

**Zusammenfassung des TK-Praktikum des sechsten Semesters  
Kommunikationsinformatik**

Praktikum

Deniz Kadiogullari und Christoph Drost

Erstgutachter: Harald Krauss



# Zusammenfassung

Kurze Zusammenfassung des Inhaltes in deutscher Sprache, der Umfang beträgt zwischen einer halben und einer ganzen DIN A4-Seite.

Orientieren Sie sich bei der Aufteilung bzw. dem Inhalt Ihrer Zusammenfassung an Kent Becks Artikel: <http://plg.uwaterloo.ca/~migod/research/beckOOPSLA.html>.



# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>v</b>
<b>Listings</b>	<b>1</b>
<b>1 GSM Versuch</b>	<b>1</b>
1.1 Allgemeine Beschreibung der Versuche	1
1.1.1 Versuchsaufbau	1
1.1.2 Die einzelnen Bauteile im Überblick	1
1.2 Visualisieren von Frequenzen	3
1.2.1 Frequenzen auflisten	3
1.2.2 Frequenzen darstellen	6
1.3 Anruf an die 2600	8
1.4 Datenmitschnitte	9
1.4.1 Anruf mitschneiden	9
1.5 Mitschnitt einer SMS	17
1.5.1 Versuchsaufbau	17
1.5.2 Versuchsdurchführung	17
<b>2 BA Versuch</b>	<b>21</b>
2.1 Allgemeine Beschreibung des Versuchs	21
2.2 Einrichten der Anlage	21
2.2.1 Einrichten der Ports	21
2.2.2 Einrichten der Routing Tabellen	22
2.3 Aufzeichnungen und Interpretationen des ISDN-D-Kanal Protokolls	23
2.3.1 Aufzeichnen des ISDN-D-Kanal Protokolls ??	23
2.3.2 Interpretieren des D-Kanal-Protokoll Mitschnitts	23
<b>3 RSP Versuch</b>	<b>29</b>
3.1 Einleitung	29
3.2 Switch und Router Konfiguration	29
3.2.1 Router start up running config	29
3.3 Packet Tracer	29
3.3.1 Versuchsaufbau	29
3.3.2 Messungen	29
3.3.3 Simulation Echo-Request/-Reply	29
3.4 Untersuchung des Paketflusses mit Wireshark	29

<b>4</b>	<b>RSC Versuch</b>	<b>31</b>
4.1	Einleitung . . . . .	31
4.2	Downlink . . . . .	31
4.3	Uplink . . . . .	31
4.4	ARFCN . . . . .	31
4.5	Untersuchung des Paketflusses mit Wireshark . . . . .	31
<b>5</b>	<b>SDH Versuch</b>	<b>33</b>
5.1	Einleitung . . . . .	33
5.2	Downlink . . . . .	33
5.3	Uplink . . . . .	33
5.4	ARFCN . . . . .	33
5.5	Untersuchung des Paketflusses mit Wireshark . . . . .	33
<b>6</b>	<b>RN Versuch</b>	<b>35</b>
6.1	Einleitung . . . . .	35
6.2	Downlink . . . . .	35
6.3	Uplink . . . . .	35
6.4	ARFCN . . . . .	35
6.5	Untersuchung des Paketflusses mit Wireshark . . . . .	35

# 1 GSM Versuch

## 1.1 Allgemeine Beschreibung der Versuche

Im folgenden handelt es sich um ein Test-Versuch GSM. GSM ist die Abkürzung für Global System for Mobile Communications und ein Standard für die volldigitale Mobilfunknetze. Wir haben ihn bereits kennen gelernt, da alle unsere Handys darauf beruhen. GSM ermöglicht die eigentliche Telefonie, eine Datenübertragung und das Versenden und Empfangen von SMS, Short Message Services. Mittlerweile wurden für die Datenübertragung leistungsfähigere Standards, wie UMTS und LTE entwickelt, jedoch ist GSM noch nicht wegzudenken.

Der Versuch soll das Verständnis für die Technik vertiefen, die den reibungslosen Ablauf unserer Handygespräche ermöglicht. Zu diesem Zweck steht uns ein System zur Verfügung, das aus einer Antenne, der Technik zur Signalverarbeitung und einem Computer mit entsprechender Software besteht.

### 1.1.1 Versuchsaufbau

Der Aufbau des Versuchs ist auf den ersten Blick leicht beschrieben: Unser System besteht aus einer Antenne [1.1.2.1](#), einem USPR [1.1.2.2](#) und einem Computer [1.1.2.3](#).

### 1.1.2 Die einzelnen Bauteile im Überblick

#### 1.1.2.1 Antenne

Dieses Bauteil wandelt die elektromagnetischen Signale in elektrische Signale um. Antennen sind in prinzipiell in allen Geräten enthalten, die etwas mit Funktechnik zu tun haben. Dazu zählen beispielsweise Radios oder auch Handys. Die Umwandlung hat den Hintergrund, dass die elektromagnetischen Single aus der Luft nicht direkt weiterverarbeitet werden können.

#### 1.1.2.2 USPR

Das USPR, Universal Radio Peripheral, ist eine geschlossene Einheit, die das Verarbeiten der Empfangenen Signale ermöglicht. Es ist modular aufgebaut, sodass ein breites Frequenzspektrum abgedeckt werden kann. Für unseren Versuch interessieren aber nur die Frequenzen des GSM. Das USPR wird im folgenden nicht weiter betrachtet, da es nicht der Gegenstand des Versuchs war, sondern diesen nur ermöglicht hat.

## 1 GSM Versuch



Abbildung 1.1: USRP mit Antenne

### 1.1.2.3 Computer

Der Computer mit seiner entsprechenden Software ist die für den Versuch am interessanteste Komponente. Er ermöglicht es, die empfangenen Funksignale grafisch darzustellen und auszuwerten. Weiterhin stellt der Computer die Protokolle für den reibungslosen Ablauf und eine vollwertige  $U_m$  Schnittstelle zur Verfügung.

Zur Bereitstellung dieser Schnittstelle und der Protokolle wird das Softwarepaket OpenBTS genutzt.

Das Ziel des Versuchs ist es, die Paketdaten mitzuschneiden, die in einem GSM Netz auftreten. Vor dem Mitschnitt soll ein Grundverständnis über die Physik hinter dem GSM Netz geschaffen werden.



## 1.2 Visualisieren von Frequenzen

### 1.2.1 Frequenzen auflisten

#### 1.2.1.1 Aufbau des Versuchs

Für die Visualisierung der Frequenzen werden die in 1.1.2 beschriebenen Komponenten benötigt. Aus der Softwarepaket OpenBTS werden die Tools lsursp, baudline und kal benötigt.

Neben der vorhandenen Hardware werden keine weiteren Geräte benötigt. Das Vorhandensein von Sendern in der Reichweite des Systems ist dennoch eine Voraussetzung.

#### 1.2.1.2 Versuchsdurchführung

Mit dem Tool lsursp wird überprüft, ob die USRP 1.1.2.2 vom System erkannt wird.

Nachdem festgestellt wurde, dass die USRP angeschlossen und vom System erkannt wurde, kann der eigentliche Versuch beginnen. Hierzu wird das Tool kal mit dem Kommando „kal -s -DCS“ aufgerufen. Kal führt einen Umgebungsscan durch, das bedeutet, dass alle Frequenzen, die im DCS 1800 Band liegen und empfangen werden können, aufgelistet werden können.

#### 1.2.1.3 Auswertung des Versuchs

Zur Erklärung, das DCS 1800 Band ist ein Frequenzband, das den Frequenzbereich um 1800 MHz nutzt. In Deutschland wurde dieser Bereich ursprünglich von den E-Netzen, also den Anbietern E-Plus und O<sub>2</sub> genutzt. Aus Kapazitätsmangel haben 1999 auch die großen D-Netz Betreiber DCS 1800 Frequenzen erworben. Der Umgebungsscan gibt also die Frequenzen aus, die mit GSM zu tun haben, auf einen Anbieter ist der Scan aber nicht beschränkt. Das Ergebnis der Umgebungsscans ist in Tabelle 1.1 aufgelistet. Jede Zeile dieser Auflistung besteht aus chan mit Frequenzen und power mit einem Wert.

Quellen  
finden

**chan** Chan steht in diesem Fall für channel oder channel number. Dieser Wert wird auch als ARFCN, Absolute Radio Frequency Channel Number, bezeichnet. Der Hintergrund ist, dass ein Teilnehmer des GSM Netzes nicht das gesamte Frequenzband benötigt. Bzw. auch, dass andere Kommunikationsteilnehmer einer Base Station ausgeschlossen werden, wenn ein Teilnehmer exklusiv das gesamte Frequenzband nutzt. Deswegen werden die Frequenzbänder in Kanäle (channels), bzw. Kanalpaare, unterteilt. Das Kanalpaar hat den Hintergrund, dass GSM für den Down- und des Uplink unterschiedliche Frequenzen nutzt. Anhand der ARFCN kann die absolute Frequenz berechnet werden, die für die tatsächliche Kommunikation genutzt wird. Die Formel

## 1 GSM Versuch

chan:	555 (1813.8 MHz + 14.632kHz)	power: 1007.18
chan:	602 (1823.2MHz - 8.896kHz)	power 481.48
chan:	619 (1826.6MHz + 572Hz)	power: 1171.37
chan:	620 (1826.8 + 347Hz)	power: 727.63
chan:	630 (1828.8MHz + 177Hz)	power: 1421.75
chan:	631 (1820.0MHz + 209Hz)	power: 2495.22
chan:	637 (1830.2MHz + 403Hz)	power: 2876.83
chan:	640 (1830.8MHz + 508Hz)	power: 36384.61
chan:	641 (1831.0MHz + 508Hz)	power: 8809.88
chan:	647 (1832.2MHz - 32.386Hz)	power: 1305.97
chan:	648 (1832.4MHz - 32470Hz)	power: 10507.76
chan:	700 (1842.8MHz + 386Hz)	power: 21662.59
chan:	701 (1843.0MHz + 455Hz)	power: 4220.36
chan:	706 (1844.0MHz + 387Hz)	power: 27836.79
chan:	709 (1844.6MHz + 2.954Hz)	power: 1148.92
chan:	713 (1845.4MHz + 621Hz)	power: 6744.54
chan:	715 (1845.8MHz + 388Hz)	power: 20091.07
chan:	755 (1853.8MHz - 20.894Hz)	power: 458.32
chan:	764 (1855.6MHz + 485Hz)	power: 19349.83
chan:	765 (1855.8MHz + 381Hz)	power: 9962.32
chan:	769 (1856.6MHz + 38.177Hz)	power: 3226.76
chan:	798 (1862.4MHz + 498Hz)	power: 994.82
chan:	802 (1863.2MHz + 498Hz)	power: 118213.39
chan:	805 (1863.8MHz + 440Hz)	power: 5598.97

Tabelle 1.1: Auflistung der empfangenen Frequenzen

dazu ist in der Abbildung 1.2 beschrieben. Auf diese Thematik wird aber im weiteren Verlauf des Versuchs weiter eingegangen.

Den Scheiß habe ich mir quasi ausgedacht. Stimmt das annähernd?

**power** Power ist die Stärke, mit der das Signal empfangen wurde. Keine Ahnung welche Einheit.

**Berechnung der Frequenzen** Wie schon erwähnt, anhand dieser Auflistung ist es möglich, den Frequenzbereich von DCS 1800 zu errechnen. Zur Berechnung eines Frequenzbereichs gibt es einige Formeln. Welche genutzt wird, hängt davon ab, welche Werte bereits bekannt sind.

Dazu was richtiges schreiben

## 1.2 Visualisieren von Frequenzen

```
ubuntu@ubuntu: ~  
-A      antenna TX/RX (0) or RX2 (1), defaults to RX2  
-g      gain as % of range, defaults to 45%  
-F      FPGA master clock frequency, defaults to 52MHz  
-v      verbose  
-D      enable debug messages  
-h      help  
ubuntu@ubuntu:~$ kal -s DCS  
kal: Scanning for DCS-1800 base stations.  
DCS-1800:  
chan: 555 (1813.8MHz + 14.632kHz)      power: 1007.18  
chan: 602 (1823.2MHz - 8.896kHz)      power: 481.48  
chan: 619 (1826.6MHz + 572Hz)      power: 1171.37  
chan: 620 (1826.8MHz + 347Hz)      power: 727.63  
chan: 630 (1828.8MHz + 177Hz)      power: 1421.75  
chan: 631 (1829.0MHz + 209Hz)      power: 2495.22  
chan: 637 (1830.2MHz + 403Hz)      power: 2876.83  
chan: 640 (1830.8MHz + 508Hz)      power: 36384.61  
chan: 641 (1831.0MHz + 325Hz)      power: 8809.88  
chan: 647 (1832.2MHz - 32.386kHz)      power: 1305.97  
chan: 648 (1832.4MHz - 32.470kHz)      power: 10507.76  
chan: 700 (1842.8MHz + 386Hz)      power: 21662.59  
chan: 701 (1843.0MHz + 455Hz)      power: 4220.36  
chan: 706 (1844.0MHz + 387Hz)      power: 27836.79  
chan: 709 (1844.6MHz + 2.954kHz)      power: 1148.92  
chan: 713 (1845.4MHz + 621Hz)      power: 6744.54  
chan: 715 (1845.8MHz + 388Hz)      power: 20091.07  
chan: 755 (1853.8MHz - 20.894kHz)      power: 458.32  
chan: 764 (1855.6MHz + 485Hz)      power: 19349.83  
chan: 765 (1855.8MHz + 381Hz)      power: 9962.32  
chan: 769 (1856.6MHz + 38.177kHz)      power: 3126.76  
chan: 798 (1862.4MHz + 498Hz)      power: 994.82  
chan: 802 (1863.2MHz + 498Hz)      power: 118213.39  
chan: 805 (1863.8MHz + 440Hz)      power: 5597.97  
ubuntu@ubuntu:~$
```

Abbildung 1.2: Screenshot des Umgebungsscans

$\begin{aligned} \text{fuplink} &= \text{Startfrequenz} + (\text{ARFCN} - \text{Offset}) * 0,2\text{MHz} \\ \text{fdownlink} &= \text{fuplink} + \text{Abstand} \\ \text{fuplink} &= \text{fdownlink} - \text{Abstand} \\ \text{ARFCN} &= (\text{fuplink} - \text{Startfrequenz} / 0,2 \text{ MHz}) + \text{Offset} \end{aligned}$
--

Tabelle 1.2: Formel zur Berechnung des Frequenzbereichs

Da in unserem Versuch die Antenne als reiner Empfänger gearbeitet hat, haben wir nur Frequenzen empfangen, die von den Sendern als Uplink Frequenzen genutzt werden.

Die empfangenen Frequenzen lassen sich auch einzelnen Providern zuordnen. Für diese Zuordnungen gibt es Pläne, welche Frequenzen an wen vergeben wurden. Auf

Was zur Berechnung schreiben

## 1 GSM Versuch

der folgenden Abbildung ist zu sehen welche Frequenzen in Deutschland von welchem Providern benutzt werden.

	von (MHz)	bis (MHz)	Kurzzeichen	Sendeleistung	Reichweite	Modulation	Gepulst	Betreiber	Sonstiges	Beschreibung
	1.710,0	1.725,0	GSM 1800 (UL)	1W ERP (Peak)	16km	GMSK	JA	Militär	Pulsung mit 217Hz. Leistung schwankt von 25mW-1W (Peak)	Mobilfunk (E-Netz)
	1.725,2	1.730,0	GSM 1800 (UL)	1W ERP (Peak)	16km	GMSK	JA	T-Mobile	Pulsung mit 217Hz. Leistung schwankt von 25mW-1W (Peak)	Mobilfunk (E-Netz)
	1.730,2	1.752,4	GSM 1800 (UL)	1W ERP (Peak)	16km	GMSK	JA	O 2	Pulsung mit 217Hz. Leistung schwankt von 25mW-1W (Peak)	Mobilfunk (E-Netz)
	1.752,8	1.758,0	GSM 1800 (UL)	1W ERP (Peak)	16km	GMSK	JA	Vodafone	Pulsung mit 217Hz. Leistung schwankt von 25mW-1W (Peak)	Mobilfunk (E-Netz)
	1.758,2	1.780,4	GSM 1800 (UL)	1W ERP (Peak)	16km	GMSK	JA	E Plus	Pulsung mit 217Hz. Leistung schwankt von 25mW-1W (Peak)	Mobilfunk (E-Netz)
	1.805,0	1.820,0	GSM 1800 (DL)	300W ERP	16km	GMSK	JA	Militär	Pulsungen mit 217Hz. Organisationskanal mit 1.736Hz. Leistungen von 0,5-300W ERP möglich	Mobilfunk (E-Netz)
	1.820,2	1.825,0	GSM 1800 (DL)	300W ERP	16km	GMSK	JA	T-Mobile	Pulsungen mit 217Hz. Organisationskanal mit 1.736Hz. Leistungen von 0,5-300W ERP möglich	Mobilfunk (E-Netz)
	1.825,0	1.847,4	GSM 1800 (DL)	300W ERP	16km	GMSK	JA	O 2	Pulsungen mit 217Hz. Organisationskanal mit 1.736Hz. Leistungen von 0,5-300W ERP möglich	Mobilfunk (E-Netz)
	1.847,8	1.853,0	GSM 1800 (DL)	300W ERP	16km	GMSK	JA	Vodafone	Pulsungen mit 217Hz. Organisationskanal mit 1.736Hz. Leistungen von 0,5-300W ERP möglich	Mobilfunk (E-Netz)
	1.853,2	1.875,4	GSM 1800 (DL)	300W ERP	16km	GMSK	JA	E Plus	Pulsungen mit 217Hz. Organisationskanal mit 1.736Hz. Leistungen von 0,5-300W ERP möglich	Mobilfunk (E-Netz)

	Militär
	T-Mobile
	O2
	Vodafone
	e-plus

Abbildung 1.3: Frequenzentabelle der Provider [frequenzPlan1800]

Wenn man die Tabelle 1.1 und die Grafik 1.3 vergleicht, ergibt sich Tabelle ?? . Auffällig ist, dass keine Signale des Anbieters Vodafone empfangen werden, dafür aber mehrere militärische Kanäle.

### 1.2.2 Frequenzen darstellen

Der Versuch 1.2.1 hat ergeben, dass im GSM Standard verschiedene Frequenzen genutzt werden. Die Tatsache, dass Kanäle mit einer Breite genutzt werden, wirft die Frage auf, was es mit diesen Breiten auf sich hat und wie Informationen übertragen werden.

#### 1.2.2.1 Aufbau des Versuchs

Der Aufbau des Versuchs entspricht grob dem Aufbau des Versuchs 1.2.1. Anstelle des Tools kal wird DSP-Buttler-Tool dsusrp genutzt.

#### 1.2.2.2 Versuchsdurchführung

Vor Beginn des Versuchs muss der Umgebungsscan aus 1.2.1 wiederholt werden. Dadurch kann sicher gestellt werden, dass die Messung mit einer aktiven, bzw. gerade gesendeten Frequenz durchgeführt wird. Nachdem eine Frequenz gefunden wurde,

## 1.2 Visualisieren von Frequenzen

chan:	555 1813.8 MHz	Militär
chan:	602 1823.2MHz	T-Mobile
chan:	619 1826.6MHz	O <sub>2</sub>
chan:	620 1826.8 + 347Hz	O <sub>2</sub>
chan:	630 1828.8MHz	O <sub>2</sub>
chan:	631 1820.0MHz	Militär
chan:	637 1830.2MHz	O <sub>2</sub>
chan:	640 1830.8MHz	O <sub>2</sub>
chan:	641 1831.0MHz	O <sub>2</sub>
chan:	647 1832.2MHz	O <sub>2</sub>
chan:	648 1832.4MHz	O <sub>2</sub>
chan:	700 1842.8MHz	O <sub>2</sub>
chan:	701 1843.0MHz	O <sub>2</sub>
chan:	706 1844.0MHz	O <sub>2</sub>
chan:	709 1844.6MHz	O <sub>2</sub>
chan:	713 1845.4MHz	O <sub>2</sub>
chan:	715 1845.8MHz	O <sub>2</sub>
chan:	755 1853.8MHz	E-Plus
chan:	764 1855.6MHz	E-Plus
chan:	765 1855.8MHz	E-Plus
chan:	769 1856.6MHz	E-Plus
chan:	798 1862.4MHz	E-Plus
chan:	802 1863.2MHz	E-Plus
chan:	805 1863.8MHz	E-Plus

Tabelle 1.3: Auflistung der empfangenen Frequenzen

beginnt der eigentliche Versuch mit dem Kommando *dbusrp-f 699219*. Dieses Kommando startet ein Analysetool, das sowohl den Frequenz- als auch den Amplitudenbereich ausgibt.

### 1.2.2.3 Auswertung des Versuchs

Das Ergebnis dieses Versuchs kann in der Grafik 1.4 betrachtet werden. Die Wellen im oberen Bereich der Darstellung sind die Darstellung im Zeitbereich, die unteren Wellen sind die Darstellung im Frequenzbereich. In der Mitte wird das Signal im Wasserfallmodell dargestellt. Das Wasserfallmodell zeigt wie sich die Grundfrequenz durch abziehen oder hinzufügen von Frequenzen verändert wird. Die Darstellung des Signal erinnert

## 1 GSM Versuch

Wat han  
mir wei  
davon?

stark an weißes Rauschen

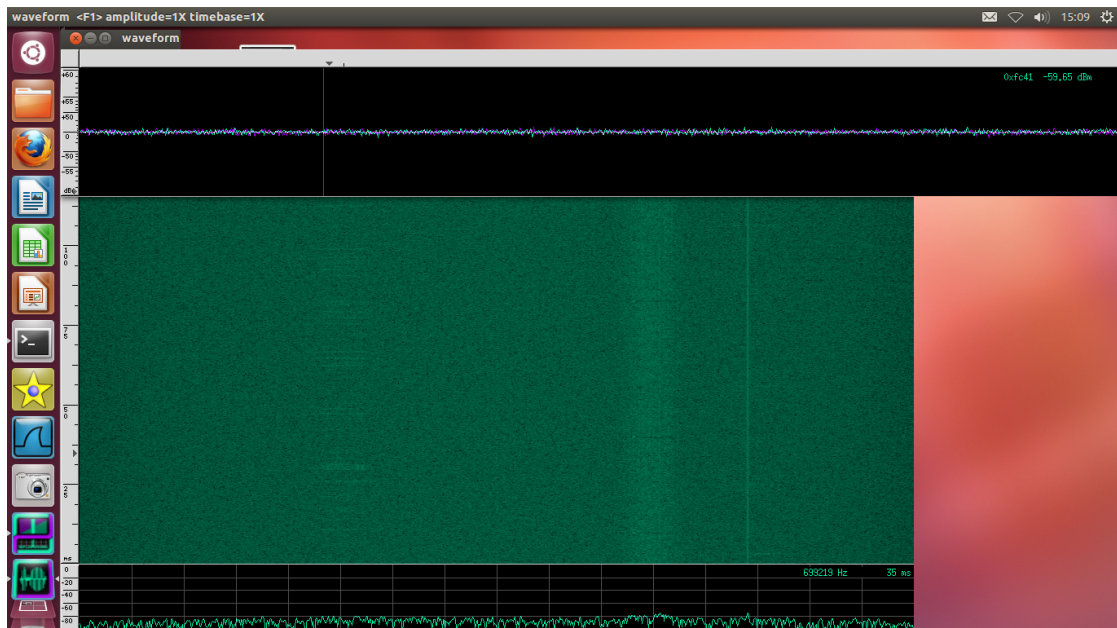


Abbildung 1.4: Eine visualisierte Frequenz

### 1.3 Anruf an die 2600

Es soll ein Anruf auf die 2600 was dem echo-Dienst entspricht durchgeführt werden. Dazu benötigen wir den am Anfang beschriebenen Versuchsaufbau sowie ein GSM-Fähiges Mobiltelefon das in dem Netz registriert ist. Als erstes muss das OpenBTS system gestartet werden dies erfolgt über mehrere Konsolen Befehle, da OpenBTS aus mehreren Komponenten besteht. Zuerst muss der Authentication-Service gestartet werden dies erfolgt durch den Befehl sipauthserve. Dannach muss die SMqueue gestartet werden die für die Weiterleitung der SMS verantwortlich ist, mit dem Befehl smqueue wird der Service gestartet. Der eigentliche OpenBTS Service muss ebenfalls gestartet werden. Dieser Dienst stellt den Kern des Systems dar, alle anderen Prozesse agieren mit diesem Prozess. Ausserdem brauchen wir noch den Asterisk Service der bereits in diesem Dokument erklärt worden ist. Diesen starten wir in einer neuen Konsole mit dem Befehl asterisk -r. Alle Befehle müssen als Superuser ausgeführt werden, sonst würden die Berechtigungen dazu fehlen. Um sich in dem Netz mit seinem eigenen Mobiltelefon registrieren zu können wählen wir das entsprechende Netz aus und erhalten unsere IMSI. Nun kann die 2600 angerufen werden und der Versuch durchgeführt werden.



## 1.4 Datenmitschnitte

Der vorangegangene Versuch sollte die physikalischen Eigenschaften des GSM Standards vermitteln. Der folgende Versuch beschäftigt sich mit den Protokolleigenschaften des GSM. Daher werden Situationen simuliert, die täglich millionenfach in den deutschen GSM Netzen stattfinden. Den Datenverkehr, den diese Situationen verursachen werden wir mitschneiden und analysieren.

### 1.4.1 Anruf mitschneiden

#### 1.4.1.1 Versuchsaufbau

Die Hardware entspricht der des Versuchs 1.2.1. Der Unterschied liegt in der Software. Diese simuliert ein eigenes GSM Netz. Als Softwarepaket wird Open BTS genutzt, was wieder mit mehreren Softwarekomponenten interagiert. Die für uns relevanten Teile des Open BTS sind im wesentlichen:

- Sipauthserve - Ist für die Authentifizierung verantwortlich
- Smqueue - Ein store-and-forward SIP Server, dient der Weiterleitung von SMS
- Asterisk - Stellt die Telefonanlage zur Verfügung

Diese Software erlaubt in Verbindung mit der Hardware den Betrieb eines eigenen GSM Netzes. Zum Mittschnitt der Daten steht das Tool Wireshark zur Verfügung. Wireshark ist ein Protocol Analyzer, damit ist es möglich einzelne Datenpakete mitzuschneiden.

Die Hardware wird um ein Mobiltelefon erweitert. Dieses ist bereits konfiguriert und registriert. Dieses Mobiltelefon ermöglicht die Kommunikation und Interaktion mit dem GSM Netz.

#### 1.4.1.2 Versuchsdurchführung

Zu Beginn des Versuchs muss die Software gestartet werden. Da die einzelnen Programme untereinander Abhängigkeiten haben, müssen sie in einem Befehl gestartet werden. Der Befehl `sudo sipauthserve & sudo smqueue & sudo OpenBTS & sudo OpenBTSCLI` startet die grundsätzliche Funktionalität des GSM Netzes. Mit dem Befehl `sudo asterisk -r` startet die Vermittlungenlage.

Nachdem die die Vorbereitungen abgeschlossen sind, steht ein GSM Netz zur Verfügung. Für den Mitschnitt der Daten muss noch Wireshark konfiguriert werden. Hierzu wird das Filter `!(udp port 5700 || udp port 5702 || icmp)` gesetzt. Ein Filter in Wireshark hat den Vorteil, dass beim Datenmitschnitt die umrelevanten Daten herausgefiltert werden können und damit die doch sehr umfangreiche Datenmenge reduziert werden kann.

Der Anruf wird auf die Telefonnummer 2600 getätigt. Diese Telefonnummer ist als echo Kanal konfiguriert, das heißt, dass das Gespräch vom Netz wieder zurück zum Teilnehmer geschickt wird.

Stimmt das denn auch? Das mit den Abhängigkeiten und gleichzeitigg starten

## 1 GSM Versuch

Nachdem der Versuch vorbereitet ist, wird über das Mobiltelefon die Nummer 2600 angerufen.

### 1.4.1.3 Auswertung des Versuchs

Der Echo Kanal funktioniert, wie beschrieben. Die Sprachdaten, die ans Netz gesendet werden, werden wieder vom Netz zurück geschickt.

Um in U(m) umwandeln so das das m klein wird

Mit geschnitten werden die Daten die über das Um Interface gesendet werden. Das Um Interface ist eine Funkschnittstelle die für die Übertragung zwischen der Mobile-Station und dem Base Transceiver Station zuständig ist. Das Um Interface arbeitet auf den Untersten drei Layer des ISO-OSI-Referenzmodell.

Durch Wireshark ist es möglich einen Einblick in die gesendeten Daten der verschiedenen Layern zu erhalten. Die Pakete die auf den verschiedenen Layern gesendet werden sind die folgenden:

- Auf Layer1 - GSMTAP
- Auf Layer2 - LAPDm
- Auf Layer3 - Resource Management Protocol (RR)

Da OpenBTS alle höheren Protokolle terminiert, erscheinen in Wireshark nicht nur die RR Nachrichten sondern auch Mobility Management und Call Management Nachrichten als Layer 3 Nachrichten welche in Wireshark als LAPDm gekennzeichnet sind. Im folgenden werden die für den Versuch wichtigen Pakete untersucht.

GSMTAP ist ein Pseudoheader für das Um Interface der in UDP Pakete gekapselt wird. Dieser übernimmt die Aufgabe der fehlerfreien Übertragung, sowie die Verteilung der Kanäle. Im Header werden Informationen zu der verwendeten Version, der Länge des Headers, der Type der übertragenden Daten, sowie die ARFCN und nähere Informationen zur Signalstärke angegeben.

```
[-] GSM TAP Header, ARFCN: 840 (Downlink), TS: 0, Channel: CCCH (0)
  Version: 2
  Header length: 16 bytes
  Payload Type: GSM Um (MS<->BTS) (1)
  Time slot: 0
  ..00 0011 0100 1000 = ARFCN: 840
  .0.. .... .... .... = uplink: 0
  Signal/Noise Ratio (dB): 0
  Signal Level (dBm): 0
  GSM Frame Number: 1335747
  Channel Type: CCCH (2)
  Antenna Number: 0
  Sub-slot: 0
[-] GSM CCCH - Paging Request Type 1
```



Abbildung 1.5: GSMTAP-Header

Unser Hauptaugenmerk legen wir auf die Pakete die als SIP gekennzeichnet sind. Hier kann man am besten nachverfolgen was gerade für eine Aktion durchgeführt wird. Da als erstes eine Verbindung aufgebaut werden muss damit man mit dem Echo-Server interagieren kann wird zuerst eine Invite Nachricht gesendet.

Bevor das jedoch passiert wird vorher auf dem Layer 2 eine Setup Nachricht gesendet. Wie man in ?? sehen kann wird hier ebenfalls die Bearer Capability angefordert. Die Bearer Capability bedeutet das ein Gerät in unserem Fall die MS eine bestimmte Transportleistung vom Netz anfordert.

```

GSM A-1/E DTAP - Setup
  Protocol discriminator: call control; call related SS messages
  .... 0011 = Protocol discriminator: Call Control; call related SS messages (0x03)
  0... .... = TI flag: allocated by sender
  .000 .... = TID: 0
  01... .... = Sequence number: 1
  ..00 0101 = DTAP call control Message Type: Setup (0x05)
  Bearer Capability 1 - (MS supports at least full rate speech version 1 and half rate speech version 1. MS has a greater preference for full rate speech version 1 than for half rate speech version 1)
  Element ID: 0x04
  Length: 6
  Octet 3
  Octets 3a - Speech versions
  Called Party BCD Number - (2600)
  Element ID: 0x0e
  Length: 3
  1... .... = Extension: No Extension
  .000 .... = Type of number: unknown (0x00)
  .... 0001 = Numbering plan identification: ISDN/Telephony Numbering (Rec. ITU-T E.164) (0x01)
  BCD digits: 2600
  Call control capabilities
  Element ID: 0x15
  Length: 2
  0000 .... = Maximum number of supported bearers: 1
  .... 0... = MCAT: The mobile station does not support Multimedia cat
  .... 0... = ENHANCED: The mobile station does not support the enhanced Network-Initiated In-call Modification procedure
  .... 0... = PCP: the mobile station does not support the Prolonged Clearing Procedure
  .... 0001 = DTMF: the mobile station supports DTMF as specified in subclause 5.3.7 of TS 24.008
  0000 .... = Spare bit(s): 0
  .... 0001 = Maximum number of speech bearers: 1
  
```

in bin  
einzu-  
binden  
<http://www.itwis.com/capability-BC.html>

Abbildung 1.6: Setup Nachricht auf dem Layer 2

Ist dieser Vorgang abgeschlossen kann der Teilnehmer durch die Invite Nachricht den Echo-Server Anfragen und ihm für den Sprach-Aufbau relevante Daten übermitteln. Diese sind, wie man im Header sehen kann, von wem die Nachricht gestellt wird, an wen sie gesendet wird, sowie die Call-ID.

## 1 GSM Versuch

```
Session Initiation Protocol (INVITE)
  Request-Line: INVITE sip:2600@127.0.0.1 SIP/2.0
    Method: INVITE
  Request-URI: sip:2600@127.0.0.1
    Request-URI User Part: 2600
    Request-URI Host Part: 127.0.0.1
    [Resent Packet: False]
  Message Header
    Via: SIP/2.0/UDP 127.0.0.1:5062;branch=z9hG4bKobts286aec235c0a789c46
      Transport: UDP
      Sent-by Address: 127.0.0.1
      Sent-by port: 5062
      Branch: z9hG4bKobts286aec235c0a789c46
    From: IMSI001011832121286 <sip:IMSI001011832121286@127.0.0.1>;tag=ekvzvzbyilfvluhd
      SIP Display info: IMSI001011832121286
    SIP from address: sip:IMSI001011832121286@127.0.0.1
      SIP from address User Part: IMSI001011832121286
      SIP from address Host Part: 127.0.0.1
      SIP from tag: ekvzvzbyilfvluhd
    To: <sip:2600@127.0.0.1>
      SIP to address: sip:2600@127.0.0.1
      Call-ID: 1246277822@127.0.0.1
    CSeq: 119 INVITE
      Sequence Number: 119
      Method: INVITE
    Contact: <sip:IMSI001011832121286@127.0.0.1:5062>;expires=3600
      Contact URI: sip:IMSI001011832121286@127.0.0.1:5062
        Contact URI User Part: IMSI001011832121286
        Contact URI Host Part: 127.0.0.1
        Contact URI Host Port: 5062
        Contact parameter: expires=3600\r\n
      Content-Type: application/sdp
      User-Agent: OpenBTS P2.8TRUNK Build Date Jun 26 2012
      Max-Forwards: 5
      P-Access-Network-Info: 3GPP-GERAN; cgi-3gpp=0010103e8000a
      Content-Length: 135
```

Abbildung 1.7: Invite Nachricht im SIP Protokoll

Vorrausgesetzt das diese Daten nicht fehlerhaft sind wird versucht eine Verbindung zu dem Server herzustellen. Diese kann man in ?? sehen. Dort wird mit dem Status-Code 100 gekennzeichnet.

## 1.4 Datenmitschnitte

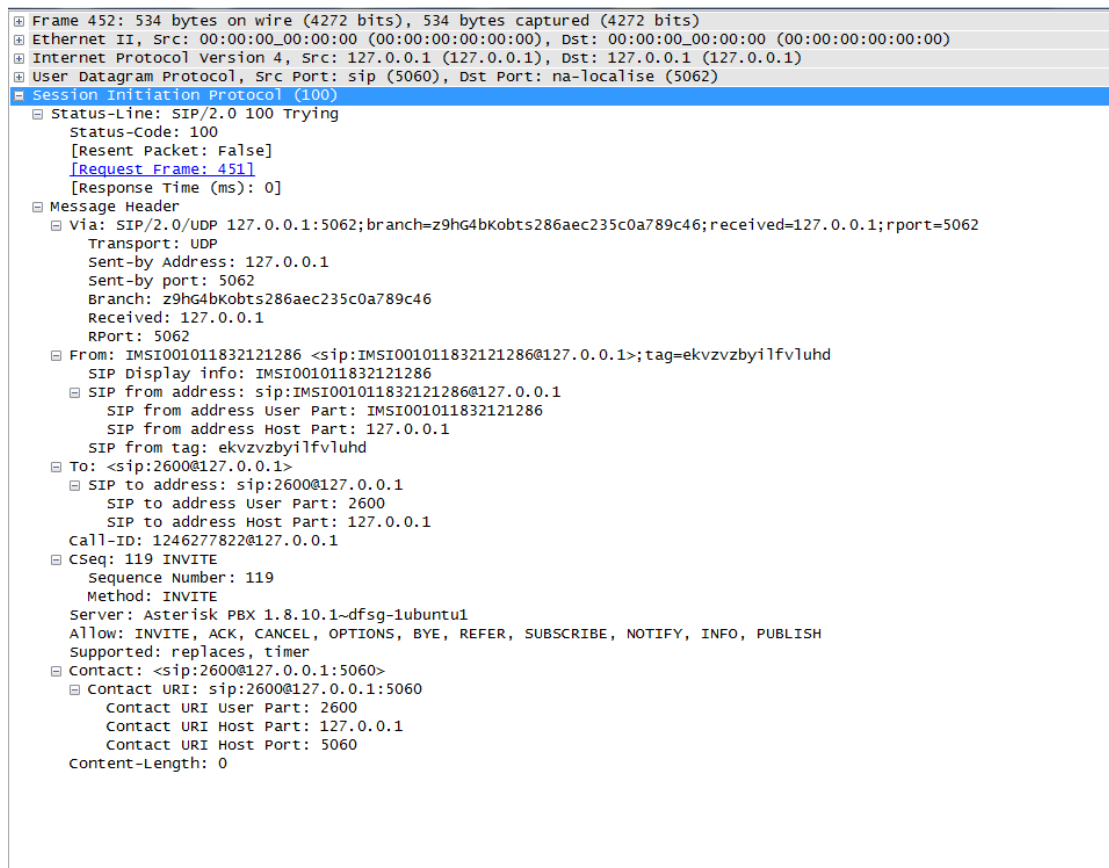


Abbildung 1.8: trying Nachricht im SIP Protokoll

Wenn die Anfrage der Verbindung erfolgreich war wird dies durch eine Ok-Nachricht bestätigt.

## 1 GSM Versuch

```

Session Initiation Protocol (200)
  Status-Line: SIP/2.0 200 OK
    Status-Code: 200
    [Resent Packet: False]
    [Request Frame: 451]
    [Response Time (ms): 1]
  Message Header
    Via: SIP/2.0/UDP 127.0.0.1:5062;branch=z9hG4bKobts286aec235c0a789c46;received=127.0.0.1;rpor
      Transport: UDP
      Sent-by Address: 127.0.0.1
      Sent-by port: 5062
      Branch: z9hG4bKobts286aec235c0a789c46
      Received: 127.0.0.1
      RPort: 5062
    From: IMSI001011832121286 <sip:IMSI001011832121286@127.0.0.1>;tag=ekvzvzbyilfvluhd
      SIP Display info: IMSI001011832121286
    SIP from address: sip:IMSI001011832121286@127.0.0.1
      SIP from address User Part: IMSI001011832121286
      SIP from address Host Part: 127.0.0.1
      SIP from tag: ekvzvzbyilfvluhd
    To: <sip:2600@127.0.0.1>;tag=as6b64f7c0
      SIP to address: sip:2600@127.0.0.1
        SIP to address User Part: 2600
        SIP to address Host Part: 127.0.0.1
      SIP to tag: as6b64f7c0
    Call-ID: 1246277822@127.0.0.1
    CSeq: 119 INVITE
      Sequence Number: 119
      Method: INVITE
    Server: Asterisk PBX 1.8.10.1~dfsg-1ubuntu1
    Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH
    Supported: replaces, timer
    Contact: <sip:2600@127.0.0.1:5060>
      Contact URI: sip:2600@127.0.0.1:5060
        Contact URI User Part: 2600
        Contact URI Host Part: 127.0.0.1
        Contact URI Host Port: 5060
      Content-Type: application/sdp
      Content-Length: 188
  Message Body
    Session Description Protocol
      Session Description Protocol Version (v): 0
      Owner/Creator, Session Id (o): root 1953701507 1953701507 IN IP4 127.0.0.1
        Owner Username: root
        Session ID: 1953701507
        Session Version: 1953701507
        Owner Network Type: IN
        Owner Address Type: IP4
        Owner Address: 127.0.0.1
      Session Name (s): Asterisk PBX 1.8.10.1~dfsg-1ubuntu1
      Connection Information (c): IN IP4 127.0.0.1
      Time Description, active time (t): 0 0
        Session Start Time: 0
        Session Stop Time: 0

```

Abbildung 1.9: Antwort auf die Einladung im SIP Protokoll  
Daraufhin erfolgt die Bestätigung und die Verbindung ist somit aufgebaut und bereit.

```

Session Initiation Protocol (ACK)
  Request-Line: ACK sip:2600@127.0.0.1 SIP/2.0
    Method: ACK
    Request-URI: sip:2600@127.0.0.1
      Request-URI User Part: 2600
      Request-URI Host Part: 127.0.0.1
    [Resent Packet: False]
    [Request Frame: 451]
    [Response Time (ms): 478]
  Message Header
    Via: SIP/2.0/UDP 127.0.0.1:5062;branch=z9hG4bKobts286aec235c0a789c46
      Transport: UDP
      Sent-by Address: 127.0.0.1
      Sent-by port: 5062
      Branch: z9hG4bKobts286aec235c0a789c46
    From: IMSI001011832121286 <sip:IMSI001011832121286@127.0.0.1>;tag=ekvzvzbyilfvluhd
      SIP Display info: IMSI001011832121286
    SIP from address: sip:IMSI001011832121286@127.0.0.1
      SIP from address User Part: IMSI001011832121286
      SIP from address Host Part: 127.0.0.1
      SIP from tag: ekvzvzbyilfvluhd
    To: <sip:2600@127.0.0.1>;tag=as6b64f7c0
      SIP to address: sip:2600@127.0.0.1
      SIP to address User Part: 2600
      SIP to address Host Part: 127.0.0.1
      SIP to tag: as6b64f7c0
    Call-ID: 1246277822@127.0.0.1
    CSeq: 119 ACK
      Sequence Number: 119
      Method: ACK
    User-Agent: OpenBTS P2.8TRUNK Build Date Jun 26 2012
    Max-Forwards: 5
    Content-Length: 0

```

Abbildung 1.10: Bestätigungs Nachricht im SIP Protokoll

Nun werden die Daten via RTP zwischen Server und User ausgetauscht. RTP ist das Real-Time-Transport Protocol das dazu dient audiovisuelle Daten über IP-basierte Netzwerke zu versenden. Wir legen nun auf das bedeutet wir möchten die Verbindung trennen. Das erfolgt durch einen Bye-nachricht (??). In dieser meldet der User dem Server das er die Verbindung nun trennt um dem Server bescheid zu geben. da ebenfalls die Layer 2 Verbindung wieder getrennt werden muss erfolgt hier eine Disconnect Nachricht wie in ??.

```

GSM A-I/F DTAP - Disconnect
  Protocol Discriminator: Call Control; call related SS messages
    .... 0011 = Protocol discriminator: Call Control; call related SS messages (0x03)
    0... .... = TI flag: allocated by sender
    .000 .... = TIO: 0
    01... .... = Sequence number: 1
    ..10 0101 = DTAP Call Control Message Type: Disconnect (0x25)
  Cause - (16) Normal call clearing
    Length: 2
    1... .... = Extension: No Extension
    .11. .... = Coding standard: Standard defined for the GSM PLMNS
    ...0 .... = Spare bit(s): 0
    .... 0000 = Location: User
    1... .... = Extension: No Extension
    .001 0000 = Cause: (16) Normal call clearing

```

## 1 GSM Versuch

Abbildung 1.11: Aufforderung zum Beenden der Layer 2 Verbindung

```
Session Initiation Protocol (BYE)
  Request-Line: BYE sip:2600@127.0.0.1 SIP/2.0
    Method: BYE
  Request-URI: sip:2600@127.0.0.1
    Request-URI User Part: 2600
    Request-URI Host Part: 127.0.0.1
  [Resent Packet: False]
  Message Header
    Via: SIP/2.0/UDP 127.0.0.1:5062;branch=z9hG4bKobts2830e292470ff33267
      Transport: UDP
      Sent-by Address: 127.0.0.1
      Sent-by port: 5062
      Branch: z9hG4bKobts2830e292470ff33267
    From: IMSI001011832121286 <sip:IMSI001011832121286@127.0.0.1>;tag=ekvzvzbyilfvluhd
      SIP Display info: IMSI001011832121286
    SIP from address: sip:IMSI001011832121286@127.0.0.1
      SIP from address User Part: IMSI001011832121286
      SIP from address Host Part: 127.0.0.1
      SIP from tag: ekvzvzbyilfvluhd
    To: <sip:2600@127.0.0.1>;tag=as6b64f7c0
      SIP to address: sip:2600@127.0.0.1
        SIP to address User Part: 2600
        SIP to address Host Part: 127.0.0.1
      SIP to tag: as6b64f7c0
    Call-ID: 1246277822@127.0.0.1
    CSeq: 120 BYE
      Sequence Number: 120
      Method: BYE
    Contact: IMSI001011832121286 <sip:IMSI001011832121286@127.0.0.1:5062>
      SIP Display info: IMSI001011832121286
    Contact URI: sip:IMSI001011832121286@127.0.0.1:5062
      Contact URI User Part: IMSI001011832121286
      Contact URI Host Part: 127.0.0.1
      Contact URI Host Port: 5062
    User-Agent: OpenBTS P2.8TRUNK Build Date Jun 26 2012
    Max-Forwards: 5
    Content-Length: 0
```

Abbildung 1.12: Aufforderung zum Beenden des Calls

Die Bye-Nachricht wird durch eine weitere OK-Nachricht bestätigt und somit ist die Verbindung vollständig abgebaut.

```
Session Initiation Protocol (200)
  Status-Line: SIP/2.0 200 OK
    Status-Code: 200
    [Resent Packet: False]
    [Request Frame: 16246]
    [Response Time (ms): 0]
    [Release Time (ms): 0]
```

Abbildung 1.13: Antwort auf die Aufforderung

## 1.5 Mitschnitt einer SMS

Wird der Versuch mit einem eigenen Mobiltelefon ausgeführt, muss sich dieses erst an am Netz anmelden. Diese Anmeldung wird vom Netz mit einer SMS bestätigt.

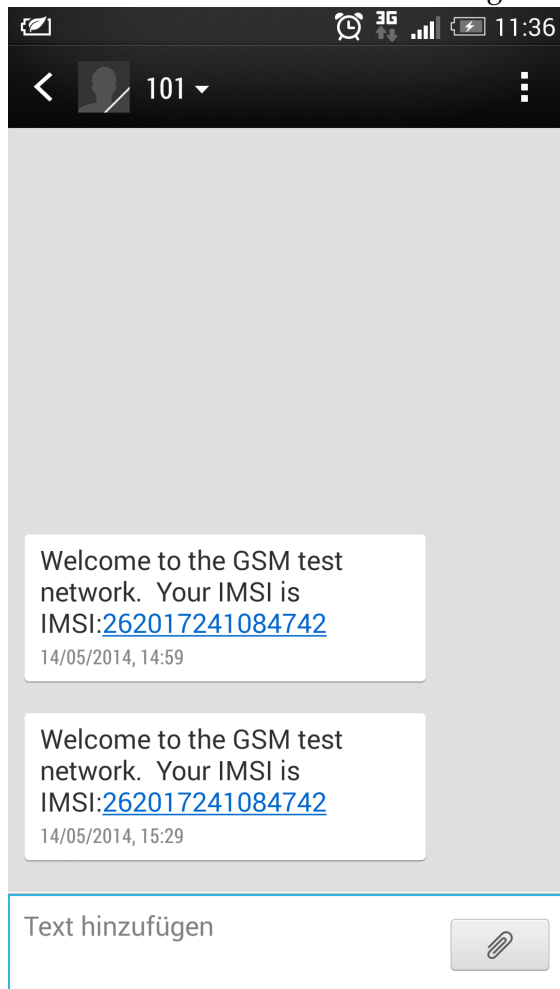


Abbildung 1.14: Quittierung der Einwahl in das GSM Netz

## 1.5 Mitschnitt einer SMS

### 1.5.1 Versuchsaufbau

Der Aufbau des Versuchs entspricht dem Aufbau 1.4.1.1.

### 1.5.2 Versuchsdurchführung

Die Durchführung entspricht der Durchführung 1.4.1.2. Anstatt des Anrufs an die Nummer 2600 wird eine SMS mit dem Inhalt „High “an die Telefonnummer 411 geschickt.

## 1 GSM Versuch

### 1.5.2.1 Auswertung des Versuchs

Die Protokolle LAPDm und GSMTAP unterschieden sich in ihrer Aufgabe nicht zu den vorher besprochenen.

Die Protokolle zu versenden der SMS sind jedoch wieder interessant. Dafür benötigten Protokolle sind

- Direct Transfer Application Part (DTAP)- dient zur Verbindungssteuerung
- Relay Protokoll (RP) - für das Routing verantwortlich
- Transport Protocol Data Unit (TPDU) - Übertragung der Nutzerdaten

Wie man sehr gut in der 1.15 sehen kann ist der gesendete Text unten in der GSM SMS TPDU ebenfalls teil der ankommenden Nachricht. Dieser beinhaltet neben dem gesendeten Text auch die Zeit in der die SMS gesendet wurde sowie die IMSI Adresse und Telefonnummer. Die Inhalte können von dem von uns gemachten Versuch abweichen da die Daten von einem älteren Versuch stammen.

In dem RP Abschnitt kann man sehen in welche Richtung die SMS gesendet wurde. Hier einmal von Netzwerk (NW) zur Mobilestation (MS) und umgekehrt. Durch das DTAP wird gekennzeichnet das es sich um eine SMS handelt.

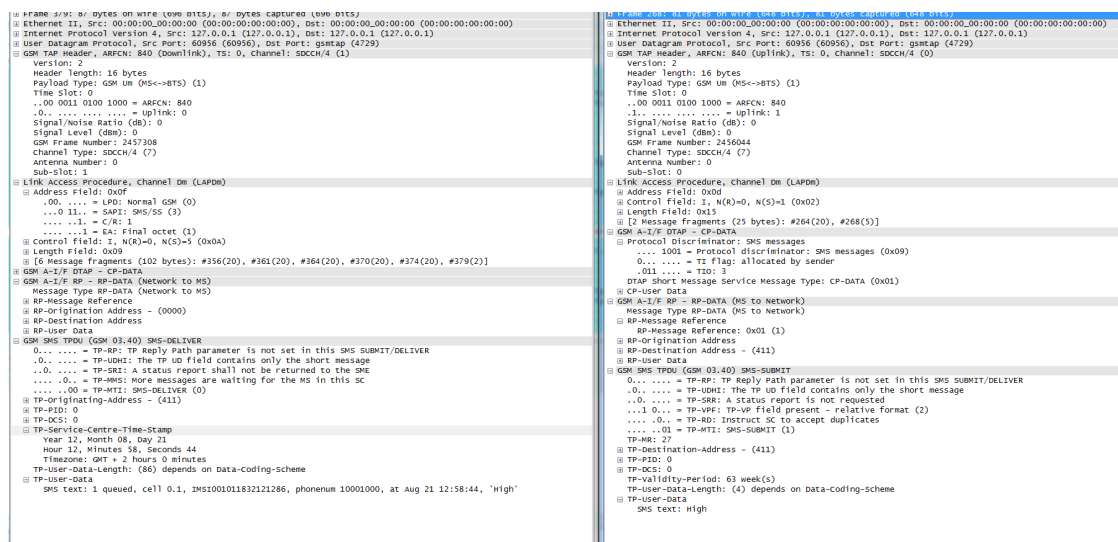


Abbildung 1.15: Gegenüberstellung Gesendet und Empfangen

Wie man in 1.16 sehen kann wird das Senden der SMS wieder von SIP initialisiert. Der Vorgang entspricht der bereits in dem Vorherigen Kapitel besprochenen Abfolge.



## 1.5 Mitschnitt einer SMS

272	54.949984	127.0.0.1	127.0.0.1	SIP	493 Request: MESSAGE sip:smc@127.0.0.1   (RP)
279	55.425348	127.0.0.1	127.0.0.1	SIP	317 Status: 202 Queued
285	55.597169	127.0.0.1	127.0.0.1	SIP	534 Request: MESSAGE sip:IMSI001011832121286@127.0.0.1:5062   (RP)
287	55.602021	127.0.0.1	127.0.0.1	SIP	378 Status: 100 Trying
381	60.613022	127.0.0.1	127.0.0.1	SIP	534 Request: MESSAGE sip:IMSI001011832121286@127.0.0.1:5062   (RP)
382	60.616438	127.0.0.1	127.0.0.1	SIP	378 Status: 100 Trying
418	62.029016	127.0.0.1	127.0.0.1	SIP	328 Status: 200 OK
560	71.656441	127.0.0.1	127.0.0.1	SIP	328 Status: 200 OK

Abbildung 1.16: SIP Abfolge des Sendens



## 2 BA Versuch

### 2.1 Allgemeine Beschreibung des Versuchs

BA, Basic Access, ist der Standardanschluss an das ISDN Netz. Er wird von den Anbietern an Privatkunden und kleine Betriebe vergeben. Basic Access bietet zwei Nutzkanäle (B-Kanäle) und einen Signalisierungskanal (D-Kanal). Obwohl die Netzbetreiber nach und nach auf reine IP Netze umstellen, hat ISDN in öffentlichen Telefonnetzen einen hohen Stellenwert. Mit der Entscheidung, dass die Ortsvermittlungsanlagen digitalisiert werden sollte, wurde 1979 ein wichtiger Grundstein für ISDN gelegt. 1987 wurde ISDN in Pilotprojekten erfolgreich getestet und schließlich 1989 flächendeckend eingeführt. ISDN bietet im Vergleich zu den analogen Übertragungstechniken den Vorteil, dass zwei Nutzkanäle gleichzeitig übertragen werden können. Zusätzliche Vorteile resultieren aus der verbesserten Sprachqualität und der schnelleren Datenübertragung. Der folgende Versuch soll das grundlegende Verständnis für ISDN vertiefen und gleichzeitig Einblicke in die Konfiguration gewähren. Der Einblick in die Konfiguration wird dadurch vermittelt, dass es zu dem Versuch gehört, die Anlage für den Versuch herzustellen. Durch Protokollmitschnitte und die bereitgestellten Unterlagen soll das grundlegende Verständnis für ISDN vermittelt werden.

### 2.2 Einrichten der Anlage

#### 2.2.1 Einrichten der Ports

##### 2.2.1.1 Aufbau des Versuchs

Als Switch steht ein Patapsco Liberator S zur Verfügung. Der Switch übernimmt die Funktionen des Netzes. Dazu gehören die TEI Vergabe, das Routing von Gesprächen und die Vergabe von Telefonnummern. Auf der Hardwareebene verbindet der Liberator die Telefone.

Zusätzlich steht ein EyeSDN Gerät. Dieses ermöglicht die Verbindung des Computers mit dem ISDN Netz und damit den Mittschnitt der Daten.

Zur Telefonie stehen zwei Telefone zur Verfügung und zum Konfigurieren des Systems ein Computer.

##### 2.2.1.2 Versuchsdurchführung

Die Hardware ist zu Beginn des Versuchs bereits verkabelt. Nachdem die SoftwareEyeSDN gestartet wurde, beginnt das Konfigurieren des Switches. Die Konfigura-

War da  
nicht  
noch  
was zur  
Stromver-  
sorgung?

## 2 BA Versuch

tion des Switches geschieht in der Anwendung Switchmanager. Dort werden die Ports eingerichtet, an denen die Telefone hängen und die Freizeichen eingestellt. Nach dem Upload wird die Konfiguration auf den Switch übertragen.

Die Einstellungen können direkt an den Telefonen getestet werden. Wenn der Hörer abgehoben wird, muss ein Freizeichen zu hören sein.

### 2.2.1.3



Abbildung 2.1: Der Liberator S [liberatorBild]

### 2.2.1.4 Auswertung des Versuchs

Nachdem die Ports konfiguriert sind, und die Konfiguration auf den Liberator S übertragen sind, sind die Telefone grundsätzlich aktiv. Beide Telefone geben nach dem Abheben des Hörers ein Freizeichen.

## 2.2.2 Einrichten der Routing Tabellen

Nach dem Versuch 2.2.1 haben die Telefone bereits Grundfunktionalitäten. Zum Telefonieren und für die weiteren Versuche fehlt aber noch die Einrichtung der Routinginformationen. Ohne die Routinginformationen ist es nicht möglich, ein Gespräch von einem Telefon zum Anderen zu leiten, als zu telefonieren.

### 2.2.2.1 Aufbau des Versuchs

Der Aufbau des Versuchs entspricht dem vom Versuch 2.2.1. Dieser ist für den aktuellen Versuch aber eine Voraussetzung.

### 2.2.2.2 Beschreibung des Versuchs

Das Routing wird im Interface Term des Liberator Fensters eingerichtet. Für das Einrichten von Routen können Profile angelegt werden, dadurch bleibt das System flexibler. Wir richten ein neues Profil ein, und konfigurieren die Routen von Telefon 1 zu Telefon 2 und umgekehrt. Wie bereits im letzten Versuch werden die Einstellungen nach dem Hochladen wirksam.

### 2.2.2.3 Auswertung des Versuchs

Nachdem die Routen konfiguriert sind, ist es möglich das Telefon 2 vom Telefon 1 an anzurufen. Bzw. ein Anruf in die Umgekehrte Richtung ist auch möglich.

## 2.3 Aufzeichnungen und Interpretationen des ISDN-D-Kanal Protokolls

### 2.3.1 Aufzeichnen des ISDN-D-Kanal Protokolls ??

#### 2.3.1.1 Versuchsaufbau

Der Versuchsaufbau entspricht dem Versuch 2.2.1. Die beiden vorangegangenen Versuche sind für diesen voraus gesetzt.

#### 2.3.1.2 Versuchsdurchführung

Die spätere Auswertung wird erleichtert, wenn der, eventuell schon laufende, Mitschnittsdienst erst gestoppt wird und alle vorhandenen Ergebnisse verworfen werden. Nachdem das geschehen ist, wird der Mitschnittdienst wieder gestartet und das Telefon 1 vom Telefon 2 angerufen. Wenn das Telefon 2 klingelt wird der Hörer abgehoben und nach einem kurzen Moment wieder aufgelegt. Dadurch wird ein Gespräch aufgebaut und wieder abgebaut. Direkt nach dem Beenden des Gesprächs wird der Mitschnittdienst wieder gestoppt. Die Auswertung des Gesprächs beginnt, wenn der Mitschnitt im Fenster mit einem Doppelklick ausgewählt wird.

### 2.3.2 Interpretieren des D-Kanal-Protokoll Mitschnitts

#### 2.3.2.1 Versuchsaufbau

Der Versuch ?? muss durchgeführt sein und die Ergebnisse müssen vorliegen. Für die Auswertung muss ein Computer mit der Software Wireshark bereit stehen.

#### 2.3.2.2 Versuchsdurchführung

Die aufgezeichneten Protokolldaten werden mit Wireshark geöffnet.

#### 2.3.2.3 Auswertung des Versuchs

Zur einfacheren und genaueren Auswertung des Versuchs sind in der Aufgabenbeschreibung 12 Fragen beschrieben.

Verbindungsaufbau  
skizzieren

1. Welche Rahmen dienen der TEI-Vergabe, und welcher TEI-Wert wird dem Telefon von der Vermittlung zugewiesen?

Die Tei vergabe wird über das Protocol TEI mit der info Itenty Request behandelt. Diese fordert einen TEi Wert zur indentifizierung des Endgerätes an.

## 2 BA Versuch

No.	Time	Source	Destination	Protocol Info
13	3601.022936	User	Network	TEI Identity Request

Frame 13 (8 bytes on wire, 8 bytes captured)  
Arrival Time: May 19, 2014 14:40:01.115936000  
[Time delta from previous captured frame: 1.999952000 seconds]  
[Time delta from previous displayed frame: 1.999952000 seconds]  
[Time since reference or first frame: 3601.022936000 seconds]  
Frame Number: 13  
Frame Length: 8 bytes  
Capture Length: 8 bytes  
[Frame is marked: False]  
[Protocols in frame: isdn:lapd:tei\_management]  
Point-to-Point Direction: Sent (0)

### ISDN

Channel: D (0)

Link Access Procedure, Channel D (LAPD)

[Direction: User->Network (0)]

Address Field: 0xfcff

1111 11... = SAPI: Layer 2 management procedures (63)

....0. .... = C/R: 0

....0 .... = EA1: 0

.... 1111 111. = TEI: 127

.... ....1 = EA2: 1

Control field: U, func=UI (0x03)

000. 00.. = Command: Unnumbered Information (0x00)

....11 = Frame type: Unnumbered frame (0x03)

TEI Management Procedure, Channel D (LAPD)

Entity: 0x0f

Reference: 30453

Msg: Identity Request (1)

1111 111. = Action: 127

....1 = Extend: 1

Die Anfrage wird durch einen Identity Assigned bestätigt und ein TEI Wert wird dem Endgerät zugewiesen. In diesem Fall wird der Wert 64 zugewiesen.

### 2.3 Aufzeichnungen und Interpretationen des ISDN-D-Kanal Protokolls

No.	Time	Source	Destination	Protocol	Info
14	3601.028984	Network	User	TEI	Identity Assigned

Frame 14 (8 bytes on wire, 8 bytes captured)  
Arrival Time: May 19, 2014 14:40:01.121984000  
[Time delta from previous captured frame: 0.006048000 seconds]  
[Time delta from previous displayed frame: 0.006048000 seconds]  
[Time since reference or first frame: 3601.028984000 seconds]  
Frame Number: 14  
Frame Length: 8 bytes  
Capture Length: 8 bytes  
[Frame is marked: False]  
[Protocols in frame: isdn:lapd:tei\_management]  
Point-to-Point Direction: Received (1)  
ISDN  
Channel: D (0)  
Link Access Procedure, Channel D (LAPD)  
[Direction: Network->User (1)]  
Address Field: 0xfeff  
1111 11.. .. = SAPI: Layer 2 management procedures (63)  
.... 1. .... = C/R: 1  
.... 0 .... = EA1: 0  
.... 1111 111. = TEI: 127  
.... 1 = EA2: 1  
Control field: U, func=UI (0x03)  
000. 00.. = Command: Unnumbered Information (0x00)  
.... 11 = Frame type: Unnumbered frame (0x03)  
TEI Management Procedure, Channel D (LAPD)  
Entity: 0x0f  
Reference: 30453  
Msg: Identity Assigned (2)  
1000 000. = Action: 64  
.... 1 = Extend: 1

#### 2. Wann ist der Aufbau der Schicht 2 abgeschlossen?

Der Aufbau ist nach dem senden eines unnumbered Acknowledge Frame der zur Bestätigung des Set Asynchronous Balance Mode Extended Frame benutzt wird

## 2 BA Versuch

No. Time Source Destination Protocol Info  
15 3601.038936 User Network LAPD U P, func=SABME

Frame 15 (3 bytes on wire, 3 bytes captured)  
Arrival Time: May 19, 2014 14:40:01.131936000  
[Time delta from previous captured frame: 0.009952000 seconds]  
[Time delta from previous displayed frame: 0.009952000 seconds]  
[Time since reference or first frame: 3601.038936000 seconds]  
Frame Number: 15  
Frame Length: 3 bytes  
Capture Length: 3 bytes  
[Frame is marked: False]  
[Protocols in frame: isdn:lapd:data]  
Point-to-Point Direction: Sent (0)  
ISDN  
Channel: D (0)  
Link Access Procedure, Channel D (LAPD)  
[Direction: User->Network (0)]  
Address Field: 0x0081  
0000 00.. .... = SAPI: Q.931 Call control procedure (0)  
.... 0. .... = C/R: 0  
.... 0 .... = EA1: 0  
.... 1000 000. = TEI: 64  
.... 1 = EA2: 1  
Control field: U P, func=SABME (0x7F)  
...1 .... = Poll: Set  
011. 11.. = Command: Set Asynchronous Balanced Mode Extended (0x1b)  
.... 11 = Frame type: Unnumbered frame (0x03)  
No. Time Source Destination Protocol Info  
16 3601.042952 Network User LAPD U F, func=UA

Frame 16 (3 bytes on wire, 3 bytes captured)  
Arrival Time: May 19, 2014 14:40:01.135952000  
[Time delta from previous captured frame: 0.004016000 seconds]  
[Time delta from previous displayed frame: 0.004016000 seconds]  
[Time since reference or first frame: 3601.042952000 seconds]  
Frame Number: 16  
Frame Length: 3 bytes  
Capture Length: 3 bytes  
[Frame is marked: False]  
[Protocols in frame: isdn:lapd:data]  
Point-to-Point Direction: Received (1)  
ISDN  
Channel: D (0)  
Link Access Procedure, Channel D (LAPD)  
[Direction: Network->User (1)]  
Address Field: 0x0081  
0000 00.. .... = SAPI: Q.931 Call control procedure (0)  
.... 0. .... = C/R: 0  
.... 0 .... = EA1: 0  
.... 1000 000. = TEI: 64  
.... 1 = EA2: 1  
Control field: U F, func=UA (0x73)  
...1 .... = Final: Set  
011. 00.. = Response: Unnumbered Acknowledge (0x18)  
.... 11 = Frame type: Unnumbered frame (0x03)



## 2.3 Aufzeichnungen und Interpretationen des ISDN-D-Kanal Protokolls

### 3. Welchen ISDN-Dienst fordert das Telefon von der Vermittlungsstelle an, welche Übertragungskapazität benötigt dieser Dienst?

Das Telefon fordert den Dienst zum Übertragen von Sprache an. Dies ist ein Kanal mit 64 kbit/s Übertragungskapazität. Im folgenden Frame kann man dies herauslesen. Es handelt sich hierbei um eine Setup Nachricht.

No.	Time	Source	Destination	Protocol	Info
192	4097.270936	User	Network	Q.931	SETUP

```
Frame 192 (24 bytes on wire, 24 bytes captured)
Arrival Time: May 19, 2014 14:48:17.363936000
[Time delta from previous captured frame: 0.022000000 seconds]
[Time delta from previous displayed frame: 0.022000000 seconds]
[Time since reference or first frame: 4097.270936000 seconds]
Frame Number: 192
Frame Length: 24 bytes
Capture Length: 24 bytes
[Frame is marked: False]
[Protocols in frame: isdn:lapd:q931]
Point-to-Point Direction: Sent (0)
ISDN
Channel: D (0)
Link Access Procedure, Channel D (LAPD)
[Direction: User->Network (0)]
Address Field: 0x0081
0000 00.. .... = SAPI: Q.931 Call control procedure (0)
.... 0. .... = C/R: 0
.... 0 .... = EA1: 0
.... 1000 000. = TEI: 64
.... 1 = EA2: 1
Control field: I, N(R)=0, N(S)=0 (0x0000)
0000 000. .... = N(R): 0
.... 0000 000. = N(S): 0
.... 0 = Frame type: Information frame (0x0000)
Q.931
Protocol discriminator: Q.931
Call reference value length: 1
Call reference flag: Message sent from originating side
Call reference value: 01
Message type: SETUP (0x05)
Bearer capability
Information element: Bearer capability
Length: 3
1... .... = Extension indicator: last octet
.00. .... = Coding standard: ITU-T standardized coding (0x00)
...0 0000 = Information transfer capability: Speech (0x00)
1... .... = Extension indicator: last octet
.00. .... = Transfer mode: Circuit mode (0x00)
...1 0000 = Information transfer rate: 64 kbit/s (0x10)
1... .... = Extension indicator: last octet
.01. .... = Layer identification: Layer 1 identifier (0x01)
...0 0011 = User information layer 1 protocol: Recommendation G.711 A-law (0x03)
Calling party number: '100'
Information element: Calling party number
Length: 5
.... 0000 = Numbering plan: Unknown (0x00)
.000 .... = Number type: Unknown (0x00)
```

### 4. Welche Kodierung des Sprachkanals wird gewählt?

ITU-T standardized coding hab ich hier aufm block stehen.

## 2 BA Versuch

5. Welche MSN-Nummer wird in welchem Rahmen übertragen?
6. Wann ist der gesicherte Aufbau der Schicht 3 abgeschlossen?
7. Welchen B-Kanal weist die Vermittlung der Verbindung zu?  
B1 Channel (0x01)
8. Wo wird die gerufene Telefonnummer übermittelt?
9. Mit welchem Rahmen bestätigt die Vermittlung die Vollständigkeit der Rufnummer und beginnt den angeforderten Teilnehmer anzuwählen?  
Alerting
10. Mit welchem Rahmen signalisiert die Vermittlung, dass der gerufene Teilnehmeranschluss ein Endgerät besitzt, das den Sprachdienst erfüllen kann und ein Rufsignal aussendet?
11. Wann sind die Schichten 3 und 2 jeweils wieder vollständig abgebaut?
12. Bei ISDN gibt es die Möglichkeit, bei einem abgehenden Ruf die Zielrufnummer vor oder nach dem Abheben des Hörers einzugeben. Wodurch unterscheidet sich die Signalisierung auf Schicht 3 (SETUP- und INFO Nachricht) der Teilnehmerschnittstelle in diesen beiden Fällen?

Haben wir den Versuch mit dem aufgelegten Hörer gemacht?

Nur dass das TODO die Tabelle nicht kaputt macht

### 2.3 Aufzeichnungen und Interpretationen des ISDN-D-Kanal Protokolls

Nachricht / Nachrichtenelement	Alert	Call Proc	Conn	DISC	Info	Rel	Rel. Comp	Setup	Setup Ack
Bearer Capability									
Cause									
Channel Identification									
Process Indicator									
Display									
Date / Time									
Calling Party Number									
Called Party Number									
Sending Complete									
Facility									
User to User Information									

Tabelle 2.1: Genutzte Nachrichten



## 3 RSP Versuch

### 3.1 Einleitung

### 3.2 Switch und Router Konfiguration

#### 3.2.1 Router start up running config

### 3.3 Packet Tracer

Mit Packet Tracer lassen sich verschiedene Hardware-Szenarien nach spielen und virtuell aufbauen. Es stehen verschiedene Hardware-Produkte zur Verfügung die sich konfigurieren lassen und beliebig verkabeln lassen. Das Folgende Szenario soll erstellt und damit gearbeitet werden.

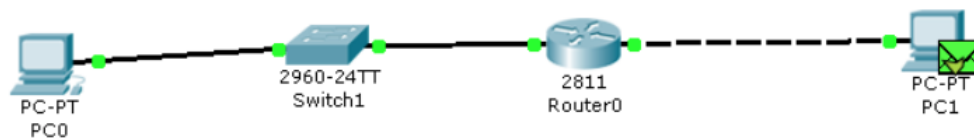


Abbildung 3.1: SMS von Netzwerk an Mobilestation

#### 3.3.1 Versuchsaufbau

#### 3.3.2 Messungen

#### 3.3.3 Simulation Echo-Request/-Reply

### 3.4 Untersuchung des Paketflusses mit Wireshark



## **4 RSC Versuch**

### **4.1 Einleitung**

### **4.2 Downlink**

### **4.3 Uplink**

### **4.4 ARFCN**

### **4.5 Untersuchung des Paketflusses mit Wireshark**





## **5 SDH Versuch**

### **5.1 Einleitung**

### **5.2 Downlink**

### **5.3 Uplink**

### **5.4 ARFCN**

### **5.5 Untersuchung des Paketflusses mit Wireshark**



## **6 RN Versuch**

### **6.1 Einleitung**

### **6.2 Downlink**

### **6.3 Uplink**

### **6.4 ARFCN**

### **6.5 Untersuchung des Paketflusses mit Wireshark**



## **Kolophon**

Dieses Dokument wurde mit der L<sup>A</sup>T<sub>E</sub>X-Vorlage für Abschlussarbeiten an der htw saar im Bereich Informatik/Mechatronik-Sensortechnik erstellt (Version 1.0). Die Vorlage wurde von Yves Hary und André Miede entwickelt (mit freundlicher Unterstützung von Thomas Kretschmer und Helmut G. Folz).