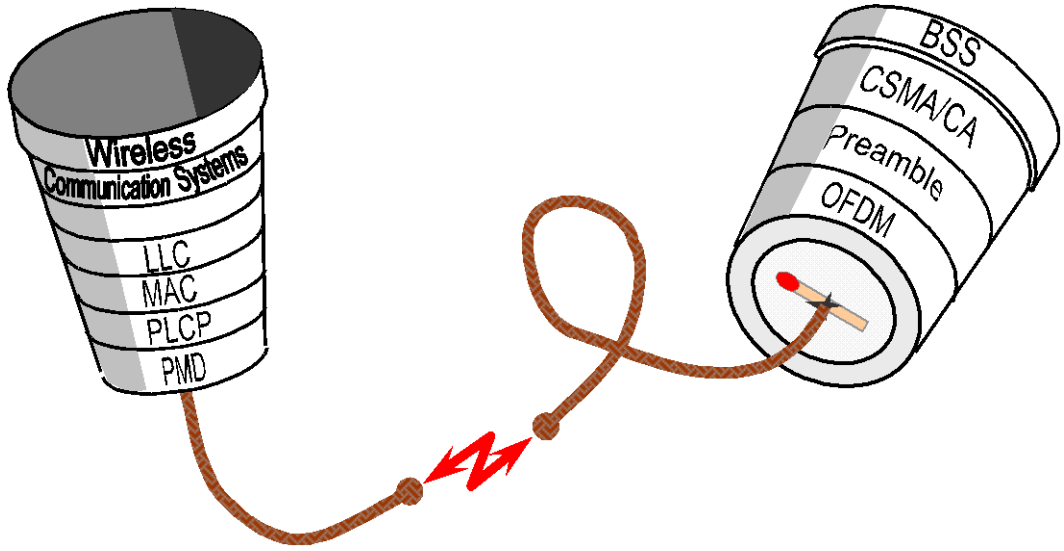
Instructions for the Lab

Wireless Communication Systems

Holger Stahl



Contents:

References ..... **ii**

Exercise 1: *SPEC* (*Spectrum Analysis*) **1-1**

Exercise 2: *DVB-T2* **2-1**

Exercise 3: *LTE-RF* (*RF Measurements*) **3-1**

Exercise 4: *LTE-Proto* (*Protocol Analysis*) **4-1**

Exercise 5: Getting Started with Wireshark **5-1**

Exercise 6: VoIP (Voice over IP) **6-1**

Exercise 7: The *MPEG-2* Transport Stream **7-1**

Hints for Productivity and Fun in the Lab:

Experiments always start with planning! The time provided for each Lab exercise should be sufficient to do it without hectic – presupposing that you already know what you have to do. As an aid, this instruction book provides some background information and preparation problems. It is expected that each student is thoroughly prepared for each exercise. The preparation problems for each exercise have to be solved and recorded into this instructions book individually by each student. If necessary, use additional resources from the library or the Internet. At the beginning of each Lab exercise block, the preparation problems will be discussed together with the Lab advisor.

References

Most of the prerequisite knowledge needed for the Lab exercises is taught in the lecture *Wireless Communication Systems*. However, some of the elective exercises (e.g. VoIP, ...) are not covered directly by the lecture topics. It is supposed that you obtain the necessary prerequisite knowledge on your own by utilizing the University library and searching for appropriate articles in the Internet. With each Lab exercise, you find cross links onto these resources, which are appropriate for the special topic:

Resources already mentioned in the Lecture Handout (page 0-5)

[Fis] W. Fischer: *Digital Video and Audio Broadcasting Technology*
In the library, you find a printed English version and a German **eBook!**

[Gho] A. Ghosh et al.: *Fundamentals of LTE* **eBook!**

[Saut] M. Sauter: *Grundkurs Mobile Kommunikationssysteme UMTS, HSDPA und LTE, GSM, GPRS und Wireless LAN* **eBook!**

[Ses] S. Sesia et al.: *LTE – The Long Term Evolution*

[Sta] W. Stallings: *Data and Computer Communications*

[Tan] A.S. Tanenbaum: *Computernetzwerke/Computer Networks* **eBook!**

[TrWe] U. Trick, F. Weber: *SIP, TCP/IP, und Telekommunikationsnetze*

Additional Resources, specialized for this Lab

[DvbT] *Informationsportal der Initiative DVB-T2 HD – Sendertabelle*.
www.dvb-t2hd.de→Downloads→Senderstandorte und Kanäle

[EtsiSpec] *3GPP Specification for UMTS* and *LTE*. ETSI, www.3gpp.org/specification-numbering
⇒ Steps for downloading a specification as *pdf*-Document:
(1) For UMTS/LTE: Click on the correct number link in the 2nd column "3G and beyond".
(2) On the next page, Click on the Specification number.
(3) On the next page, Open the tab "Versions".
(4) Scroll down to the desired release (e.g. "Release 10" for LTE), and choose the newest version of that release.
(5) Click the ETSI Symbol in one of the last columns for this version!
(6) On the next page, Click the Symbol in the right upper part of the page.

[Niv] [Niv] Homepage niviuk.free.fr
⇒ Several tables and interactive tools for LTE and other mobile communication systems

[Rau] C. Rauscher: *Fundamentals of Spectrum Analysis*. Rohde&Schwarz, Munich, 2005
⇒ 10 copies available in the library

[Sie2] G. Siegmund: *Technik der Netze, Band 2 – Neue Ansätze (VoIP, ...)*, Offenbach, 2010
⇒ 5 copies (in German) available in the library

[WiS] *Wireshark*. *www.wireshark.org*, ⇒ Freeware tool for the analysis of network protocols

Exercise 1: *SPEC* (*Spectrum Analysis*)

In this exercise, you will learn some basics about performing measurements with a Spectrum Analyser.

References: [Rau]

1.1 Equipment

Figure 1-1: Handheld Spectrum Analyser ROHDE&SCHWARZ *FSH3*

- Spectrum Analyser ROHDE&SCHWARZ *FSH3* (see manual on page 1-10 and in the *Online Community* of the University)
- Analog Video Test Signal Generator GRUNDIG VG1000
- Television Measurement Receiver KWS *AMA200*
- Omni-directional multi-band antenna *Maxi-Saver*
- Loudspeakers connected to the AF (*Audio Frequency*) output of *FSH3*

1.2 Background

1.2.1 How a Spectrum Analyser Works

Representation of signals in the time domain and in the frequency domain

[From the *FSH3* manual, see the *Online Community* of the University].

Basically, a signal can either be analysed in the time domain or in the frequency domain. In the time domain, how the signal varies with time can be observed on an oscilloscope, for example. In the frequency domain, a Spectrum Analyser can be used to display the frequency components of a signal.

Both modes are essentially equivalent, because applying the Fourier Transform to any signal converts it into its spectral components. However, depending on the signal characteristics to be measured, one method is usually more appropriate than the other. Just by glancing at an oscilloscope, it is possible to tell whether a measurement signal is a sine signal, a square wave with a certain on/off ratio or a saw tooth. However, it is not at all obvious what the harmonic content of the signal is or if low-level signals are superimposed. This is easy to see with a Spectrum Analyser.

Figure 1-2 shows an example for the two measurement techniques: In the time domain, an oscilloscope is showing a section of a signal which is approximately a square wave. The same signal viewed with a Spectrum Analyser shows a line spectrum, i.e. the fundamental and the harmonics.

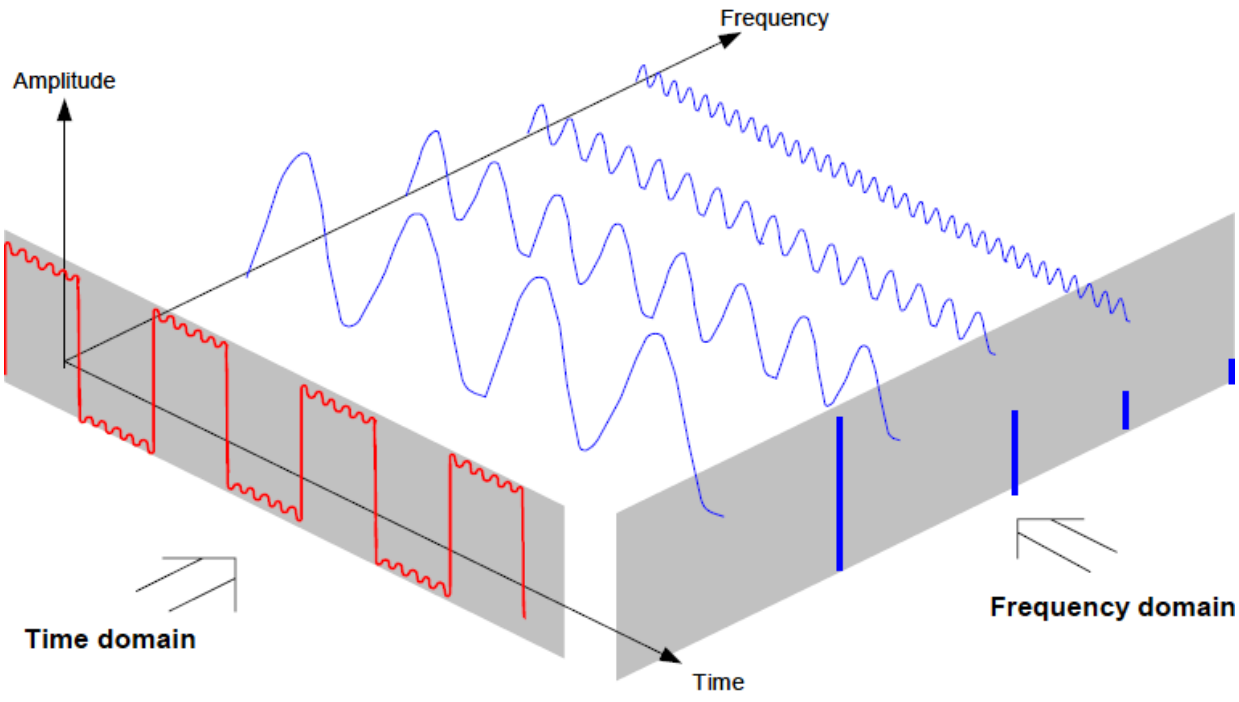


Figure 1-2: Representation of a signal in the time and frequency domain. Example: A square wave signal

The periodic square wave in the time domain can be Fourier transformed to the frequency domain. In the case of a square wave there is a fundamental (= frequency of the square wave) and its odd harmonics. Using a narrow bandpass filter, the Spectrum Analyser makes measurements in the frequency domain. Only at those frequencies where there is a signal, there a reading which gives the amplitude of the frequency component.

Block diagram of a Spectrum Analyser

Figure 1-3 shows a functional block diagram of a spectrum analyser like the *FSH3*. Please note, that this diagram has been strongly simplified in order to explain the influence of the most important parameters, which can be adjusted by the user at the front panel. These parameters have been printed in **bold** ⇩.

Usually, a spectrum analyser is capable of investigating an RF signal in two modes: In the time domain and in the frequency domain. The latter mode is called *Normal Mode*, the first *Zero Span Mode*.

(1) Analyser in *Normal Mode*

For splitting up an RF signal into its spectral components, a band pass filter is used that sweeps over the frequency range that is to be observed. This band pass filter is called the *Resolution Filter* and is specified by the *Resolution Bandwidth* (RBW). Instead of sweeping the centre frequency of this filter, it operates on a fixed IF (*Intermediate Frequency*). In order to analyse the amplitude of a certain frequency component, the RF signal is shifted in the frequency domain by mixing (i.e. multiplying) it with a sinus wave from an LO (*Local Oscillator*).

The output of the *Resolution Filter* is rectified (*Envelope Detector*) and smoothed by a *Video Filter*. The bandwidth of the latter is called VBW (*Video Bandwidth*) and can be used to smooth out noisy signals, equivalent to averaging with regard to time. This output signal of the *Video Filter* vertically deflects the writing point of the display. Usually, a logarithmic scaling is applied, which can be adjusted by the parameter *Range*.

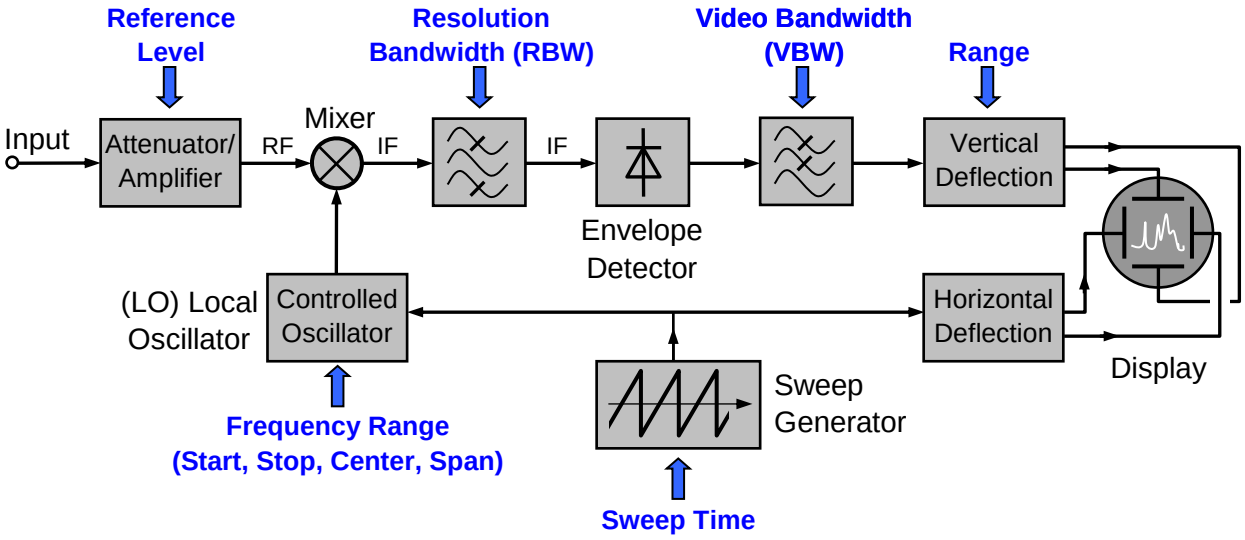


Figure 1-3: Block diagram of a Spectrum Analyser in *Normal Mode*

The horizontal deflection of the writing point on the screen is controlled by a *Sweep Generator*. Simultaneously, the *Sweep Generator* also adjusts the analysed frequency by changing the frequency of the LO. Like with an oscilloscope, the writing point moves periodically from the left edge to the right edge of the screen, showing the amplitude at the respective frequency. The period of the *Sweep Generator* is called the *Sweep Time* (SWT). The frequency range to be analysed is defined by a couple of frequencies, either *Start*/*Stop* or *Center*/*Span*.

(2) Analyser in Zero Span Mode

In *Zero Span Mode*, the analyser is tuned to a fixed frequency in order to analyse the transient structure of the input signal, instead of sweeping. The block diagram is the same as shown above for *Normal Mode*, but the LO operates on a fixed frequency instead of being controlled by the *Sweep Generator*.

1.2.2 Analog TV broadcast signals

During this exercise, you will also investigate analogue TV signals. Aerials on the university building *block R* receive TV signals from terrestrial transmitters and from satellites, which are distributed via cable to the Communication Systems Lab. The following explanations give a brief introduction into the European TV standard in order to understand the measurements:

The basic principle of analogue Television

To record a picture, its luminance is scanned line by line from left to right, from top to bottom. In the TV receiver, the cathode-ray writes the same picture line by line onto the screen. 25 times per second, a completely new picture is transmitted. Since these 25 Hz would be recognized by human eyes as a flicker, instead of a complete picture, two half-pictures are written to the screen. These two pictures are interlaced accurately, in order to complement each other perfectly:

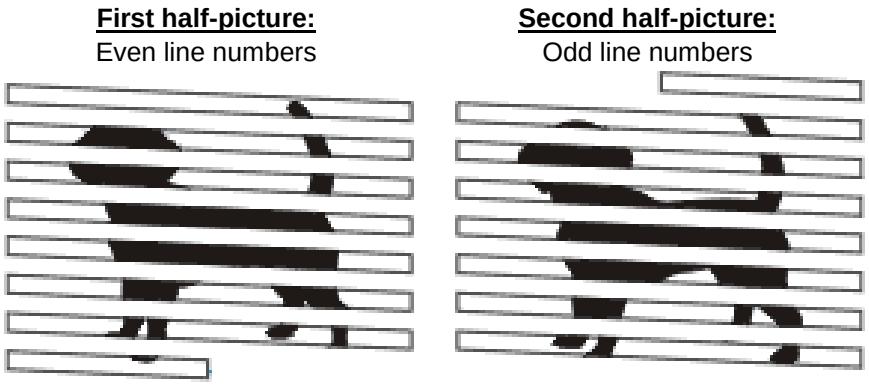


Figure 1-4: A picture is written line-by-line in two subsequent half-pictures for even and odd line numbers, respectively

Information needed by a TV receiver

For transmitting a complete colour TV channel including stereo sound, the following information has to be transmitted:

(1) Luminance

This means the 'black and white' information of the picture contents. The luminance is transmitted by AM (*Amplitude Modulation*) one-sideband modulation with a bandwidth of 5 MHz. As shown in Figure 1-5, a high amplitude of 75% means 'black' and a low amplitude of 10% means 'white'. The maximum amplitude of the carrier (100%) is reserved for line and picture synchronisation.

(2) Sound

For stereo sound, two audio channels are needed. As shown in Figure 1-6, two carriers transport the mono signal *Left+Right* and the sole *Right* signal. The modulation scheme is FM (*Frequency Modulation*).

(3) Colour

For compatibility reasons, the colour signal is transmitted on a special carrier at 4.43 MHz above the centre frequency of the channel. The modulation scheme of the European PAL (*Phase Alternating Line*) standard is PM (*Phase Modulation*) in combination with AM. See Figure 1-6.

(4) Synchronization and blanking

To indicate the beginning of a new line (at the left corner of the screen) or a new picture (at the upper edge of the screen), synchronisation pulses are provided inside the luminance signal as shown in the following figure. Each time a line ends, a pulse is sent that exceeds the maximum valid luminance

value of 75% amplitude. The duration of this pulse determines, if a line synchronisation or a picture synchronisation has to be done.

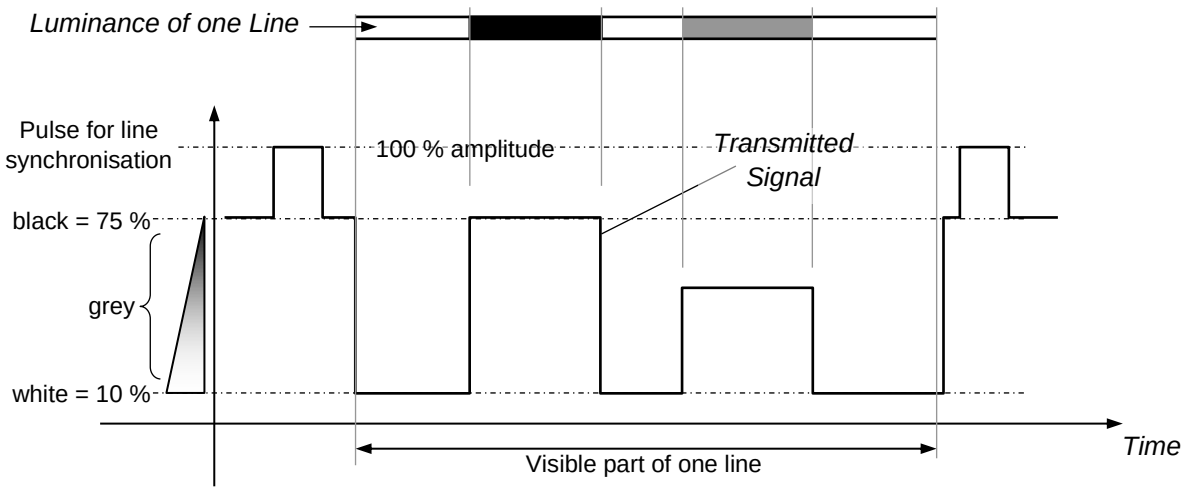


Figure 1-5: Luminance of one line and the transmitted luminance and synchronisation signal vs. time

Spectrum of a TV signal

The following figure shows how the TV information is spectrally packed into one TV channel. The main bandwidth is occupied by the picture carrier, which contains the luminance signal and the synchronisation signals. In order to save bandwidth, this AM signal usually is transmitted in form of a so-called *Vestigial Sideband Signal* (VSB), with the lower half of the spectrum suppressed by a filter:

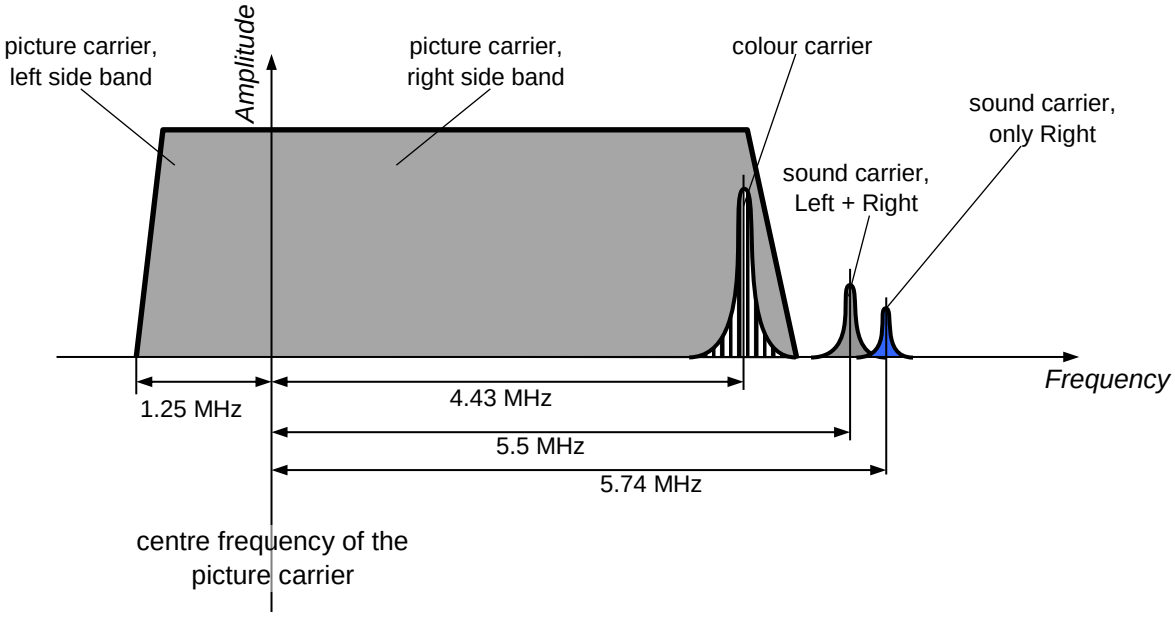


Figure 1-6: Contents of a Spectrum of a TV signal

1.3 Preparation Problems (to be answered before doing the Lab!)

a) **Spectral Analysis**: Look at Figure 1-3:

⇒ What is the difference between *Normal Mode* and *Zero Span Mode*?

Normal Mode:	**Zero Span Mode:**

⇒ What happens, if the parameter **SWT** is set too short? Find the answer in the book [Rau]!

⇒ Get the book [Rau], and find out how the parameter **SWT** should depend on **RBW**!

⇒ List two advantages of using a high *Resolution Bandwidth*?

⇒ List two advantages of using a low *Resolution Bandwidth*?

b) **Operation of the Analyser *FSH3***: See manual on page 1-10!

⇒ Figure 1-3 shows 9 parameters of the analyser that significantly influence the operation. Find the *FSH3* menus on page 1-10 for adjusting these parameters and put circles around them.

c) **Audio and Video Broadcast**

⇒ Which is the frequency range of the FM broadcast radio band ('UKW') in Germany?

⇒ Give the modulation schemes for the following information that is contained in a TV signal:

Luminance:
Audio L+R:
Audio R:
Colour:
Synchronisation:

d) **Signal Properties of Digitally Modulated Wireless Communication Signals**

⇒ In Experiment 3, we will examine various signals of digital communication systems. In order to identify these signals, find out some properties of ***DVB-T2***, ***GSM***, ***UMTS,*** & ***LTE*** signals:

	DVB-T2	**GSM**	**UMTS/W-CDMA**	**LTE**
Channel Bandwidth				
Form of the Spectrum				
Bursty Power? Burst Duration?				

1.4 Practical Part

Experiment 1 : Signal Analysis in the Spectral Domain (50 min)

a) **Getting Started with the Spectrum Analyser *FSH3***

⇒ Press Preset to reset all instrument parameters to default values.

⇒ Get familiar with the display. Where are the following parameters displayed on the screen?

- Center Frequency (*Center*)
- Span Frequency (Span)
- Resolution Bandwidth (*RBW*)
- Video Bandwidth (*VBW*)
- Sweep Time (*SWT*)
- Reference Level (*Ref Level*)

⇒ Set *RBW* = 100 kHz (BW→MANUAL RES BW) and write down the *SWT*:

SWT for RBW = 100 kHz:

⇒ Set *RBW* = 10 kHz and write down the *SWT*:

SWT for RBW = 10 kHz:

⇒ How does the *SWT* depend the *RBW*?

Explain the reason for this dependency, being automatically considered by the FSH3:

b) **FM Audio Broadcast**

We will look at radio broadcast signals now:

⇒ Connect the Omni-Directional Antenna to the Analyser's input and place it outside the window.

⇒ Connect the *FSH3* input to the socket.

⇒ Adjust the parameters *Span, RBW, VBW, RefLvl,* and *Range* in order to have a good overview of the FM radio band.

⇒ How many channels are there in the examined FM band?

⇒ What is the approximate bandwidth that an FM channel occupies?

10 , **100** or **1000 kHz**? Put a circle around the correct answer.

⇒ Set the Marker (MARKER→MARKER) to 98.5 MHz and demodulate to the loudspeakers.

Choose an appropriate *RBW* and appropriate parameters (MARKER→MARKER DEMOD) for demodulation.

c) **Television Broadcast**

Analyse the spectrum of the GRUNDIG Test Signal Generator at a frequency of approx. 90 MHz.

⇒ What is the exact *Center Frequency* of the signal?

⇒ Does the video test signal generator actually transmit the luminance signal in form of a *Vestigial Sideband Signal* (VSB)? Give reason!

⇒ Does the test signal generator actually transmit a stereo sound signal? Demodulate it!

⇒ Does this TV channel broadcast a colour or a black/white program? Give reason!

Experiment 2 : Analysis of Signals in *Zero Span Mode* (40 min)

Use the *Zero Span Mode* of the analyser in order to see the AM-demodulated luminance signal:

a) **Analysis of picture lines**

⇒ Put the machine into *Zero Span Mode* at a *Center Frequency* observed in Experiment 1c), and set the AMPT→RANGE to *Linear 0–100%.*

⇒ Set *SWT* = 1 ms and *RBW* = 1 MHz.

⇒ Set the *Trigger Mode* (SWEEP→TRIGGER) to *Video* and adjust the *Trigger Level* in order to get a stable readout.

⇒ Find the synchronisation pulse and show it to the Lab advisor.

⇒ What is the reason for defining low amplitudes to show bright regions and high amplitudes to show black regions of the picture line?

⇒ Find out the horizontal deflection frequency from the readout: f_{hor} =

⇒ Switch the *Video Generator* to *VIDEOSIGNAL* = 

, and explain the spectrum!

b) **Analysis of a half-picture**

⇒ Set *SWT* = 50 ms and *RBW* = 100 kHz and adjust the Trigger in order to get a stable readout.

⇒ Find out the vertical deflection frequency.

f_{vert} =

⇒ How many lines does a (complete) picture have?

Experiment 3 : Identification of wireless signals being received in the Lab (20 min)

In this section, we will try to identify signals of digital communication systems, which can be received in the frequency range 500 MHz to 2.2 GHz. This can be done in three steps: First **(a)**, a coarse scan is done over the complete bandwidth, in order to find areas in this frequency range with activity. Then **(b)**, we will "zoom" into the spectral details of the areas that have been found before. Last **(c)**, we try to find transient structures in the signal amplitude.

a) **Coarse Spectral Scan of the Spectrum from 500 MHz to 2.2 GHz**

⇒ Reset all *FSH3* adjustments by pressing Preset.

⇒ In order to increase the sensitivity of the input, activate the *FSH3*'s built-in preamplifier by SETUP → Hardware Setup → Preamp: *ON* and
adjust the attenuator to SETUP → Hardware Setup → Dynamic Range: *LOW NOISE*

⇒ Connect the Omni-Directional Antenna to the Analyser's input again.

⇒ Adjust *RefLvl*, *Start* and *Stop Frequency* appropriately.

⇒ Which areas of activity can you make out?

Frequency Range (MHz)							
	From						
	To						

b) **Observation of Spectral Details**

⇒ "Zoom" into the areas of activity (by decreasing the *SPAN*) and guess from their spectral bandwidth and the form of the spectral amplitude distribution, which communication standard the signals belong to (write the standard below the table columns above!).

⇒ Note down one exemplary channel frequency for each of the standards GSM, DVB-T, and UMTS:

Standard	**DVB-T2**	**GSM**	**UMTS**	**LTE**
Channel Centre Frequency				

⇒ Discuss your observations with the lab advisor!

c) **Search for Transient Structures**

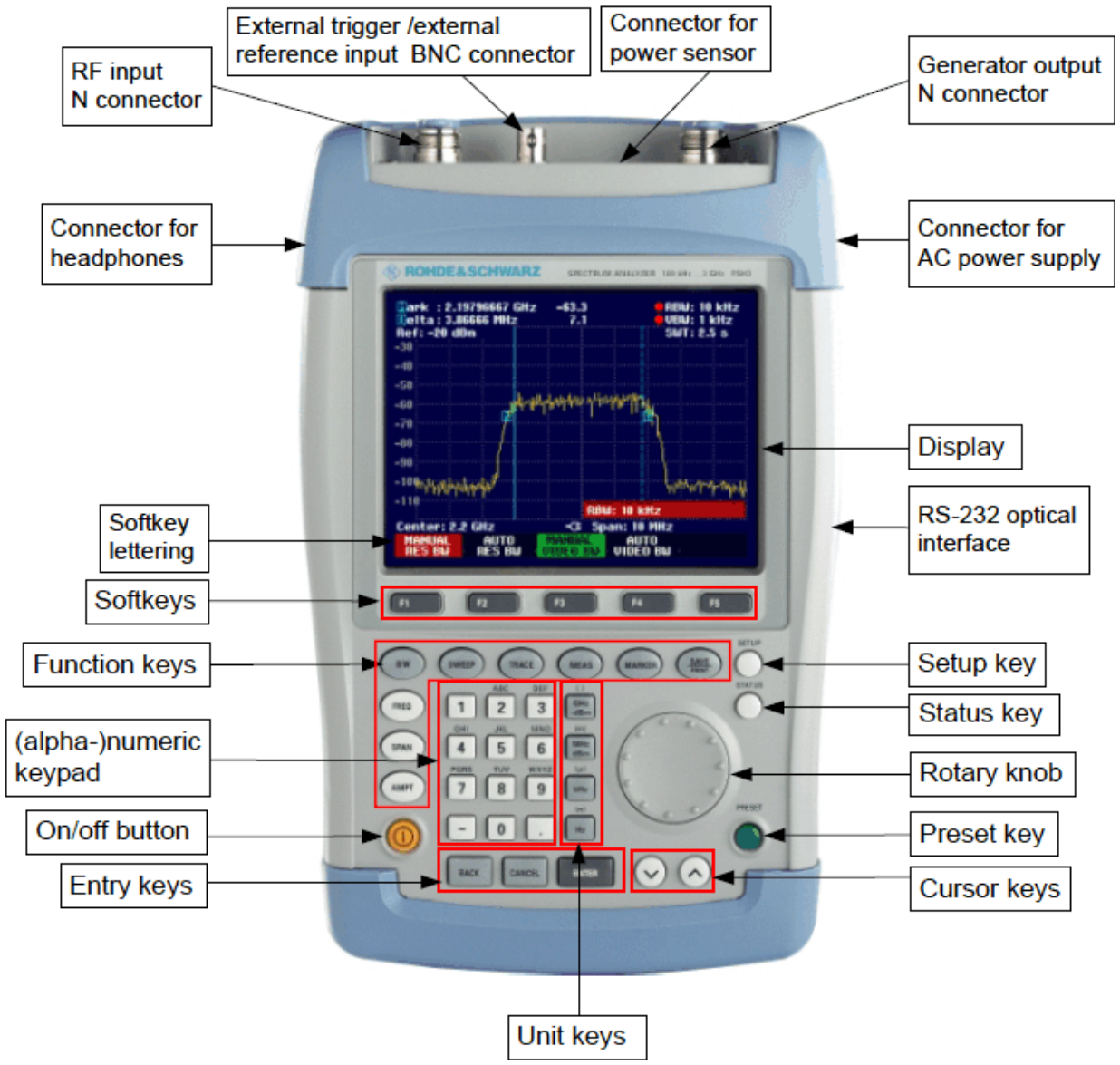
⇒ How does a transient (i.e. "pulsed" or "bursty") signal power look in *Normal Mode* and in *Zero Span Mode* of the Analyser?

⇒ Look again into the preparation question 1.3d): Which of the three communication standards should show a visible transient structure in the signal power? Verify the presence of that structure qualitatively and quantitatively and show your results to the lab advisor!

1.5 User Manual of the Spectrum Analyser R&S *FSH3*

We will use the handheld spectrum analyser *FSH3* to analyse *Bluetooth*, GSM, and broadcast signals in the time and in the frequency domain. The instrument used in the Lab also comprises a tracking signal generator, which for instance allows measurements of the (amplitude) frequency response of cables.

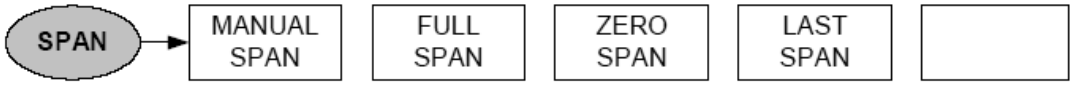
Front view



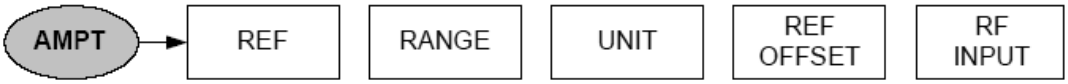
Menu Overview

Frequency entry

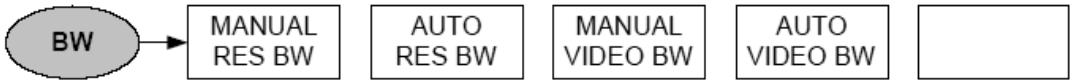
Frequency span entry



Level entry



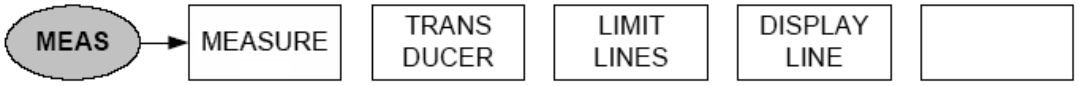
Bandwidth entry



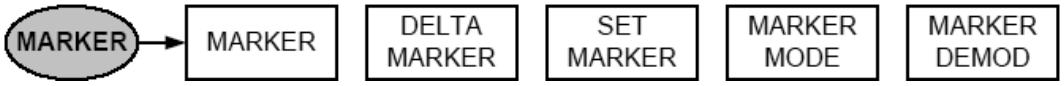
Sweep entry

Trace settings

Measurement functions



Markers



Exercise 2: *DVB-T2*

This exercise will impart some basics of the DVB-T(1) and DVB-T2 (*Digital Video Broadcast*) standards.

References: [Fis], [DvbT]

2.1 Equipment

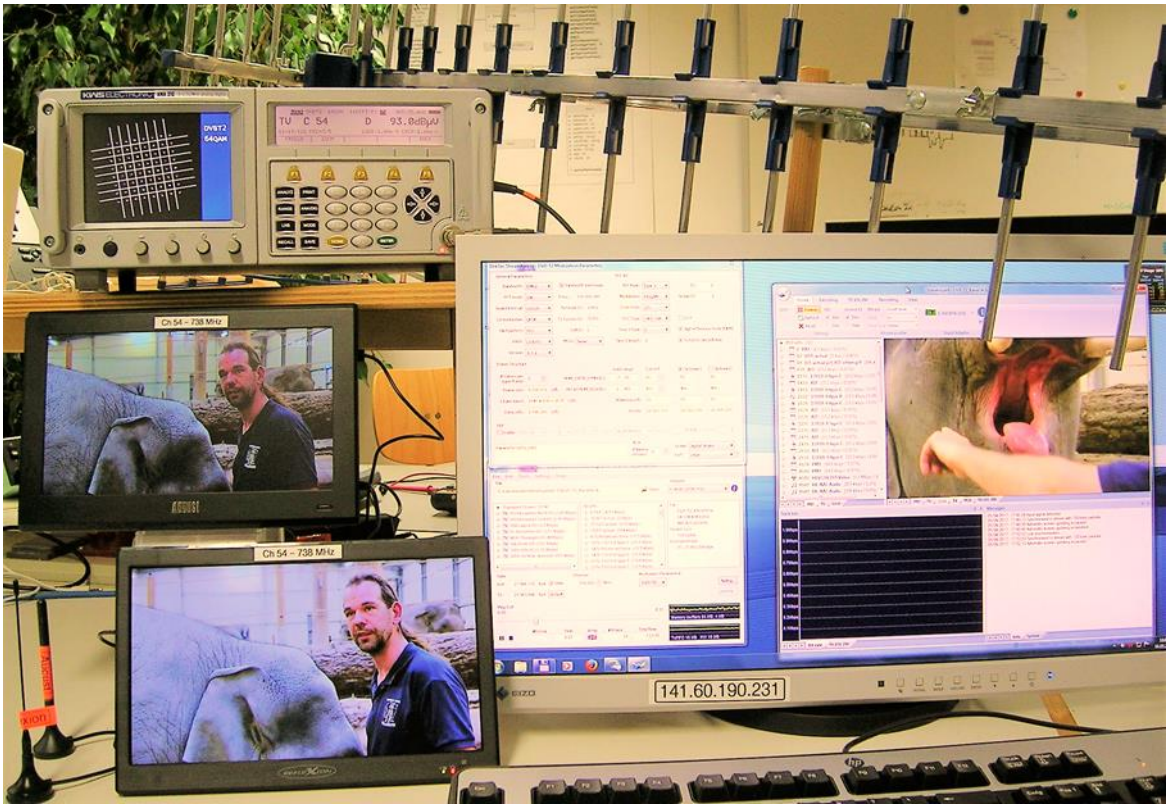


Figure 2-1: Hardware being used to explore DVB-T(1) and -T2 signals and their transmission channel

- PC equipped with:
  - ⇒ PCI VHF/UHF modulator card *DekTec DTA-115* (with λ/4 dipole conn. to RF power outlet)
  - ⇒ Software *DekTec StreamXpress* v3.19.1
  - ⇒ Software *DekTec StreamXpert* v2.1.4
- DVB-T Receivers:
  - ⇒ Measurement Receiver *KWS AMA310* (with BNC cable connected to ASI-out, and carry case)
  - ⇒ DVB-T Mini Receiver *August* (with power adapter)
  - ⇒ DVB-T Mini Receiver *Reflexion* (with power adapter)
- Antennas:
  - ⇒ Amplified VHF/UHF indoor aerial *One4All*
  - ⇒ High-gain roof antenna (18 dB *Yagi-Uda*) for the UHF range

2.2 Background

2.2.1 The COFDM Physical Layer of a DVB-T(1) Transmitter

Figure 2-2 shows the main components of the DVB-T (1st generation) physical layer, which was the basis for the development of DVB-T**2**. Before the TS (*Transport Stream*) data is actually modulated by the IFFT (*Inverse Fast Fourier Transform*) and an I/Q modulator, it has to be forearmed against the impairments of the transmission channel, i.e. fading and interference:

⇒ The FEC (*Forward Error Coding*) adds redundancy in order to correct bit errors in the receiver. DVB-T(1) uses a *hybrid FEC* scheme, combining a polynomial block coding algorithm (*Reed-Solomon* "outer FEC") with a *Convolutional Coder* ("inner FEC"), and a subsequent *Puncturing* process. Both coders have fixed code rates of $r = 188/204$ and $r = ½$. In order to adapt the overall code rate to the channel quality, the *Puncturing* block dismisses bits according to an adjustable regular pattern. The resulting overall code rate ranges between ($½ \cdot 188/204$) and ($^{7}/_{8} \cdot 188/204$).

⇒ The *Interleaving* "mixes" the order of the bits in order to spread the information and the redundancy over all subcarriers of the OFDM bandwidth. This is extremely important to support the error correction inside the receiver, because bit errors in "bad" subcarriers can be eliminated by the redundancy in "good" subcarriers.

Generally, the robustness of OFDM (*Orthogonal Frequency Division Multiplex*) can only be achieved together with a strong FEC and the subsequent interleaving! Therefore the name COFDM (Coded OFDM).

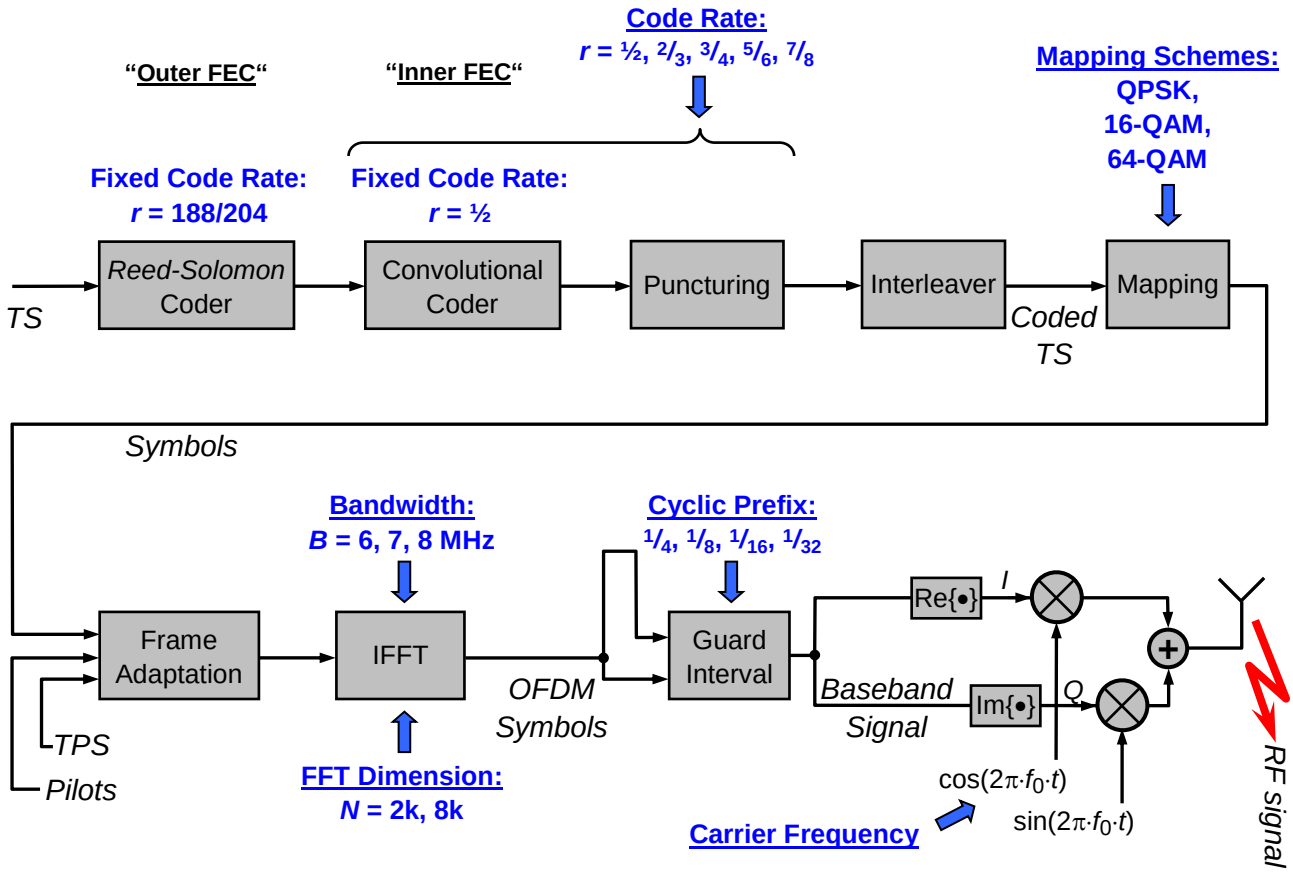


Figure 2-2: Simplified Block Diagram of the DVB-T**(1)** physical layer (usually called "COFDM Modulator") – The **bold text** and the arrows ⇨ mark the configuration parameters.

2.2.2 Hard- and Software Components Used in this Exercise

Figure 2-3 shows the main components of this exercise: The heart of the exercise are a TV transmitter based on the modulator card *DekTec DTA-115*, and the TV measurement receiver *KWS AMA310*:

- The **PCI card *DekTec DTA-115*** can be used in two different modes:
  - ⇒ Modulation of an MPEG-2 *Transport Stream* as a DVB-T(1) or DVB-T2 signal. The modulation parameters can be controlled by the play-out software *DekTec StreamXpress* (also used in the *MPEG-2* exercise). This software also contains a fading simulator, which simulates a multipath channel and/or an SFN (*Single Frequency Network*).
  - ⇒ Capturing of an MPEG-2 *Transport Stream*, provided by the ASI (*Asynchronous Serial Interface*) output of the *AMA310*. The TS is passed to the MPEG-2 analysis software *DekTec StreamXpert*.
- The measurement receiver *KWS AMA310* gets the RF signal to be analysed either from the real broadcast network *Wendelstein*/*Olympiaturm* or from the *DTA-115*.
- Two ordinary customer receivers *August* and *Reflexion* can be used to compare the reception quality.

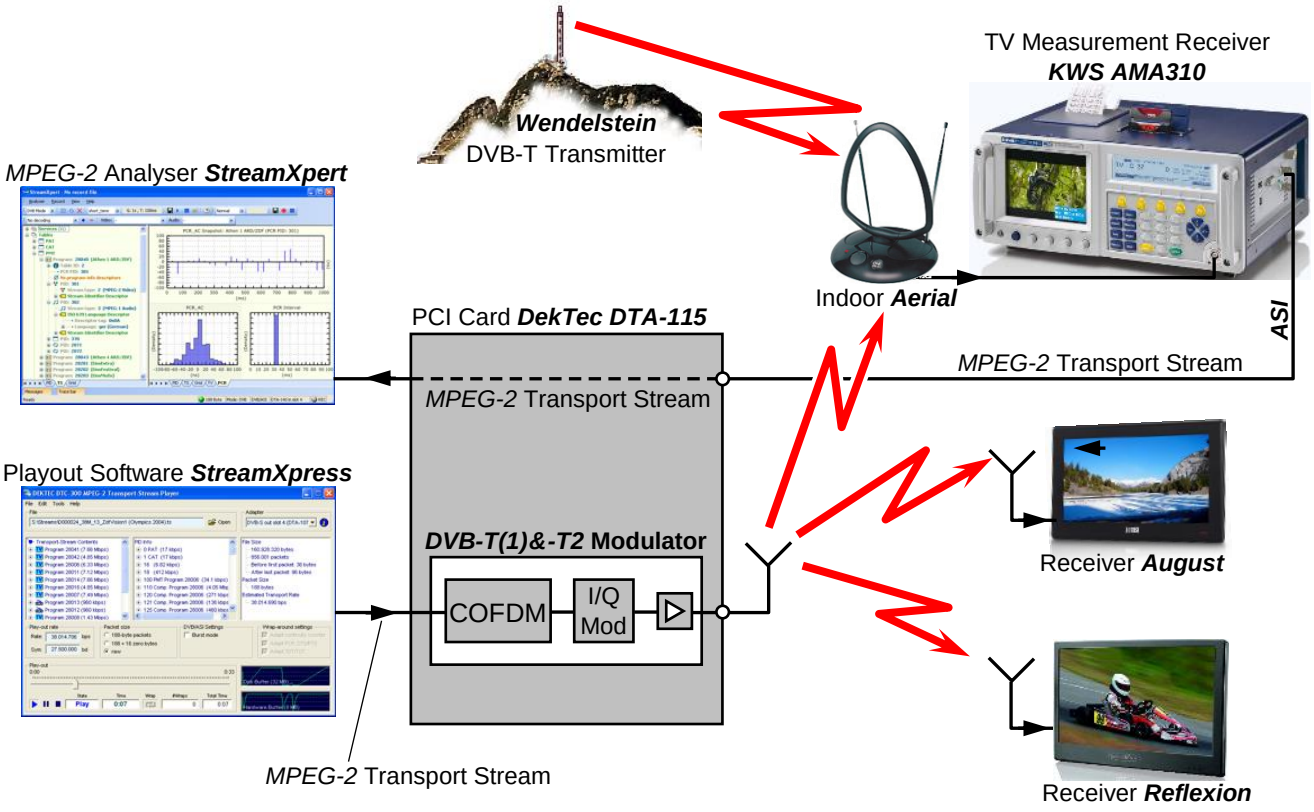


Figure 2-3: Block diagram of the hard- and software components in this *DVB-T2* exercise

2.2.3 Transmission Parameters of DVB-T(1)

The following diagram (from [Fis]) illustrates the interrelation between the various frequencies and bandwidths of a COFDM transmission due to the DVB-T(1) standard:

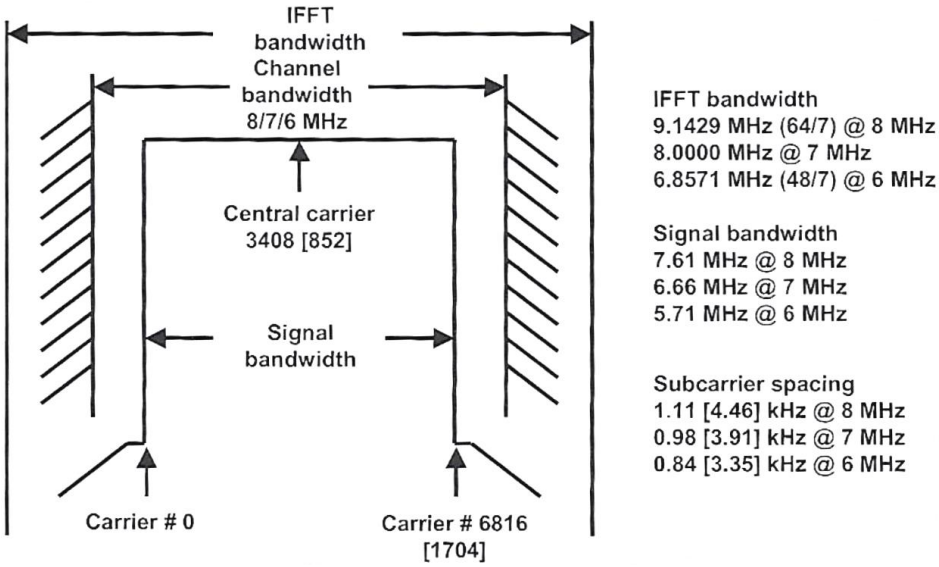


Figure 2-4: Spectral parameters of a DVB-T(1) signal in the two modes: 8k and [2k]. Taken from [Fis]

2.3 Preparation Problems (to be answered before doing the Lab!)

a) TV-Broadcast with DVB-T2 in the Region of *Upper Bavaria*

⇒ Which is the *channel bandwidth* used for TV broadcast in Germany?

⇒ Find out the frequency channels and the modulation parameters of the *Wendelstein*:

Channel Number	Frequency / MHz	Transmission Mode (FFT-Dimension)	Mapping	*CP* (Cyclic Prefix)	Code Rate	Resulting Data Rate / Mbit/s

⇒ Which parameters are responsible for the higher data rates of the channels 26, 35, & 48?

b) DVB-T(1) modulation parameters

⇒ The data rate of the "ARD Regional" TS is 21.56 Mbit/s. Which modulation parameters would be necessary, in order to transmit this TS over an 8-MHz-wide DVB-T(1) channel instead?

Transmission Mode:	Mapping	CP	Code Rate

⇒ Calculate the maximum net data rate (available for an *MPEG-2* TS) of a DVB-T**(1)** modulator:

f_{bmax} =

c) DVB-T2 modulation parameters

⇒ List four differences of the DVB-T2 modulator, compared to the block diagram of Figure 2-3.

1. **2.** **3.** **4.**

⇒ Which additional values exist for the "basic" parameters of DVB-T2 compared to DVB-T(1)?

Transmission Mode:	**Bandwidth:**	**Mapping:**	**CP:**	**Code Rate:**

⇒ Briefly explain the idea of diversity by "rotated constellations" in DVB-T2:

⇒ DVB-T2 defines multiple PLPs (*Physical Layer Pipes*). What for?

d) Physical Measurements in DVB-T(1) and DVB-T2

⇒ What are the definitions of the noise ratios CNR and MER? Look them up in [Fis]!

⇒ Calculate the thermal noise for a TV channel ($B = 8$ kHz) at room temperature and at an impedance of $R = 75\ \Omega$:

In the unit ‚V':
In the unit ‚dBµV':

e) Multipath Propagation and SFNs (*Single Frequency Networks*)

⇒ How does a COFDM spectrum look, which suffers from a *Delay Spread* $\Delta\tau = 1\ \mu s$?

⇒ Look up the distance *R-Bau*↔*Wendelstein*, and *R-Bau*↔*Olympiaturm* in *Google Earth*:

Distance to Wendelstein:	**Distance to Olympiaturm:**

⇒ Calculate the Delay Spread, if both stations could be received simultaneously:

2.4 Practical Part

Preparation: Open the directory **`Lab_R1.16`**, on the **`Desktop`** of your computer. Here, enter the **`zip`**-archive **`WiCS`**, and copy the directory **`DVB-T2`** directly to your **`Desktop`**. All files, which you will use and manipulate for this exercise should be contained in this directory!

Experiment 1 : Getting Started – Analysis of Terrestrially Received TV Signals (40 min)

You will analyse the TV signals from the transmitter "Wendelstein", using an indoor antenna.

a) **Spectrum analysis of the UHF band**

⇒ Let the *KWS AMA310* perform a spectrum analysis (use softkeys RANGE and ANALYZ) over the UHF TV frequency band. Can you receive all DVB-T2 channels, which should be available in Rosenheim? Hint: Use the →/← keys to find out the frequencies of the spectral peaks.

⇒ Perform a narrowband spectral analysis (softkeys SPAN→>>>→30MHZ) of channel 56. How can you recognize that this is an OFDM transmission? Can you see any fading effects?

b) **Receiving TV services** on frequency channel 56

⇒ How many services are contained in the *MPEG-2* TS of that channel? Hint: To receive (i.e. demodulate/decode) a signal, press ENTER. Place the antenna to get a signal quality with MER ≥ 15 dB.

⇒ Find out the major modulation (i.e. COFDM) parameters of the signal on **channel 56**: Hint: Display the parameters by pressing the soft keys >>>→PARAMETERS (2 pages → !).

Bandwidth:	**Transmission Mode:** ☐ Extended Carrier Mode?	**Mapping** ☐ Rotated?	**CP:**	**FEC-Code Rate:**
L1 FEC Type	**L1 Post Signalling Mapping**	**PP (Pilot Pattern):**	**# used PLPs:**	**DVB-T2 Version:**

c) **Analysis of the MPEG-2 Transport Stream** using the MPEG-2 Analyser software *StreamXpert*

⇒ Make sure that the *ASI* output of the *KWS AMA310* is connected to the input of the PCI card.

⇒ During the reception of a program, start *StreamXpert*:

- On the left side, you can display the TS contents, ordered by the logical tables and services (use the tab PID).
- On the right side, you can display the TS contents, ordered by their PIDs (use the tab TS).

⇒ In which range are the data rates of video and audio PES' (*Packetized Elementary Streams*)?

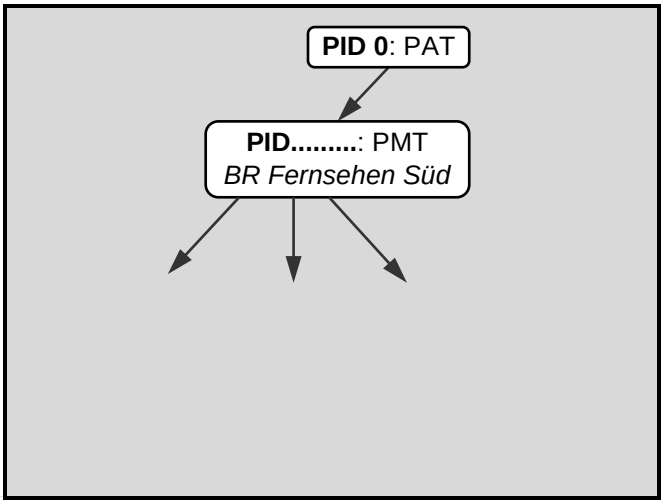
Data rate Video PES:	**Data rate Audio PES:**

⇒ Are the data rates of the services constant?

⇒ How can the decoder find the PIDs of all available audio & video streams?

Draw the sub tree for the program "BR Fernsehen Süd", containing all *Services* and *Tables* for Audio, Video, & Teletext, together with their PID.

Start with the PAT (*Program Association Table*) @ **PID 0**.



⇒ Are there any services which come with multiple audio streams? Why?

⇒ [Fis]

⇒ Display the PCR (*Program Clock Reference*) Analyser (PCR tab) for any of the services: In which intervals are the PCR time stamps sent? What is the maximum Jitter of the PCR?

Interval:	**Jitter:**

d) For the next experiments, **record 30 s of the TS** into a file: In *StreamXpert*, select a filename in your directory **`DVB-T2`** on the **`Desktop`** by Recording→  , and then  .

Experiment 2 : *COFDM*-Parameters of the *DVB-T(1)* Physical Layer (20 min)

Now we will use the *DTA-115* as a signal generator, transmitting the pre-recorded Transport Stream.

a) **Problem analysis in the constellation diagram**

⇒ Use the software *StreamXpress* to re-transmit the recorded '.ts' file by DVB-T**(1)** on channel 54 or channel 55 (f_c = 738 MHz or 746 MHz), depending on the work place.
Make sure that the transmission is done with the DVB-T(1) parameters prepared on page 2-4!

⇒ Set the RF output to max. power: Settings→RF Output Control→Main Output Level = –3dBm.

⇒ Receive the signal with the *AMA310*, and display the constellation diagram (>>>→CONST).
Hint: On the *AMA310*, you have to manually switch the MODULATION to DVB-T.

⇒ Which problem can be observed in the constellation diagram? Discuss possible solutions!

⇒ Before you continue, solve the problem with the help of the lab advisor!

b) **Constellation Diagram of Single Subcarriers**

⇒ Observe the SINGLE CARriers with the following indexes. What do they represent?

Index **3408**:	Index **3**:

⇒ Estimate the number of symbols that is superimposed in the single carrier constellation view.

c) **Switch on the two TV sets**, and check if they can receive the signal as well.

Note: The DVB-T(1) channels are pre-set to the storage locations 9…16 on both TV-sets!

⇒ Adjust the values of the parameters *Mapping*, *CR*, *CP*, and *Mode* in order to get the maximum net data rate ('*Rate Out'* in *StreamXpress*) of the DVB-T**(1)** physical layer?
Compare the observed value to your results of the preparation question on the top of page 2-5!

⇒ Do the two TV sets (*August* and *Reflexion*) support this high data rate?

⇒ How can the net data rate ('*Rate Out*') differ from the '*Rate TS*'?
What happens internally in *StreamXpress* when you de-activate the 'RMX' checkbox?

⇒ Activate 'RMX' again, and reset the parameters to obtain *Rate Out* = 22.394 Mbit/s!

Experiment 3 : Impact of the Terrestrial Propagation Channel (60 min)

Let's simulate a few channel impairments and see how the receivers cope with that.

a) **Fading** (*Inter Symbol Interference*)

⇒ Press fading… → Multiple Transmission Paths Simulation → Enable, and define two paths with 3 dB attenuation each, the 2nd with a CONSTANT DELAY of $\Delta\tau = 1\ \mu s$.

⇒ Does the spectrum (ANALYZ) look like in your preparation?

⇒ In *StreamXpress*, set one of the paths to CONSTANT_DOPPLER @ a speed of 0.1 km/h.
Explain your observations in the spectrum!

b) **Time Dispersion** in a simulated SFN (*Single Frequency Networks*)

⇒ How much *Delay Spread* $\Delta\tau_{max}$ (two paths with 3 dB attenuation) can the two TV sets handle?

$\Delta\tau_{max}$

⇒ On the *AMA310*, receive TV again, and press >>> → IMPULSRES to observe the impulse response of the channel. Which path length difference Δd is simulated here?

Δd =

⇒ Increase the delay of the second path to $\Delta\tau = 250\ \mu s$ and its attenuation to 26 dB.
Of which type is the noise that you see in the constellation diagram: Additive or multiplicative?
Discuss possible reasons for that effect with the lab advisor, before you continue!

c) **AWGN (*Additive White Gaussian Noise*) for DVB-T(1)**

⇒ Now, switch off Multiple Transmission Paths Simulation, and Enable AWGN. Observe:

- The constellation diagram (@ big display of *AMA310*)
- The adjusted *CNR/SNR* value
- The subjective picture and sound quality on the two TV sets (*August* and *Reflexion*)

⇒ Up to which noise level are the receivers able to operate satisfactorily? At which noise ratios are impairments visible / is a reception impossible (= *Picture Loss*)?

Receiver	Threshold for first impairments:	Threshold for *Picture Loss*
August	*SNR =*	*SNR =*
Reflexion	*SNR =*	*SNR =*

⇒ Which of the two receivers performs better?

d) **Reception over DVB-T2 with Channel Impairments**

⇒ In *StreamXpress*, disable the channel simulator and switch the Modulation to DVB-T2! Use the same frequency channels 54/55 as for the previous exercises.

Hint: Set the modulation parameters observed in b) on page 2-6.so that "Rate Out" matches "Rate TS" approximately. All adjustments are explained in the *StreamXpress* manual, which can be found in your directory **`DVB-T2`** on the **`Desktop`**.

⇒ Also switch the three receivers *AMA310*, *Reflexion*, and *August* to DVB-T2.
Note: Choose Modulation→DVB-T2 on the *AMA310*. Choose program storage 1…8 on the *Reflexion*. Ask the advisor to change modulation on *August*. Please do not use the station scan!

⇒ Now activate AWGN again and observe the constellation diagram and the picture quality, and the *MER* value (@ small right LCD of *AMA310*), depending on the adjusted *SNR*/*CNR*:

Receiver	Threshold for first impairments:		Threshold for *Picture Loss*	
August	*SNR =*	*MER =*	*SNR =*	*MER =*
Reflexion	*SNR =*	*MER =*	*SNR =*	*MER =*

⇒ Which of the two receivers performs better?

⇒ Activate "Rotated Constellation" now and observe the Fall-off-the-Cliff *SNR* for both receivers. Also try to do this experiment with the presence of spectral fading (two paths of equal attenuation with CONSTANT_DELAY $\Delta\tau = 1\ \mu s$). Does the rotation lead to any SNR gain?

$SNR_{\text{with spectral fading/without Rotation}}$ =

$SNR_{\text{with spectral fading/with Rotation}}$ =

⇒ **Gain =**

Experiment 4 Measurements in a real SFN of *Wendelstein* & *Olympiaturm* (20 min)

a) **Cleanup:** Please remove your directory **`DVB-T2`** from the **`Desktop`** of your computer now.

b) Ask the advisor to prepare the *AMA310* and the directional *Yagi* antenna for a measurement from the 2nd floor of the west side of the R-building.

⇒ Measure the impulse response on channel 48. Enter the *Delay Spread* and the *Path Length Difference* between *Wendelstein* and *Olympiaturm*:

Delay Spread $\Delta\tau$**:**	***Path Length Difference*** Δd**:**

⇒ Compare the results to your preparation on page 2-5. Are there differences?

Exercise 3: *LTE-RF* (*RF Measurements*)

The PHY of the 4G mobile communication standard LTE (*Long Term Evolution*) employs the transmission schemes OFDMA and MIMO, in order to obtain high data rates, low latency, and high spectral efficiency in mobile environments. We will observe these principles by doing RF measurements by operating an LTE modem against the radio communication tester ROHDE&SCHWARZ *CMW500*.

References: [Gho] (eBook!)**,** [Saut] (in German), [Ses], [EtsiSpec] (36 series)

3.1 Equipment

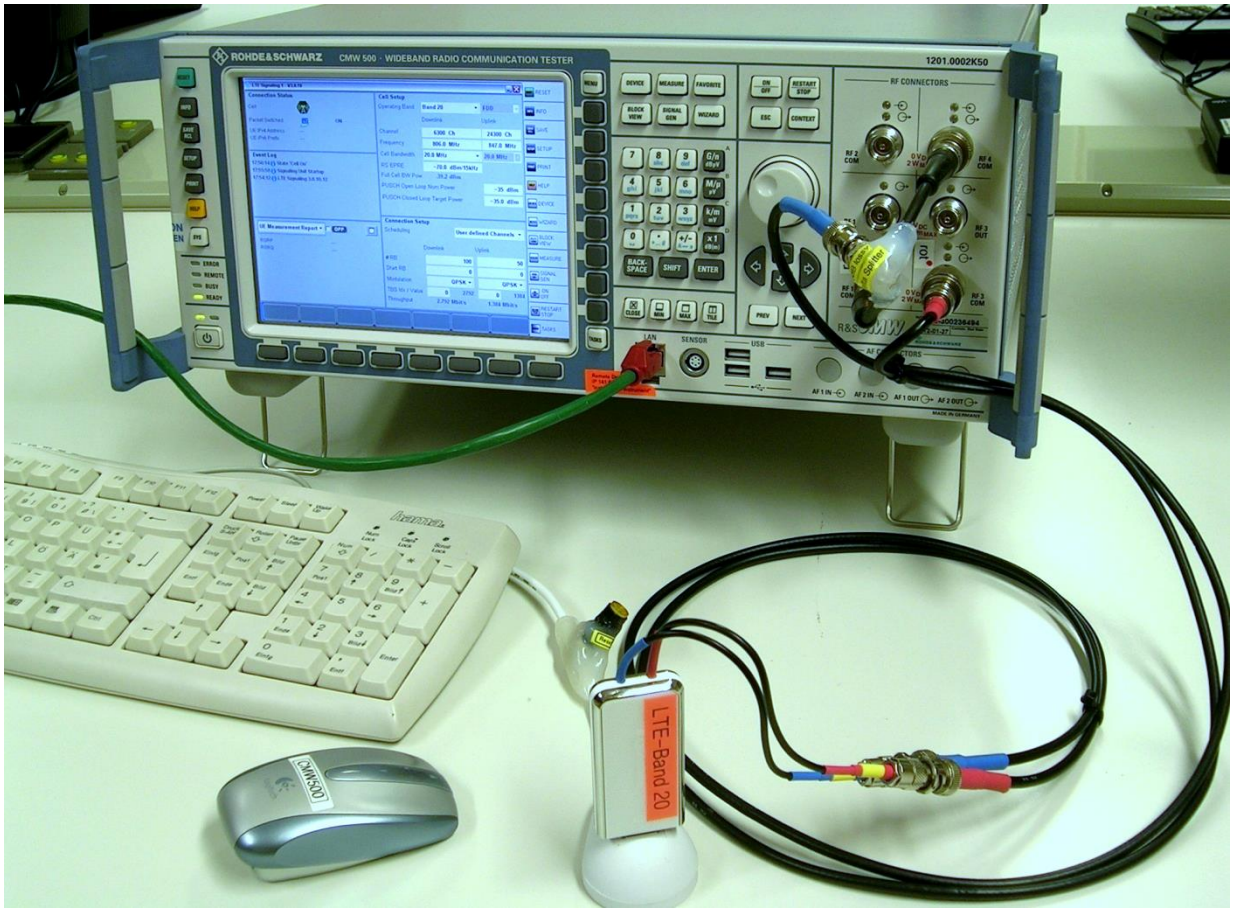


Figure 3-1: *LTE-RF* experimental setup

- Communication Tester ROHDE&SCHWARZ *CMW500* (User Manuals for the *CMW500* and for the options being employed in this exercise can be found in the University's *Online Community* **.**
- LTE modem SAMSUNG *GT-B3740*, with connector cables mounted to the two antenna ports.
- Please bring...
  - ⇒ **The LTE lecture handout** and
  - ⇒ **a pocket calculator!**

3.2 Background

3.2.1 The Wideband Radio Communication Tester R&S *CMW500*

The R&S CMW 500 is a multi-mode protocol tester that simulates a mobile communication network. It is used to test mobile devices at all stages of development, e.g. in T&D, at network operators and in certification tests. The instrument is capable of generating one cell with MIMO (Multiple In Multiple Out) technology or two independent cells.

The figure below shows the user interface and the connectors on the front panel of the tester:

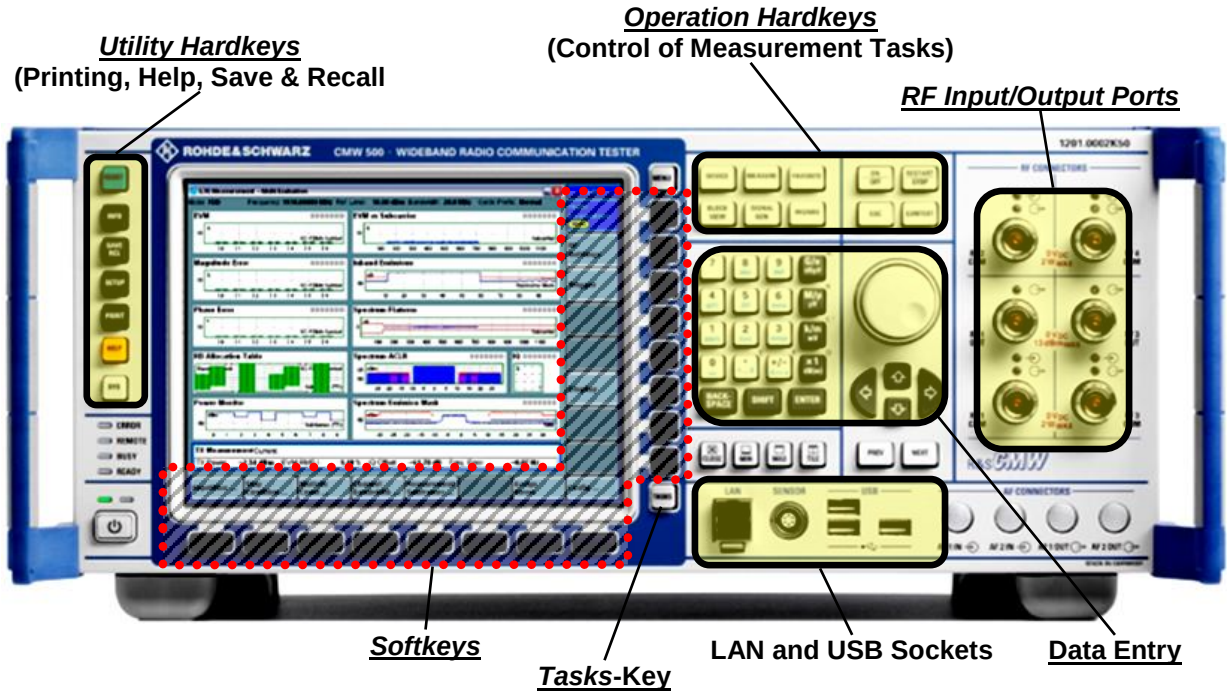


Figure 3-2: Front panel of the R&S CMW500

Some details of the CMW500's operation philosophy:

- Key TASKS (on the bottom left of the display):
  The tester offers various functions for signal generation and for measurements and signal analysis, called *Tasks*. Between those *Tasks* that are currently loaded in the memory, the user can select the one which will be displayed with its GUI on the screen by pressing the TASKS key.
- *Softkeys* (on the bottom and right of the screen):
  When the GUI of a *Task* is displayed, it offers additional configuration menus by *Softkeys*, the label of which is shown beside or above the key. Generally, with the softkeys right of the screen, the user determines the available menus offered by the softkeys on the bottom of the screen.
- Key ESC (within the *Operation Hardkeys*):
  Several *Hard-* and *Softkeys* open additional windows positioned on top of the basic task window. Use the ESC key to close those windows.
- Key PRINT (within the *Utility Hardkeys*):
  ⇒ Allows to copy a screenshot file to the hard-drive of the tester or to a connected USB stick.
- Key HELP (within the *Utility Hardkeys*):
  ⇒ Opens a context help window for the active task or function of the tester.

3.2.2 Functions of the R&S *CMW500* used in the Lab Experiments

The following block diagram displays the measurement functions and its internal and external connections that are used during the experiments of this lab exercise:

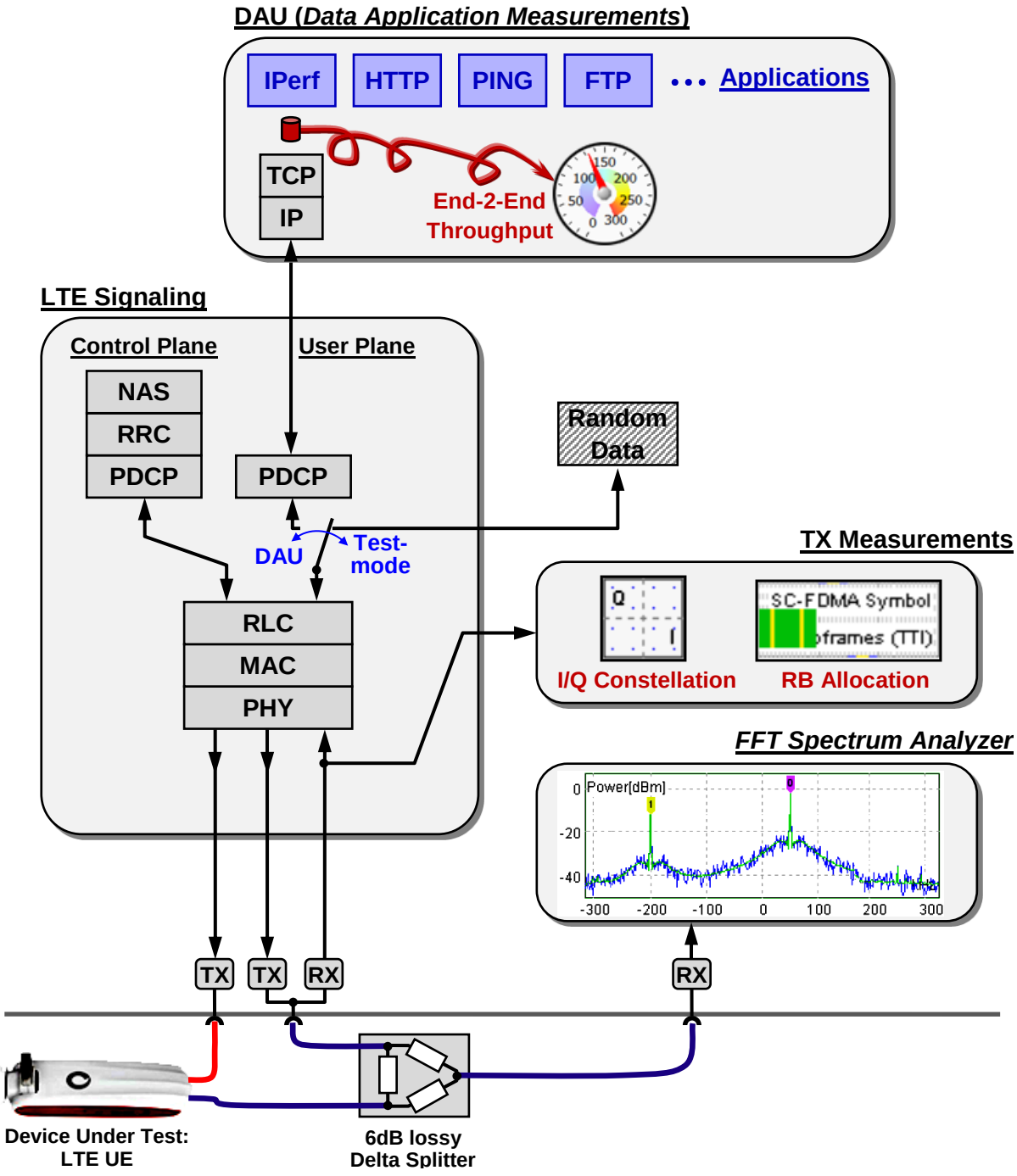


Figure 3-3: Simplified Block Diagram of the R&S CMW500

- The central task is *LTE Signaling*, which simulates the *eNodeB* and the EPC with NAS functions.
- Throughput measurements can be done either on the RLC/MAC layer (the *BLER* task) or on the application layer by using the DAU (*Data Application Unit*). The DAU is a completely separate PC embedded within the CMW500, running TCP/IP and several client and server applications.
- The signals can be analysed by the tasks *TX Measurements* and an *FFT Spectrum Analyser*.

3.3 Preparation Problems (to be answered before doing the Lab!)

a) Setup and Operation of the CMW500:

⇒ Which *Hardkey* should be preferably used to close a window?

⇒ What for is the DAU (*Data Application Unit*) in the CMW500?

⇒ Which signal direction (UL or DL) can be observed by the task *TX Measurement*?

⇒ What for is the splitter needed in Figure 3-3?

⇒ In which position do you have to put the switch *DAU↔Testmode* in Figure 3-3, in order to transmit IP Packets?

⇒ What is the purpose of the **open source tool *IPerf***? Find out from the Internet!

b) The Downlink Resource Grid

Visit the Web page [Niv]→Store→LTE→RadioResourceGrid and observe the downlink resource grid for the bandwidth B = 5 MHz and a *Normal Cyclic Prefix*:

⇒ What is the duration of an OFDM-Symbol in the downlink?

⇒ How are the RBs and REs numbered in time and frequency domain, respectively?

⇒ Where is the PCFICH located in time and frequency domain?

Time Domain:	**Frequency Domain:**
Slot no.:	***Subcarriers:***
OFDM Symbol:	

⇒ How often are the PBCH and the SCHs transmitted per 10 ms *Radio Frame*?

PBCH:	**SCH:**

⇒ What is the bandwidth of the PBCH und the SCHs?

PBCH Bandwidth:	**SCH Bandwidth:**

⇒ How do the DMRS (*Demod. Reference Signals*) differ for the antenna ports 0 and 1?

⇒ On which physical downlink channel(s) is *System Information* (log. channel BCCH) sent?

Channels:

c) The PUCCH (*Physical Uplink Control Channel*)

⇒ Under which circumstances does the UE send a PUCCH in the uplink of a certain subframe, instead of a PUSCH (*Physical Uplink Shared Channel*)?

d) UE Performance

⇒ Which *UE Categories* support MIMO operation?

UE Categories:

⇒ Download [EtsiSpec] 36.213 "Physical Layer Procedures" (see page ii for detailed instructions!). Inside this document, look at Table 7.1.7.2.1-1:

⇒ Which TBS (*Transport Block Size*) is needed in a 2×2 MIMO system, to obtain a data rate of 102.048 Mbit/s system with $N = 100$ RBs in the frequency domain?

⇒ Which *TBS Index* is necessary for that?

e) FFT Spectral Analysis:

⇒ List two advantages one disadvantage of performing spectral analysis by using the FFT (*Fast Fourier Transform*), compared to a sweep-based analysis.

Advantage: **Disadvantage:**

⇒ Assume the FFT analysis of a complex base band signal of $T = 205\,\mu s$ duration, being sampled @ $f_s = 10$ MHz. What is the FFT-Dimension N, the frequency resolution Δf, the total *SPAN*?

FFT-Dimension N =	**Frequency Resolution Δf =**	***SPAN* =**

⇒ By using a non-rectangular window function, the effective window duration is reduced, and the resolution bandwidth is increased. Explain why!

Reason:

3.4 Practical Part

Experiment 1 : Getting Started with the *R&S CMW500* (30 min)

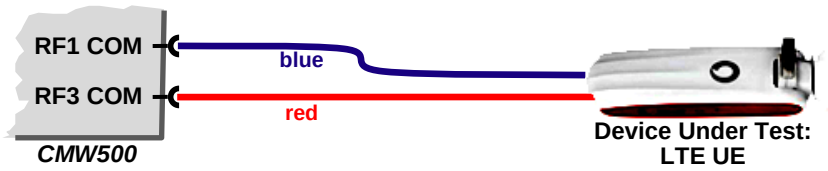


Figure 3-4: Connection Plan for Experiment 1

a) Configuration of the LTE "Call Box"

⇒ Start *Remote Desktop* on the PC, in order to control the *R&S CMW500*.

⇒ Press SAVE RCL and Recall the configuration `WiCS_Lab\LteRf-Experiments1to3`.

⇒ Switch to TASKS → LTE Signalling.

⇒ Which channel is used? Does the signal with its bandwidth fit into *Frequency Band* 20?

Channel Frequency:	**Channel Bandwidth:**

⇒ What is the total transmit power? What is the formula to calculate this Full Cell BW Pow. from the power per Resource Element RS EPRE?

⇒ Configure the network identity by Config... → Network → Identity.

⇒ Activate the *eNodeB* by pressing ON OFF.

b) Connection Setup of the UE

⇒ Connect the UE (using the USB cable) to the PC. If the *Connection Manager* shows that the network has been found, press the big central *Connect* Key.

⇒ TASKS → LTE Signaling will show *Attached*.

⇒ Look at the UE Measurement Report (if necessary, it has to be switched *On* before): What is the current RSRP in the Downlink? Does it differ from the transmit power RS EPRE?

RSRP =

⇒ Check the drop down menu UEMeasurementReport/UECapabilities: Write down the Category of the UE and the supported operating bands. Which maximum data rate should be possible in the Downlink/Uplink? Have a look into the lecture handout, subchapter "MIMO Operation" What is the UE's IMSI (menu item UE Info)?

UE Category:	**Data Rates (DL/UL):**	**IMSI:**

⇒ The UE has two antenna ports for receiving, but it can only transmit on one of these ports in the Uplink. In BLOCK VIEW → Show Active Routing click on "LTE SIG 1", in order to find out which of the two cables (red or blue) is the transmit port:

⇒ Before starting the next Experiment 2, disable the UE by unplugging its USB connector!

Experiment 2 : Analysis of Downlink Signals (70 min)

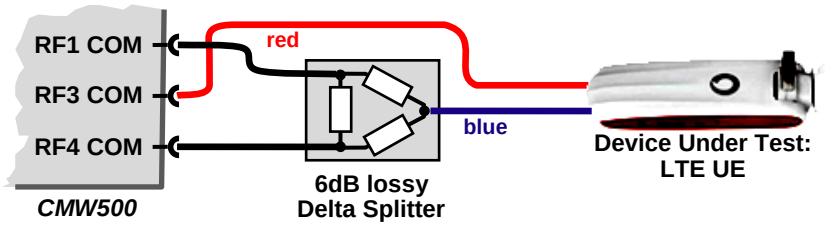


Figure 3-5: Connection Plan for Experiment 2 and Experiment 3

a) FFT Spectrum Analyser Setup

⇒ Select TASKS → GPRF Measure → FFT Spectrum, and switch on the analyser.

⇒ In BLOCK VIEW → Show Active Routing, find out the RF routing of the FFT Spectrum analyser input.

⇒ Close BLOCK VIEW and adjust FFT → Frequency Span..., in order to see the entire signal bandwidth.

Frequency Span =

⇒ Adjust the FFT Length... in order to see a window of the signal with approx. 400 µs duration.

FFT Length =

⇒ What are the frequency steps Δf between adjacent spectral FFT samples?
Why is the displayed RBW of the analyser significantly bigger?

Δf =	***RBW =*** **Because... :**

⇒ Adjust Trigger → Trigger Offset... to 40 µs. Which part of the resource grid do you recognize?

b) Identification of Physical Downlink Channels

⇒ Leave Trigger Offset... = 40 µs and look at the time and spectral domain graphs:

- What is the duration of one OFDM symbol (read out in the time domain!)?
- What is the spacing of the reference symbols in the spectral domain (use Marker → Ref Marker... and another relative Marker 1...)?
- Which two physical channels can be identified in the spectrum?
- How many OFDM symbols (i.e. time) does the PDCCH span: 1, 2 or 3?

⇒ Use the Markers to find out the bandwidth of the physical channels P-SCH, S-SCH, and PBCH:

Phys. Channel	Bandwidth
P-SCH	
S-SCH	
PBCH	

⇒ In the observed scenario, also a PDSCH allocation can be found, which carries *System Information*. Browse through the 10 ms *Radio Frame* in order to search for this PDSCH allocation:

- Which subframe(s) is/are used for transmitting the *System Information* and the associated scheduling DCI (*Downlink Control Information*)?

Sub-frame(s)

- Which RBs in these subframes are used?

⇒ Discuss all results with the advisor!

c) Allocation of User Data

⇒ Enable the UE by connecting its USB connector and **attach** (i.e. *connect*) **it to the *CMW500***.

⇒ In TASKS→LTE Signaling→Conn. Setup→Scheduling→Edit All→DL Stream 1, allocate 1.8 MHz bandwidth in subframe no. 3. How many RBs are that?

Note: No automatic link adaptation is performed by the CMW500!

⇒ Verify the effect in FFT Spectrum and show it to the advisor, before you continue.

d) Additional Reference Symbols for MIMO Operation

⇒ In TASKS→LTE Signaling, activate MIMO Operation in the Downlink: Config...→Scenario: *Two RF Out Ports*. Note: You have to disconnect the UE before!

⇒ Observe the modified BLOCK VIEW→Show Active Routing of "LTE SIG 1" now!

⇒ Connect the input of the FFT Analyser to the output of the second antenna port and observe the additional *Demodulation Reference Symbols*. Use Markers, to find out the frequency shift!

Shift between antenna ports 0 and 1 =

Experiment 3 : Analysis of Uplink Signals (40 min)

a) Identification of Physical Uplink Channels

⇒ In TASKS→LTE Signaling, switch off the second port: Config...→Scenario→*Standard Cell*.

⇒ Enable the UE and attach it.

⇒ Select the TASKS→LTE TX Meas.→Multi Evaluation and activate it (press RESTART STOP).
Twelve different measurements are displayed simultaneously.

⇒ Which physical channel can be seen in the RB Allocation Table view?
What do the yellow regions in the table signify?

⇒ Modify the Uplink Scheduling in TASKS→LTE Signaling→Connection Setup→Scheduling →Edit All→UL, in order to force the occurrence of a PUCCH in UL subframe no. 8!

⇒ In TASKS→LTE TX Meas.→Multi Evaluation, verify the effect in the RB Allocation Table view: Which RBs are occupied by the PUCCH? What size does the PUCCH region have?

⇒ Now observe the UL with TASKS→FFT Spectrum by adjusting RF Settings→Frequency:
Verify the frequency hopping within one subframe!

⇒ Where can you see the leaking DC Carrier?

⇒ In TASKS→LTE Signaling→Scheduling→Edit All→DL Stream 1, select DL subframe no. 4 and set the no. of RBs to '0'. Observe the spectrum of the PUCCH? What happened?

⇒ Discuss the results with the advisor!

b) Constellation Diagrams

⇒ In TASKS→LTE TX Meas.→Multi Evaluation select the view IQ Constellation, and watch the PUSCH. Which Modulation Scheme is used?

⇒ In the IQ Constellation view, select Multi Evaluation→Measurement Subframes and adjust Measurement Subframe = *1*. Which modulation scheme do you see here?

⇒ Discuss the results with the advisor!

Experiment 4 : MAC Throughput Measurement (15 min)

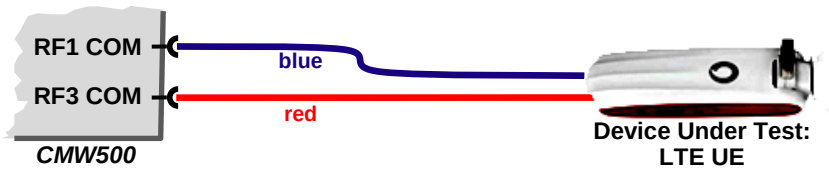


Figure 3-6: Connection Plan for Experiment 4 and Experiment 5

a) Maximum possible SISO throughput

⇒ Press CLOSE multiple times, to close all windows and tasks.

Then recall the configuration `WiCS_Lab\LteRf-Experiments4to5` by pressing SAVE RCL.

⇒ How many RBs are used? Does the entire signal bandwidth fit into *Frequency Band* 20?

# of RBs:	**Signal Bandwidth:**

⇒ Enable the UE and attach it.

⇒ In TASKS→LTE Signaling, vary Modulation and TBS Idx, in order to find the MCSs supported by the UE, which result in the maximum throughput in the Downlink.

Max. Throughput DL =

b) Maximum possible MIMO throughput

⇒ Detach the UE and activate MIMO Operation again: Config...→Scenario: *1 Cell – 2 RF Out*.

⇒ Vary #RBs, and TBS Idx, in order to find the MCSs supported by the UE, which result in the maximum throughput of 102.048 Mbit/s in the Downlink, and 51.024 Mbit/s in the Uplink.

Downlink	***#RBs*:**	?	?	**Uplink**	***#RBs*:**	?	?
	***TBS Idx*:**	**23**	**21**		***TBS Idx*:**	**22**	**21**

Experiment 5 : IP-Based *End-to-End* Data Transmission (25 min)

a) DAU (*Data Application Unit*) Setup

⇒ Now, disable the UE and switch off the *eNodeB*, before configuring TASKS→LTE Signaling to exchange data *End-to-End*: Config...→Connection→Connection Type: *Data Application*.

⇒ Attach the UE and compare its *IP*-Address, shown on the CMW500 in LTE Signaling, and shown on the PC by running the command tool `ipconfig`.

UE IPv4 Address	**... on *CMW500*:**	**...on the PC:**

⇒ Select the task Data Meas, and click on Configure Services→Overview, in order to note down the IP address of the DAU.

DAU IPv4 Address	

⇒ In TASKS→Data Meas→Configure Services, switch on HTTP. Close the window by ESC.

⇒ Verify the setup by accessing the DAU from a Web-Browser on the PC!

b) PING test

⇒ Check, if the DAU answers to a PING command from the PC. What is the round-trip time?

Latency (*Round Trip Time*) =

⇒ Now check, if the PC answers to PING commands from the DAU:
In TASKS→Data Meas→Ping, run a test by RESTART STOP. What is the round-trip time?

Latency (*Round Trip Time*) =

c) Throughput Measurements using the tool *IPerf*

⇒ On the PC, run the program *IPerf* and start a TCP server, measuring in the Downlink.

⇒ On the CMW500, in TASKS→Data Meas→IPerf, configure a downlink test (Client→Use) of a Test Duration = 20 s. Start the test. What is the measured throughput?

Throughput in the Downlink:

⇒ Can the throughput be increased, if you run 2 TCP connections in parallel (Parallel conn.)?

Max. possible Throughput in the Downlink:

⇒ On the PC, run *Wireshark* and check, how many parallel TCP connections are really used:

No. of parallel connections:

Exercise 4: *LTE-Proto* (*Protocol Analysis*)

In this lab exercise, we will analyse some basic protocol scenarios of the 4G mobile communication standard LTE (*Long Term Evolution*). The scenarios have been pre-recorded by a setup similar to the one in Exercise 3: *LTE-RF*, and will be analysed offline by a special message Analyser software. You will learn the differences between *Service Primitives* and *Protocol Data Units*, and how to set appropriate filters in the message Analyser, in order to observe and understand the cooperation of all protocol layers and entities in an LTE system.

References: [Gho] (eBook!)**,** [Saut] (in German), [Ses]

4.1 Equipment

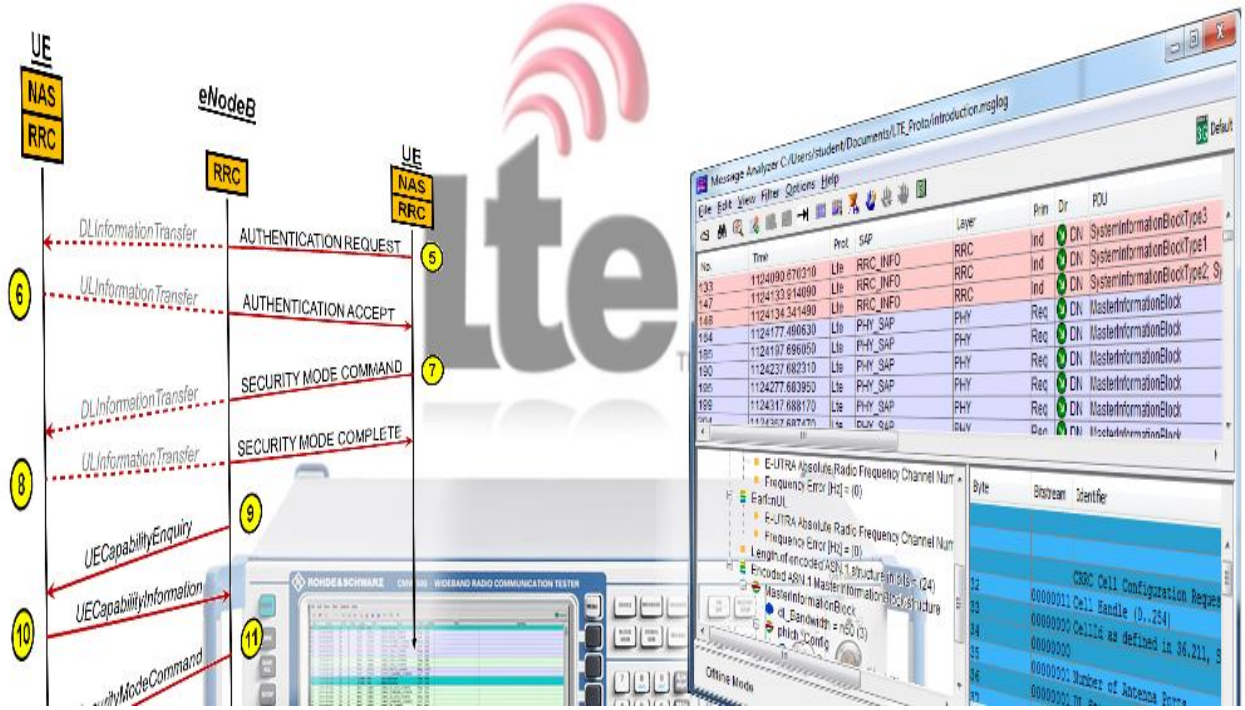


Figure 4-1: The ROHDE&SCHWARZ *Message Analyser* for the *CMW500*

- PC equipped with...
  - ⇒ ROHDE&SCHWARZ *Message Analyser*
  - ⇒ License Manager and Hardlock-Dongle
- Please bring...
  - ⇒ **The LTE lecture handout** and
  - ⇒ **a USB storage (e.g. memory stick)** for bringing the downloaded LTE specifications (see 4.3e) for taking message sequences home.

4.2 Background

4.2.1 SPs (*Service Primitives*) and PDUs (*Protocol Data Units*)

Within a communication system, the actual data flow happens vertically between the different layers of the OSI reference model for communication. The interfaces between the layers are called SAPs (*Service Access Points*), because protocol instances above the interface consume services, whilst the instance below the interface provides that services.

The messages exchanged at the SAPs are called *Service Primitives*. Four general types of primitives are distinguished during the lifetime of an interaction: *Request*, *Indication*, *Response*, and *Confirmation* – depending on their direction and on the side (active or passive):

***Service Primitives* can be used for two purposes:**

(5) Configuration of the lower layer instance

(6) Transport Container for PDUs (*Protocol Data Units*), being exchanged with the peer instance:

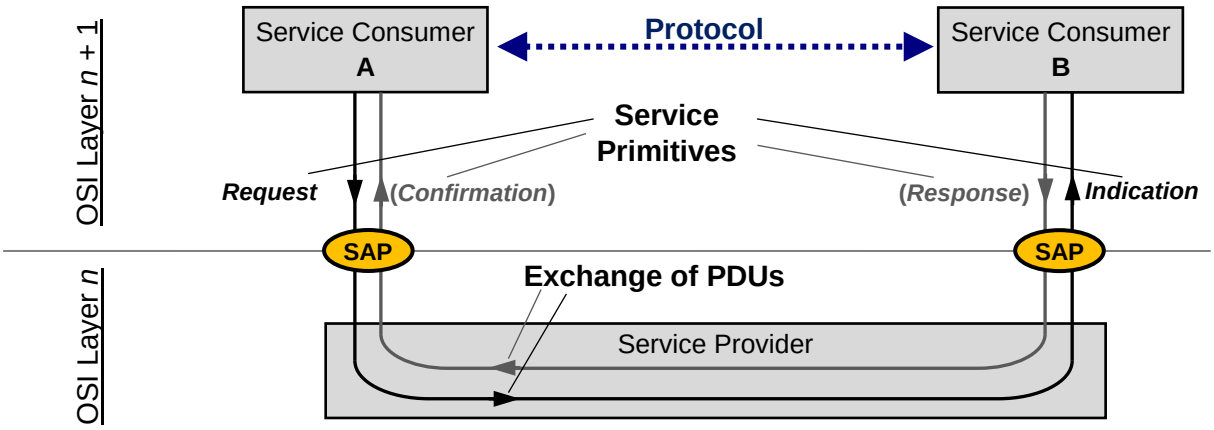


Figure 4-2: The four types of *Service Primitives*

Note that the *Message Analyser* used in this lab exercise distinguishes only between *Requests* (down) and *Indications* (up), independently of the side (active or passive).

4.2.2 LTE Security

Independence of NAS and AS Security

On the LTE air interface, both the AS (*Access Stratum*) and the NAS (*Non-Access Stratum*) are protected by independent security algorithms:

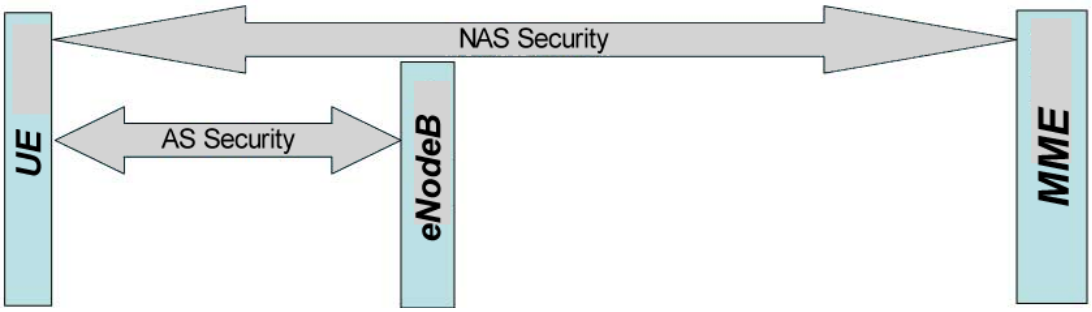


Figure 4-3: Independence of NAS and AS Security (modified www.3glteinfo.com/category/3gpp/lte/lte-security)

- **NAS security**:
  NAS signalling PDUs between the UE and the MME are *Integrity Protected* and *Ciphered* with an extra NAS security header.
- **AS security**:
  ⇒ RRC messages between the UE and *eNodeB* are *Integrity Protected* and *Ciphered*.
  ⇒ U-Plane (User) data is only *Ciphered*, but not *Integrity Protected*.

The independency of AS and NAS security guarantees a contiguous protection of NAS signalling messages, even during handovers from one RAT (*Radio Access Technology*) to another.

Standardized Algorithms for *Ciphering* and *Integrity Protection*

In the LTE Release 8, both for AS and for NAS security, three different versions of *Ciphering* and *Integrity Protection* have been defined, respectively:

- **Integrity:** "0000" – EIA0 (*Null Integrity Protection*), i.e. *Integrity Protection* is switched off
  "0001" – 128-EIA1 SNOW 3G
  "0010" – 128-EIA2 AES

  For NAS messages, the use of the *Null Integrity Algorithm* is only allowed in exceptional cases. Our exercises are performed in a scenario without *Integrity Protection*, which is tweaked by a special non-standardized NAS-Algorithm "0111" that is accepted by the test USIM card but actually does nothing.
- **Ciphering:** "0000" – EEA0 (*Null Ciphering Algorithm*), i.e. *Ciphering* is switched off
  "0001" – 128-EEA1 SNOW 3G based algorithm
  "0010" – 128-EEA2 AES based algorithm

4.2.3 LTE Message Recording using the ROHDE&SCHWARZ CMW500

In order to analyse communication protocols, the messages can be captured in two different manners:

(1) Observing the traffic directly on the medium:

This principle is applied, for instance, when the traffic on an Ethernet interface shall be captured and analysed by a message analyser like *Wireshark*:

⇒ Only PDUs (*Protocol Data Units*) are captured, no *Service Primitives*.

⇒ All traffic going over the medium is captured without any exception.

(2) Observing the traffic at selected interfaces within one of the communication systems:

As explained below, in this lab exercise you will analyse messages that have been observed at various SAPs, so-called PCOs (*Points of Control and Observation*) within the CMW500:

⇒ *Service Primitives* are captured, some of them containing PDUs as well.

⇒ Internal configuration messages (which do not occur in form of PDUs) will also be captured.

⇒ Only traffic going through these special SAPs in this special device is captured.

The following block diagram illustrates the position of the observed SAPs. For this exercise, only the *Control Plane* has been observed, i.e. only Signaling and Configuration data can be captured.

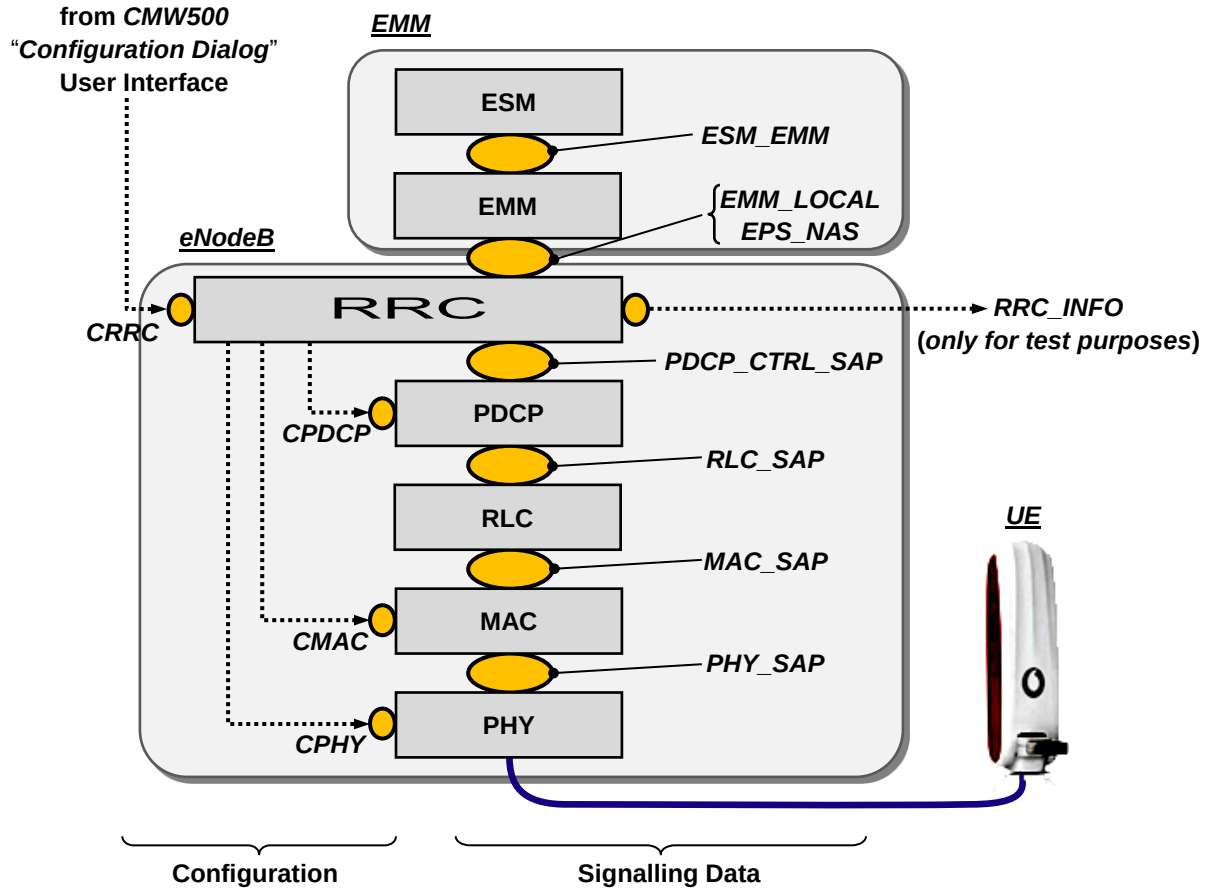


Figure 4-4: Position of the relevant observed SAPs in the *Control Plane* of the CMW500

Note: In this exercise, the data capture has already done before, i.e. you will analyse pre-recorded data.

4.2.4 Introduction to the ROHDE&SCHWARZ *Message Analyser*

The captured data, so-called *message logs*, will be analysed by a universal ROHDE&SCHWARZ *Message Analyser* software, which has been equipped with all necessary MDDB (*Message Definition Data Bases*) for the LTE protocols. The user interface is similar to *Wireshark*, there are three main windows:

(1) Sequence Representation View:

Shows all recorded messages in chronological order. Each line represents a message, the columns show message number, time stamp, SAP, and other details, which can be defined by the user.

(2) Structure View:

Shows the decoded contents of the *Service Primitive* that has been selected in the window above, in an "Explorer" tree-like hierarchical manner.

Unlike *Wireshark*, only one protocol layer is evaluated for every message. For understanding the vertical message flow through the different protocol layers, different *Service Primitives* (i.e. one for each SAP) have to be observed in the protocol stack.

(3) Details View:

Shows the same contents of the *Service Primitive* as the Structure View, but in sequential manner.

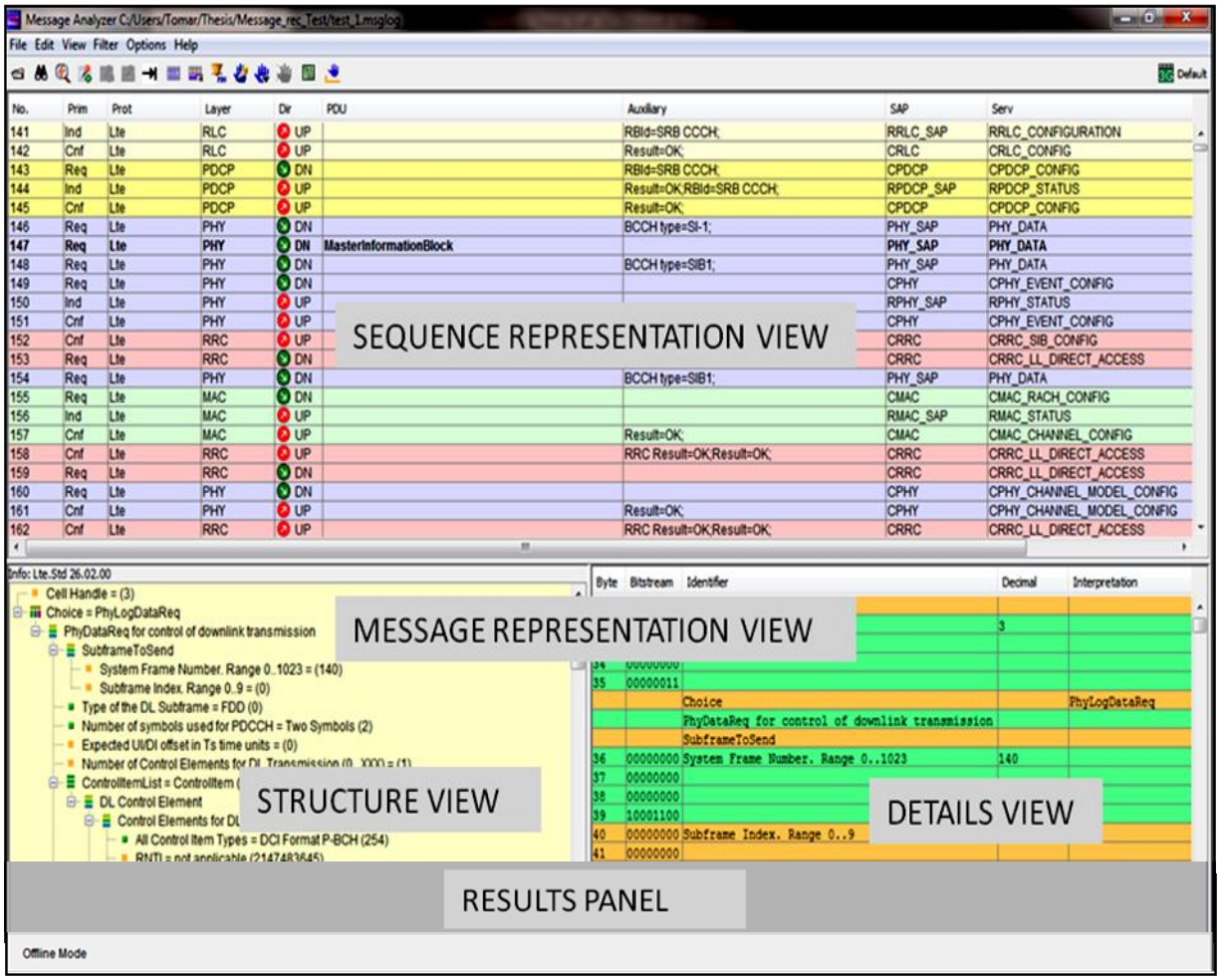


Figure 4-5: GUI of the Message Analyser

The Message Analyser provides a powerful tool to define display filters, in order to reduce the displayed messages. Furthermore it supports plain-text search, a tool to set bookmarks, and the parent/children tool to get an overview of messages that belong to the same data flow in the protocol stack.

4.3 Preparation Problems (to be answered before doing the Lab!)

a) *Service Primitives* and PDUs in the protocol stack of the *CMW500*:

⇒ List two differences between protocol analysis by *Wireshark* and by the R&S *Msg. Analyser*:

⇒ Which SAPs of the *CMW500* (Figure 4-4) have to be observed, in order to see...

RRC PDUs:	**MAC PDUs:**

b) System Information

⇒ Which role does the **"*ASN.1*"** play for the 3GPP protocols? Look it up in the Internet!

⇒ Download [EtsiSpec] 36.331 "RRC Protocol" (see page ii for detailed instructions!), and look up the Definition of SIB1 (*SystemInformationBlockType1*) in sect. 6.2:
What does the Information Element '*q-RxLevMin*', being explained in sect. 6.3 exactly mean?

⇒ Which SIBs contain information about neighbour cells?

c) Attachment of a UE

⇒ How many messages have to be exchanged during the *Random Access* procedure?
Which of these messages can carry Layer 3 data?

# of messages for Random Access:	**Messages available for L3 data:**

⇒ Which Layer 3 messages have to be sent by the UE during the *Attachment* process, indicating the change of its states:

		Layer 3 Message?
RRC:	**IDLE→CONNECTED**	
ECM:	**IDLE→CONNECTED**	
EMM:	**DEREGISTERED→REGISTERED**	

d) LTE/SAE Security

⇒ Security includes *Authentication*, *Integrity Check*, and *Ciphering*. Look up these three terms in the Internet and explain them briefly:

Authentication:	**Integrity Check:**	**Ciphering:**

⇒ What is the purpose of the parameters *RAND*, *AUTN*, and *RES*, which are used during the so-called *Challenge Response* Authentication? Look them up in the Internet!

***RAND*:**	***AUTN*:**	***RES*:**

⇒ What is the IMEISV? Look it up in the Internet!

⇒ Which protocol layer(s) is/are responsible for AS security?

e) Specification Files to be brought into the lab exercise

⇒ Download [EtsiSpec] 24.301 "*NAS Protocols*" as well. Store the file on your USB storage (e.g. memory stick), together with the previously downloaded [EtsiSpec] 36.331 "RRC Protocol". Bring the files into the lab, so you have them available during the lab experiments.

4.4 Practical Part

Preparation: Open the directory **`Lab_R1.16`** on the **`Desktop`** of your computer. Here, enter the **`zip`**-archive **`WiCS`**, and copy the directory **`LTE_Proto`** directly to your **`Desktop`**. Open the log file **`SimpleAttach.msglog`**, containing an LTE signaling scenario between the CMW500 and a UE.

Experiment 1 : Getting Started with the *R&S Message Analyser* (50 min)

a) Familiarization with the Views of the ROHDE&SCHWARZ *Message Analyser*

In the *Sequence Representation View*, scroll down and compare the column "RFN" (*Radio Frame Number*) with the column "Time": What is the unit of the timestamps in column "Time"?

⇒ Search ( ) for the message that contains the PDU "*EMM Attach Accept"*:
Which message no. and which Service Primitive type (*Request*, *Indication*,...) is that?

Message No.:	**Primitive Type:**	**Direction (UL/DL):**

b) Service Primitives and PDUs

⇒ In [EtsiSpec] 24.301 "*NAS Protocols*" and look up the definition of the EPS message "***Attach Accept***" in sect. 8.2.1.1. Each of the NAS (*Non-Access Stratum*) PDUs starts with a '*Protocol Discriminator*', which identifies this message as an EMM PDU.

⇒ In the *Details View* of the *Message Analyser*, spot the Identifier '*Protocol Discriminator'*.

⇒ How many bytes are used for the PDU itself, how many for its *Service Primitive* header?

PDU:	**Service Primitive:**

⇒ Look up in the [EtsiSpec] 24.301, which two values are possible for the mandatory IE (*Information Element*) '***EPS attach result***'? Which value do you observe in the message log?

Possible values:	**Actual value in the message log:**

⇒ Inside the "*EMM Attach Accept*" PDU, you can find another mandatory IE called '*ESM Message Container*". What is its exact purpose? Which PDU does it contain in the message log?

Purpose:	**Contents:**

⇒ What is the IP-Address being assigned to the UE by the "*ESM Activate Default EPS Bearer Context Request*" PDU? Compare that value with the hexadecimal view in the *Results Panel* and the *Details View* of the Message Analyser:

IP-Address	**Decimal Representation:**	**Hexadecimal Representation:**

⇒ Show your results to the advisor!

c) Control Messages

Now we will observe the *Primitives* at the Configuration SAPs of the protocol instances:

⇒ In the *Sequence Representation View* of the Message Analyser, observe the message #120: Which SAP is used and what is the purpose of this *Service Primitive*?

SAP:	**Purpose:**

⇒ Which Bandwidths are configured for the Down- and Uplink?

Downlink:	**Uplink:**

⇒ In the *Structure View*, right-click to open the context menu of the *Information Element* "*DlLte-Bandwidth*" = 50 and select Add Column, in order to show an extra column in the *Sequence Representation View*.

⇒ Observe the newly created table column. Which protocol layers of the *eNodeB* (i.e. the *CMW500* tester) are configured to this downlink bandwidth?

Layers:

Experiment 2 : Broadcasted Information and Random Access (90 min)

a) System Information

Look at the MIBs (*Master Information Blocks*), passed regularly from the MAC to the PHY:

⇒ Activate the PDU filter ( ), in order to reduce the view to *Protocol Data Units* only

⇒ What is the transmission interval (in ms) of the MIB?
For that purpose, display the SFN in a separate column of the *Sequence Representation View*, and compare it to the RFN (*Radio Frame Number*)!

⇒ What is the downlink bandwidth (in RBs) indicated on the PBCH?

Now look at the SIBs (*System Information Blocks*), shown at the test SAP ***RRC_INFO***:

⇒ What is the PLMN (*Public Land Mobile Network*) ID?

MCC:	**MNC:**

⇒ Which is the minimum necessary power that the UE has to receive this *eNodeB*, in order to select it as its serving cell?

⇒ Are there any neighbour cells announced (give reason!)?

b) Setting View Filters

Now, we will have a brief look at MAC PDUs:

⇒ Open the filter editor by → Add Empty Filter... → New Generic Statement.... On the left side of this window, *include* conditions can be added, the right side shows *exclude* conditions.

⇒ Add an include statement by selecting → Prot: *LTE*, in order to see only MAC PDUs observed at the PHY_SAP. Save and apply ( ) the filter as `Lte_onlyPhySap.fil`.

⇒ Which MAC PDUs can be seen at the beginning of the scenario (msg. no. <1000), and which are exchanged at the end of the scenario (msg. no. >4000)?

Beginning:	**End:**

c) Random Access of the UE

⇒ Switch off the message filter and disable the PDU filter again to see all messages.

⇒ Search for the indication "*Prach is Received*" from MAC → RRC:
How many µs later is the RAR (*Random Access Response*) message sent?

"*Prach is received*" Message no.:	***RAR* Message no.:**	

⇒ Which RA-RNTI ("*RNTI*") is used to scramble the CRC of this RAR message? Which T-RNTI ("Temporary ID for UE") is assigned to the UE at the very end of the message?

RA-RNTI:	**T-RNTI:**

⇒ Select again, in order to display exclusively *Service Primitives* that bear PDUs.
Why do the messages observed before disappear now?

⇒ Find out the Number of the so-called "message 3", which is received in the UL from the UE with its individual *T-RNTI*: Which PDUs of the different layers MAC, RLC, PDCP, and RRC can you identify? Note: Use the help of the tab Parent/Children in the *Results View*!

MAC:	**RLC:**	**PDCP:**	**RRC:**

⇒ Which MAC PDU completes the *Random Access* procedure (search "*Contention Resolution*"), indirectly assigning the final C-RNTI?

Message no:	**C-RNTI:**

⇒ How long (time in ms) did the complete *Random Access* procedure take?

Experiment 3 : Layer 3 Messages for Attachment (60 min)

a) Layer 3 Messages for Attachment (1/3)

In the following, you should compare the PDUs in the message sequence chart below with the PDUs shown in *Message Analyser*.

⇒ Label them with their chronological message no.. See the example below for (1), being labelled with #138 and #139 !

⇒ Define and apply ( ) a new filter `Lte_onlyLayer3.fil` that shows all LTE messages, excluding PHY, MAC, RLC, and RRC (because RRC PDUs are already observed, when they pass the PDCP_CTRL_SAP).

⇒ Which of the PDUs appears in more than one message?

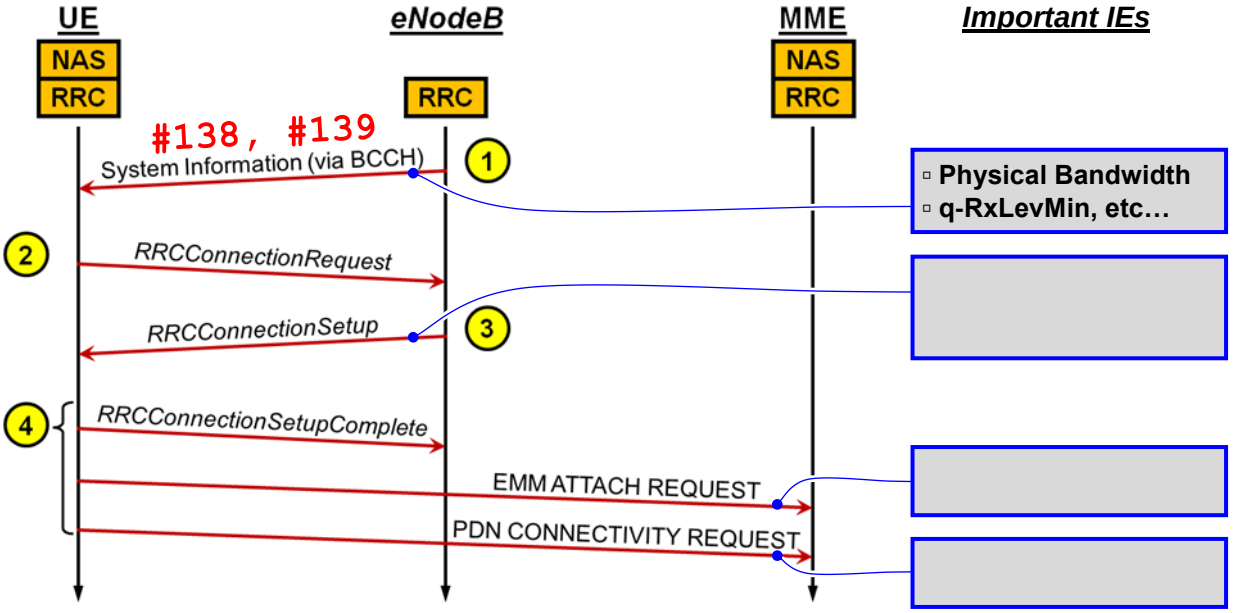


Figure 4-6: 1st part of the Layer 3 messages for Attachment

3 ⇒ The main purpose of the *RRCConnectionSetup* is the configuration of all UE Layers PHY, MAC, RLC, PDCP, and the RRC itself:

How many SRBs (*Signalling Radio Bearers*) are installed in the RRC?

Are there any DRBs (*Dedicated Radio Bearers*) installed?

4 ⇒ Which of the messages #2388 or #2733 transports the IMSI, which one the desired IP version?

IMSI:	IP-Version:

b) Layer 3 Messages for Attachment (2/3): Security Configuration

⇒ Label all PDUs in the message sequence chart below with their chronological message no.:

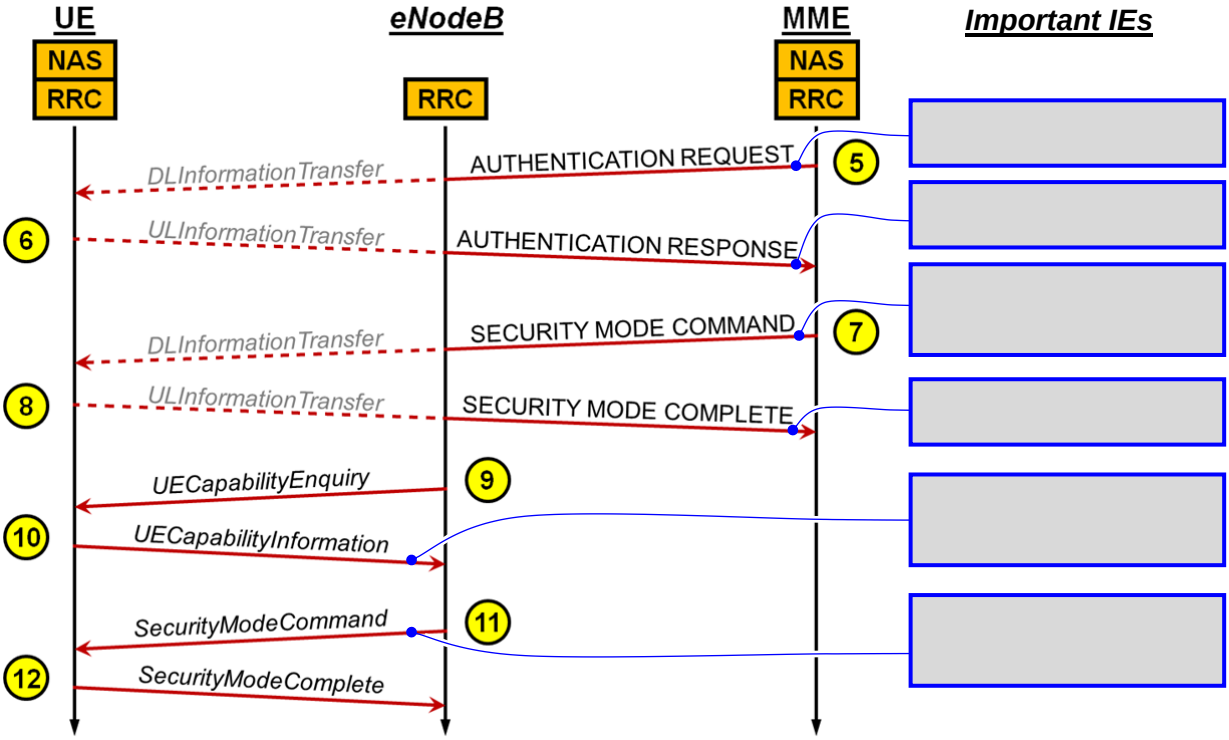


Figure 4-7: 2nd part of the Layer 3 messages for Attachment: Security Procedures

⇒ How many bytes do the parameters *RAND*, *AUTN*, and *RES* occupy during the Authentication?

5 & 6

RAND:	**AUTN:**	**RES:**

⇒ Which of the security algorithms listed on page 4-3 is used for the NAS security?

***Integrity Protection*:**	**Ciphering:**

⇒ What is the IMEI of the UE (sent with the *Security Mode Complete* message)?

IMEI:

⇒ Which of the security algorithms listed on page 4-3 is used for the AS security?

11 & 12

***Integrity Protection*:**	**Ciphering:**

⇒ All RRC *Service Primitives* in the Uplink carry an *Integrity verification result* as the last element. Compare the results before and after the configuration of the NAS security, in order to check if security is activated or not.

Result before configuration of NAS security:	**Result afterwards:**

c) Layer 3 Messages for Attachment (3/3)

⇒ Label all PDUs in the message sequence chart below with their chronological message no..

⇒ Which of the PDUs appear in more than one message?

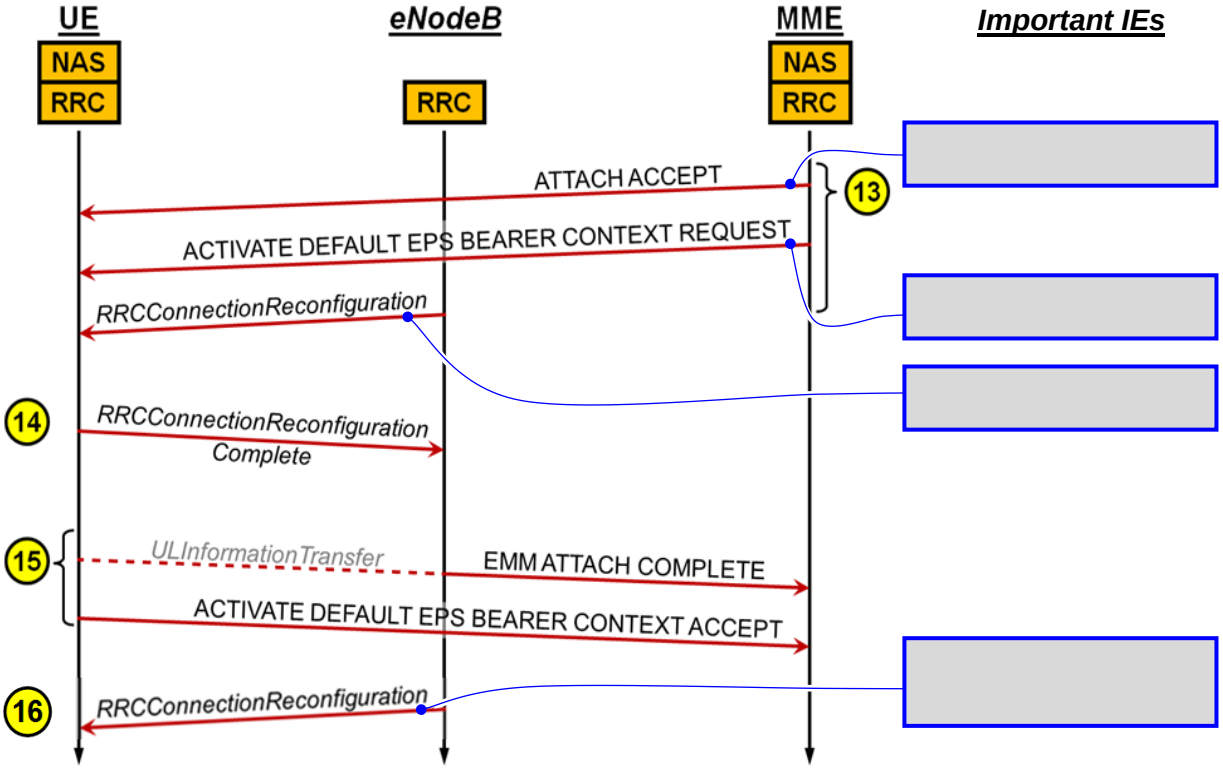


Figure 4-8: 3rd part of the Layer 3 messages for Attachment

⇒ Which of the PDUs assigns the GUTI (*Global Unique Temporary Indentifier*) to the UE?
13 & 14 What is the M-TMSI part of the GUTI that can be found in the message log?

PDU:	M-TMSI:

⇒ Which of the PDUs assigns the IPv4 Address to the UE? Which value does it have?

PDU:	IP Address:

⇒ The *RRCConnectionReconfiguration* message alters/extends the adjustments made during the *RRCConnectionSetup* 3. Are there any *Dedicated Bearers* added?

⇒ Does the UE have to do periodic TAUs (*Tracking Area Updates*)?
Look at the value of the *Timer T3412* !

15 ⇒ Which previous requests of the UE are completed by the messages *EMM Attach Complete* and *Activate Default EPS Bearer Context Accept*?

16 ⇒ Message #3714 shows a second *RRCConnectionReconfiguration* PDU. Look at the *MeasurementReports* after that message. Which re-configuration is therefore done by message #3714?

⇒ How many different cells are measured and which reporting period is configured?

d) Conclusion

⇒ Go through Figure 4-6 to Figure 4-8. In the grey/blue boxes on the right side, enter the most important parameters being contained in the messages as *Information Elements*. See the example for message 1 on page 4-10 !

⇒ How much time did the complete *Attachment* procedure take?

⇒ Finally, export all Entries of the *Sequence Representation View* as an active html page to your personal memory stick (material for your examination preparation :-):
With the filter `Lte_onlyLayer3.fil` being set, press Ctrl-A, in order to mark all messages, and then File→Export to HTML....

Exercise 5: Getting Started with Wireshark

This exercise should help you to get familiar with protocol analysis using the software *Wireshark*.

References: [WiS], [Tan], [Sta]

5.1 Equipment

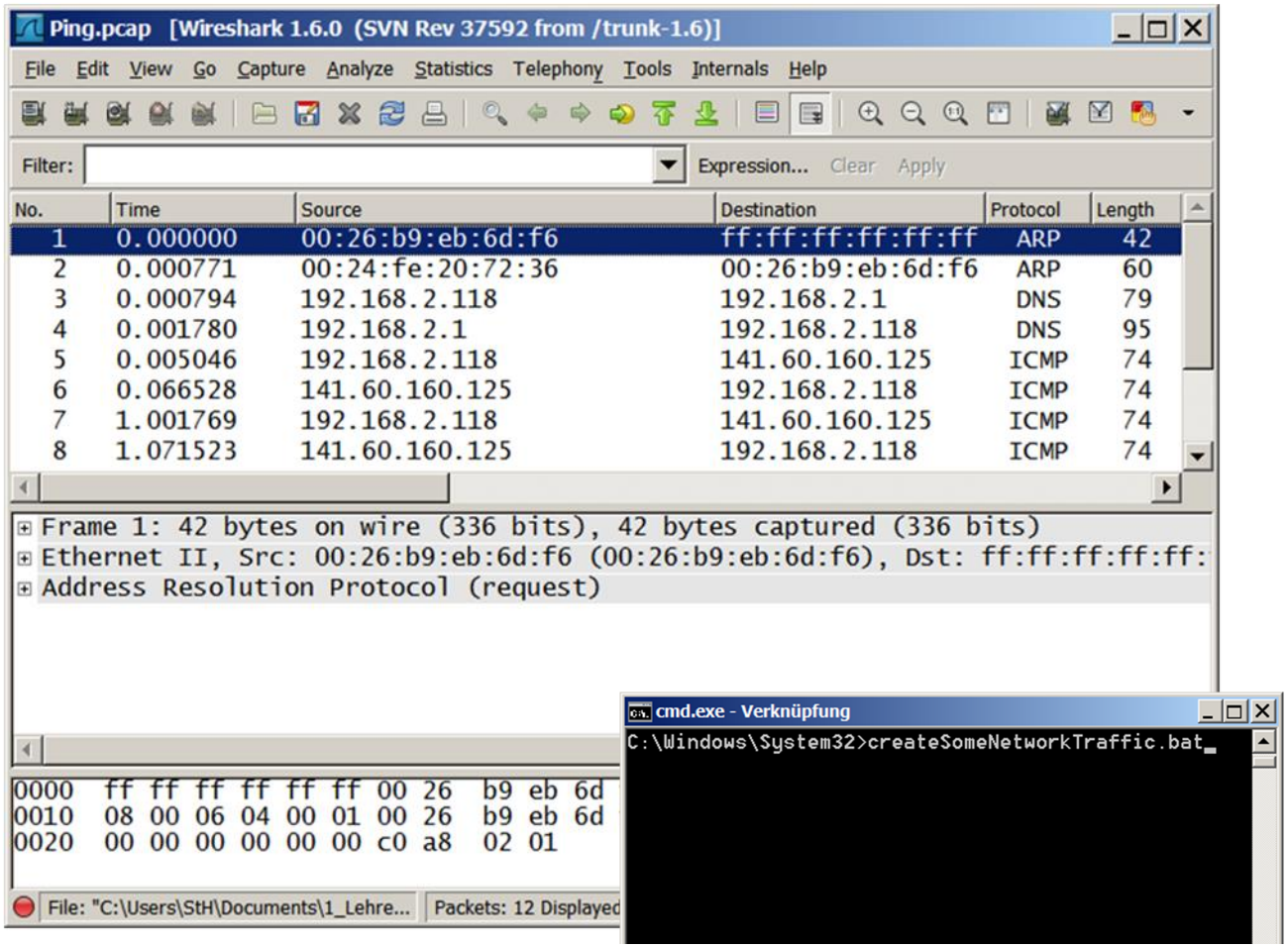


Figure 5-1: For this exercise, we don't need more than *Wireshark* and the *Windows* command tool...

- PC with the software *Wireshark* (see short manual on page 5-3).

5.2 Preparation Problems (to be answered before doing the Lab!)

a) Get familiar (again) with the software *Wireshark* (see a short description on page 5-3)!

b) The IP (*Internet Protocol*) provides a so-called *packet-switched service*. Find out three properties, which distinguish IP from a *circuit-switched service*:

1)
2)
3)

c) Find out the names of the following three address types and give an example for each:

⇒ How is a network interface (e.g. an *Ethernet* card of a PC) addressed?

Address Type:	**Example:**

⇒ How is a computer system anywhere in the global internet addressed?

Address Type:	**Example:**

⇒ How can a service (FTP, HTTP, eMail, NTP) available on a specific system be differentiated?

Address Type:	**Example:**

d) In the Practical Part, we will run the following batch file:

```
ipconfig /flushdns
arp -d
ping www.db.de
```

Find out what the commands in each of the three lines mean and explain its exact effect with the given parameters:

Line 1:
Line 2:
Line 3:

5.3 Practical Part

Preparation: Open the directory **`Lab_R1.16`**, on the **`Desktop`** of your computer. Here, enter the **`zip`**-archive **`WiCS`**, and copy the directory **`Wireshark`** directly to your **`Desktop`**. This directory contains the file(s) which you will use and manipulate for this exercise.

Experiment 1 : Capture of a Web-Site Transmission (20 min)

a) Capture some self-generated network traffic:

⇒ Start the software *Wireshark* and start capture.

⇒ Run the batch file **`createSomeNetworkTraffic.bat`** and wait until it is processed.

⇒ Stop capture. Now, *Wireshark* should have collected many hundred *Ethernet* frames!

b) Now, **try to reduce the amount of displayed data**:

⇒ List at least 5 protocols, which are used on this network beside *Ethernet*, *IP*, and *TCP*.

⇒ Set an appropriate filter to display only those frames that contain IP packets.

⇒ Set an appropriate filter to display only those frames that contain IP packets with either the source or the destination address of your local machine (see label attached to the monitor).

⇒ Which MAC-Address does your local PC use?

: : : : :

Experiment 2 : Analysis of the Wireshark Trace (20 min)

Set appropriate filters in *Wireshark* (see page 5-3) to answer the following questions:

⇒ Which is the numerical destination IP address for the *PING* Requests?

IP address =

⇒ What is the value of the parameter '*Type*' inside the ICMP (*Internet Control Message Protocol*) segment for *PING Requests* and *PING Replies*?

***PING Request* '*Type*':**

⇒ How many bytes of payload (i.e. data) does the first *PING Request* contain?

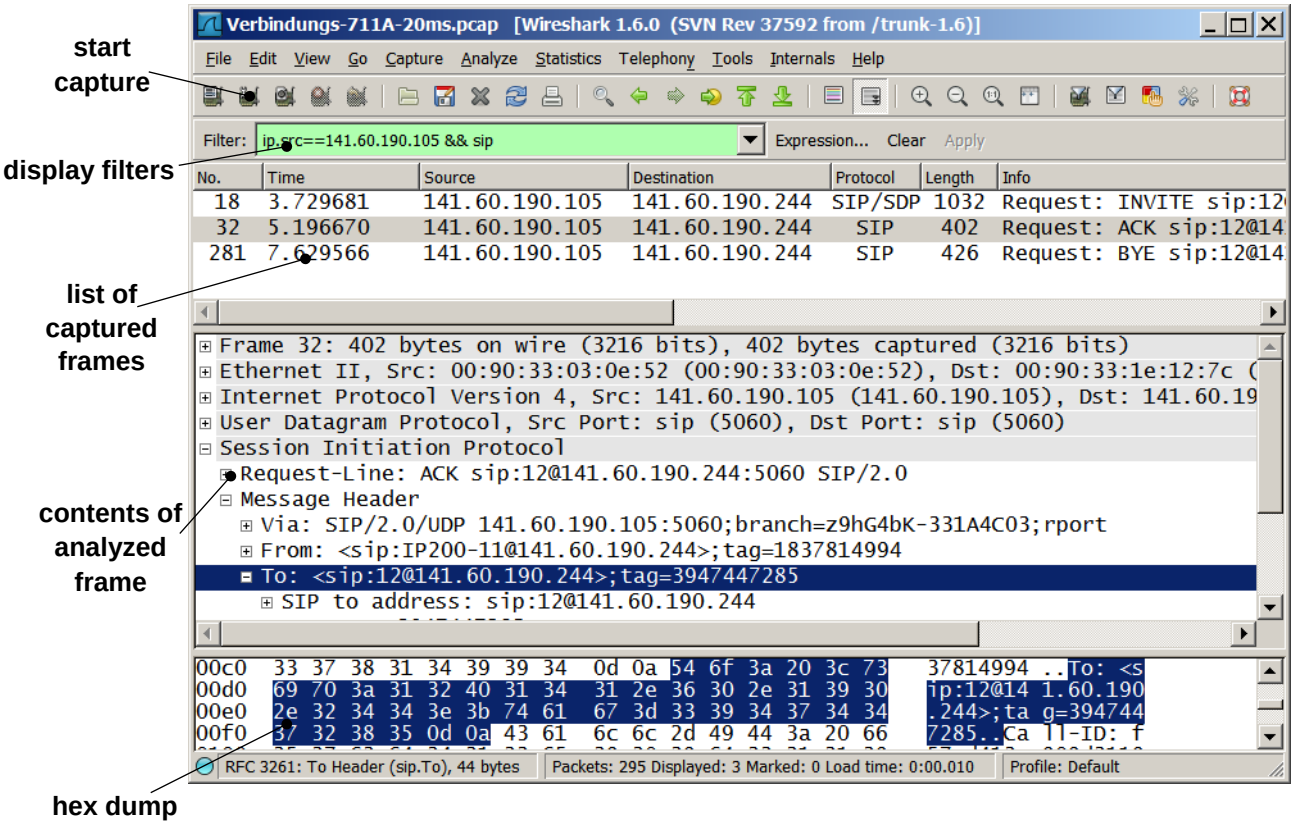
⇒ Does *PING* work? If not (a "timeout" is returned): What is the reason for that?

Cleanup: Please delete the directory **`Wireshark`** from the **`Desktop`** of your computer.

5.4 Short Manual of the Network Analyser Software *Wireshark*

Wireshark is a freeware tool for network administrators with two purposes:

- Capturing of frames that are received from any (wired, wireless, or virtual) network card.
- Analysis of these captured frames by a comfortable user interface. Hundreds of different protocols have been included until now, including most IE³ protocols and the complete TCP/IP family:



Start capture: Capturing of frames is done by starting and stopping the tool.

List of captured frames: After capture, a numbered list of all Layer 1 frames appears in the upper window. The highest protocol found in each frame is displayed in the column *Protocol*, followed by a short summary of the contents.

Contents of analysed frame: The frame, which is selected in the list of captured frames will be analysed in the window below. Each protocol that can be found inside that frame will be listed here in an upside-down, tree-like notation. Pressing the crossed expand buttons gives more info about the selected protocol contents.

Hex dump window: The third window shows a dump of the frame in hexadecimal form and (on the right) in ASCII form as well. The region of the selected protocol will be highlighted here.

Display filters are used to hide non-relevant frames. Enter Boolean expressions here. Some examples:

Filter string	Effect
ip	Only frames containing an IP packet are displayed
ip.addr == 12.34.56.78	Only frames containing packets with the IP address 12.34.56.78 (source or destination) are displayed
(tcp.flags.syn == 1) && (ip.addr == 12.34.56.78)	Only frames containing TCP segments with IP address 12.34.56.78 and SYN flag set are displayed

Exercise 6: VoIP (Voice over IP)

VoIP (*Voice over Internet Protocol*) services use IP(*Internet Protocol*)-based packet switched networks in order to transport synchronous (i.e. *circuit-switched*) data. The most popular VoIP standard is the text-based SIP (*Session Initiation Protocol*), the basics of which will be introduced in this lab exercise.

References: [TrWe] (German, in Library, strongly recommended!), [WiS]

6.1 Equipment

Figure 6-1: *VoIP* experimental setup

- PC with the software *Wireshark* (see short manual on page 5-3)
- Two IP telephones *Innovaphone IP 200* (see manual on page 6-13 and in the University's *Online Community*)
- Two IP telephones *Innovaphone IP 241*
- Registrar/Gateway *Innovaphone IP 302*, equipped with PBX (*Private Branch Exchange*) software
- *Netgear* Ethernet Hub
- Patch cables, power adaptors, etc.
- **Please bring a pocket calculator!**

6.2 Background

The SIP standard was developed at Columbia University in the mid-90's, and standardized in 1999 by the IETF (*Internet Engineering Task Force*). A special feature of SIP is the similarity to the HTTP (*Hypertext Transfer Protocol*): All signalling messages are text-based.

Also, SIP plays a central role for offering user services (like voice, file transfer, instant messaging, etc.) in the 3rd and 4th generation of mobile communication – In so-called *Next Generation Networks* (NGNs), all user traffic shall be transported via SIP in the *IP Multimedia System* (IMS).

6.2.1 Structure of a SIP network

SIP uses a Client-Server architecture. The most important elements are listed below:

(1) User Agents

User Agents (UAs) are the terminals of the SIP network, e.g. telephones. Two user agents can communicate with each other, if their addresses are known to each other. The terminal initiating a transaction is called UAC (*User Agent Client*), the other (passive) is the UAS (*User Agent Server*).

Each UA is identified by a unique address, the so-called URI (*Uniform Resource Identifier*). The (simplified) format of an URI is similar to eMail address:

```
<username>@<hostname>
```

The permanent URI should be world-wide unique, so the user can be called from world-wide.

(2) Registrar Server

The *Registrar Server* is the central element of a SIP network that enables mobility of the users: Each UA that wants to participate in the network, has to register with the *Registrar Server* before:

In order to allow mobility of the users, the same user may run his UA on different hardware and in different local area networks at different times. In that case, the UA will identify itself by a temporary URI (like shown above, but with `hostname` set to the current IP address). In order to map incoming calls for a certain permanent URI to the current location of the UA, the *Registrar* has to be informed about both the temporary and the permanent URI of the UA. The pair of permanent (i.e. world-wide unique) and temporary URIs will then be stored into a *Location Server*, where it can be accessed by the *Proxy Server*.

This registration is invoked by the *REGISTER* request, which will be explained in chapter 6.2.3.

(3) Proxy Server

The *Proxy Server* can be seen as the switchboard exchange of SIP. Instead of setting up a session directly between two UAs, the *Proxy* acts as intermediate agent, performing both the counterpart

- of a *Server* (for the calling UAC), and
- of a *Client* (for the called UAS).

In the case of mobile (i.e. location-independent) UAs, the *Proxy Server*

- accepts sessions (i.e. INVITE requests) to a permanent URI,
- then looks up the associated temporary URI of the UAS in the *Location Server*,
- and finally forward the INVITE request to the temporary destination of the UAS.

Proxy Servers can be organized hierarchically and distributed. Since all signalling traffic between the UAs is forwarded by the Proxies also taking the routing decisions for the SIP messages.

In many cases (as well in this lab exercise!) *Registrar* and *Proxy Server* are collocated in the same hardware, also implementing the *Location Server* and a PBX (*Private Branch Exchange*).

(4) Gateway

A VoIP network would be boring, if it could only be operated isolated. The *Gateway* connects it to *circuit-switched* networks like ISDN.

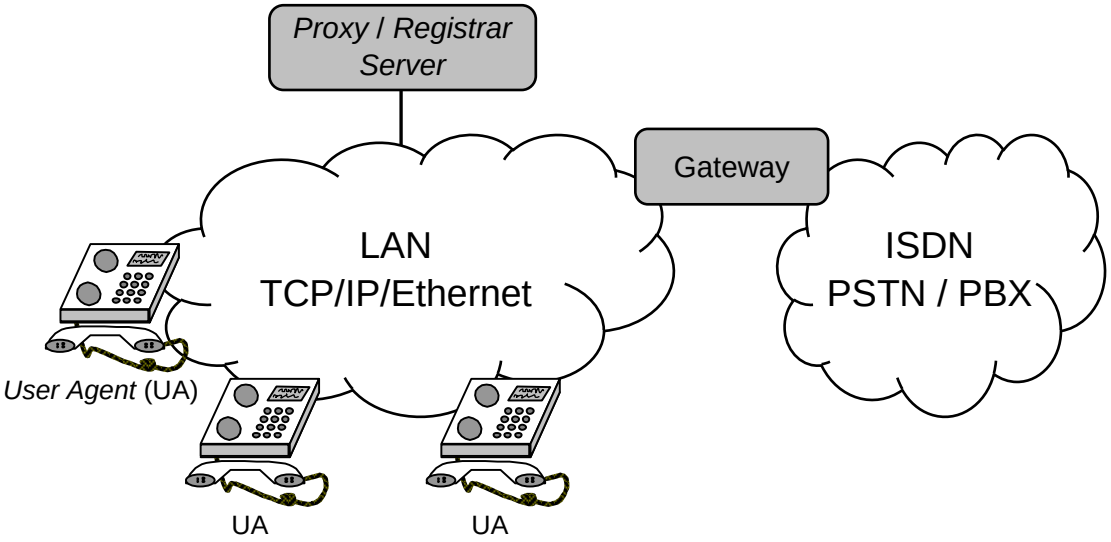


Figure 6-2: Example for a small SIP-Network with three terminals, a *Proxy*/*Registrar* Server and an ISDN Gateway

6.2.2 The SIP Protocol Stack

Figure 6-3 shows the protocols being used for SIP audio or video sessions, in the *OSI Reference Model for Communication*:

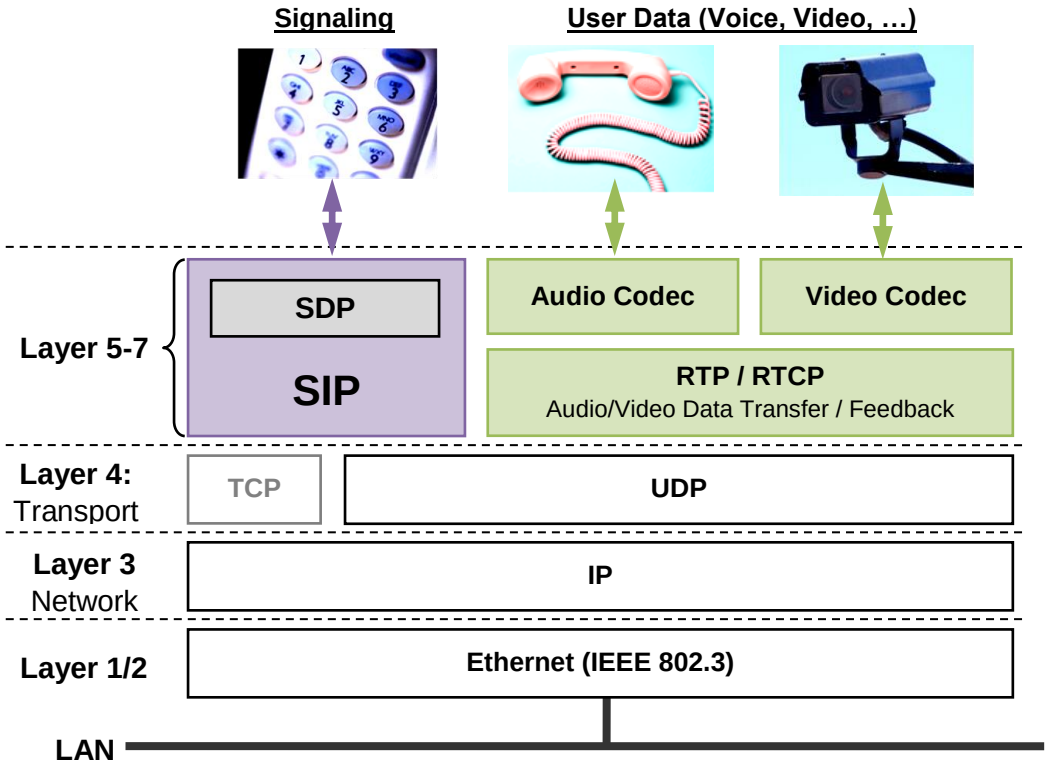


Figure 6-3: Protocols used together with SIP

- **SIP** is used for all signalling messages, like registration, session setup and shutdown are controlled by text-based messages. See section 6.2.3 for a list of SIP *Requests* and *Responses*. Normally SIP uses UDP for the underlying transport protocol, but optionally TCP can be used as well.
- The **SDP** (*Session Description Protocol*) is used for the negotiation of the type of user data to be exchanged (e.g. audio and/or video), the Codec, ports, and the direction of the transmission. The SDP messages can be embedded inside SIP *Requests* of the method '*INVITE*', and SIP *Responses* of the type "200 OK".
- The **RTP** (*Real Time Transfer Protocol*) transfers the user data stream, once a session has been established. By sequentially numbering the real-time data packets, RTP takes care of the correct order at the receiver and decides to dismiss packets which are lost or arrive too late. Each stream transports uni-directional data of one type. Each data stream is initiated by one **RTCP** (*Real-Time Control Protocol*) message.

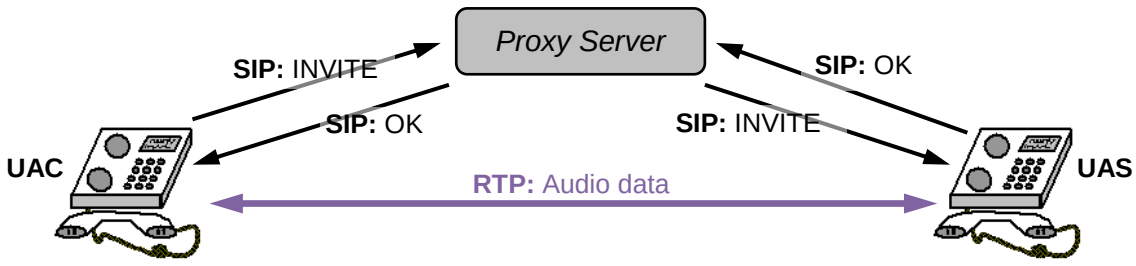


Figure 6-4: Example: Protocols used for setting up a telephone session with the presence of a *Proxy Server*

6.2.3 SIP Messages

All messages are text-based, consisting of the following parts:

- **Start Line**: Indicates the method (e.g. INVITE, REGISTER, OK, etc.) and the destination URI (e.g. the UAS for the INVITE message).
- **Message Header**: Contains parameters for the respective method, separated by semicolons.
- **An Optional Message Body**: Contains optional information, like the embedded SDP messages, see section 6.2.2 above.

SIP Requests

Requests are sent from the UAC to the UAS. Some methods are shown below:

Table 6-1: Some SIP Requests

INVITE	The UAC initiates a new call by inviting the called party UAS to a session.
ACK (nowledgement)	Sent by the UAC in order to acknowledge the session establishment
BYE	Indicates that one of the UAs wants to close the session (hang-up)
REGISTER	Used by a UA to inform the *Registrar* Server about its permanent and temporary URI.

SIP-Responses

Responses are sent from the UAS to the UAC in order to provide status information, as an answer to prior *Requests*. Unlike *Requests*, *Responses* are not specified by a method, but by a three-digit number. Some types are shown below:

Table 6-2: Some SIP Responses

100 Trying	The Request has been successfully received, but not yet processed entirely.
180 Ringing	Indicates that a UAS that has received an INVITE Request is ringing
200 OK	The prior *Request* has been successfully received and processed.
4xx **Request Failure**	An error occurred while processing the request

6.2.4 Protocol Scenarios

Registration Example

As mentioned above, a UA informs the *Registrar* about its temporary URI. Note the two parameters of the '*REGISTER*' message, indicating the permanent ("To") and the temporary ("Contact") URI:

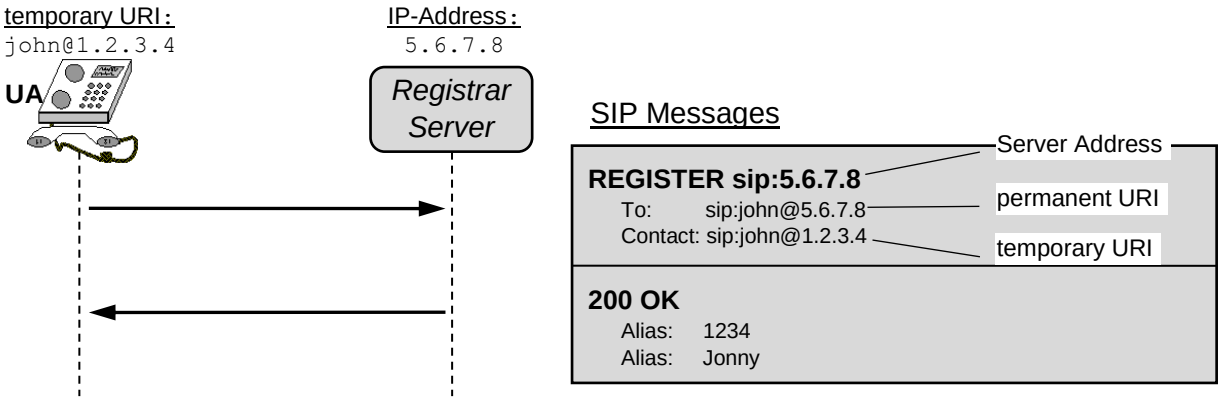


Figure 6-5: MSC for the Registration of a UA with a *Registrar*

The *Response* "OK" optionally carries Aliases, like nick names or conventional telephone numbers.

Establishment of a Voice Session between a *UA Client* and a *UA Server*

A SIP session is established by a Three Way Handshake: The 'INVITE' Request of the UAC is acknowledged by an '200 OK' Response from the UAS. The third necessary message leading to the RTP session is an 'ACK' Request from the UAC, which actually confirms the '200 OK' message. The Response '180 Ringing' indicates that the 'INVITE' Request has reached its destination:

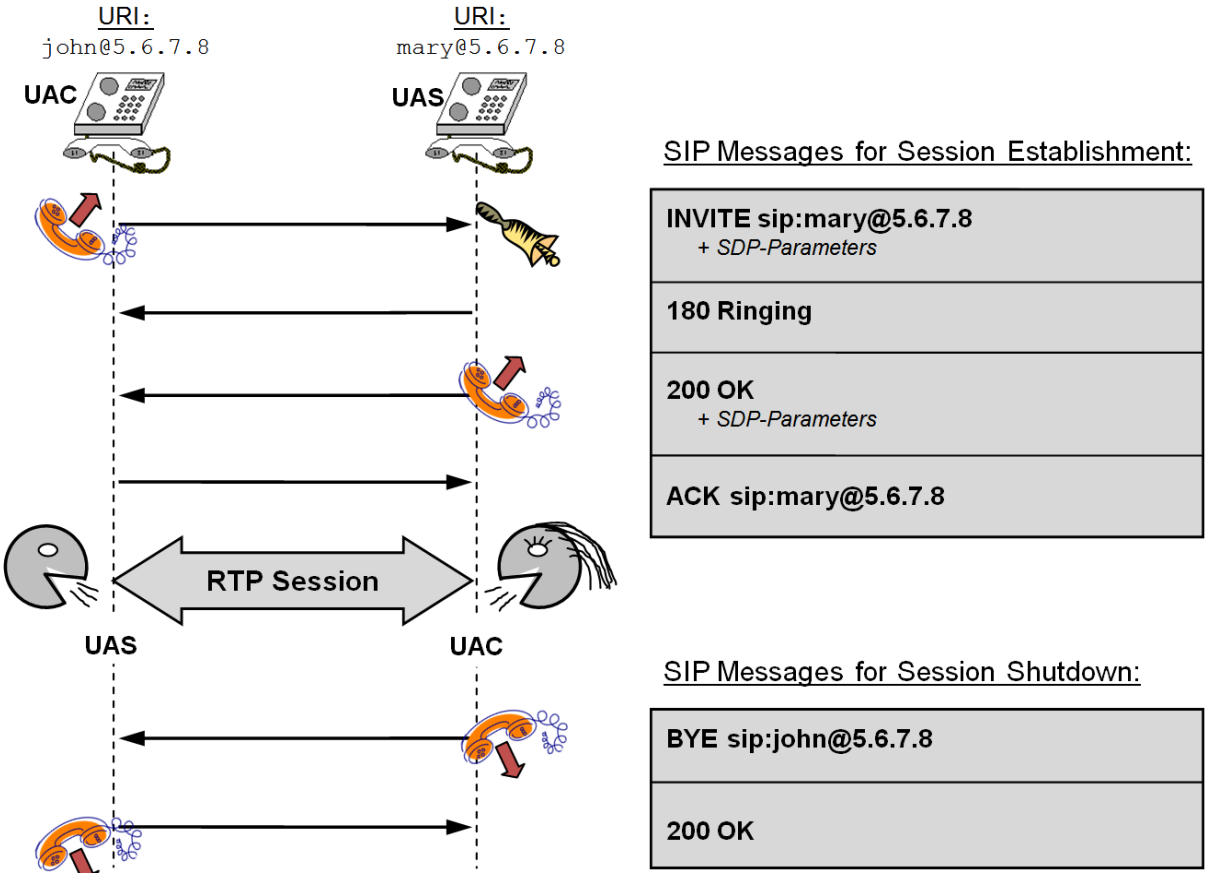


Figure 6-6: MSC for a Session Establishment

6.2.5 Speech Codecs

For VoIP, many different Audio Codecs have been defined. They influence the perceived speech quality (bandwidth, delay, and other subjective effects resulting from data compression), and the necessary data rate for a VoIP session. In this lab exercise, we will use two Codecs G.711 and G.729A:

Table 6-3: Some speech Codecs defined for VoIP

Codec		Transfer Data Rate	Audio Bandwidth	Coding Delay
G.711a	PCM	64 kbit/s	0.3 … 3.4 kHz	0.125 ms
G.729a	CS-ACELP	8 kbit/s	0.3 … 3.4 kHz	15 ms

6.3 Preparation Problems (to be answered before doing the Lab!)

a) Have a look at the manual for the telephone IP200 on page 6-13.

b) As shown in Figure 6-7, we use an old-fashion *Ethernet* Hub instead of a Switch, in order to connect the SIP components and the PC running *Wireshark*. Why?

c) SIP Network Structure

⇒ Explain the difference between a *Registrar* and a *Proxy Server* in your own words:

⇒ Telephone calls can be performed with or without *Registrar* and *Proxy Servers*.
List two advantages of using these servers:

d) SIP Protocols

⇒ Look at the protocol scenario for a session establishment in Figure 6-6. Why do ***John*** and ***Mary*** exchange their roles as UAC and UAS during the scenario?

⇒ Look at the setup of the practical exercise in Figure 6-7. What will be sensible permanent URIs of the four UAs?

⇒ How are the temporary and the permanent URI coded into the '*REGISTER*' *Request*?

Temporary URI:	**Permanent URI:**

e) Several different **speech Codecs** can be used. List one advantage for each of the Codecs below:

G.711a:	**G.729a:**

6.4 Practical Part

The following figure shows the setup of the experiment. There are 4 terminals, and one box housing the *Proxy/Registrar/Gateway Server*. The *Ethernet* Hub allows the PCs to trace every communication inside the network:

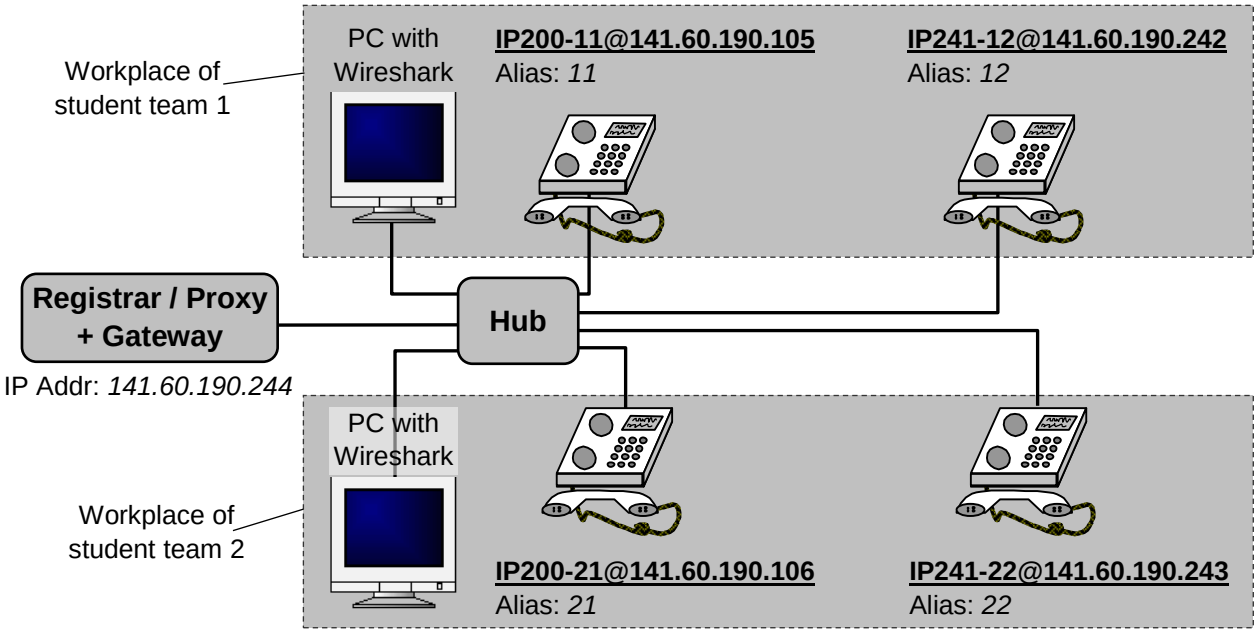


Figure 6-7: Schematics for the VoIP exercise. The UAs are labelled by their temporary URIs

Please note: Each student team has two telephones, one of type ***IP200*** and one of type ***IP241***, with individual IP addresses. In order to keep this lab instructions universal, instead of the actual IP addresses of the team, the text refers to the respective telephone types (i.e. ***IP200*** or ***IP241***) of the student team.

Experiment 1 : Examination of basic SIP Protocol Scenarios (40 min)

a) Direct Call setup using the temporary (= current) URI

⇒ Try to call the different telephones in the network from your ***IP241***.

b) Configuration of calls via the *Registrar/Proxy Server*

⇒ Open the Web GUI of your ***IP200***: Open *Firefox*, and open the Webpage with the IP address of your ***IP200*** telephone (Username: *admin*; password: *ip200*).

- Verify that the correct *Registrar/Proxy* is configured (Configuration → Registration1 → Registration → Primary Server Address).
- Verify that the correct UA name of the URI is configured (Configuration → Registration1 → Registration → User ID).

⇒ Open the Web GUI of the *Registrar/Proxy/Gateway* (IP address 141.60.190.244, user: *admin*; password: *WiCS-ip302*).

- Verify that the two telephones (UAs) of your team are recognized by the *Registrar* (Administration → PBX → Objects → PBX).
- Add a telephone number as an *alias* by clicking at the respective *Long Name* of the telephone and then fill the field *Number* in the window that opens. Click OK.
- Go to Administration → PBX → Registration, and wait until your UAs appear with *No*.

⇒ Now you can set up calls by dialling two-digit numbers!

c) Examination of the registration process with *Wireshark*

⇒ Open *Wireshark* and set a display filter "sip && ip.addr==<Address of your *IP200*>".

⇒ Remove the power supply of the ***IP200*** from the mains socket, start capturing in *Wireshark*, and plug in the power supply again. Wait for a successful registration and stop capturing.

⇒ Compare the recorded trace with Figure 6-5. What are the differences?

⇒ SIP messages are text-based! In *Wireshark*, select the '*REGISTER*' message, select the "Session Initiation Protocol" in the 2nd window, then choose the context menu item to Export Selected Packet Bytes... into a '`.txt`' file. Open that file with *Notepad*.

d) Examination of a session establishment:

⇒ Now use the software *Wireshark* to capture a complete call scenario from your ***IP200*** to your ***IP241***, including call establishment.

⇒ Change the display filter
- ... to see traffic from/to your ***IP241*** as well,
- ... to see RTP packets as well!

⇒ Complete the following MSC by...
- ... entering the IP addresses of the UAC and the UAS, and...
- ... all ***SIP*** messages until the session is established.

Ignore any traffic caused by '*REGISTER*', '*SUBSCRIBE*', and '*NOTIFY*' messages.

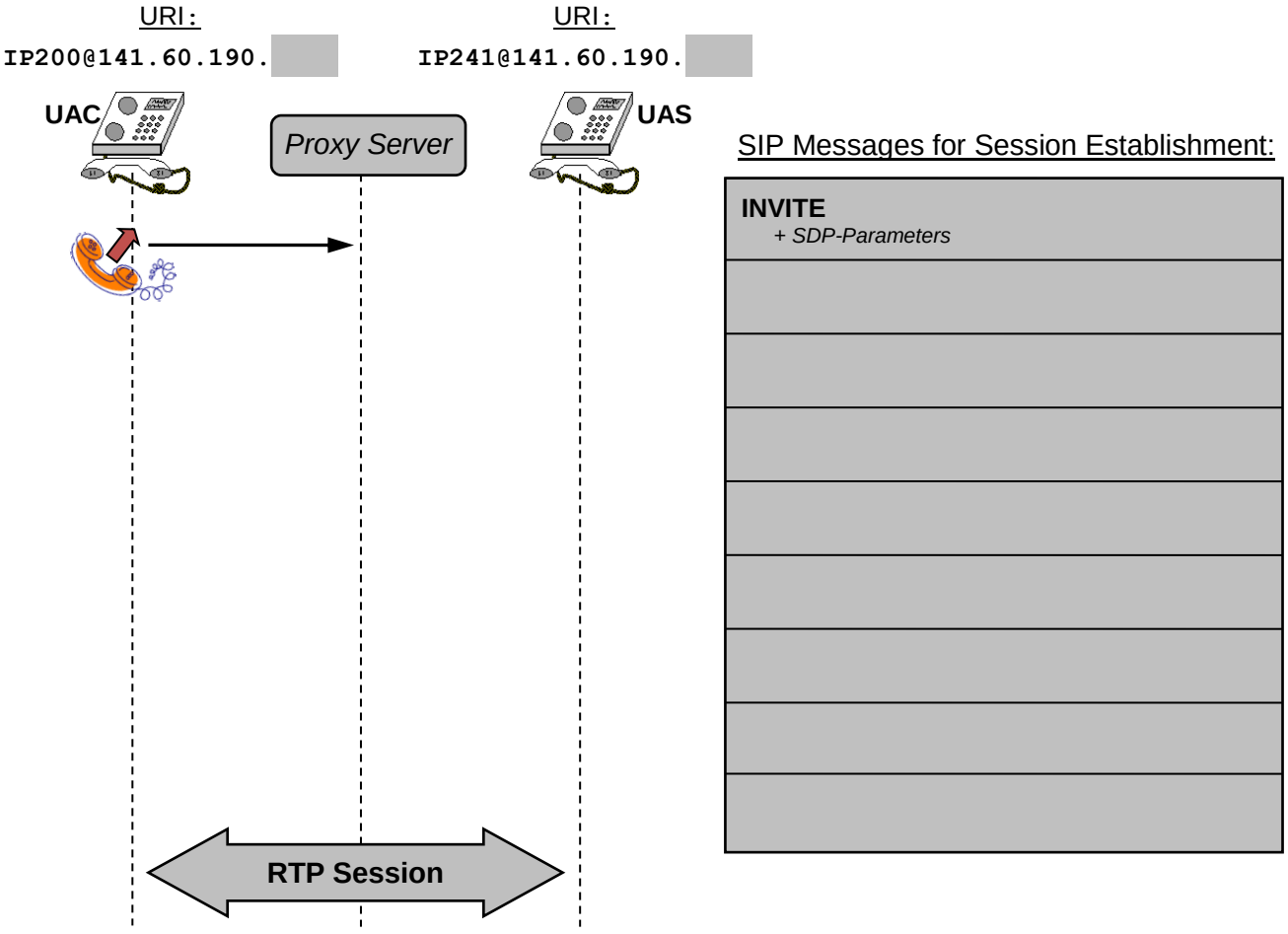


Table 6-4: MSC (*Message Sequence Chart*) for the establishment of a VoIP session

⇒ Verify the result by letting *Wireshark* draw a message sequence chart: Select an arbitrary frame of the VoIP conversation, and then Telephony→VoIP Calls→Select All→Flow .

⇒ What is the difference compared to the direct call setup in Figure 6-6?

⇒ Are the RTP packets also forwarded by the *Proxy Server*?

Do not close the *Wireshark* trace, because we need it in the next experiment.

Experiment 2 : Examination of the SDP and RTP Protocols (20 min)

a) SDP (Session Description Protocol)

⇒ Observe the embedded SDP messages in the '*INVITE*' *Request* and in the associated *Response*: How does the receiver know that this will be a VoIP session?

b) RTP (Real-Time Transfer Protocol)

⇒ Change the display filter, in order to see only RTP packets in the direction ***IP200*** to ***IP241***.

⇒ Which Codec is used for the payload?

⇒ Look at the increments of the fields '*Sequence number*' and '*Timestamp*'. What do these fields mean?

***Sequence number*:**	***Timestamp*:**

⇒ Replay this RTP stream in *Wireshark* by selecting an arbitrary RTP packet, and then Telephony→VoIP Calls→Select All→Player .

Do not close the *Wireshark* trace, because we need it in the next experiment.

Experiment 3 : Audio Codecs and Transmission Parameters (40 min)

a) **Audio Codec G.711A** (same Wireshark trace as before)

⇒ Calculate the net data rate of the speech channel from ***IP200*** to ***IP241***.

⇒ Compare this value with Table 6-3. If there is a difference, discuss it with the advisor.

b) Audio Codec G.729A

⇒ Change the Codec of your ***IP200*** to G.729A
(Configuration→Registration1→Registration→General Coder Preference: 'G729A').

⇒ Capture a new *Wireshark* trace for a VoIP session from ***IP200*** to ***IP241***.

- Check the subjective speech quality: Does it differ significantly from G.711A?

⇒ Calculate again the net data rate of the speech channel from ***IP200*** to ***IP241***.

Hint: One of the parameters that you have to find out for the calculation, is the number of packets send per time interval (e.g. 1 second). In Wireshark the keys **<Ctrl>+T** toggle the "time" column from absolute to relative times and vice versa.

⇒ Find out the gross data rate (including the overhead of all protocols):

⇒ How big is the relative protocol overhead in % (related to the pure audio payload)?

c) Effects of the packet size

Changing the payload size of the RTP packets can significantly reduce the protocol overhead:

⇒ Change the payload size of your ***IP200*** from 20 ms to 200 ms (Configuration →Registration1→Registration→General Coder Preference→Framesize [ms]: '200').

⇒ How big is the relative protocol overhead now?

⇒ Check out, what is the subjective disadvantage of using large RTP packets in a VoIP session!

Experiment 4 : Interconnection of a VoIP Network (30 min)

Note: This Experiment has to be done in common (both VoIP teams together)!

Now let's use a *Gateway* to connect the IP and the ISDN world. The ***IP302*** box also integrates a *Gateway*. Figure 6-8 shows the *Gateway* configuration GUI on the left side.

Three configuration steps of the *Gateway* have to be done:

a) Set up an **Interface** to the ISDN network (real hardware socket on the ***IP302***)

b) Set up a virtual **VoIP** interface

c) Set up a **Route** from the VoIP network into the ISDN network and vice versa.

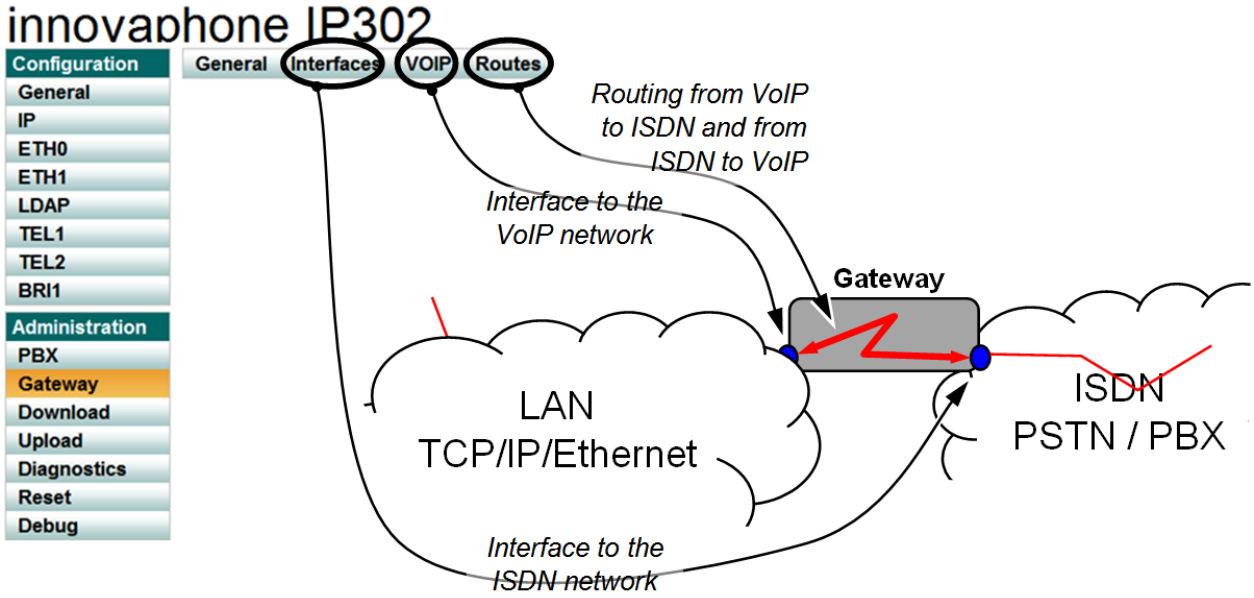


Figure 6-8: The Gateway between IP and ISDN and its configuration menu on the *IP302* GUI

a) The ISDN Interface

⇒ Ask the Lab advisor to connect the ISDN cable of the lab to the *IP302* box.

⇒ Check the physical ISDN connection (Configuration → BRI1 → State → Physical State: 'up').

⇒ Name the interface meaningfully (Administration → Gateway → Interfaces → BRI1 → Name).

b) The VoIP Interface

⇒ Define a prefix digit (e.g. the '**0**') that triggers outgoing VoIP calls via the *Gateway*: Administration → PBX → Objects → PBX → Gateway-VoIP → Number

⇒ Configure the *GW1* interface to register on the local *Registrar Server* (IP address 127.0.0.1): In Administration → Gateway → VoIP → GW1, set a meaningful Name, the Server Address, and in Alias List the exact Name: 'VoIP'.

⇒ Verify the successful registration of the *Gateway* at the *Registrar* (Administration → PBX → Registrations).

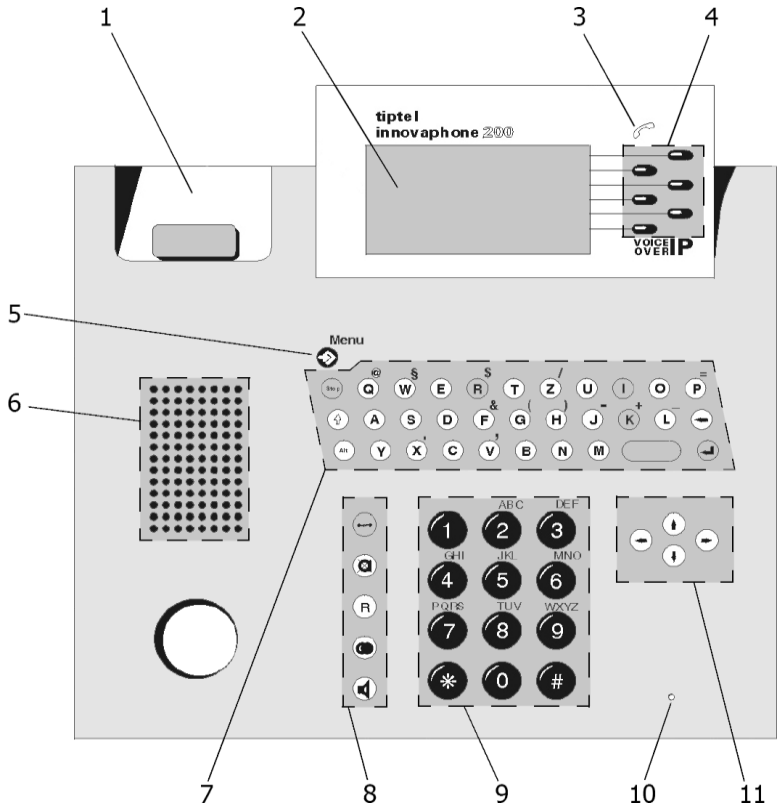
c) Routes for incoming and outgoing calls

⇒ In Administration → Gateway → Routes, click on  in order to insert a new route from the ISDN interface to the VoIP interface. Set a meaningful Description and configure the 'Number Out' field of a telephone, which shall ring in case of incoming calls.

⇒ Verify the successful configuration of the route by dialling the ISDN number **08031-805-1774**!

⇒ Add another route for outgoing calls from the VoIP interface to the ISDN interface and check if it works. Experiment with the '*Number In*' field, in order to swallow the prefix digit '**0**'...

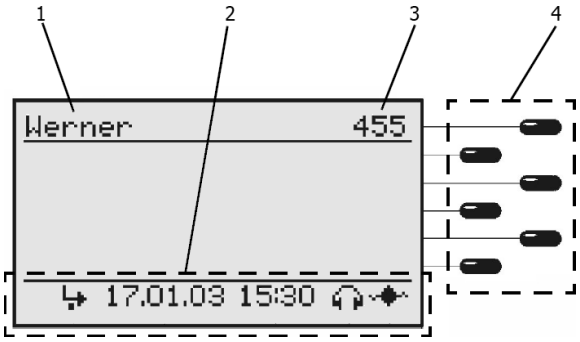
6.5 User Manual of the IP Telephone INNOVAPHONE IP 200



Pos.	Symbol	Description and key function
1		Handset support
2		Display
3		LED
4		Menu selection and Function keys
		The six Function keys next to the display are used to select menus, to execute functions assigned to them or to directly dial a subscriber displayed there.
5	Menu	The "Input/Menu" key is used to open the main menu or save changes.
6		Loudspeaker
7		Character keypad
	Stop	The Stop key is used to cancel a function and return to the next higher menu.
		The Shift key is used in combination with a subsequent key to enter capital letters.
		The Backspace key is used to delete characters to the left of the cursor while making an entry.
	Alt	The Alt key can be used in combination with a subsequent key to enter the special character above the key as well as additional special characters.
	Space	Space key for entering spaces
		The Enter key is used to complete an entry or select the current line of a list.

Pos.	Symbol	Description and key function
8		Function block
		The Clearing key is used to terminate calls.
		The Mute key is used to switch off/on the microphone.
		The Refer-back key is used to enable the functions "Hold" and "Switch". In idle mode, the Refer-back key is used to call up the list of missed calls.
		The Redial key is used to call up the list of 100 phone numbers last dialled.
		The Loudspeaker key is used to switch on/off the hands-free system.
9		Dial keypad
	❶ ... ❾	Digit keys for entering phone numbers
	✱ #	The Star and Hash keys have special functions with touch-tone dialling.
10		Microphone
11		Arrow keys for navigating within the menu

Display:



Pos.	Symbol	Designation
1		Name (H.323 ID or nickname of the innovaphone PBX configuration)
2		Status line; gives information on the current status of the telephone using the following symbols:
	17.01.03	Date
	15:30	Time
		No connection to the gatekeeper
		Connection established to the gatekeeper
		Loudspeaker on
		Hands-free system activated
		Microphone switched off (symbol flashes)
		Headset activated
		Call diversion activated
		Handset activated
		Telephone blocking activated
3		Own call number (E.164)
4		Function keys for selecting functions directly

Exercise 7: The *MPEG-2* Transport Stream

In this exercise you will get an introduction to the structure of MPEG-2 Transport Streams. Inside a *LabVIEW* program for transmitting a pre-recorded Transport Stream as a *DVB-T(1)* signal, you will dissect and re-assemble the MPEG-2 data in real-time.

Reference: [DvbT]

7.1 Equipment

Figure 7-1: Receiving an MPEG-2 Transport Stream that has been transmitted by a *LabVIEW* program

- PC equipped with:
  - ⇒ PCI UHF modulator card *DekTec DTA-110* (with cables connected to the 2 RF outputs)
  - ⇒ Software *DekTec StreamXpress* v3.0.9
  - ⇒ Software *LabVIEW*, for accessing the API of *DTA-110* (find API documentation in the University's *Online Community*)
- DVB-T(1) Receiver:
  - ⇒ DVB-T(1) Mini Receiver *Orbit* (with power adapter, antenna adapter, and remote control)
  - ⇒ Measurement Receiver *KWS AMA300*

7.2 Background

7.2.1 Hard- and Software being Used for this Exercise

Figure 7-2 shows the main components of this exercise: The PCI card *DekTec DTA-110* is used to modulate an MPEG-2 Transport Stream as a DVB-T(1) signal. Two possibilities exist for the control of the PCI card and the generation of the Transport Stream:

⇒ The play-out software *DekTec StreamXpress* (also being used in the DVB-T2 exercise).

We use this program to analyse the contents of the Transport Stream to be replayed from a file, and its PIDs (*Packet Identifiers*). The data rate of the Transport Stream can be analysed as well.

⇒ The *LabVIEW* program *Ts2DvbT.vi*, the source code of which is shown in Figure 7-5.

We use this *LabVIEW* VI (*Virtual Instrument*) and its Sub VIs in order to dissect and re-assemble the MPEG-2 TS (*Transport Stream*), before it is replayed by the modulator *DTA-110*. This allows us to read a TS from a file, manipulate it, and observe the result at the receivers.

The RF signal produced by the *DTA-110* can then be analysed by up to two receivers:

⇒ An ordinary DVB-T(1) receiver *Orbit*

⇒ The measurement receiver *KWS AMA300*, which will be used for the last Experiment 5 (*Manipulation of Service Description Tables*), because it immediately shows changes of the *Service Names* provided in the TS. Furthermore, the *AMA300* displays errors in the MPEG-2 TS.

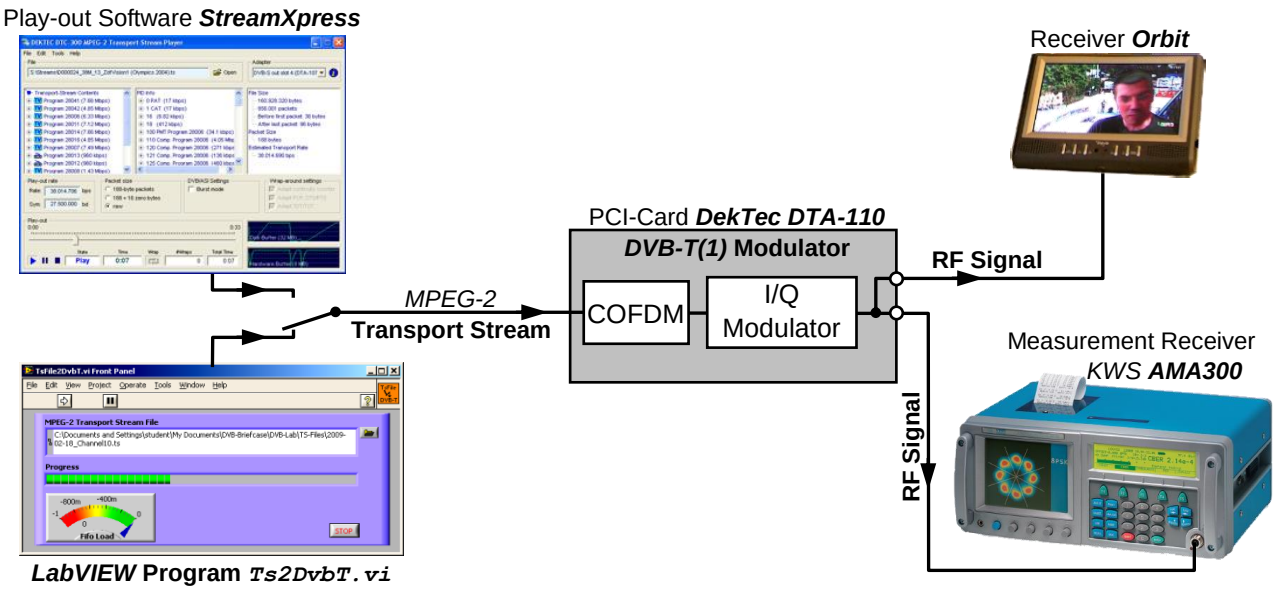


Figure 7-2: Block diagram of the *MPEG-2* exercise – two MPEG-2 transmitters and two *DVB-T(1)* receivers

7.2.2 Programmatic Control of the Modulator *DekTec DTA-110*

For the modulator card *DTA-110*, an API (*Application Programming Interface*) is provided, which allows C++ code to control the modulator and send data to it. This (so-called DTAPI) library contains around 130 methods organized in eight C++ classes. A description of all methods and their usage can be found in the University's *Online Community*. Figure 7-3 shows an example of such a description, for the method `DtOutpChannel::Write()`, which is used to send a block of MPEG-2 data to the card:

DtOutpChannel::Write

Write data bytes to the output channel.

```
DTAPI_RESULT DtOutpChannel::Write (
  [in] char*  pBuffer,        // Pointer to data to be written to hardware
  [in] int    NumBytesToWrite // Number of bytes to be written to hardware
);
```

Parameters

pBuffer

Pointer to the buffer containing the data to be written to the output channel. The pointer must be aligned to a 32-bit word boundary.

NumBytesToWrite

Number of bytes to be written to the output channel. The buffer size must be positive and a multiple of four.

Figure 7-3: The API method `DtOutpChannel::Write()`, see the University's Online Community

In order to simplify the use of the DTAPI, all C++ methods needed to transmit DVB-T(1) signals, have been adopted by *LabVIEW* VIs (*Virtual Instruments*). Hence, the entire functionality of the modulator can be accessed by graphical, data-driven programming. Figure 7-4 shows the LV version of `Write()`:

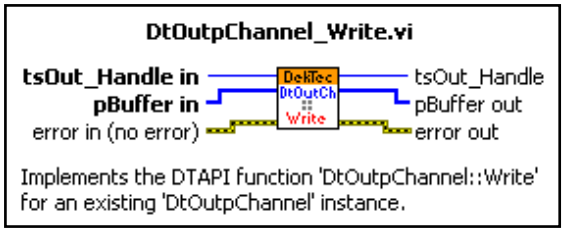


Figure 7-4: The associated *LabVIEW* VI, which adopts the original API method `DtOutpChannel::Write`

Below, the Block Diagram of the *LabVIEW* program `Ts2DvbT.vi` is shown. It is able to read a Transport Stream from a file, and transmit it in form of a DVB-T(1) signal. In order to manipulate the MPEG-2 packets and their contents, they can be dissected, before they are passed to the `Write` function:

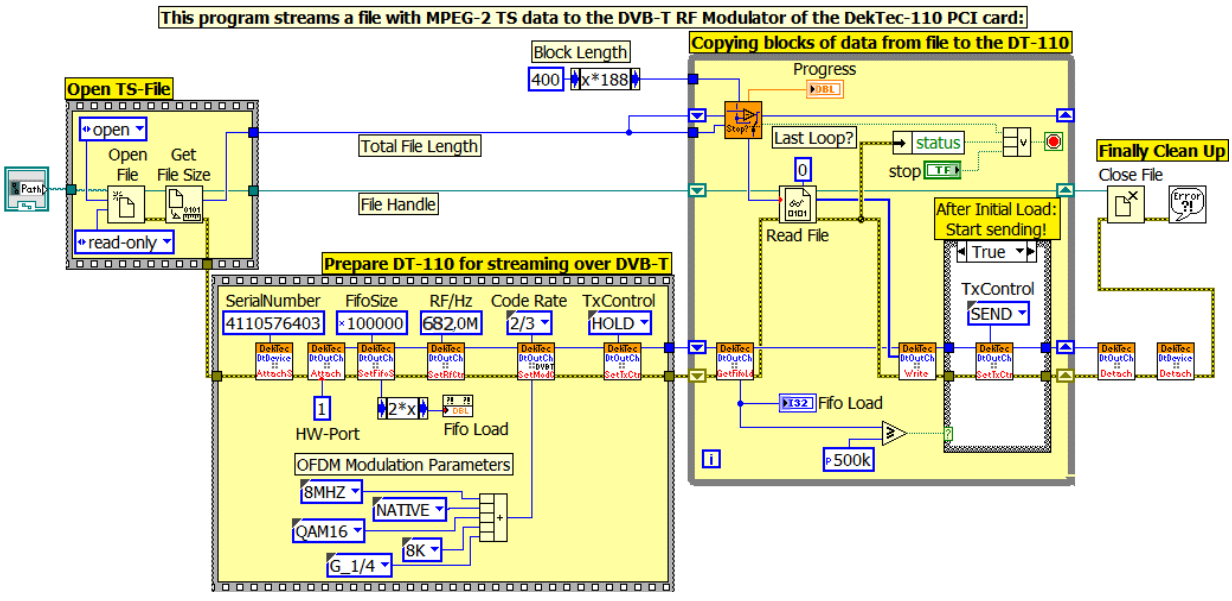


Figure 7-5: *LabVIEW* program `Ts2DvbT.vi` – *Block Diagram*

7.2.3 Hierarchical Structure of the MPEG-2 Transport Stream

The Transport Stream bears the audio & video PES (*Packetized Elementary Stream*), and supplementary information in a hierarchical structure, which can be seen as a "micro protocol stack":

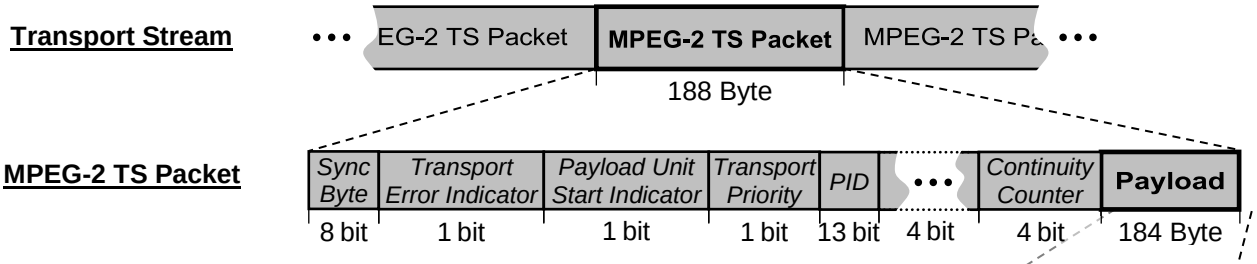


Figure 7-6: Structure of an MPEG-2 Transport Stream Packet

Additional information supplements the audio & video PES:

⇒ PSI (*Program Specific Information*) tables – necessary for demultiplexing the PIDs, and

⇒ SI (*Service Information*) tables – support and simplification of the receiver operation.

An SI example is the SDT (*Service Descriptor Table*), which contains names of the services that can be found in that TS. This information is not necessary for operation, but it helps the user to find the programs and configure his receiver. In this exercise, we will analyse and manipulate exactly those SDTs:

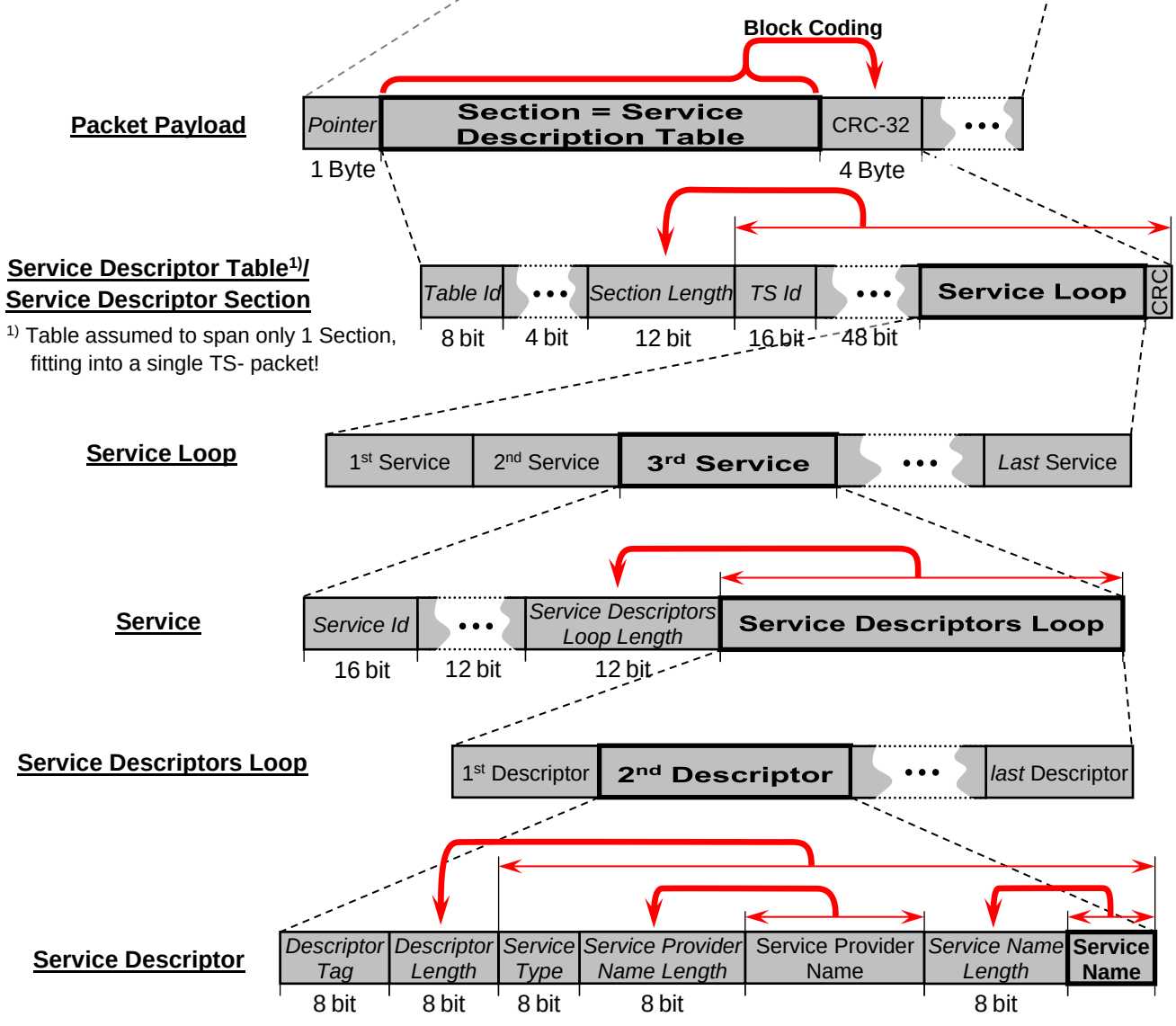


Figure 7-7: Simplified hierarchical structure of an SDT inside the payload of an MPEG-2 TS packet

7.3 Preparation Problems (to be answered before doing the Lab!)

a) Study the Block Diagram of the *LabVIEW* program *Ts2DvbT.vi*:

⇒ What are the reasons for using *Error Clusters* as so-called *Flow-Through* parameters?
Hint: Ask those of your colleagues, which are already *LabVIEW* experts!

⇒ On which TV channel and with which *DVB-T(1)* parameters will the program transmit?

⇒ Why is the transmitter put to TxControl HOLD, until the filling of the FIFO exceeds a load of 500 kB?

⇒ Which are the three conditions that stop the execution of the loop?

1.:	**2.:**	**3.:**

⇒ How can the user recognize that the program runs fast enough to keep up with the modulator?

⇒ Which place in the program would be appropriate to observe/manipulate the TS?

b) By which factor is the physical layer data rate increased, if the *CR* is changed from 2/3 to 5/6 ?

c) Complete the diagram of Figure 7-6 by entering…

⇒ … the value of the *Sync Byte* in the *TS Packet*, and the size of the *PID* in the *TS Packet*.

⇒ What for is the bit *Payload Unit Start Indicator*?

d) Find out the purpose, the PID, and the typical repetition rate of the following PSI and SI tables:

Table Type	**Purpose**	**PID**	**Rep. Rate**
PAT (= *PSI*) (*Program Association Table*)			
NIT (= *SI*) (*Network Information Table*)			
SDT (= *SI*) (*Service Descriptor Table*)			

e) Complete Figure 7-7, showing the structure of the SDT (*Service Descriptor Table*):

⇒ Enter the two possible values for the '*Table Id'*.

⇒ Enter the number of bytes in the *Service Loop*, depending on the '*Section Length*' value.

⇒ Over how many bytes is the CRC-32 calculated, depending on the '*Section Length*' value?

f) Implementation of a CRC (*Cyclic Redundancy Check*) generator in *LabVIEW*

The *Service Descriptor Table* is terminated by a CRC-32. Below you find an example for a simple CRC generator consisting of only three LFSRs (*Linear Feedback Shift Registers*):

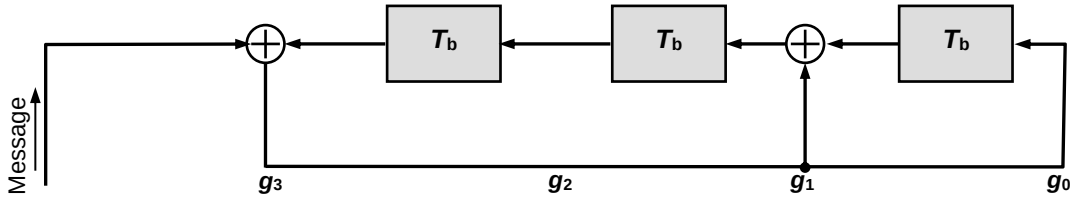


Figure 7-8: Simple Example for a CRC Generator with LFSRs

⇒ What is the generator polynomial $G(u)$? **G(u) =**

In the following figure, a *LabVIEW* program is shown that implements this simple CRC generator. The shift registers are represented by a *32-bit unsigned integer* which can be shifted and XORed:

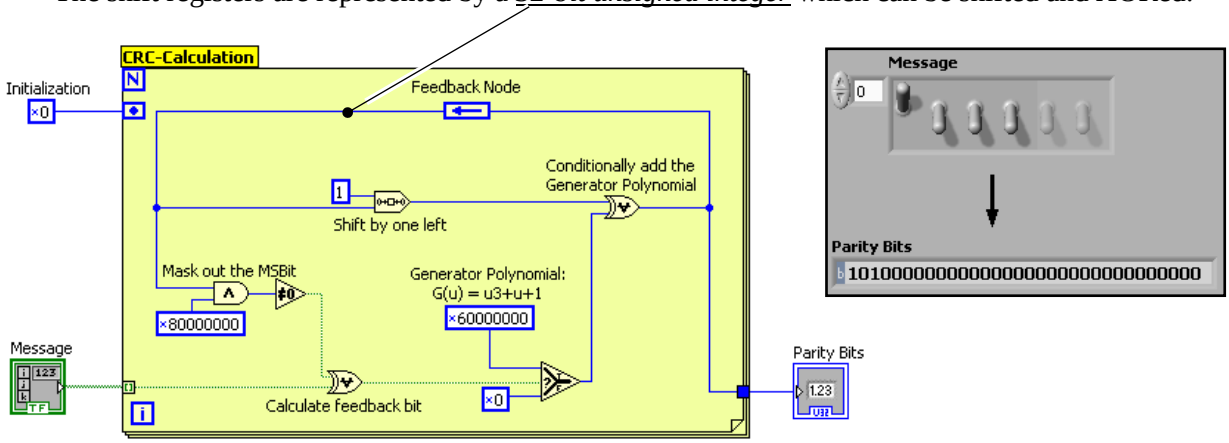


Figure 7-9: *LabVIEW* implementation of the simple CRC shown in Figure 7-8: *Block Diagram* left, *Front Panel* right

⇒ How many parity bits will be generated? What are these bits for the example message '1000'?

The number '0x60 00 00 00' specifies the generator polynomial.

⇒ Why are only two bits set in the number, although the polynomial has three coefficients ≠ '0'?

⇒ Look into the Internet (e.g. *Wikipedia*): Find out the definition of the CRC-32 generator polynomial G(*u*), and the *32-bit unsigned integer* number that represents the code in the *LabVIEW* program (binary and hexadecimal):

Generator P. *G*(*u*) =
⇒ Binary number =
⇒ Hex Number =

⇒ CRC-32 requires the shift registers to be initialized by '1'es? How to do this in the code above?

7.4 Practical Part

Preparation: Open the directory **`Lab_R1.16`**, on the **`Desktop`** of your computer. Here, enter the **`zip`**-archive **`WiCS`**, and copy the directory **`MPEG-2`** directly to your **`Desktop`**. All files, which you will use and manipulate for this exercise should be contained in this directory.

Experiment 1 : Getting Started with the Transmit Software *StreamXpress* and *LabVIEW* (30 min)

a) **MPEG-2 analysis and manipulation** with the software *StreamXpress*:

⇒ Start the program *StreamXpress* and replay the stream file **`Channel35.ts`**. Switch on the *Orbit* receiver, which should display the program.

⇒ Which services are contained in this TS? Fill in the missing data into the table below:

String	1st Service	2nd Service	3rd Service	4th Service
Name of the Service				
PID of Video PES				
PID(s) of Audio PES				

⇒ What is the total data rate of this stream (*Data Rate TS*) / of the baseband ("*Data Rate Out*")?

Data rate TS =	*Data rate Out* =

⇒ Change the *CR* parameter to *CR* = 5/6 and compare the changes of the data rate to your results of preparation question 7.3b):

Data rate TS =	*Data rate Out* =	Factor *Out*/*TS* =

⇒ Find out the PID of *Null Packets*:

PID =

In the following, you will analyse and manipulate the stream with a *LabVIEW* program. You can find that in the directory **`MPEG-2\LabVIEW`** on the **`Desktop`** of your computer.

b) Run the ***LabVIEW* stream transmitter `Ts2DvbT.vi`**:

⇒ The stream should be replayed correctly by the receivers.

⇒ Now, all bytes shall pass the *SubVI* **`Dissect_RawTS.vi`**, before they are written into the modulator. Insert this *Sub VI* appropriately into **`Ts2DvbT.vi`** !

Note for the rest of this exercise:
All files **`Dissect_*.vi`** can be found in the sub-directory **`LV_Mpeg2-Dissector`**.

⇒ Run the stream transmitter again. Is the *Sync* Byte present with its correct value?

Experiment 2 : Explore the PIDs (*Packet Identifiers*) in the Stream (30 min)

`Dissect_RawTS.vi` calls **`Dissect_TsPacket.vi`**, and passes the TS data 'packet-by-packet'. Open **`Dissect_TsPacket.vi`**:

a) Let's analyse the PIDs of the TS packets:

⇒ Find out, how often the PAT (*Program Association Table* with PID "0x0") is repeated in the TS. Hint: Enable the call to `Beep.vi` in the *Case* of PAT packets.

⇒ What is the approximate repetition rate for the NIT (*Network Information Table*) and the SDT (*Service Descriptor Table*)? Compare your results to the preparation 7.3d).

***Rep. rate PAT*:**	***Rep. Rate NIT*:**	***Rep. Rate SDT*:**

b) For the next experiment, we need a definition of a ***Null Packet***.

⇒ Add a *Case* for this PID and set a *Breakpoint* inside that *Case Structure*.

⇒ After the program has hit the *Breakpoint*, copy the control '*TS Packet (188 Bytes) in*' and change it into a constant, in order to be used in the next experiment. Verify that it comprises exactly 188 Bytes!

Experiment 3 : Stuffing the Transport Stream with *Null Packets* (30 min)

Now we **change the data rate of the Transport Stream** by using the recorded *Null Packet* constant:

⇒ In **`Ts2DvbT.vi`**, change the *CR* to '5/6', and verify the expected artefacts on the *Orbit* TV!

⇒ How many *Null Packets* have to be added per iteration, to match the baseband data rate?

⇒ In **`Dissect_RawTS.vi`**, add the necessary number of *Null Packets* at the output. Verify the expected result on the *Orbit* TV!
Hint: Use the disabled 5×5 copier as a sample pattern, how to append multiple packets!

Experiment 4 : Analysis and Manipulation of the SDT (*Service Descriptor Table*) (40 min)

a) Now, you will **analyse SDT packets** with the **`Dissect_Service*.vi`** *VIs*:

⇒ Familiarize yourself with the three **`Dissect_Service*.vi`** *VIs* and compare them to the hierarchical structure of the SDT in Figure 7-7.

⇒ In **`Dissect_TsPacket.vi`**, add a new *Case*, in order to catch SDT packets.

⇒ Strip the *Pointer* (see Figure 7-7) off the SDT packet payload, and pass it to the input of the *SubVI* **`Dissect_ServiceDescriptorTable.vi`**. The manipulated packet payload, which is returned at the output of this *SubVI*, shall be joined together with the *Pointer* again.

Hint: Use the functions `Split 1D Array` to strip the *Pointer*, and `Build Array` to join it!

⇒ Run the transmitter and check the *Table ID* in the *Front Panel* of **`Dissect_ServiceDescriptorTable.vi`**: Do we catch only SDTs or maybe BATs as well?

***Table ID* =**

⇒ What is the size of the SDT section? Does it fit into a single TS packet, including the CRC?

***Section Length* =**	***Total Length* incl. Pointer & CRC =**

⇒ In order to analyse the services, open the *SubVI* **`Dissect_ServiceLoop.vi`**. Then run **`Dissect_ServiceDescriptorTable.vi`** directly with its previous input data.

⇒ In the *Block Diagram* of **`Dissect_ServiceDescriptorsLoop.vi`**, enable the *Delay for Slow Motion*. Then run **`Dissect_ServiceLoop.vi`** directly with its previous input data.

⇒ How many Services can be found in this TS?

⇒ Find out the names of the services and their respective provider:

	1st Service	2nd Service	3rd Service	4th Service
Service Provider Name				
Service Name				

⇒ How many *Service Descriptors* exist per *Service*?

b) **Change `Dissect_ServiceDescriptorsLoop.vi`** in a way that it replaces the *Names* of the 1st, 2nd, 3rd *Service* by the names of your team members!

⇒ Run **`Dissect_ServiceDescriptorTable.vi`** directly and check in "slow motion", if your manipulations take effect.

⇒ Now disable the *Delay for Slow Motion* and run the transmitter against the *Orbit* TV receiver. Why does the receiver not show the new *Service Names*?

⇒ Connect the *KWS AMA300* receiver to the second RF output of the DVB-T(1) modulator. Does it decode the new *Service Names* correctly? What is the problem?

Experiment 5 : Generation of a CRC-32 (*Cyclic Redundancy Check*) (20 min)

a) **Familiarize yourself with the *VI* `Dissect_UpdateCRC.vi`**:

⇒ Run the *VI* directly and verify that it produces the same result as in Figure 7-9.

⇒ Change the generator polynomial and the initialization value in order to calculate a CRC-32!

b) **Implementation and test of the CRC-32**:

⇒ Integrate **`Dissect_UpdateCRC.vi`** into **`Dissect_ServiceDescriptorTable.vi`**. Hint: The CRC must be calculated over the right *Section Length* (i.e. without the *Pointer*!)

⇒ Temporarily comment out (`Diagram Disable`) the modification of the *Service Names* in **`Dissect_ServiceDescriptorsLoop.vi`**, and run **`Dissect_ServiceDescriptorTable.vi`** directly. Check, if the last 4 bytes of the *SDT Section Payload* are the same for input/output!

⇒ Run the transmitter against the *KWS AMA300* and the *Orbit* receiver. Does everything work?

Cleanup: Please delete the directory **`MPEG-2`** from the **`Desktop`** of your computer.