- slots to spread TX : 8
   16: 94 ------0- The cell is barred  : no
   16: 94 -------0 Call reestabl.i.cell: allowed
   17: 00 -----0-- Emergency call EC 10: allowed
   17: 00 00000--- Acc ctrl cl 11-15: 0 = permitted, 1 = forbidden
   17: 00 ------00 Acc ctrl cl  8- 9: 0 = permitted, 1 = forbidden
   17: 00 -------0 Ordinary subscribers (8)
   17: 00 ------0- Ordinary subscribers (9)
   17: 00 -----0-- Emergency call (10): Everyone
   17: 00 ----0--- Operator Specific (11)
   17: 00 ---0---- Security service (12)
   17: 00 --0----- Public service (13)
   17: 00 -0------ Emergency service (14)
   17: 00 0------- Network Operator (15)
```

Abbildung 6.10: Konsolen-Output der Analyse mit *go.sh capture_ 927.0M_ 112.cfile* (E-Plus)

```
001: 08 20 00 88 a5 00 00
   0: 59 010110-- Pseudo Length: 22
   1: 06 0------- Direction: From originating site
   1: 06 -000---- 0 TransactionID
   1: 06 ----0110 Radio Resouce Management
   2: 1a 00011010 RRsystemInfo2
   3: 00 ---x---- BCCH alloc. seq. num: 0
   3: 00 00------ Bitmap 0 format
   7: e4 1------- BCCH Allocation    : ARFCN 96
   7: e4 -1------ BCCH Allocation    : ARFCN 95
   7: e4 --1----- BCCH Allocation    : ARFCN 94
   7: e4 -----1-- BCCH Allocation    : ARFCN 91
   8: 02 ------1- BCCH Allocation    : ARFCN 82
  13: 04 -----1-- BCCH Allocation    : ARFCN 43
  14: 40 -1------ BCCH Allocation    : ARFCN 39
  15: 06 -----1-- BCCH Allocation    : ARFCN 27
  15: 06 ------1- BCCH Allocation    : ARFCN 26
  16: 08 ----1--- BCCH Allocation    : ARFCN 20
  17: 20 --1----- BCCH Allocation    : ARFCN 14
  19: 88 1------- BCCH carrier with NCC = 7 is permitted for monitoring
  19: 88 ----1--- BCCH carrier with NCC = 3 is permitted for monitoring
  20: a5 10------ Max. of retransmiss : 4
  20: a5 --1001-- slots to spread TX : 12
  20: a5 ------0- The cell is barred  : no
  20: a5 -------1 Cell reestabl.i.cell: not allowed
  21: 00 -----0-- Emergency call EC 10: allowed
  21: 00 00000--- Acc ctrl cl 11-15: 0 = permitted, 1 = forbidden
  21: 00 ------00 Acc ctrl cl  8- 9: 0 = permitted, 1 = forbidden
  21: 00 -------0 Ordinary subscribers (8)
  21: 00 ------0- Ordinary subscribers (9)
  21: 00 -----0-- Emergency call (10): Everyone
  21: 00 ----0--- Operator Specific (11)
  21: 00 ---0---- Security service (12)
  21: 00 --0----- Public service (13)
  21: 00 -0------ Emergency service (14)
  21: 00 0------- Network Operator (15)
  22: 00 00000000 Acc ctrl cl  0- 7: 0 = permitted, 1 = forbidden
```

Abbildung 6.11: Konsolen-Output der Analyse mit *go.sh capture_927.0M_112.cfile* (E-Plus: BCCH)

Installation

GSSM setzt ein funktionstüchtiges GNU Radio voraus. Es kann mit dem Befehl *./bootstrap && ./configure && make && sudo make install* kompiliert und installiert werden. Anschließend muss Wireshark mittels des beiliegenden Patchs modifiziert,[9] kompiliert und installiert werden. Es wurde die Version 0.99.5 aus dem Wireshark SVN-Repository[10] verwendet. GSSM benutzt zur Übertragung der Pakete an Wireshark ein TUN-Interface. Das entsprechende Kernel-Modul *tun.ko* muss geladen sein.[11] Dieses TUN-Interface kann mit Hilfe des Programmes *mktun* erstellt werden.[12]

Anwendung

Wie in Kapitel 6.2 beschrieben, muss zuvor die Frequenz einer aktiven BTS ermittelt werden. Die Signalstärke sollte dabei möglichst hoch sein, um ein bestmögliches Analyseergebnis zu erzielen. Der ermittelte Frequenzwert muss manuell in das Python-Skript *gssm_ usrp.py* im Ordner *gssm/src/* eingetragen werden. Ebenfalls in diesem Skript sollten die Parameter *Decim* auf 112 und *Gain* je nach Empfangstärke auf 80-100 geändert werden. Mit dem Befehl *./mktun gsm* wird das benötigte TAP-Interface erstellt. Anschließend kann das Skript gestartet werden: *./gssm_ usrp.py*. Es wird eventuell eine Vielzahl von Fehlern angezeigt, die nicht weiter beachtet werden müssen. Um die live mitgeschnittenen Daten sichtbar zu machen, muss Wireshark gestartet werden und das erstellte GSM-Interface zur Überwachung ausgewählt werden. Eine Aufzeichnung der Funkzelle bei 927.0 MHz befindet sich auf der CD im Ordner */GSM-Analyse/GSSM* unter dem Namen *wireshark_ capture_ GSSM_ 927MHz*. Abbildung 6.12 zeigt Wireshark mit einen Ausschnitt der ermittelten GSM-Pakete.

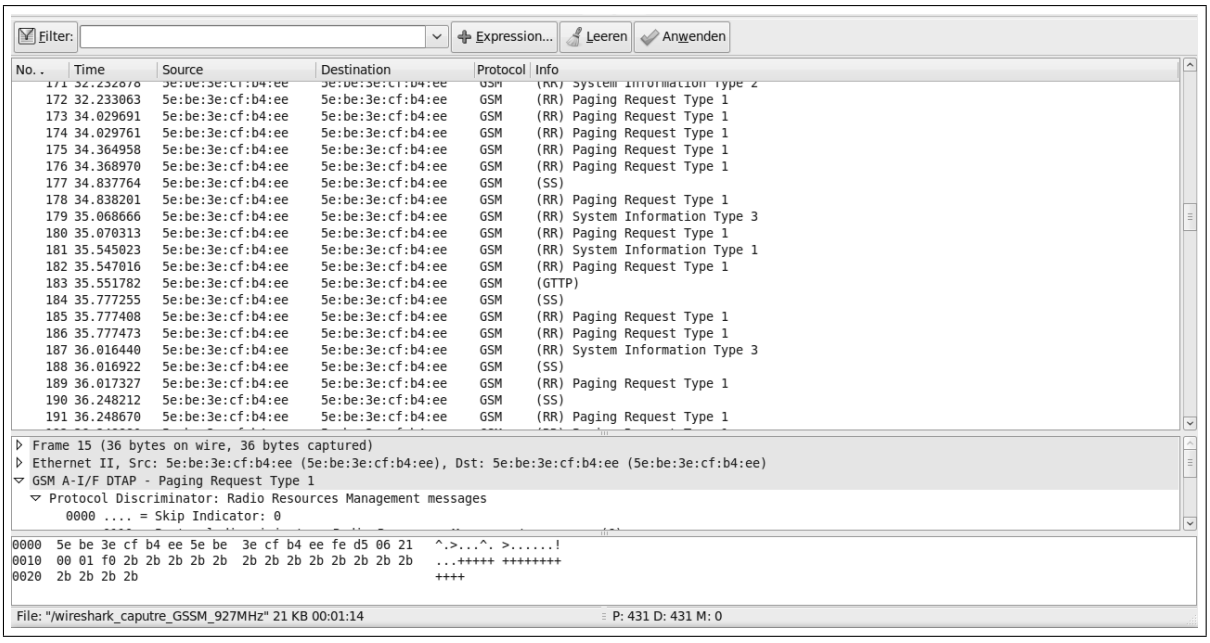

Abbildung 6.12: GSSM und Wireshark bei 927.0 MHz

[9] *patch -p1 < /src/gssm/patch/wireshark-0.99.5-gssm.patch*
[10] Revision 30195, *http://anonsvn.wireshark.org/wireshark/releases* [Online; letzter Aufruf 29.09.2009]
[11] kann mittels *lsmod grep tun* überprüft werden
[12] *sudo ./usr/local/bin/mktun gsm*

Schwächen und Fehler

GSSM ist lediglich in der Lage, GSM-Pakete von der BTS zur MS zu überwachen, aber nicht umgekehrt (nur Downlink). Der momentane Entwicklungstand der Software hat das Alpha-Stadium noch nicht verlassen, so dass mit einer Vielzahl an Fehlern und Problemen gerechnet werden muss. Die U_m-Schnittstelle wurde bisher auch nur teilweise implementiert und viele Pakete können von Wireshark nicht korrekt identifiziert werden, da sie über eine abweichende Protokoll-Beschreibung verfügen. Die Identifizierung kann beispielsweise durch einen unterschiedlichen Frame-Type fehlschlagen. Auch die Schnittstelle zwischen GSSM und Wireshark arbeitet äußerst unzuverlässig, weil es bisher kein standadisiertes GNU Radio Interface für Wireshark gibt. Für zukünftige Versionen wäre es außerdem wünschenswert, dass die GSM-Frequenz automatisch eingestellt wird und nicht erst noch mühsam manuell ermittelt und in das Python-Skript eingetragen werden muss.

6.4 GSM Decodierung mit Nokia 3310 und Wireshark

Das Mobiltelefon Nokia 3310 ist in der Lage, GSM-Nachrichten aus einem „Gammu Trace Log“ zu dekodieren. Es ist möglich, Signalisierungsprozesse auf Layer 2 (LAPDm) in Sende- und Empfangsrichtung sichtbar zu machen. Diese Tatsache beruht darauf, dass die Entwickler eine Loggingfunktion eingebaut hatten, um ihre DCT3-Firmware debuggen zu können. Es wurde allerdings vergessen, diese Loggingfunktion wieder zu entfernen, so dass alle Nokia 3310 Mobiltelefone über die Möglichkeit verfügen, diese zu aktivieren. In den folgenden Abschnitten wird dieser Aktivierungsvorgang genauer beschrieben. Die generierten XML-Dateien können mit Wireshark geöffnet und analysiert werden. Alternativ kann zur Analyse der aufgezeichneten Dateien auch das Programm *Gsmdecode* des Chaos Computer Clubs verwendet werden. Es ist genau wie Wireshark in der Lage, GSM-Nachrichten zu dekodieren. Dieses Programm kann über die Webseite *www.airprobe.org* heruntergeladen werden.[13]

Abbildung 6.13: Nokia 3310 und MBUS NK-33 Datenkabel samt Seriell-zu-USB-Konverter

[13] *https://svn.berlin.ccc.de/projects/airprobe/attachment/wiki/tracelog/gsmdecode-0.7bis.tar.gz* [5] [Online; letzter Aufruf 29.09.2009]

6.4.1 Voraussetzungen und Installation

Hardwareseitig wird ein Nokia 3310 Mobiltelefon und ein spezielles MBUS NK-33 Datenkabel benötigt.[14] Abbildung 6.13 zeigt ein Nokia 3310 mit entsprechendem Datenkabel.
Eine genaue Installationsanleitung für die Software kann dem AirProbe Wiki des CCC entnommen werden.[15] Es werden die Pakete *gammu*,[16] Wireshark[17] und *dialog* benötigt. Gammu wurde wie folgt konfiguriert (Konfigurationsdatei unter *∼/.gammurc* oder mit dem Befehl *gammu-config*):

```
[gammu]
port = /dev/ttyUSB0
model = 6110
connection = mbus
synchronizetime = yes
logfile =
logformat = nothing
use_locking = yes
gammuloc =
```

6.4.2 Echte Netze analysieren

Abbildung 6.14 zeigt einen Wireshark Screenshot. Es ist ein Gesprächsaufbau des Nokia 3310 Mobiltelefons mit eingesteckter T-Mobile-Karte zu sehen. Die entsprechenden Pakete konnten parallel mittels Gammu-Software aufgezeichnet werden. Das dazugehörige Log-File kann mit Wireshark betrachtet werden und ist auf der beigelegten CD im Ordner */GSM-Analyse/Nokia3310/Wireshark/* unter dem Namen *Rufaufbau_Pakete_Nokia3310_im_echten_T-Mobile_Netz.xml* zu finden. Die aufgezeichneten Daten decken sich mit den auf der AirProbe-Webseite zur Verfügung gestellten Vergleichsdateien *call_1525.xml* und *call_init.xml*.[18]
Darüber hinaus wurde noch eine SMS-Nachricht mittels Nokia E71 an das Nokia 3310 gesendet und mittels Gammu-Tracelog aufgezeichnet. Abbildung 6.15 zeigt einen Ausschnitt der generierten Pakete. Die vollständige Aufzeichung ist auf der CD im Ordner */GSM-Analyse/Nokia3310/Wireshark/* unter dem Namen *SMS_Nokia_E71_an_Nokia_3310_echtes_Vodafonenetz.xml* zu finden und kann mittels Wireshark 1.2.1 betrachtet werden. Der Nachrichteninhalt der SMS konnte auf Grund der eingeschalteten Verschlüsselung nicht eingesehen werden.

6.4.3 OpenBTS-Netz analysieren

Abbildung 6.16 zeigt den Gesprächsaufbau eines iPhone 2Gs zu einem Nokia 3310 über die OpenBTS. Vergleicht man nun die aufgezeichneten XML-Dateien *Rufaufbau_Pakete_Nokia3310_im_echten_T-Mobile_Netz.xml* und *Gespräch_Nokia_3310_zu_iPhone2G_OpenBTS.xml* kann man feststellen, dass bei OpenBTS keine Verschlüsselung verwendet wurde. Dieser Unterschied und der detaillierte Gesprächsaufbau wurde in Kapitel 5.2.2 in Abbildung 5.8 verdeutlicht.
Auch bei einer gesendeten SMS-Nachricht war zu sehen, dass keine Verschlüsselung stattgefunden hat. Dazu wurde mittels der OpenBTS Konsole an das Nokia 3310 eine SMS-Nachricht geschickt.

[14] N-33 Nokia Cable 3310, 3330, 3390 with MBUS Interface (compatible), *http://ucables.com/ref/NK-33* [Online; letzter Aufruf 29.09.2009]
[15] Tracelog - AirProbe, *https://svn.berlin.ccc.de/projects/airprobe/wiki/tracelog* [5] [Online; letzter Aufruf 29.09.2009]
[16] *http://www.gammu.org/wiki/index.php?title=Download* - Version: 1.26.1
[17] Version 1.2.1
[18] siehe CD: */GSM-Analyse/Nokia3310/Vergleichsdateien_airprobe.org/*

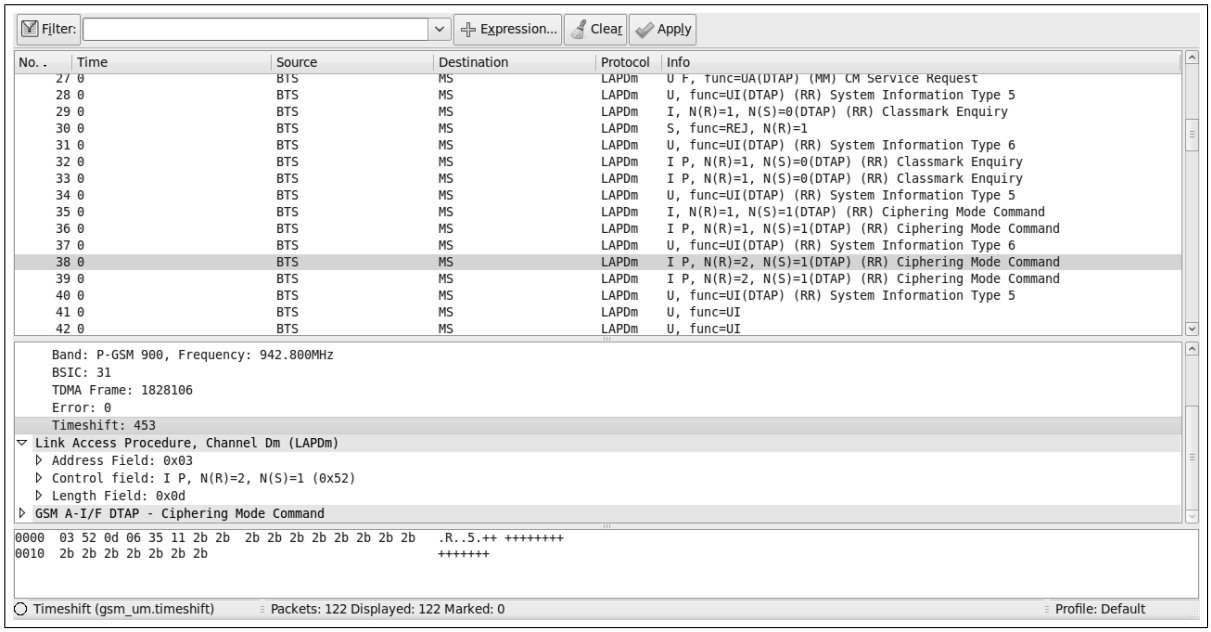

Abbildung 6.14: Wireshark-Mitschnitt: Gesprächsaufbau im T-Mobile Netz eines Nokia 3310

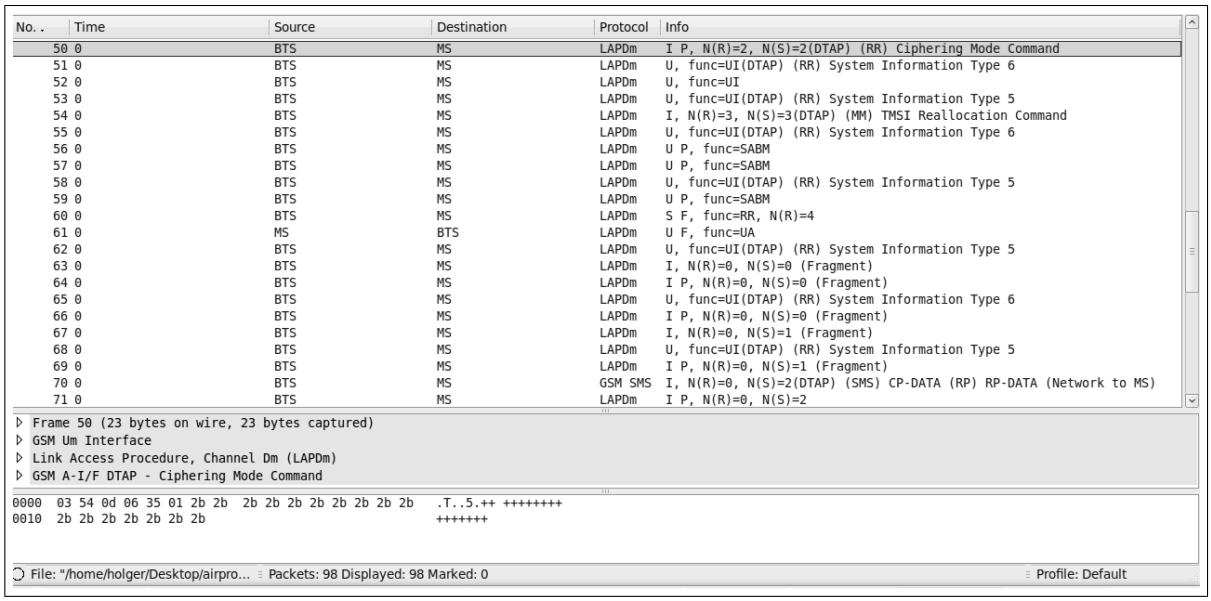

Abbildung 6.15: Wireshark-Mitschnitt: SMS im Vodafone-Netz an Nokia 3310

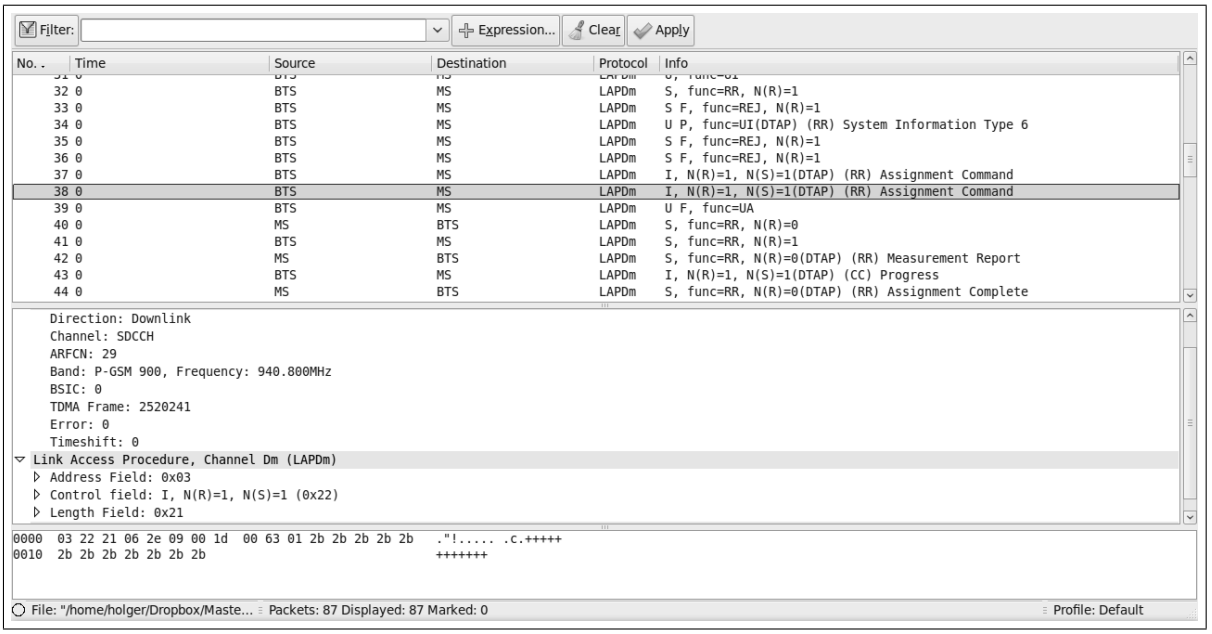

Abbildung 6.16: Wireshark-Mitschnitt: Gespräch über OpenBTS (iPhone/Nokia3310)

Der Vorgang und die dabei ablaufenden Signalisierungsprozesse wurden aufgezeichnet.[19] Der Inhalt dieser Nachricht konnte in Wireshark unverschlüsselt mitgelesen werden (siehe Abbildung 6.17).

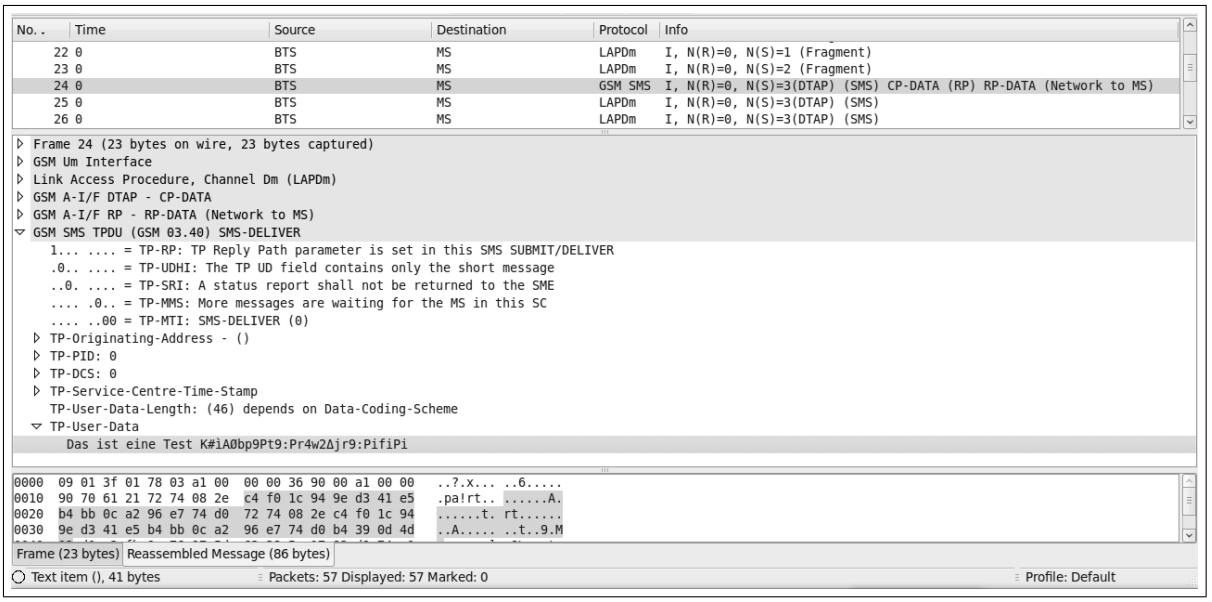

Abbildung 6.17: Wireshark-Mitschnitt: SMS von OpenBTS an Nokia 3310

[19] auf der CD unter dem Namen *SMS_von_OpenBTS_an_Nokia_3310* im Ordner */GSM-Analyse/Nokia3310/Wireshark/*

6.4.4 Handover-Szenario

Mit Hilfe von OpenBTS, einem Nokia 3310 und Wireshark kann für Lehrzwecke ein mögliches Handover-Szenario konstruiert und anschaulich vermittelt werden. OpenBTS verfügt noch nicht über die Möglichkeit, ein Handover zu einer anderen OpenBTS-Zelle oder zu einer Original-Mobilfunkzelle durchzuführen. Dies ist erst für zukünftige Versionen >2.4 geplant. Es ist allerdings möglich, Mobiltelefone aus Original-GSM-Zellen per Handover während eines Gesprächs zu einem Zellwechsel zu OpenBTS anzuregen. Das Handover wird letztendlich nicht durchgeführt, da OpenBTS die erhaltenen Handover-Befehle noch nicht verarbeiten kann. Die Idee dabei ist, durch OpenBTS eine ARCFAN aus einer benachbarten Zelle der Original-Zelle zu imitieren und somit auf dieser Frequenz die besten Sende- und Empfangsstärke zu bekommen. In Kapitel 7.1 wird beschrieben, wie es mit den richtigen Einstellungen der Parameter möglich ist, Original-Funkzellen der Mobilfunkbetreiber zu simulieren. Die Identifikation dieser Parameter erfolgt dazu beispielsweise mittels des Nokia Netzmonitors (siehe Kapitel 6.1) oder AirProbe (siehe Kapitel 6.3). Es wird eine benachbarte Zelle gewählt, die über extrem schlechte Empfangswerte verfügt. Die OpenBTS-Zelle wird mit demselben (M)CC und (M)NC betrieben wie diese Original-Zelle mit schlechtem Empfang (optional sollte der Short-Name ebenfalls dem der Original-Zelle entsprechen, z.B. Vodafone). Der wichtigste einzustellende Parameter, der mit oben genannten Methoden ermittelt wurde, ist die ARFCN und somit die Frequenz. Diese muss exakt der der Original-Zelle entsprechen. Eine Beispielskonfiguration könnte somit wie folgt aussehen:

- **ARFCN**: 57 $\equiv$ 946.4 MHz
- **GSM.MNC $\equiv$ (M)NC**: 02
- **GSM.MCC $\equiv$ (M)CC**: 262
- **GSM.ShortName**: Vodafone

Die Mobiltelefone messen in regelmäßigen Abständen während des Gesprächs die Empfangsstärke (Sendestärke der BTS) und teilen das Ergebnis der BTS mit (*Measurement Report*), die ihrerseits diese Werte an den BSC weiterleitet. Da die OpenBTS-Zelle eine Zelle nachahmt, die vorher extrem schlechte Empfangswerte hatte, steigt die gemessene Empfangstärke bei einem Mobiltelefon, das sich im unmittelbaren Empfangsbereich der OpenBTS-Zelle befindet. Idealerweise ist die OpenBTS-Zelle nun die GSM-Zelle mit den besten Empfangswerten. Dieses Szenario entspricht quasi einer schnellen Bewegung aus dem Empfangsbereich der aktuellen Zelle in den Empfangsbereich einer weit entfernten Zelle, in diesem Fall zur OpenBTS-Zelle. Der BSC möchte, auf Grund der ihm mitgeteilten Messwerte des Mobiltelefons ein Handover einleiten und schickt dazu an das Mobiltelefon einen Handover-Befehl (Handover Command). Dieser enthält verschiedenste Konfigurationsparameter wie Frequenz und Timeslot der Zelle, in den das Mobiltelefon wechseln soll, und diverse Synchronisationsparameter. OpenBTS kann im momentanen Entwicklungsstand dem Handover-Wunsch des Mobiltelefons noch nicht nachkommen, da -wie bereits erwähnt- die Implementierung dazu noch fehlt. In zukünftigen Versionen ist allerdings eine Hand over-Implementierung geplant. Ein Handover von OpenBTS zu einer Original-Zelle ist somit ebenfalls momentan nicht möglich; hier werden nicht einmal die notwendigen Handover-Befehle generiert. Das komplette Szenario ist in Abbildung 6.18 noch einmal verdeutlicht. Das Mobiltelefon ist in BTS1 eingebucht und misst in regelmäßigen Abständen die Empfangswerte zu allen BTS aus der Nachbarschaftsliste der eingebuchten BTS. Diese Messwerte (Measurement Reports) werden der aktuellen BTS mitgeteilt, die wiederum diese an den zuständigen BSC weiterleitet. Durch OpenBTS wird eine Zelle aus der Nachbarschaft (BTS3, ARFCN 57) vorgetäuscht, die bisher über recht schlechte Empfangswerte verfügte (-98dB). Der gemessene Empfangspegel steigt auf -60 dBm, was auch der BTS bzw. dem BSC in einem Measurement Report mitgeteilt wird. Dieser möchte ein Handover veranlassen und schickt entsprechende Handover-Befehle an das

Mobiltelefon.[20] Abbildung 6.19 zeigt einen Ausschnitt der erzeugten Wireshark-Pakete.[21]

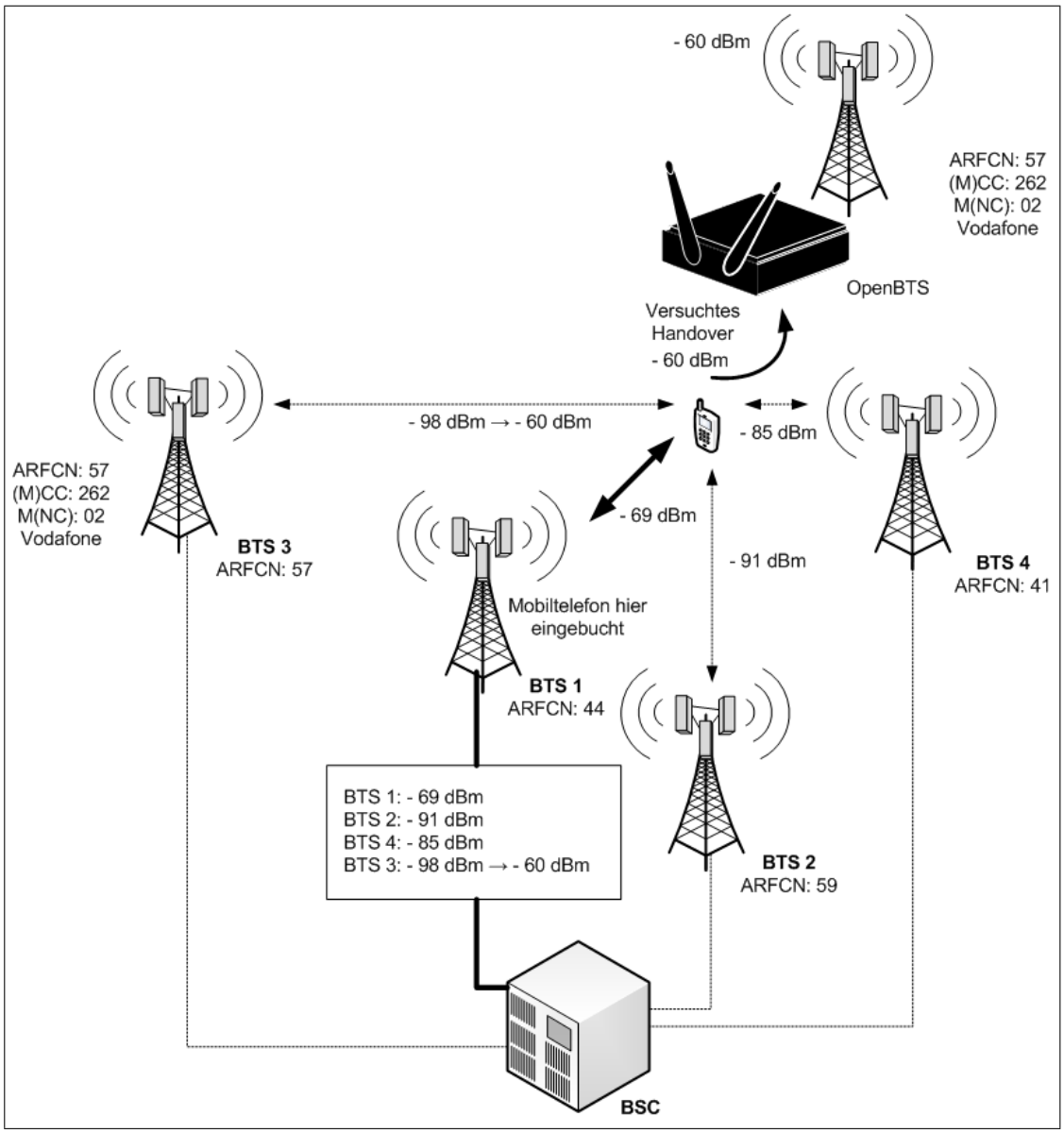

Abbildung 6.18: Handover Vodafone zu OpenBTS (simulierte Vodafone Funkzelle)

6.4.5 Erzwungener Zellwechsel

Das im vorherigen Abschnitt 6.4.4 beschriebene Handover-Szenario beschreibt in leicht modifizierter Form auch einen erzwungenen Zellwechsel. Die Voraussetzung dazu ist, dass ein Mobiltelefon gerade kein Gespräch führt und sich lediglich im Standby-Zustand befindet, um auf ein mögliches Gespräch oder eine Textnachricht zu warten. Das Mobiltelefon überprüft in regelmäßigen Abständen die Broadcast-Kanäle der Nachbarzellen seiner aktuellen Zelle, in das es

[20] Diese mit dem Nokia 3310 aufgezeichneten Dateien eines solchen Handover-Szenarios sind auf der CD im Ordner *GSM-Analyse/Nokia3310/Wireshark* unter den Namem *Gespraech_zwischen_Nokia_3310_zu_Nokia_E71_und_Handover_Vodafone(AFCRN_64)_zu_OpenBTS (AFCRN_107).xml* zu finden.

[21] Es wird Version >1.2.1 von Wireshark benötigt

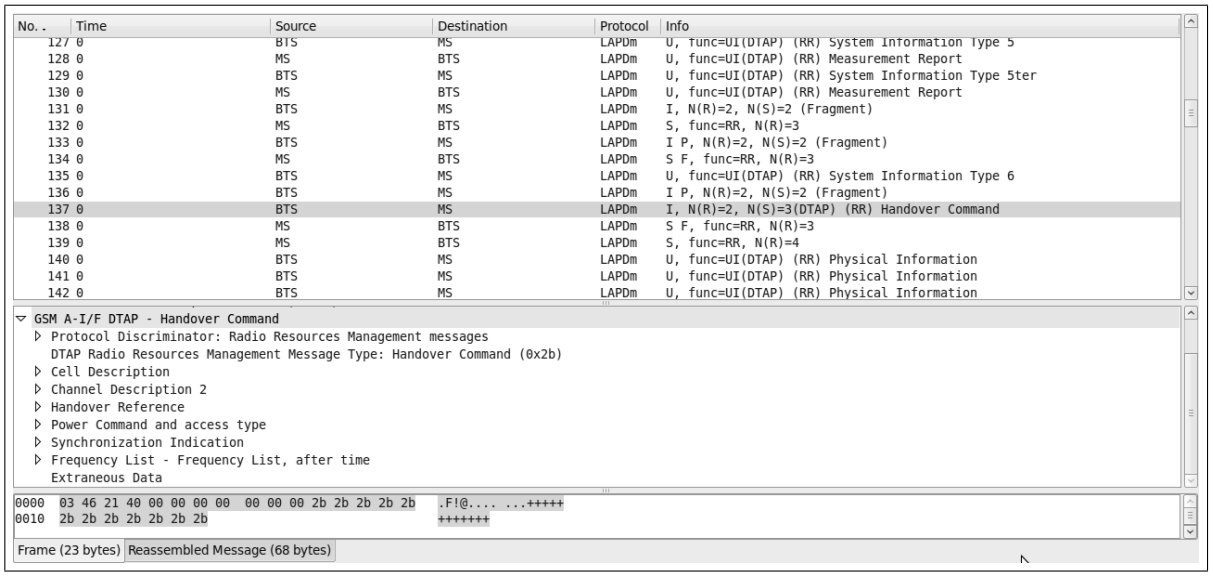

Abbildung 6.19: Wireshark-Mitschnitt: Versuchtes Handover Vodafone zu OpenBTS

eingebucht ist. Eine entsprechende Liste der Nachbarzellen erhält es in regelmäßigen Abständen ebenfalls über den Broadcast-Kanal seiner aktuellen Zelle. Wird nun eine Zelle aus dieser Nachbarschaftsliste durch OpenBTS nachgeahmt, verfügt sie plötzlich im unmittelbarem Empfangsradius des Mobiltelefons über die besten Empfangswerte. Das Mobiltelefon wird von sich aus einen Zellwechsel durchführen, da es in regelmäßigen Abständen den Empfangspegel misst. Dieser Zellwechsel kann sehr gut mit dem Nokia Netzmonitor[22] (siehe Abschnitt 6.1) beobachtet und nachvollzogen werden. Das Nokia 3310 Mobiltelefon war zunächst in eine Funkzelle mit der AFCRN 64 eingebucht. OpenBTS wurde auf einer Frequenz aus der Nachbarliste dieser Zelle betrieben (AFCRN 79) und nach kurzer Zeit wieder gestoppt. Das Mobiltelefon führte zunächst den Zellwechsel durch und begann nach der Deaktivierung der OpenBTS mit einer Netzsuche. Letztendlich wählte das Mobiltelefon eine neue Original-Zelle mit der AFCRN 68. Dieser Vorgang und zwei weitere Zellwechsel wurden mittels Gammu-Tracelog und dem Nokia 3310 aufgezeichnet und sind auf der CD im Ordner *GSM-Analyse/Nokia3310/Wireshark* zu finden.[23]

6.4.6 Ergebnis und Vergleich

Das schon etwas ältere Nokia 3310 Mobiltelefon bietet eine zuverlässige Analysemöglichkeit, verschiedene Abläufe in GSM sichtbar zu machen. Gerade der Rufaufbau eines Gespräches, Nachrichten auf den Broadcast-Kanälen, Handover, Zellwechsel und auch Dienste wie SMS können mittels Wireshark sichtbar gemacht werden. Da OpenBTS noch keine Verschlüsselung für SMS-Nachrichten einsetzt, kann der Inhalt dieser sogar im Klartext eingesehen werden (siehe Abbildung 6.17). Das Nokia 3310 bietet neben der Möglichkeit mittels Netzmonitor (siehe Kapitel 6.1) GSM-Abläufe, Einstellungen und verwendete Parameter sichtbar zu machen, durch diese beschriebene Analysemethode die Chance, mittels Wireshark Abläufe in einem GSM-Netz gerade für Lehrzwecke näher zu bringen. Besonders die notwendigen Schritte und gesendeten bzw. empfangenen Pakete zur Registrierung eines Mobiltelefons, eines Gesprächsaufbaus oder das Versenden einer SMS-Nachricht können schrittweise nachvollzogen werden.

[22] Monitor Mode 04 und 05, um den Empfangspegel der benachbarten Zellen zu überwachen

[23] unter den Namen: *Zellwechsel_ Vodafone(AFCRN_ 64)_ zu_ OpenBTS(AFCRN_ 79)_ OpenBTS_ beendet_ zurück_ ins_ Vodafonnetz(AFCRN_ 68)_ nach_ Suchvorgang.xml*, *Zellwechsel_ Vodafone_ OpenBTS.xml* und *Zellwechsel_ Vodafone_ OpenBTS(2).xml*

7 Sicherheitsimplikationen

In den folgenden Abschnitten, sollen mögliche Probleme im Bereich Sicherheit kurz beleuchtet werden. Hierbei handelt es sich um verschiedene Angriffsszenarien auf die GSM-Infrastruktur, die gerade für Demonstationszwecke von Interesse sein können. OpenBTS spielt hierbei eine wichtige Rolle, da es zum Beispiel für den in Abschnitt 7.1 vorgestellten IMSI-Catcher verwendet werden kann. Aber auch Angriffe auf die Verschlüsselung (Abschnitt 7.3) und auf die GSM-Hintergrundinfrastruktur sind denkbar (Abschnitt 7.4).

7.1 OpenBTS als IMSI-Catcher

Der Authentifizierungsvorgang im GSM-Netzwerk enthält eine entscheidende Schwachstelle. Kapitel 2.3.2 beschreibt, wie dieser Authentifizierungsvorgang im Detail vonstatten geht. Das entscheidende Problem dabei ist, dass sich der Benutzer zwar gegenüber dem GSM-Netzwerk authentifiziert, die entsprechende BTS allerdings gegenüber dem Benutzer nicht verfiziert wird. Es handelt sich nur um eine einseitige Authentifizierung. Diese bildet auch die Grundlage für den Einsatz eines IMSI-Catchers.
Bereits im Jahre 1996 wurde von der deutschen Firma Rohde und Schwarz der erste IMSI-Catcher „GA 090“ vorgestellt [11]. Die ursprüngliche Idee war die Identifizierung eines Benutzers über die IMSI. Dazu war selbstverständlich die Hilfe des Netzanbieters nötig, der in der Lage ist, zu einer IMSI den entsprechenden Teilnehmer zu ermitteln. Bereits ein Jahr später war es mittels des Modells GA 900 möglich, ausgehende Gespräche aufzuzeichnen. Ein solches Gerät kostet zwischen 200.000 und 300.000 Euro.[1] Mit Hilfe von OpenBTS ist es möglich, einen solchen IMSI-Catcher für rund 1500 Euro zu realisieren (für die benötigte Hardware siehe Kapitel 4). Alle softwareseitig benötigten Parameter, um ein echtes Netz nachzuahmen, können wie in Kapitel 6.1 beschrieben ermittelt werden. Wichtig dabei ist, dass eine nicht gleiche Cell-ID und nicht gleiche Location Area verwendet werden, da sonst Mobiltelefone keine neue reguläre Registrierung durchführen. In der Masterarbeit von Dennis Wehrle, „Open Source IMSI-Catcher“ werden eine prototypische Implementierung eines IMSI-Catchers basierend auf OpenBTS beschrieben und genauere Zusammenhänge und Hintergründe erklärt [30]. Mobilfunkteilnehmer buchen sich ohne deren Kenntnisnahme in den IMSI-Catcher ein und können überwacht werden. Alle eingebuchten Mobiltelefone und deren dazugehörende IMSI werden protokolliert. Sämtliche ausgehenden Gespräche können mit dem IMSI-Catcher mittels Asterisk-Server beispielsweise durch folgendes Script im Audioformat wav aufgezeichnet werden:

```
exten => _0.,1,SetVar(CALLFILENAME=${TIMESTAMP}-${CALLERIDNUM}-${EXTEN})
exten => _0.,2,Monitor(wav,${CALLFILENAME},m)
exten => _0.,3,Dial(CAPI/408150:b${EXTEN:1},,tT)
exten => _0.,4,Congestion()
```

Abbildung 7.1 veranschaulicht die Funktionsweise von OpenBTS als IMSI-Catcher.

[1] IMSI-Catcher, *http://de.wikipedia.org/wiki/IMSI-Catcher* [Online; letzter Aufruf 25.10.2009]

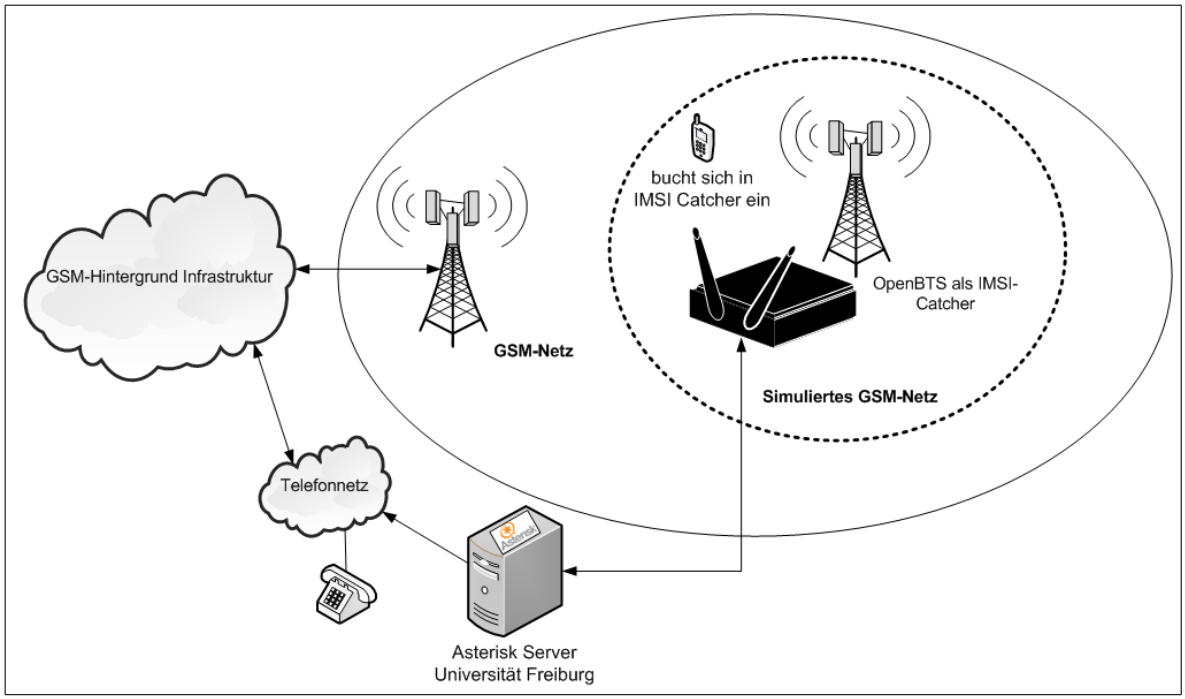

Abbildung 7.1: Funktionsweise eines IMSI-Catchers basierend auf OpenBTS

7.2 Ortung

Auf Grund der zellbasierten Struktur von GSM liegt es nahe, eine Lokalisierung eines Mobilfunkteilnehmers durchzuführen. Die Genauigkeit der Ortung kann gerade im ländlichen Raum allerdings nur auf mehrere Kilometer eingegrenzt werden, so dass zusätzliche Verfahren notwendig sind, um diese Genauigkeit zu erhöhen. Frederick Löser beschreibt in seiner Studienarbeit „Position System mit GSM“ [21] verschiedene Ansatzmöglichkeiten zur Lokalisierung mobiler Endgeräte mit GSM. Für ein Mobiltelefon ist es nötig, in regelmäßigen Abständen seine aktuelle Position in Form von Location Area und Cell-ID an die BTS zu übertragen. Diese Übertragung kann genutzt werden, um eine möglichst exakte Lokalisierung durchzuführen. Aus diesen Lokalisierungsvorgängen können dann Bewegungsprofile eines Mobilfunkteilnehmers erstellt und ausgewertet werden, was eine erhebliche Bedrohung des einzelnen Mobilfunkteilnehmers zur Folge hat. Mittels einer sogenannten lautlosen SMS kann im GSM-Netz eine genaue Ortung eines Mobiltelefons, ohne Kenntnisnahme eines Verbindungsaufbaus durch den Nutzer, erzwungen werden. Einige Strafverfolgungsbehörden schicken Verdächtigen verstärkt geheime Kurzmitteilungen aufs Mobiltelefon, um so ihren Aufenthaltsort herauszufinden und Bewegungsprofile zu erstellen.[2] Darüber hinaus kann die Fähigkeit zur Ortung für verschiedenste Dienste genutzt werden. Einige Betreiber bieten Infodienste als Location Based Services an. Mittels dieser Dienste können beispielsweise nahe gelegene Restaurants, Bahnhöfe oder Sehenswürdigkeiten schnell gefunden werden. Allerdings ist es damit genau so möglich, Kinder oder auch den Ehepartner zu überwachen und deren bzw. dessen genauen Aufenthaltsort zu bestimmen.

[2] Staatsanwaltschaft kritisiert „Spitzel-SMS“ der Polizei, *http://www.heise.de/newsticker/Staatsanwaltschaft-kritisiert-Spitzel-SMS-der-Polizei–/meldung/35915* [Online; letzter Aufruf 30.09.2009]

7.3 Verschlüsselte Gespräche entschlüsseln

Auf der Konferenz „Hacking at Random“ im August 2009 in den Niederlanden hat Karsten Nohl die Pläne zum Knacken der Verschlüsselung des GSM-Standards genauer erläutert [32]. Sofern eine entsprechende Hardware zur Verfügung steht, könnte jedes beliebige Gespräch, das sich in deren Reichweite befindet, abgehört werden. Neben normalen Telefongesprächen ließen sich auch SMS-Nachrichten mitlesen. Die Gefahr für Unternehmen und Privatpersonen, mit den ausspionierten Gesprächen und Informationen erpresst zu werden, steigt. Laut Aussage von Karsten Nohl soll der Hack so „unglaublich einfach“ sein, dass er unsere tägliche Mobiltelefon-Nutzung massiv beeinflussen könnte. Er hat sämtliche notwendigen Informationen und Tools für jedermann frei zugänglich auf seiner Webseite veröffentlicht.[3] Mit Hilfe der Community sollen die geheimen Schlüssel vorausberechnet und in einem Codebook gespeichert werden. Da dies auf normalem Wege selbst beim nicht als besonders sicher geltenden A5/1-Standard mit einem PC 100.000 Jahre dauern und 128 Peta-Byte an Speicher kosten würde, greift die Software von Nohl auf einige Kniffe zurück. Mit Hilfe moderner Grafikkarten mit CUDA-Unterstützung[4] für die Berechnung soll diese über das Internet auf viele Rechner verteilt werden. Die Speicherung erfolgt mit Hilfe spezieller Verfahren in stark komprimierter Form als Codebook. Nohl fordert die Community auf, am Projekt teilzunehmen, um in möglichst kurzer Zeit die Tabellen fertig zu stellen. Knapp 200 Computer sollen ausreichen, um sie in wenigen Monaten zu generieren. Sollte dieses Projekt erfolgreich sein und die Tabellen etwa per Tauschbörsen verbreitet werden, müssten die Mobilfunkbetreiber vermutlich auf sicherere Verschlüsselungsstandards ausweichen. Dazu müssten allerdings auch alle Mobiltelefone das Verfahren unterstützen, was bei den meisten ein Firmware-Update nach sich zieht. Dies dürfte sich bei vielen Mobilfunkteilnehmern kaum realisieren lassen, da es entweder erst gar nicht möglich ist oder nur mit erheblichem Aufwand durchführbar ist. Auch die Banken müssten in Folge davon ihr sicher geglaubtes mobile-Tan-Verfahren vermutlich einstellen, da das Entschlüsseln von SMS-Nachrichten dann kein größeres Problem mehr darstellt.

7.4 Unverschlüsselte GSM-Backend-Struktur

Eine Verschlüsselung der Gespräche findet meist nur auf der Luftschnittstelle zwischen Mobiltelefon und BTS statt (siehe Abbildung 2.3 in Kapitel 2.3). Oftmals werden in der GSM-Hintergrundinfrastruktur unverschlüsselte Richtfunkverbindungen verwendet. Diese können mit kommerziellen oder Eigenbau-Geräten abgehört werden. Sämtliche Gespräche sollten als normale ISDN-Verbindung herausgefiltert werden können.[5] Eine Karte mit Richtfunkstrecken von Vodafone im Großraum Karlsruhe ist auf der Homepage von Norbert Hüttisch zu finden.[6]. Diese Richtfunkstrecken könnten als mögliche Ziele geeignet sein. Hierbei handelt es sich um eine theoretische Idee. Ein Test in der Praxis und Konstruktion der benötigten Hardware stehen aktuell noch aus. Es sei an dieser Stelle ausdrücklich darauf hingewiesen, dass das Abhören einer Richtfunkstrecke ohne Einverständnis des Betreibers einen Straftatbestand erfüllt.

[3] Creating A5/1 Rainbow Tables, *http://reflextor.com/trac/a51* [Online; letzter Aufruf 1.10.2009]

[4] Die Compute Unified Device Architecture (CUDA) ist eine von Nvidia entwickelte Technik zur Beschleunigung wissenschaftlicher und technischer Berechnungen.

[5] Bundesamt für Sicherheit in der Informationstechnik, *https://www.bsi.bund.de/ContentBSI/Publikationen/Broschueren/gsm/index_ htm.html* [Online; letzter Aufruf 29.09.2009]

[6] Richtfunkübersicht Raum Karlsruhe, *http://www.nobbi.com/rifu.html* [Online; letzter Aufruf 29.09.2009]

7.5 SMS-Spam

Gelingt es, verschiedene Teilnehmer in ein durch OpenBTS simuliertes Netz „zu locken“, können diese mit beliebig vielen SMS-Nachrichten überflutet werden. Ein mögliches Szenario könnte sein, dass ein Kaufhaus eine OpenBTS-Zelle betreibt, um darüber sämtliche Kunden über mehr oder weniger gute, aktuelle Angebote zu informieren. Ob der Betreiber eines solchen Kaufhauses allerdings die dafür nötige Sende- und Empfangsgenehmigung auf dem entsprechenden Frequenzband erhalten würde, bleibt dahingestellt. Im Gegensatz zu dem weit verbreiteten E-Mail Spam, der gültige Empfänger E-Mailadressen voraussetzt, ist es nicht nötig, gültige Mobilfunknummern zu besitzen. Die Mobilfunkteilnehmer müssen sich lediglich im Sende- bzw. Empfangsradius einer OpenBTS-Zelle befinden und in diese eingebucht sein. Mittels eines Jammers könnnen alle Frequenzen, bis auf die der OpenBTS gestört werden, so dass die Mehrzahl der Mobiltelefone sich automatisch in die OpenBTS-Zelle einbucht [30].

7.6 Konfiguration „Over the Air“

Die Idee einer OTA-Attacke ist es, Mobiltelefone per SMS zu manipulieren. Hierzu könnte beispielsweise der Austausch des WAP-Gateways, Internet oder SMS-Gateways oder auch die Manipulation der Mobiltelefon Firmware gehören. So könnten die Telefonnummern in dem Telefonbuch eines Mobiltelefons durch extrem teure 0900er-Nummern getauscht werden und der Benutzer würde das vermutlich erst feststellen, wenn er bereits einen teuren Anruf getätigt oder einen überteuerten SMS-Dienst benutzt hat. Die Änderung des GPRS-Gateways kann dafür sorgen, dass der Benutzer beim Surfen durch das Internet auf speziell präparierte Webseiten durch den manipulierten Gateway umgeleitet wird und unbemerkt bei einer Eingabe durch den Mobilfunkteilnehmer persönliche Daten inklusive Bankdaten und Kreditkartendaten abgefangen werden. Mittels nachgeahmtem Originalnetz durch OpenBTS können Mobilfunkteilnehmer in dieses Netz gelockt werden, und per SMS kann eine solche Manipulation durchgeführt werden. Die Möglichkeit, SMS-Nachrichten zu verschicken, ist ab OpenBTS Version 2.3 gegeben. Nähere Informationen zu OTA können dem Airprobe Wiki[7] des Chaos-Computer-Clubs Berlin entnommen werden. Ein Praxistest mittels OpenBTS steht noch aus. Es müsste die Möglichkeit geschaffen werden, SMS-Nachrichten direkt binär zu verschicken. Eine mögliche Idee wäre es, die Software Now SMS/MMS Gateway 2009 der Firma NowMobile.com Limited[8] zu verwenden, um die benötigte OTA-Nachricht zu generieren. Diese Nachricht müsste binär-codiert von OpenBTS an das entsprechende Mobiltelefon versendet werden.

[7] OTA - airprobe, *https://svn.berlin.ccc.de/projects/airprobe/wiki/OTA* [Online; letzter Aufruf 29.09.2009]
[8] Now SMS/MMS Gateway & MMSC, *http://www.nowsms.com/* [Online; letzter Aufruf 07.10.2009]

8 Ausblick

Über den in dieser Arbeit vorgestellten Betrieb einer eigenen GSM-Zelle, die Abläufe in einem GSM-Netz und die Analyse unterschiedlichster Aspekte und Sicherheitsimplikationen hinaus sind noch einige weitere Fragestellungen für zukünftige Arbeiten und Projekte von Interesse.

Ziel eines weiteren Projektes oder einer Masterarbeit könnte die Entwicklung eines Netzwerkmonitors für GSM-Netze auf Basis von AirProbe und einem USRP sein. Als Grundlage für einen möglichen Funktionsumfang könnte der in Kapitel 6.1 vorgestellte Netzmonitor der Nokia Mobiltelefone dienen. Besonders für Demonstrationszwecke und Location Based Services auf Mobilfunkbasis ist die Entwicklung eines solchen Netzwerkmonitors von Bedeutung.[1]
Gerade unter dem Aspekt der Einsetzbarkeit in der Lehre ist ein Vergleich zwischen OpenBTS (basierend auf dem USRP) und OpenBSC[2] (basierend auf einer BS-11 MicroBTS der Firma Siemens) interessant. Die in dieser Arbeit vorgestellten Analysemethoden und Demonstationsmöglichkeiten verschiedenster Abläufe und sich daraus ergebender Sicherheitsimplikationen können unter unterschiedlichen Aspekten mit OpenBSC und einer BS-11 MicroBTS verglichen werden.[3] Eventuell werden weitere Abläufe und sicherheitskritische Aspekte aufgedeckt und anhand praktischer Demonstrationen nachvollziehbar gemacht.
OpenBTS befindet sich aktuell noch in der Entwicklung, und zukünftige Versionen werden über zusätzliche Funktionen wie ein mögliches Handover zwischen mehreren BTS, die Nutzung von multiplen Frequenzen oder auch eine vollständige SMS-Funktionalität verfügen.[4] Gerade letztere könnte für das in Kapitel 7.6 vorgestellte Angriffsszenario von erhöhtem Interesse sein. Im Bereich SMS besteht die Gefahr, in Frankreich diese SMS abzufangen und zu lesen, da sie unverschlüsselt übertragen werden.[5] Auch die Erweiterung von OpenBTS um Datendienste wie GPRS und EDGE stehen noch aus. Nach Einschätzung der Entwickler wird dazu keine zusätzliche Hardware benötigt [8]. Darüber hinaus sollen die zugrunde liegende Hardware (USRP) und vor allem der interne Taktgeber des USRP durch eine speziell auf OpenBTS optimierte Version entwickelt und ausgetauscht werden. Konkret bedeutet das vor allem, dass in den nächsten Monaten und Jahren durch den zunehmenden Funktionsumfang von OpenBTS noch weitere Abläufe in einem GSM-Netzwerk demonstriert und nachvollzogen bzw. auf sicherheitskritische Aspekte untersucht werden können. Der Betrieb mehrerer OpenBTS-Zellen und Mobilitätsfunktionen von Gesprächsvermittlungen innerhalb der verschiedenen Asterisk-Server bis hin zum Handover von laufenden Gesprächen ist eine weitere Idee für zukünftige Projekte. Der Aufbau eines echten zellulären Netzes brächte auch noch einen zusätzlichen Vorteil. Es würde gelingen, eine punktgenaue Ortung der verschiedenen Mobilfunkteilnehmer zu erstellen oder auch über die Zeit gesehen ganze Bewegungsprofile aufzuzeichnen oder sonstiges Benutzerverhalten zu überwachen (siehe Kapitel 7.2).
In Zukunft soll dass GSM-Netz durch das wesentlich sichere UMTS-Netz abgelöst werden, was

[1] Netzwerkmonitor für GSM-Netze auf Basis einer OpenSource BTS *http://www.ks.uni-freiburg.de/php_ arbeitdet.php?id=167* [Online; letzter Aufruf 04.10.2009]
[2] OpenBSC, http://bs11-abis.gnumonks.org/trac/wiki/OpenBSC [Online; letzter Aufruf 30.09.2009]
[3] Vergleich zweier OpenSource GSM-Lösungen unter dem Aspekt des Einsatzes in der Lehre *http://www.ks.uni-freiburg.de/php_ arbeitdet.php?id=171* [Online; letzter Aufruf 04.10.2009]
[4] OpenBTS/Plan - GNU Radio, *http://gnuradio.org/trac/wiki/OpenBTS/Plan* [Online; letzter Aufruf 30.09.2009]
[5] Interception of GSM Cellphones, *http://www.spyworld-actu.com/spip.php?article288* [Online; letzter Aufruf 25.10.2009]

allerdings noch einige Jahre dauern wird. Die Netzabdeckung von UMTS ist signifikant im ländlichen Bereich noch mehr als unzureichend. Auch in dem UMTS-Netz wird wird sich die Frage stellen, inwiefern es wirklich sicherer als das GSM-Netz ist. Mögliche Abläufe und Vorgänge in einem UMTS-Netz können für Lehr- und Forschungszwecke von ähnlichem Interesse sein wie die in dem bis dato genutzten GSM-Netz. Durch den Betrieb eines eigenen UMTS-Netzes könnten die übertragenen Daten besser in Bezug auf Funktionalität des Netzes analysiert und mögliche Sicherheitsaspekte herausgearbeitet werden. Die Sicherheitsarchitektur ist der des GSM-Netzes sehr ähnlich. Es wurden allerdings einige Verbesserungen eingeführt und beispielsweise die Krypto-Algorithmen A3, A5 und A8 durch die Blockchiffre „Kasumi“ (offiziell: A5/3) ersetzt [18]. Die Key-Länge beträgt nun 128 Bit und eine Ende-zu-Ende-Verschlüsselung ist möglich. Die einseitige Authentifizierung wurde abgeschafft und auch das Netz muss sich gegenüber dem Benutzer authentifizieren. Der Einsatz eines IMSI-Catchers ist in einem UMTS-Netz nicht mehr direkt möglich. Mit Hilfe eines UMTS-Störsenders können aber die Funkzellen im Sendebereich des Störsenders gezwungen werden, automatisch auf den niedrigeren Standard GSM herunter zu schalten. Dann könnte ein IMSI-Catcher wieder unerkannt eingesetzt werden. Eine weitere mögliche Schwachstelle könnten die am VLR vorgesehenen Abhörstellen für Behörden sein. Ob und inwieweit diese ein mögliches Sicherheitsrisiko mit sich bringen, bleibt noch offen.
Durch Kontrolle über Funktionen, die sich auf dem Layer 3 eines Mobilfunktelefons abspielen, ist eine bestimmte Art von DOS-Attacken möglich. Damit lassen sich theoretisch ganze Funkzellen für unbestimmte Zeit außer Kraft setzen.[6] Dieser Angriff und vorangehend genannten Probleme und Angriffsszenarien zeigen, dass GSM und wohl auch bald UMTS in den nächsten Jahren im Bereich Lehre, Forschung und Sicherheit von gesteigertem Interesse sein wird.

[6] The OpenBTS Chronicles *http://openbts.blogspot.com/2009/06/telecoms-dirty-big-secret.html* [Online; letzter Aufruf 01.10.2009]

Literaturverzeichnis

[1] 3rd Generation Partnership Project (3GPP): *3GPP Specification: 45.005.* WWW-Dokument, *http://www.3gpp.org/ftp/Specs/html-info/45005.htm.* [Online; letzter Aufruf 17.09.2009].

[2] Baumgarten, Uwe: *Systeme der drahtlosen Kommunikation.* In: *Mobile Verteilte Systeme.* Oldenbourg, 2009.

[3] Beman Dawes, David Abrahams: *Boost C++ Libraries.* WWW-Dokument, *http://www.boost.org/.* [Online; letzter Aufruf 03.05.2009].

[4] Burgess, David: *David Burgess on OpenBTS - a DIY GSM Air-Interface!* WWW-Dokument, *http://blog.ecomm.ec/2009/02/david-burgess-on-openbts.html.* [Online; letzter Aufruf 25.10.2009].

[5] CCC Berlin: *AirProbe.* WWW-Dokument, *https://svn.berlin.ccc.de/projects/airprobe.* [Online; letzter Aufruf 06.08.2009].

[6] Cho, Young-Jae: *Sicherheit in GSM Netzen.* PDF-Dokument, *http://www7.informatik.uni-erlangen.de/~dressler/lectures/seminar-mobilsec-ws0708/Cho.pdf*, 2008. [Online; letzter Aufruf 30.09.2009].

[7] Christian Cordes, Christoph Mohry: *GSM - Global System for Mobile Communication.* PDF-Dokument, Studienarbeit am LfKs Universität Freiburg, *http://www.ks.uni-freiburg.de/download/papers/telsemWS05/G2-GSM/HA_GSM2_Mohry_1.pdf*, 2006. [Online; letzter Aufruf 15.10.2009].

[8] David A. Burgess, Harvind S. Samra: *The Open BTS Project.* PDF-Dokument, *http://www.ahzf.de/itstuff/papers/OpenBTSProject.pdf*, 2008. [Online; letzter Aufruf 25.10.2009].

[9] ETSI - European Telecommunications Standards Institute: *Digital cellular telecommunications system (Phase 2+); Radio subsystem synchronization.* PDF-Dokument, Version 8.4.0, *http://ham.zmailer.org/oh2mqk/GSM/GSM-05.10.pdf*, 1999. [Online; letzter Aufruf 15.10.2009].

[10] European Telecommunications Standards Institute - ETSI: *GSM UMTS 3GPP Numbering Cross Reference.* WWW-Dokument, *http://webapp.etsi.org/key/queryform.asp.* [Online; letzter Aufruf 20.06.2009].

[11] Fox, Dirk: *Der IMSI-Catcher.* PDF-Dokument, *http://www.secorvo.de/publikationen/imsicatcher-fox-2002.pdf*, 2002. [Online; letzter Aufruf 25.10.2009].

[12] Global Mobile Suppliers Association: *GSM/3G Network Update/GMSA.* PDF-Dokument, *http://gsmworld.com/documents/Final_report.pdff*, 2009. [Online; letzter Aufruf 27.10.2009].

[13] GNU Radio: *The GNU Software Radio.* WWW-Dokument, *http://gnuradio.org/.* [Online; letzter Aufruf 02.09.2009].

[14] GNU Radio: *OpenBTS/BuildingAndRunning - GNU Radio.* WWW-Dokument, *http://www.gnuradio.org/trac/wiki/OpenBTS/BuildingAndRunning*, 2008. [Online; letzter Aufruf 23.08.2009].

[15] Hamza, Firas Abbas: *The USRP under 1.5X Magnifying Lens!* PDF-Dokument, *http://gnuradio.org/trac/attachment/wiki/UsrpFAQ/USRP Documentation.pdf*, 2008. [Online; letzter Aufruf 30.09.2009].

[16] Ingenieur-Büro Marx: *Überblick über die GSM Subsysteme.* WWW-Dokument, *http://www.embsys.de/gsm/Overview.html.* [Online; letzter Aufruf 19.08.2009].

[17] Jörg Eberspächer, Christian Bettstetter, Hans-Jörg Vögel Christian Hartmann: *GSM.* In: *GSM - Architecture, Protocols and Services*, Seiten 4–27, Chichester, West Sussex, United Kingdom, 2009. Wiley.

[18] Kipfelsberger, Tobias: *Sicherheit in GSM und UMTS.* PDF-Dokument, Seminararbeit an der Universität Koblenz-Landau, *http://www.uni-koblenz.de/~steigner/seminar-net-sec/sem9.pdf*, 2004. [Online; letzter Aufruf 26.10.2009].

[19] Lehrstuhl für Nachrichtentechnik: *Beispiele von Nachrichtensystemen - Global System for Mobile Communications.* PDF-Dokument, Technische Universität München, *http://www.lntwww.de/downloads/Beispiele%20von%20Nachrichtensystemen/Aufgaben/Kapitel3/*, 2008. [Online; letzter Aufruf 26.10.2009].

[20] Loula, Alexsander: *OpenBTS - Installation and Configuration Guide.* PDF-Dokument, Version 0.1, *https://81.56.142.154/Cour/These/OpenBTS/OpenBTS_Guide_En_v0.1.pdf*, 2009. [Online; letzter Aufruf 26.10.2009].

[21] Löser, Frederik: *Position System mit GSM.* PDF-Dokument, Studienarbeit am LfKs Universität Freiburg, *http://www.ks.uni-freiburg.de/download/studienarbeit/WS05/03-06-GSMposition-Loeser/03-06-positionsystemGSM-FLoeser.pdf*, 2006. [Online; letzter Aufruf 25.10.2009].

[22] Management, OpenBTS/Mobility. WWW-Dokument, *http://sourceforge.net/apps/trac/openbts/wiki/OpenBTS/Mobility.* [Online; letzter Aufruf 13.10.2009].

[23] nobbi.com - Norbert Hüttisch: *Monitorsoftware Tapir-G.* WWW-Dokument, *http://www.nobbi.com/monitor/index.html.* [Online; letzter Aufruf 09.09.2009].

[24] Petz, Elisabeth: *GSM - Global System for Mobile Communications.* PDF-Dokument, Diplomarbeit an der Universität Wien, *http://www.diplom.de/db_hobsons/diplomarbeiten5607.html*, 2001. [Online; letzter Aufruf 25.10.2009].

[25] R. Riemer c/o WAM: *UMTSlink.at.* WWW-Dokument, *http://umtslink.at.* [Online; letzter Aufruf 06.08.2009].

[26] RadioWare: *The USRP Board - RadioWare Project.* WWW-Dokument, *http://radioware.nd.edu/documentation/hardware/the-usrp-board.* [Online; letzter Aufruf 04.08.2009].

[27] Robert Flick, Fabian Uehlin: *OpenBTS.* PDF-Dokument, Projektarbeit an der Fachhochschule Kaiserslautern, *http://www.fabian-uehlin.de/OpenBTS-Dokumentation.pdf*, 2009. [Online; letzter Aufruf 13.08.2009].

[28] Sauter, Martin: *Von UMTS und HSDPA, GSM und GPRS zu Wireless LAN und Bluetooth Piconetzen.* In: *Grundkurs Mobile Kommunikationssysteme*, Wiesbaden, 2008. Friedr. Vieweg und Sohn Verlag.

[29] Shaw Computerhard- & Software: *PC Software & Datenkabel für Siemens Handys.* WWW-Dokument, *http://www.s25atonce.de.* [Online; letzter Aufruf 09.09.2009].

[30] Wehrle, Dennis: *Open Source IMSI-Catcher.* PDF-Dokument, Masterarbeit am LfKs Universität Freiburg, 2009.

[31] Wikimedia Foundation, Inc.: *Überblick über die GSM Subsysteme.* WWW-Dokument, *http://de.wikipedia.org/wiki/Global_System_for_Mobile_Communications.* [Online; letzter Aufruf 10.07.2009].

[32] WinFuture.de: *GSM-Lücke macht alle Handy-Telefonate abhörbar.* WWW-Dokument, *http://winfuture.de/news,49297.html.* [Online; letzter Aufruf 08.09.2009].

[33] Yacoub, Michel Daoud: *Wireless Technology.* In: *Wireless Technology: Protocols, Standards, and Techniques (Hardcover)*, Seiten 136–137. CRC; 1. Edition, Dezember 2001.

Abkürzungsverzeichnis

Abkürzung	**Beschreibung**
AC	Authentification Center
ACC	Access Control Class
AGCH	Access Grant Channel
BCCH	Broadcast Control Channel
BCH	Broadcast Channel
BSC	Base Station Controller
BSS	Base Station Subsystem
BSSAP	Base Station Subsystem Application Part
BTS	Base Transciever Station
BTSM	BTS Management
CCCH	Common Control Channel
CCH	Control Channel
CKSN	Ciphering Key Sequence Number
CM	Call Management
CUDA	Compute Unified Device Architecture
DCCH	Dedicated Channel
DCS	Digital Cellular System
DECT	Digital Enhanced Cordless Telecommunications
EIR	Equipment Identity Register
FACCH	Fast Associated Control Channel
FCCH	Frequency Correction Channel
FPGA	Field Programmable Gate Array
GMSC	Gateway Mobile Switching Center
GPRS	General Paket Radio Service
GSM	Global System for Mobile Communications
HLR	Home Location Register
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
LAC	Location Area Code
LAI	Location Area Identifier
LAPD	Link Access Procedure D-Channel

Abkürzung	**Beschreibung**
MCC	Mobile Country Code
MM	Mobility Management
MNC	Mobile Network Code
MOC	Mobil Originating Call
MS	Mobile Station
MSC	Mobile Switching Center
MSISDN	Mobile Station ISDN-Number
MSRN	Mobile Subscriber Roaming Number
MTC	Mobile Terminated Call
MTP	Message Transfer Part
NCC	Network Colour Code
NSS	Network Subsystem
PCH	Paging Channel
PCS	Personal Communications Service
PLMN	Public Land Mobile Network
PSTN	Public Switched Telefon Network
RACH	Random Access Channel
RR	Radio Resource Management
RSSI	Received Signal Strength Indication
SACCH	Slow Associated Control Channel
SCCP	Signalling Connection Control Part
SCH	Synchronization Channel
SDCCH	Stand-Alone DCCH
SDR	Software-Defined Radio
SIM	Subscriber Identity Module
SRI	Send Routing Information
SSS	Switching Subsystem
TCH	Traffic Channel
TMSI	Temporary Mobile Subscriber Identity
TRAU	Transcoding und Rate Adaption Unit
UMTS	Universal Mobile Telecommunication System
USRP	Universal Software Radio Peripheral
VLR	Visitor Location Register
WAP	Wireless Application Protocol

Abbildungsverzeichnis

2.1 Logo der „GSM Association“ . . . 3
2.2 Struktur eines GSM-Netzes in der Theorie und Praxis [17] . . . 5
2.3 Struktur eines-GSM Netzes . . . 7
2.4 SIM-Karte . . . 8
2.5 Authentifizierungsprozess im GSM-Netzwerk unter Verwendung der Algorithmen A3 und A8 (Vorlage nach [6]) . . . 11
2.6 Größenordnungen der verschiedenen GSM-Netzkomponenten . . . 13
2.7 Übersicht der GSM-Netzschnittstellen . . . 13
2.8 GSM-Rahmenstruktur . . . 15
2.9 Aufbau eines Normal Burst . . . 16
2.10 Übersicht über die verschiedenen Bursts (Vorlage nach [25]) . . . 17
2.11 Hierarchische Übersicht der logischen Kanäle . . . 17
2.12 Hierarchie der Rahmenstruktur . . . 19
2.13 Nutzung der Timeslots im Downlink [28] . . . 20
2.14 GSM Protokoll Architektur . . . 22
2.15 Registrierungsprozess beim Einschalten des Mobiltelefons (Vorlage nach [25]) . . . 24
2.16 Location Update (Vorlage nach [25]) . . . 26
2.17 Ablauf eines Mobile Terminated Calls . . . 27
2.18 Eingehender Anruf in GSM (Vorlage nach [25]) . . . 28
2.19 Ausgehender Anruf in GSM . . . 29
2.20 Ausgehender Anruf bzw. Aufbau eines Sprachkanals (TCH) (Vorlage nach [28]) . . . 30
2.21 Handover-Entscheidung . . . 30

3.1 System-Überblick: Hard- und Software [20] . . . 31
3.2 GNU Radio, OpenBTS und Asterisk Logo . . . 32

4.1 USRP1 mit Gehäuse . . . 36
4.2 USRP Motherboard mit 2 RFX900 Daughterboards . . . 37
4.3 DBSRX und RFX900 Daughterboard . . . 38
4.4 Signalgenerator FA-SY 1 zusammengebaut samt Gehäuse . . . 40
4.5 Konfigurationsparameter für FA-SY 1 mittels *USB_Sync.exe* . . . 41
4.6 Übersicht der verwendeten Mobiltelefone . . . 42

5.1 Übersicht über den Aufbau der Hardware-Komponenten . . . 44
5.2 USRP mit externem Taktgeber und Antennen . . . 45
5.3 Übersicht über das Zusammenspiel der Softwarekomponenten . . . 46
5.4 OpenBTS-Konsole mit Übersicht über die möglichen Befehle . . . 46
5.5 Registrierungsprozess beim Einschalten des Mobiltelefons in OpenBTS . . . 47
5.6 OpenBTS Logfile mit *Loglevel = INFO* . . . 48
5.7 Wireshark-Mitschnitt: OpenBTS Registrierungsprozess . . . 48
5.8 Mobile Originated Call und Call Clearing Prozedur (Vorlage nach [27]) . . . 49

6.1 Nokia Netzmonitor . . . 52

6.2 Siemens C35 und s25@once!-Software: T-Mobile Netzinformationen (Location Area, Cell ID, Empfangsqualität) . . . 54
6.3 Siemens C35 und Tapir-G Netzmonitor - T-Mobile Netzinformationen . . . 54
6.4 Spektrumsanalyse im Bereich 924-932 MHz mittels GNU Radio-Skript: *usrp_fft.py* 55
6.5 Frequenzspektrum eines BTS-Kanals bei 927.0 MHz (Decim=60) . . . 55
6.6 Frequenzspektrum eines BTS-Kanals bei 927.0 MHz (Decim=112) . . . 56
6.7 Frequenzspektrum eines BTS-Kanals bei 927.0 MHz mittels USRP2 . . . 56
6.8 ARFCAN - Frequenz Umrechner . . . 57
6.9 OpenBTS bei 940.8 MHz mittels USRP2 . . . 58
6.10 Konsolen-Output der Analyse mit *go.sh capture_927.0M_112.cfile* (E-Plus) . . . 59
6.11 Konsolen-Output der Analyse mit *go.sh capture_927.0M_112.cfile* (E-Plus: BCCH) 60
6.12 GSSM und Wireshark bei 927.0 MHz . . . 61
6.13 Nokia 3310 und MBUS NK-33 Datenkabel samt Seriell-zu-USB-Konverter . . . 62
6.14 Wireshark-Mitschnitt: Gesprächsaufbau im T-Mobile Netz eines Nokia 3310 . . . 64
6.15 Wireshark-Mitschnitt: SMS im Vodafone-Netz an Nokia 3310 . . . 64
6.16 Wireshark-Mitschnitt: Gespräch über OpenBTS (iPhone/Nokia3310) . . . 65
6.17 Wireshark-Mitschnitt: SMS von OpenBTS an Nokia 3310 . . . 65
6.18 Handover Vodafone zu OpenBTS (simulierte Vodafone Funkzelle) . . . 67
6.19 Wireshark-Mitschnitt: Versuchtes Handover Vodafone zu OpenBTS . . . 68

7.1 Funktionsweise eines IMSI-Catchers basierend auf OpenBTS . . . 70

Tabellenverzeichnis

2.1 Historischer Überblick wichtiger Netzwerktechnologien im Bereich der Mobilkommunikation [[19], [31], [21]] 4
2.2 Übersicht über die verwendeten Frequenzbänder [21], [31] 6
2.3 Übersicht der GSM-Schnittstellen [25] 14
2.4 Übersicht über die verschiedenen Bursts 16
2.5 Übersicht OSI- und GSM-Schichtenmodell 21

4.1 Features des EP1C12 FPGAs 37
4.2 Übersicht der Daughterboards RFX900, RFX1800 und DBSRX 39
4.3 Übersicht über die mit OpenBTS verwendeten Mobiltelefone 42

5.1 Unterschiede zwischen GSM-Netz und OpenBTS 50

6.1 Nokia NetMonitor DCT-3-Übersicht 52
6.2 Mit Netzmonitor gescannte Werte der verschiedenen Mobilfunkbetreiber (Ort: Rechenzentrum der Universität Freiburg) 53
6.3 Nokia E71 mit PhonNetInfo-Software für das E-Plus Netz 53

B.1 OpenBTS Revisions History 86

A Installationsanleitung GNU Radio

1. Aktuellen Sourcecode aus dem GNU Radio SVN[1] herunterladen:
   svn co http://gnuradio.org/svn/openbts/trunk/ openbts

2. Folgende Pakete sind essentiell notwendig, und nicht vorhandene Pakete können mit folgendem Befehl nachinstalliert werden: *sudo apt-get -y install swig g++ automake1.9 libtool python-dev fftw3-dev libcppunit-dev libboost1.35-dev sdcc-nf libusb-dev libsdl1.2-dev python-wxgtk2.8 subversion guile-1.8-dev libqt4-dev python-numpy ccache python-opengl libgsl0-dev python-cheetah python-lxml doxygen qt4-dev-tools libqwt5-qt4-dev libqwtplot3d-qt4-dev python-qwt5-qt4*

3. Zusätzliche, eventuell optional benötigte Pakete installieren: *sudo apt-get -y install gkrellm wx-common libwxgtk2.8-dev alsa-base autoconf xorg-dev g77 gawk bison openssh-server emacs cvs usbview octave*

4. QWT 5.0.2 installieren; dieses kann mittels *wget* von
   http://superb-east.dl.sourceforge.net/sourceforge/qwt/qwt-5.0.2.tar.bz2 heruntergeladen werden.

   ```
   tar jxf qwt-5.0.2.tar.bz2 //Dateien entpacken
   cd qwt-5.0.2
   ```

   Die Datei *qwtconfig.pri* muss editiert werden:
   Unter Unix Version „INSTALLBAS“ den Pfad */usr/local* eintragen (vorher */usr/local/qwt-5.0.2*); und „doc.path“ zu *$$INSTALLBASE/doc/qwt* ändern (vorher
   textit$$INSTALLBASE/doc);

   ```
   qmake
   make
   sudo make install
   cd ..
   ```

5. Boost installieren; dieses kann von der Entwicklerseite heruntergeladen werden [3]

   ```
   cd boost_1_38_0
   BOOST_PREFIX=/opt/boost_1_37_0
   ./configure --prefix=$BOOST_PREFIX
   --with-libraries=thread,date_time,program_options
   make
   sudo make install
   cd ..
   ```

6. GNU Radio downloaden, kompilieren und installieren:

[1] *http:/www.gnuradio.org/wiki* [13]

```
svn co http://gnuradio.org/svn/gnuradio/trunk gnuradio
cd gnuradio
# As per the instructions for installing Boost
export LD_LIBRARY_PATH=$BOOST_PREFIX/lib
./bootstrap
./configure --enable-gr-wxgui --enable-grc --enable-usrp --enable-gr-usrp
--enable-gr-audio-alsa --enable-gnuradio-examples --enable-gnuradio-core
--enable-gr-audio-oss
make
sudo make install
```

7. Um das USRP benutzen zu können, müssen noch folgende Rechte eingestellt werden

```
sudo addgroup usrp
sudo addgroup <YOUR_USERNAME> usrp
echo 'ACTION=="add", BUS=="usb", SYSFS{idVendor}=="fffe",
SYSFS{idProduct}=="0002", GROUP:="usrp",
MODE:="0660"' > tmpfile
sudo chown root.root tmpfile
sudo mv tmpfile /etc/udev/rules.d/10-usrp.rules
```

Es kann anschließend mit folgendem Befehl getestet werden, ob eine Verbindung zum USRP möglich ist: *ls -lR /dev/bus/usb | grep usrp*. Dies ist der Fall, wenn eine Ausgabe der Form *rw-rw—- 1 root usrp 189, 514 Mar 24 09:46 003* erscheint.

Im Ordner */usr/local/share/gnuradio/examples/usrp* befindet sich ein Pythonskript, mit dem der maximale Datendurchsatz gemessen werden kann.

```
cd /usr/local/share/gnuradio/examples/usrp/
./usrp_benchmark_usb.py

Testing 2MB/sec...
usb_throughput = 2M
ntotal = 1000000
nright = 998435
runlength = 998435
delta = 1565 OK

Testing 4MB/sec...
usb_throughput = 4M
ntotal = 2000000
nright = 1998041
runlength = 1998041
delta = 1959 OK

Testing 8MB/sec...
usb_throughput = 8M
ntotal = 4000000
nright = 3999272
runlength = 3999272
```

```
delta = 728 OK

Testing 16MB/sec...
usb_throughput = 16M
ntotal = 8000000
nright = 7992153
runlength = 7992153 d
elta = 7847 OK

Testing 32MB/sec...
usb_throughput = 32M
ntotal = 16000000
nright = 15986239
runlength = 15986239
delta = 13761 OK

Max USB/USRP throughput = 32MB/sec
```

Der maximale Datendurchsatz zwischen PC und USRP beträgt 32 MB pro Sekunde.
Kommt es beim Ausführen der Pythonskripte zu einem Import-Error der Form: *Import-Error: libgnuradio-core.so.0: cannot open shared object file: No such file or directory* kann folgender Befehl helfen:

```
export PYTHONPATH=/usr/local/lib/python2.5/site-packages
sudo ldconfig
```

B OpenBTS-Changelog

Name	Kommentare
1.0 (none)	completed L1, L2
1.1 Arnaudville	
1.2 Breaux Bridge	GNU Build, very early assignment
1.3 Carencro	first post-injunction release
1.4 Donaldsonville	fixed Ubuntu build error
1.5 Eunice	fixed L2 bugs related to segmentation, removed incomplete SMS directory, moved „abort“ calls into L3 subclasses
1.6 New Iberia	import of all 2.2 improvements to non-SMS release
2.0 St. Francisville	SMS support, file-based configuration
2.1 Grand Coteau	DTMF support, fixed more Linux-related build errors, -lpthread, TLMessage constructor, expanded stack to prevent overflows in Linux, moved gSIPInterface to main app, fixed iterator bug in Pager
2.2 Houma	added LEGAL notice, removed Assert classes, stop paging on page response, fixed Pager-spin bug, fixed Transceiver spin bugs, fixed 2^{32} microsecond rollover bug, reduced stack footprints in Transceiver, fixed SMS timestamps, check LAI before using TMSI in LUR, reduced memory requirement by 75%, removed PagerTest, fixed stale-transaction bug in paging handler, fixed USRP clock rollover bug, faster call connection, new USRPDevice design
2.3 Jean Lafitte	check for out-of-date RACH bursts, better TRX-GSM clock sync, formal logging system command line interface, emergency call setup
2.4 Kinder	fixed BCCH neighbor list bug, support for neighbor lists, fixed support for non-local Asterisk servers, cleaner configuration management, more realtime control of BCCH parameters, proper rejection of Hold messages, fixed L3 hanging bug in MTDCheckBYE
2.5 Lacassine	imported Joshua Lackey patches, put readline in CLI, SIP fixes from Anne Kwong, SIP fixes from testing with SMS server, timing advance control

Tabelle B.1: OpenBTS Revisions History

C Übersicht der CD

Auf der beigelegten CD befinden sich neben der Ausarbeitung im PDF Format, auch die in Microsoft Visio 2007 erstellten Grafiken in gepackter Form. Der Ordner */GSM-Analyse* beinhaltet die für AirProbe notwendigen Änderungen am GSM-Receiver, Vergleichsdateien der AirProbe-Webseite und die mit dem Nokia 3310 erstellten Analysedateien diverser Abläufe im GSM- bzw. OpenBTS-Netz. Im Ordner */OpenBTS* sind die verschiedenen OpenBTS Versionen und Asterisk Konfigurationsdateien zu finden.

Nr.	Datei
1	/Bilder/Visio_Grafiken.zip
2	/GSM-Analyse/AirProbe/capture_927.0M_112.cfile
3	/GSM-Analyse/AirProbe/capture_929.6M_112_E-Plus_IMSIs_TMSIs.cfile
4	/GSM-Analyse/AirProbe/gsm-receiver.tar.gz
5	/GSM-Analyse/GSSM/GSSM_Wireshark_927MHz.png
6	/GSM-Analyse/GSSM/wireshark_capture_GSSM_927MHz
7	/GSM-Analyse/Nokia3310/Vergleichsdateien_Airprobe.org/call_1525.xml
8	/GSM-Analyse/Nokia3310/Vergleichsdateien_Airprobe.org/call_init.xml
9	/GSM-Analyse/Nokia3310/Vergleichsdateien_Airprobe.org/sms.xml
10	/GSM-Analyse/Nokia3310/Vergleichsdateien_Airprobe.org/sms2.xml
11	/GSM-Analyse/Nokia3310/Wireshark/Gespräch_Nokia_3310_zu_Iphone2G_OpenBTS.xml
12	/GSM-Analyse/Nokia3310/Wireshark/Gespräch_Nokia_E71_zu_Nokia_3310_echtes_Vodafonenetz.xml
13	/GSM-Analyse/Nokia3310/Wireshark/Gespraech_zwischen_Nokia_3310_zu_Nokia_E71_und_Handover_Vodafone(AFCRN_64)_zu_OpenBTS(AFCRN_107).xml
14	/GSM-Analyse/Nokia3310/Wireshark/Registrierungsprozess_bei_OpenBTS_mit_manueller_Netzsuche.xml
15	/GSM-Analyse/Nokia3310/Wireshark/Registrierungsprozess_bei_Vodafone_mit_manueller_Netzsuche.xml
16	/GSM-Analyse/Nokia3310/Wireshark/Rufaufbau_Pakete_Nokia3310_im_echten_T-Mobile_Netz.xml
17	/GSM-Analyse/Nokia3310/Wireshark/SMS_Nokia_E71_an_Nokia_3310_echtes_Vodafonenetz.png
18	/GSM-Analyse/Nokia3310/Wireshark/SMS_Nokia_E71_an_Nokia_3310_echtes_Vodafonenetz.xml
19	/GSM-Analyse/Nokia3310/Wireshark/SMS_von_OpenBTS_an_Nokia_3310.xml
20	/GSM-Analyse/Nokia3310/Wireshark/Vodafone_Netzsuche.xml
21	/GSM-Analyse/Nokia3310/Wireshark/Wireshark_Gespraech_zwischen_Nokia_3310_zu_Nokia_E71_und_Handover_Vodafone(AFCRN_64)_zu_OpenBTS(AFCRN_107).png
22	/GSM-Analyse/Nokia3310/Wireshark/Wireshark_OpenBTS_Gespräch_zwischen_Nokia3310_und_iPhone.png
23	/GSM-Analyse/Nokia3310/Wireshark/Wireshark_Openbts_SMS_an_Nokia3310.png
24	/GSM-Analyse/Nokia3310/Wireshark/Wireshark_Registrierungsprozess_bei_OpenBTS_mit_manueller_Netzsuche.png
25	/GSM-Analyse/Nokia3310/Wireshark/Wireshark_T-Mobile_Gesprächaufbau_Nokia3310.png
26	/GSM-Analyse/Nokia3310/Wireshark/Zellwechsel_Vodafone(AFCRN_64)_zu_OpenBTS(AFCRN_79)_OpenBTS_beendet_zurück_ins_Vodafonnetz(AFCRN_68)_nach_Suchvorgang.xml
27	/GSM-Analyse/Nokia3310/Wireshark/Zellwechsel_Vodafone(AFCRN_64)_zu_OpenBTS(AFCRN_79)_OpenBTS_beendet_zurück_ins_Vodafonnetz(AFCRN_68)_nach_Suchvorgang.xml~
28	/GSM-Analyse/Nokia3310/Wireshark/Zellwechsel_Vodafone_OpenBTS (2).xml
29	/GSM-Analyse/Nokia3310/Wireshark/Zellwechsel_Vodafone_OpenBTS.xml
30	/Open Source GSM BTS Setup und Analyse-Holger Bertsch-Universität Freiburg-Oktober 2009.pdf
31	/OpenBTS/Asterisk-Server Konfiguration/extensions.conf
32	/OpenBTS/Asterisk-Server Konfiguration/sip.conf
33	/OpenBTS/OpenBTS-Konfigurationen/OpenBTS (O2).config
34	/OpenBTS/OpenBTS-Konfigurationen/OpenBTS (T-Mobile).config
35	/OpenBTS/OpenBTS-Konfigurationen/OpenBTS (Vodafone).config
36	/OpenBTS/OpenBTS-Konfigurationen/OpenBTS.config
37	/OpenBTS/OpenBTS-Konfigurationen/OpenBTS_2.4.config.example
38	/OpenBTS/Registrieungsprozess/Manuelle_Registrierung_Nokia_3310_DEBUG.out
39	/OpenBTS/Registrieungsprozess/Manuelle_Registrierung_Nokia_3310_INFO.out
40	/OpenBTS/Registrieungsprozess/Manuelle_Registrierung_Nokia3310_INFO.png
41	/OpenBTS/Registrieungsprozess/Registrierungsprozess_bei_OpenBTS_mit_manueller_Netzsuche.xml
42	/OpenBTS/Registrieungsprozess/Wireshark_Registrierungsprozess_bei_OpenBTS_mit_manueller_Netzsuche.png
43	/OpenBTS/Versionen/openbts-1.5Eunice.tar.gz
44	/OpenBTS/Versionen/openbts-1.6NewIberia.tar.gz
45	/OpenBTS/Versionen/openbts-2.3JeanLafitteOE.tar.gz
46	/OpenBTS/Versionen/openbts-2.4Kinder.tar.gz
47	/OpenBTS/Versionen/openbts-2.5LacassinePrerelease.tar.gz