Methods for interference mitigation in wireless networks

Metody pro snížení rušení
v bezdrátových sítích

Ing. Tomáš Dulík

Doctoral degree program: Engineering Informatics
Supervisor: doc. RNDr. Vojtěch Křesálek, CSc.

Doctoral thesis
2012

Tomas Bata University in Zlín
Faculty of Applied Informatics

ABSTRACT

In this dissertation thesis, the co-location interferences in the most popular wireless networks today – the 802.11 WLAN – are explored.

The main topic of this thesis are the interferences caused by operation of several co-located radio devices on the adjacent frequency channels. The interference properties of DSSS and OFDM baseband modulations are discussed, as well as the impact of the RF transmit/receive chain. For evaluating the interferences in controlled experiments, RF emulation testbed was designed and built and several example measurements were done.

In the final part of the dissertation, methods for interference mitigation in wireless networks are proposed and their feasibility discussed.

Keywords: DSSS, OFDM, Cross Technology Interference.

ABSTRAKT

Tato disertační práce se zabývá rušením u dnes nejpopulárnějších bezdrátových sítí standardu WLAN 802.11.
Hlavním tématem tezí je vzájemné rušení několika zařízení umístěných v jedné (ko)lokaci, které pracují na sousedních frekvenčních kanálech.

Práce diskutuje vlastnosti modulací DSSS a OFDM vzhledem ke vzájemnému rušení v základním pásmu, stejně jako vliv vysokofrekvenční části vysílacího/přijímacího řetězce. Pro praktické ověření metod, potlačujících vzájemné rušení, byla navržena a sestavena vysokofrekvenční měřící a testovací síť, na které bylo posléze provedeno několik příkladů měření reálných zařízení.

Závěrečná část práce obsahuje popis několika metod pro omezení vzájemného rušení včetně analýzy jejich proveditelnosti.

Klíčová slova: DSSS, OFDM, interference, vzájemné rušení různých systémů, odolnost proti rušení.

ACKNOWLEDGEMENTS

I want to dedicate this work to our Lord who did not let me down when all the infernal powers disturbed and troubled me. I had to finish this thesis in the same time when my two grant projects had to be finalized. During the same period, we also had to submit 5 new project proposals. Most of these months, I was absolutely overwhelmed and exposed to all kinds of problems that normally do not happen, including the whole university network outages happening just in the time of my remote measurements.

I want to thank my beloved family for deallocating me from all their plans and activities until the last sentence of this work was finished.

I also thank to:

- my supervisor Vojtěch Křesálek for many fresh and inspiring ideas of how this work could be exploited in future projects. The sense of doing something potentially useful was the best motivation I could get.
- The head of our department Roman Jašek and all the other faculty colleagues, who tried to heal my atrophy of academic ambitions by pushing me constantly up to the state where there was no way out but to sit down and write this until the end.
- Petr Dvořák and Peter Janků for helping me with the testbed mechanical construction.
- All the bus drivers from ČSAD Vsetín for their smooth driving, because substantial parts of this text were written on the bus line between Zlín and my home town.

This project was supported by the CESNET grant 351/2009 and the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

Printed version of this thesis is available at the library of Tomas Bata University in Zlín, nám. T. G. Masaryka 5555, 760 01 Zlín, Czech Republic.

Motto:

“*If you want to make God laugh, tell him about your plans.*”
Woody Allen's paraphrase [121] of an old Yiddish proverb [124].

TABLE OF CONTENT

1 INTRODUCTION

This work was motivated by a real world problem we have discovered while working on BSc. and MSc. projects described in [33], [40], [55], [71] and [108]: when carefully monitoring the wireless networks we have built, we have often encountered high packet loss not corresponding to any foreign source of interference. We have tried to analyse the possible reasons of this phenomenon in [17] and [111] only to find that the worst interference sources were our own co-located radios running on adjacent and sometimes even alternate channels.

When experimenting with the various settings of the WLAN devices, we have found that the DSSS devices are quite immune against nearby transmissions in adjacent channels, while OFDM stations become non-functional even in adjacent channel power levels much lower than those routinely present in our deployments.

Although the adjacent/alternate channel protection and rejection parameters values are clearly defined in the 802.11 standards, it is impossible to find any WLAN product which would declare the actual values of these parameters in various operation modes. Moreover, it is not possible to find any WLAN product which would declare values of ALL parameters defined in the 802.11 standards.

As I wanted to find the right answers and right solutions to these problems, I decided to take the following steps:

1. Build a specialized RF testbed for WLAN measurements, which would allow measuring and testing the behaviour of real radio devices.
2. Study the interference properties of DSSS and OFDM technologies and analyse them both experimentally and theoretically.
3. Based on the findings, propose or analyze some methods for increasing the interference immunity of co-located wireless devices:
    1. “Know your enemy” (and know yourself).
    2. Coordinated Dynamic Frequency Selection (DFS) / Channel Agility and Transmitter Power Control (TPC).
    3. Active noise cancellation.
    4. Improving the ACPR at the transmitter.
    5. Distributed OFDM symbol-level synchronization.

2 STATE OF THE ART

During the last decade, the digital baseband modulation technologies became ubiquitous in all areas of human living. Many of these technologies must share very limited resource – the license-free frequency bands. Although most digital communication standards can handle presence of well-defined interference types in the channel, it is impossible to specify and solve all possible modes of coexistence between different technologies.

The mutual interference between different wireless technologies is sometimes called "*Cross Technology Interference*" (CTI) and it is probably the most frequent source of problems in the current wireless world.

Surprisingly, the area of CTI between DSSS and OFDM for the specific case of 802.11 standards is totally unexplored. As 802.11 WLAN is the most popular wireless standard today, this is a nice research challenge. This thesis will try to fill this gap by exploring the interference between DSSS and/or OFDM and also try find methods for the interference mitigation. Some results of this work could be useful also outside the 802.11 domain, because OFDM is used in DVB, DSL, WiMax (802.16e), PAN UWB (802.15.3a), digital radio, 3GPP LTE downlink, etc., while DSSS is used in WLAN 802.11b, ZigBee (802.15.4), GPS, CDMA, cordless phones and many proprietary (non-standardized) modulation schemes.

2.1 Literature review

2.1.1 RF testbeds

As the design of RF testbeds is more practical engineering than scientific problem, few publications target this topic. A nice, but older survey of academic WLAN testbeds was published in [63].

There are lots of testbed projects like ORBIT [86] or EWANT [96], which use a free space medium. However, this approach is unsuitable for exact, controlled measurements we need to carry out.

The most inspiring paper for this work was [11], which, aside of detailed description of the testbed itself, provided references to other similar testbed projects, like the MeshTest [22]. Compared to these two projects, our budget was severely limited and therefore it was not possible to implement a full switching matrix allowing interconnections with arbitrary attenuation. Although it would be nice to have, most of the experiments are doable with the limited configuration built for this thesis.

2.1.2 Cross-device and cross-technology interference and its mitigation

If Google is to be taken as a metric tool for a term popularity, then “Cross Technology Interference” seems to be quite unpopular topic.

The most recent research paper [66] targets interferences from Wi-Fi to ZigBee networks in the 2.4 GHz band by enhancing the ZigBee MAC layer. The paper contains a lot of useful references, but none of them targets physical layer problems.

In these thesis, we want to analyse the CTI more deeply, starting from the physical layer. Here, we must submerge in the signal processing theory behind the baseband modulation methods - OFDM and DSSS and also behind the radio frequency domain.

2.2 Commercial products with interference mitigation solutions

2.2.1 GPS-synchronized frame-level TDM

It seems that the only external interference mitigation method used in commercial 802.11 products today is GPS-synchronized frame-level time division multiplex (TDM).

One of the first wireless networking products using GPS-based synchronization was Motorola Canopy and its cluster management module [75].

Ubiquiti Networks, Inc. recently came out with frame-level GPS-synchronized TDM in their WLAN 802.11n product called Rocket M5 GPS [89]. The product uses GPS as precise time reference and UDP/IP for coordination and signalling. Some details about its inner workings are described in [88].

WiMax 802.16 uses GPS synchronization for separating the upstream a downstream traffic – see [93] for an overview.

2.2.2 Noise cancellation techniques

I could not find any relevant publications on the topic of cross-device RF noise cancellation in any application domain. All publications and patents I found refer only to inner-device noise cancellation - e.g., Atheros, one of the primary manufacturers of the WLAN chipsets, has the “Ambient noise immunity” system – see [97] and [112], but it cancels only the noise which is generated by the chip itself.

2.3 Patents

There are many patents covering the area of OFDM interference mitigation, e.g.:

- US Patent 7813701 [109] proposes a simple principle: when the system is idle, the receiver builds statistics of the its channel frequencies affected by interferences. Then the system allocates the data to be transmitted to the OFDM sub-carriers currently being affected by the lowest levels of interference while allocating no data to the sub-carriers with the highest interference levels.

  Review: This idea is simple enough to be functional and it could help in the case of adjacent channel interference, which can damage narrow bands on the side of the spectrum. The cost is throughput decreased just by the rate of unusable sub-carrier bands.

- US Patent 7630290 [36] presents a system which combines the principles of DSSS with OFDM. The data from the input are spread by an orthogonal Hadamard code sequence and the result is fed into the OFDM modulator.

  Review: the cost of this idea is obvious – compared to the "pure" OFDM, the system throughput is lowered by factor of N, where N is the length of the Hadamard code.

- US Patent 7242720 [110] proposes using a MIMO configuration of multiple transmitting and multiple receiving antennas, which help dealing with the multipath fading effects.

  Review: This patent was filled in 2002 and if I understand the description correctly, it was completely outdated by the 802.11n standard whose specification started in 2004 and was released in 2007.

- US Patent 6487253 [59] is dealing with OFDM channel estimation. The classic approach for the channel estimation by using few pilot tones (which is also used by 802.11 OFDM preambles) can be crippled by interferences that would make the estimation partially or completely wrong. Thus, the authors propose using not only pilot tones, but also a complete OFDM training symbol, which would eliminate the vulnerability of the few pilot tones.

  Review: this patent was filled in 1999, the same year when 802.11a standard was released, which defines preambles with both the pilot tones and the training sequence.

3 THESIS OBJECTIVES

The objectives of this thesis are:

1. Build a specialized RF testbed for WLAN measurements, which will allow measuring and testing the behaviour of the real radio devices.
2. Analyse the dark and chaotic space of cross-device and/or cross-technology interferences. In this part, an analyse of the OFDM and DSSS transmit/receive chains will be presented.
3. Propose new or analyze existing methods for interference mitigation usable within DSSS and OFDM transceivers. There is no single "heal the world" solution and therefore, this work will concentrate on the most severe interference scenario of several radios co-located densely on a radio tower.
   The methods discussed in these theses are:
   1. "Know your enemy and know yourself": this method is based on knowing the interferences properties and knowing the parameters of the radio devices. Then, a set of configuration adjustments can be done to mitigate the interferences.
   2. Coordinated Dynamic Frequency Selection (DFS) and Transmitter Power Control (TPC): this is an extensions to the DFS and TPC procedures defined by the 802.11 standard, which could be implemented by coordinating the frequencies and transmit power levels between the co-located radio devices.
   3. Active noise cancellation: the active noise cancellation is widely used in headphones products, where the method profits from low sampling rates and cheap DSP chips. For local cross-device radio interferences, an active noise cancellation method using RF phase shifter and programmable RF attenuator is described.
   4. Improving the ACPR at the transmitter – several methods for baseband and RF band are discussed and analyzed.
   5. Distributed OFDM symbol-level synchronization.

4 WLAN 802.11: THE BASIC PRINCIPLES

Before introducing the core of the dissertation, the basic principles of the the baseband modulations used in 802.11 standard – the DSSS (Direct Sequence Spread Spectrum) and OFDM (Orthogonal Frequency Division Multiplex) – need to to be presented.

Side note: this whole chapter should be skipped by experienced researchers and developers. I considered its inclusion but finally left it here because this thesis should also serve as an application note and textbook for the RF emulation testbed. I believe that the theory, principles and rationale behind the 802.11 DSSS and OFDM must be understood by the students working with the testbed, therefore I wanted to put together very short and graphical, yet complete explanation of these modulation methods and their interferences.

4.1 Spread spectrum: the basic theory

The spread spectrum rationale is based on the Shannon-Hartley theorem:

$$C = B \log_2\left(1 + \frac{S}{N}\right) \qquad (1)$$

Where

- C is the channel capacity in bits per second,
- B is the bandwidth of the channel in hertz,
- S is the total received signal power over the bandwidth in watt or volt^2,
- N is the total noise or interference power over the bandwidth in watt or volt^2.

This single equation is the corner stone of the whole telecommunication industry, because it clearly and simply defines the maximum theoretical channel capacity in presence of noise.

As we can see, the best way to achieve higher channel capacity is to increase the channel bandwidth, because increasing the signal to noise ration (SNR) yields only logarithmic growth of the capacity. Moreover, we can not beef up the signal power arbitrary, because the maximum power levels for a given frequency band are defined by government regulatory bodies. And we can not remove the noise from channel. So in fact, there are only two variables we can operate on effectively: the channel frequency bandwidth and the channel capacity.

But this does not mean we should resign on fighting the infernal interfering noise. In the area of interference mitigation, the Shannon-Hartley theorem brings another important implication:

besides extending the bandwidth to achieve higher capacity, we can choose to increase bandwidth while keeping the capacity same. This inflicts reduction of the SNR required to reach the capacity on the given channel. The SNR can be expressed by "reversing" the Shannon-Hartley equation:

$$\frac{S}{N}=2^{\frac{C}{B}}-1 \tag{2}$$

Example:

Let us suppose we have a cheap spectrum band which is wide enough to accommodate more than our communication data rate. A realistic example:

In 802.11b, the channel width is 22MHz, and the basic data rate is only 1 MHz.

Using (2), we calculate the SNR as:

$$\frac{S}{N}=2^{\frac{1}{22}}-1=0.032$$

This means that the energy of the channel noise can be theoretically 1/0.032 = 31.2 times greater than the energy of the signal.

This foretells the spread spectrum principle: we can achieve greater noise immunity with slower data rates if we can use greater frequency bandwidth.

The spread spectrum *processing gain* is then defined as:

$$N=\frac{B_{SS}}{B} \tag{3}$$

Where B_{ss} is the bandwidth of the signal after spectrum was spread and B is the original bandwidth.

However, the most important note comes at the end: Shannon-Hartley theorem was constructed for ideal channels with white noise. In real world, we do not fight the white noise, our interferences are both narrow-band and wide band with various time/frequency characteristics.

4.2 Spreading spectrum of rectangular signals

How to spread the spectrum after all? There are several methods, even 802.11 standard specifies two of them: "Frequency Hopping Spread Spectrum" (FHSS) and "Direct Sequence Spread Spectrum" (DSSS). Because FHSS does not allow data rates over 2MBit/s, all current WLAN products in 2.4GHz support only the DSSS and therefore we will analyze this method here.

The DSSS uses the most easiest method to broaden spectrum of a rectangular impulse – which is the contracting (shortening) the rectangular data impulse length. The explanation of this principle is worth of the following page.

Let us take an example of periodic rectangular signal as shown on the picture below:

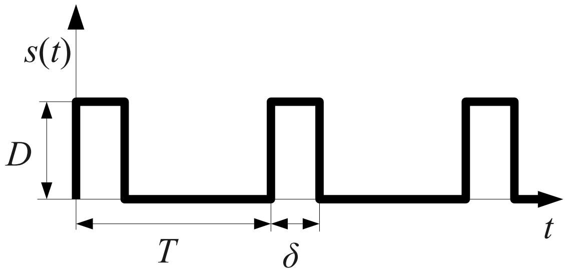


Figure 1: Periodic rectangular signal

The k-th Fourier coefficient of such signal can be inferred[1] as:

$$c_k = D.\frac{\delta}{T}.sinc\left(\frac{\delta}{2}k\,\omega\right) \tag{4}$$

Where *sinc*(..) function is defined as

$$sinc(x) = \begin{cases} \frac{\sin(x)}{x}, & x \neq 0 \\ 1, & x = 0 \end{cases} \tag{5}$$

These equations define the exact shape of the signal amplitude spectrum, because we can easily find the null points in the spectrum: $c_k = 0$ if $\frac{\delta}{2}\omega = \pi$. Thus $\omega = \frac{2\pi}{\delta}$ and so $2\pi f = \frac{2\pi}{\delta}$ therefore $f = \frac{1}{\delta}$.

As for the amplitudes, the amplitude of *k*-th harmonic is $2|c_k|$, which is

$$|U(\omega)| = 2|c_k| = 2\mathrm{D}\frac{\delta}{T}\left|sinc\left(\frac{\delta}{2}k\,\omega\right)\right|$$

Now we can draw the amplitude spectrum:

[1] For complete inference, see example 1.11. on page 16 in [12]

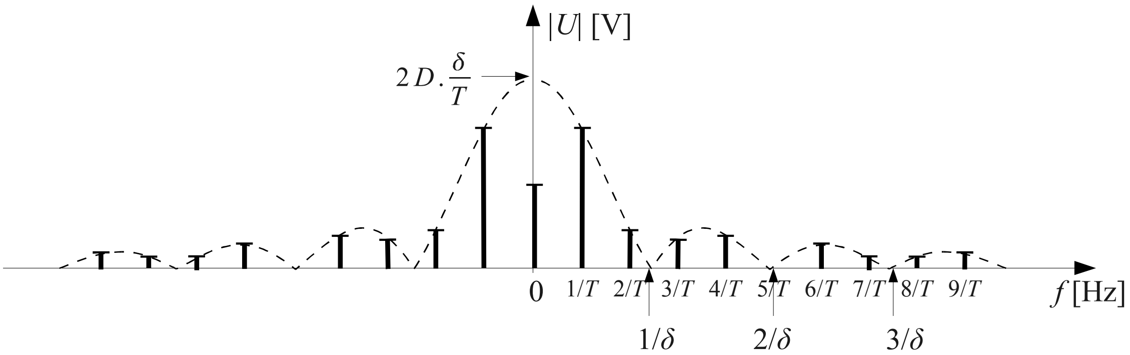


Figure 2: Amplitude spectrum of periodic rectangular signal

Note:

- the *sinc*() function side lobes decay down to infinite frequencies. We obviously do not want to transmit signal with infinite spectrum, so we will try to filter the sides lobes as much as possible and keep only the main spectrum lobe.
- the spectrum is composed of discrete points whose distance is 1/T from each other. That also means that non-periodic rectangular impulse has a continuous spectrum because its T=∞.

Now if we shorten the impulse length *δ*, then the null points 1/*δ, 2/δ,* ... will grow proportionally. This means that the spectrum will expand, exactly as we intended.

If we shorten the impulse duration, what about the period?

Lets consider the situation, when we only shorten impulse 3 times but keep the same period. As we can see on figure 3, the amplitude of the pulses was reduced 3 times, but we now have more discrete points in the main spectral lobe.

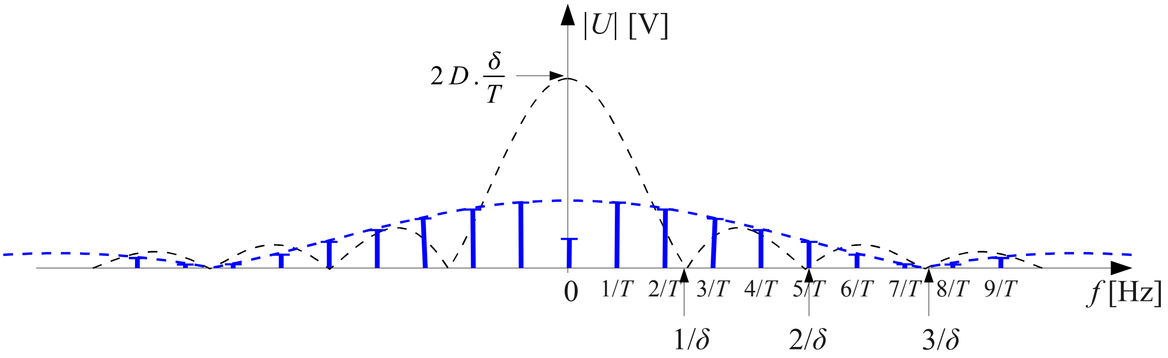


Figure 3: Amplitude spectrum of periodic rectangular signal with pulse width=1/3 δ

But if we shorten the period to 1/3 T, we will get only 3 discrete points in the main lobe, as shown on figure 4.

Thus, when implementing a spread spectrum modulation, we must consider both the impulse duration and signal period, because we have to consider the possible interferences in both time

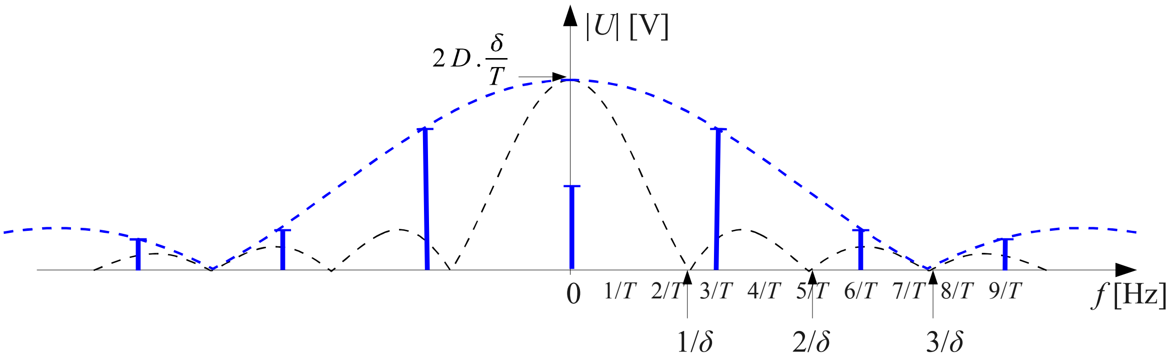


Figure 4: Spectrum of periodic rectangular signal with pulse width=1/3 δ and period=1/3 T

and frequency domains:

1. with a long period we get more spectral components in the main lobe and thus better immunity against narrow band interferences, but the immunity in the time domain will be weak: any interference impulse similar to our transmitted impulse will damage our transmission.
2. With shortened period, we get fewer spectrum harmonic components and thus weaker immunity against narrow band interferences.

It might seem that whatever possibility we choose, it will not work well for both interference types. Fortunately, our options for processing the communication signals are not limited to time and/or frequency domains. In any communication system, the signals are just carriers of information we want to transmit. Therefore, for solving the problem mentioned above, we can use methods from the theory of codes.

4.3 Direct Sequence Spread Spectrum and its spreading codes

As mentioned above, for spreading the spectrum, we must shorten the transmitted (data) impulses. However, at the receiver site, we need to be able to distinguish these shortened impulses from similar interfering impulses, and we must consider not only interference caused by other transmitters, but also our own signals delayed by multipath propagation effects.

The solution for this problem is not to use a single impulse, but a defined sequence of impulses. This sequence is called "*chipping sequence*" (because it "chops" the data bits into smaller chips) or "*pseudo-noise sequence*" or "*pseudo-noise code*" (PN code).

For fighting the interference caused by multipath propagation, the chipping sequence should have good autocorrelation property. This means that correlation of the sequence with its delayed versions must be zero or close to zero.With such a sequence, the receiver can detect the proper signals by using a correlator, which eliminates the foreign signals and noise. The correlator in fact spreads the narrow-band interfering signals so they look like noise. A (bit naïve) visualization of this operation is shown on the figure below.

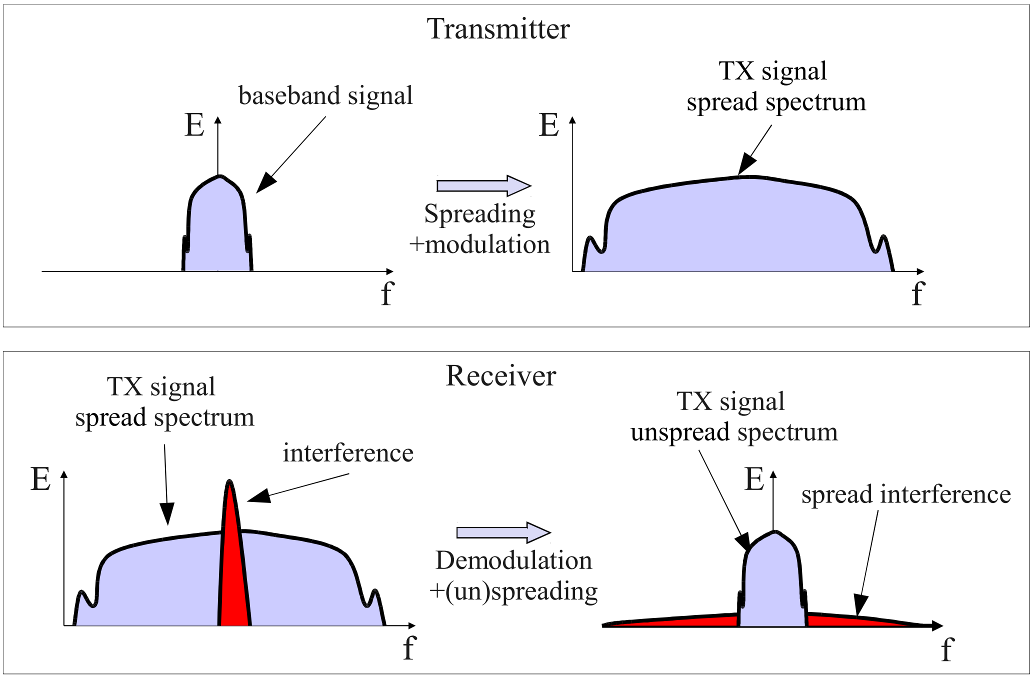


Figure 5: Spread spectrum and interference

Transmitter DSSS implementation is even easier – the coder just replaces the original (slow) data bits with the chosen chipping sequence.

The last question to answer is how to find the magic chipping sequence?

There are many codes suitable for this purpose, e.g. Kasami codes, Gold codes, Walsh-Hadamard sequences, Maximum-length sequences (m-sequences) [61] etc. The selection depends on the application requirements.

For WLAN, the 802.11 committee had the following requirements [4]:

- the code must be short for achieving good throughput with 22 MHz channel,
- the code must be long enough for achieving good multipath performance,
- the code division multiple access is not required.

After a lot of tests and experiments (see [4] for an example), the 802.11 committee decided to use the Barker code with 11 chips.

Other (military, enterprise telecommunications) applications have different requirements, e.g.:

- securing the communication against eavesdropping and securing the receiver against radio jamming (intentional interference made artificially in order to disrupt the communication). To achieve this, the chipping sequences are generated by pseudorandom generators, whose generating polynomial and initial register values are known only by the participating parties.

- achieving huge processing gain for long-haul links (e.g., GPS). For this purpose, very long sequences are used. E.g., GPS uses 8190 or 10230 chips sequence.
- allowing concurrent transmissions of multiple radios in a single frequency band (Code Division Multiple Access – CDMA). The concurrent access to the common media brings a new requirement: multiple sequences are needed and their cross-correlation signal must have minimal energy very close to zero.

In this work, we concentrate only on WLAN networks and thus we will not pursue the other spread spectrum technologies.

4.4 DSSS in 802.11

For the reasons mentioned above, the 802.11 standard selected single 11 bit Barker code [1,0,1,1,0,1,1,1,0,0,0] as the chipping sequence. In the transmitter, the chipping sequence is simply XORed with the data bits as shown on the figure below:

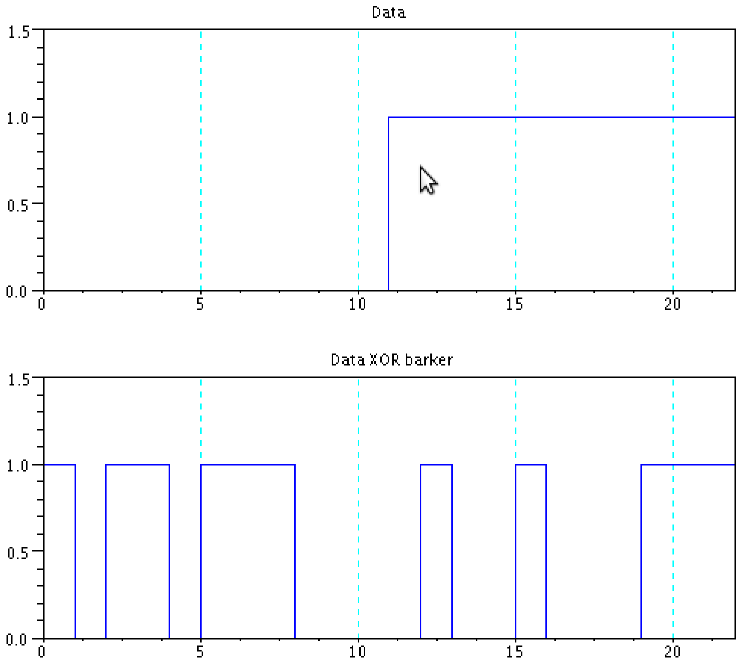


Figure 6: Barker code in 802.11

The 11-bit barker code has good autocorrelation properties – this is shown on the next figure:

This figure was obtained in Scilab by simply calling the autocorrelation function on the (oversampled) Barker sequence, but as discussed in [4], this is not enough for proper judging of the autocorrelation properties: in the real world, each symbol is preceded and followed by other symbols and according to 802.11, the consecutive symbols should be mutually phase-rotated.

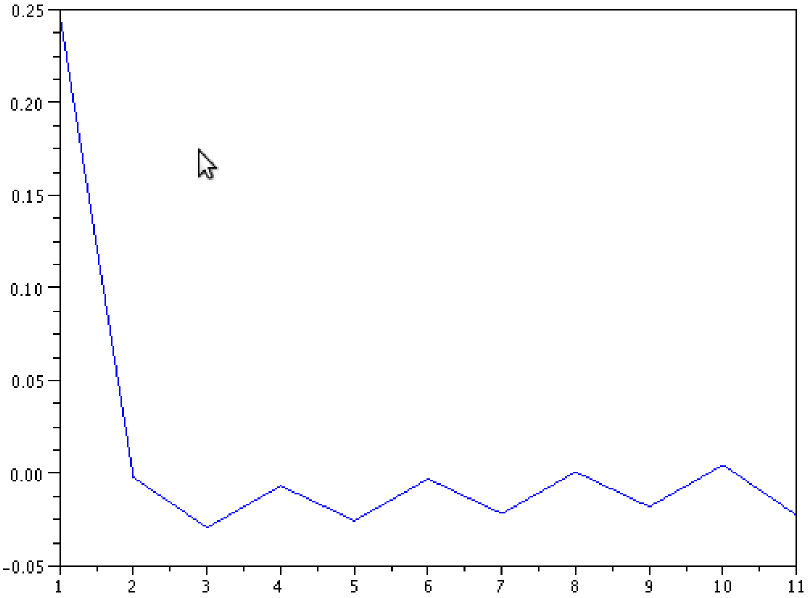


Figure 7: 802.11 Barker code autocorrelation

Therefore, the autocorrelation properties of the chipping sequence were thoroughly simulated and also measured by the 802.11 committee participants on real-world implementation with good results.

In the receiver, the demodulated I/Q signals are sampled by A/D converters and correlated with the Barker sequence. The correlator can be found on figure 4 in [126]. A typical correlator product is shown on figure 12 in [49].

4.5 802.11 OFDM

The OFDM theory is explained in many textbooks and tutorials, but many of them are very inaccurate when they try to explain the orthogonality. Therefore, an explanation from other point of view will be provided in this chapter.

The basics of OFDM are the same as with single-carrier modulations. The difference is that OFDM is modulating multiple carriers at once. The figure 8 shows example of amplitude and phase spectrum of 5 carriers modulated by QPSK.

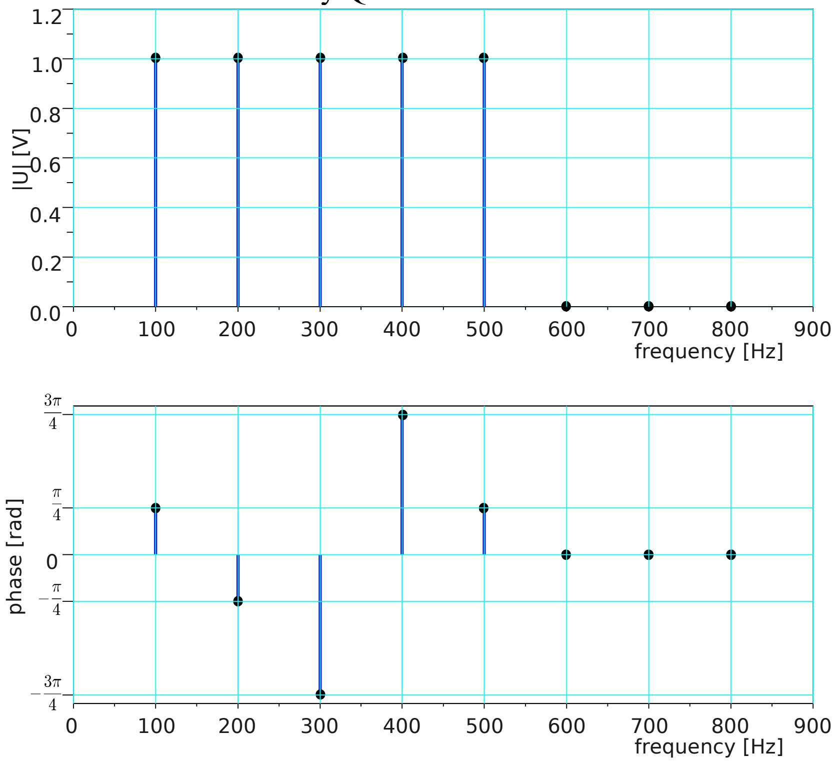


Figure 8: OFDM: 5 carriers modulated with QPSK

As we can see, the frequency of each carrier is an integer multiple of the first carrier frequency 100 Hz.

The modulated carriers in the time domain are on figure 9. From this time domain representation, we clearly see the rule for OFDM symbol duration: it must be the same or longer than the first carrier period.

$$\delta_{symbol} \geq T_{1\text{st}\,carrier} \tag{6}$$

This is because the 1st carrier period is the longest and we need to transmit it whole. Typically, we do not need to make the symbol duration much longer than this - the longer the symbol period is, the lower is the symbol frequency and throughput. We do not have to fear of problems with first carrier detection at receiver – the side carriers are often not used for data transmission anyway, because they are adversely affected by the channel filtering.

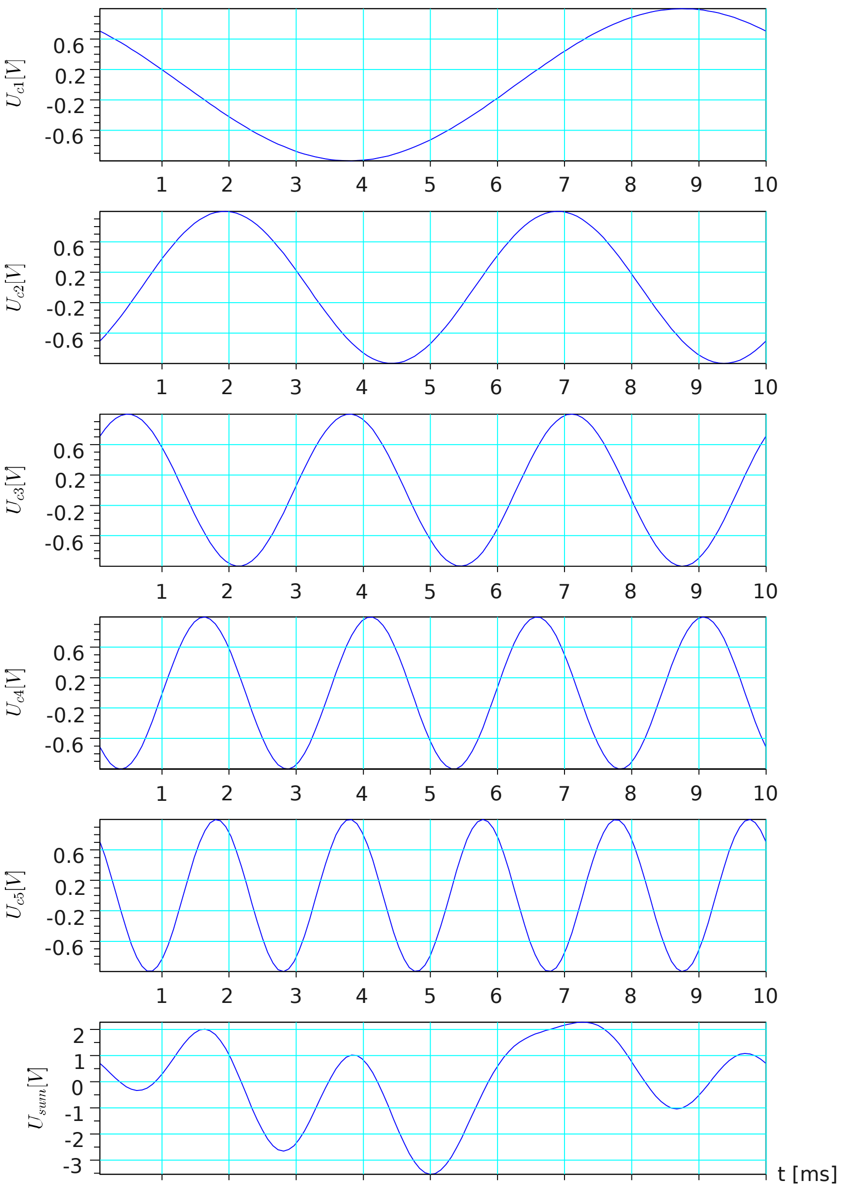


Figure 9: OFDM subcarriers in the time domain

Because the frequencies of the other carriers are integer multiples of the 1st carrier frequency, it is clear that if condition (6) is fulfilled, then all carriers will have (at least) an integer number of cycles in each OFDM symbol and thus the OFDM symbol will be periodic itself. The OFDM symbol periodicity is shown on the last sub-plot of figure 9.

Therefore, if we analyse the spectrum of an OFDM symbol captured in its exact boundaries, we get the perfect spectrum with discrete frequency points shown on figure 8.

However, in the real world communication link, it is impossible to implement ideal (pure) sampling, ideal non-drifting sampling clocks and ideal symbol synchronization.

Therefore, a real receiver would have to analyse symbols containing sections of previous or following symbols. How does OFDM cope with this problem?

First, we should look at consecutive OFDM symbols in the time domain. On the figure 10, there are two consecutive 5-carrier QPSK-modulated OFDM symbols:

$$\left[\frac{\pi}{4},-\frac{\pi}{4},-\frac{3\pi}{4},-\frac{3\pi}{4},\frac{\pi}{4}\right] \text{ and } \left[\frac{\pi}{4},-\frac{3\pi}{4},-\frac{\pi}{4},\frac{3\pi}{4},\frac{\pi}{4}\right] .$$

We can see a sharp spike in their join point (*t*=10 ms).

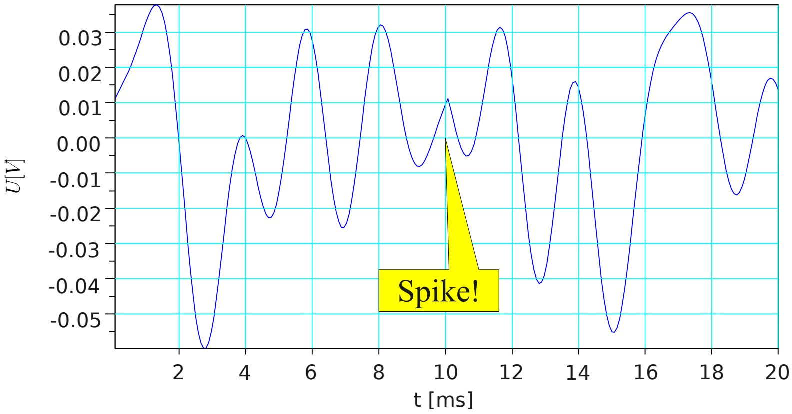


Figure 10: Two OFDM-QPSK symbols

The spike is source of unwanted harmonics and the question is – will these harmonics interfere with some or all of the OFDM subcarriers?

At first sight, the analysis of this problems looks very complex. But it is not: because an OFDM symbol is a linear composition of single harmonics carriers, we can do the analyse with a single OFDM carrier. The result should then hold for multiple carriers as well.

To make the things even simpler, we will modulate the carrier by BPSK. The result for 3 single-carrier OFDM symbols is on the figure 11. As we can see, the OFDM signal on that picture is in fact result of simple cosine multiplied by the square signal below.

Multiplication of complex harmonic signal (carrier frequency) with another signal is described in 4.6.1 subchapter "I/Q modulator as complex multiplier or complex spectrum shifter". The result of this operation is the signal spectrum shifted to the frequency of the carrier – see equation (25).

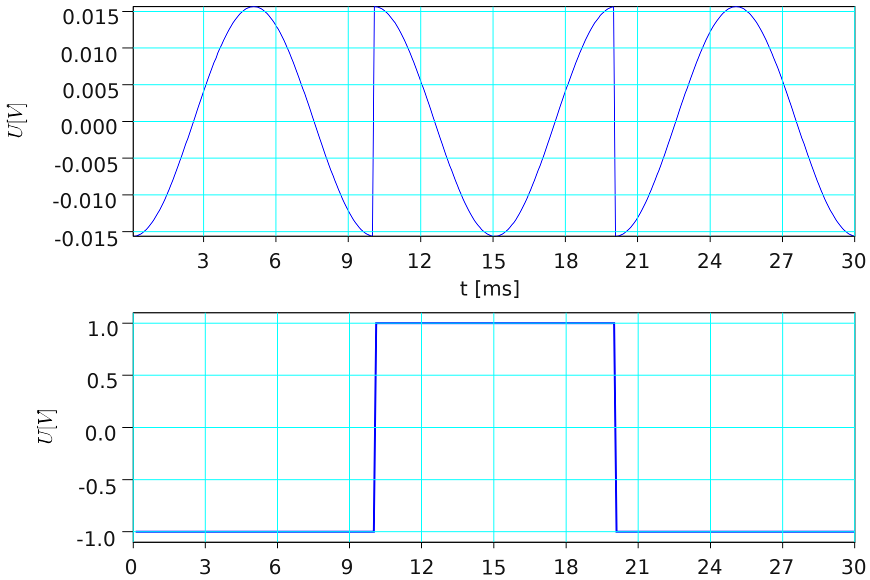


Figure 11: Single carrier OFDM-BPSK with 3 symbols.
Square signal below could be used to generate it by multiplying with simple cos(..).

We already know the spectrum of square periodic signal – it is described in chapter 4.2 “Spreading spectrum of rectangular signals“. The spectrum envelope has the shape of the *sinc*(..) function with null points spaced by $1/\delta_{\text{symbol}}$ and is composed of discrete pulses with spacing 1/T, where T is period of the square (symbol) signal.

We can generalize these facts for the case of 2 symbols (T=2 δ_{symbol}) in each of the 3 OFDM carriers and draw their **individual** spectra:

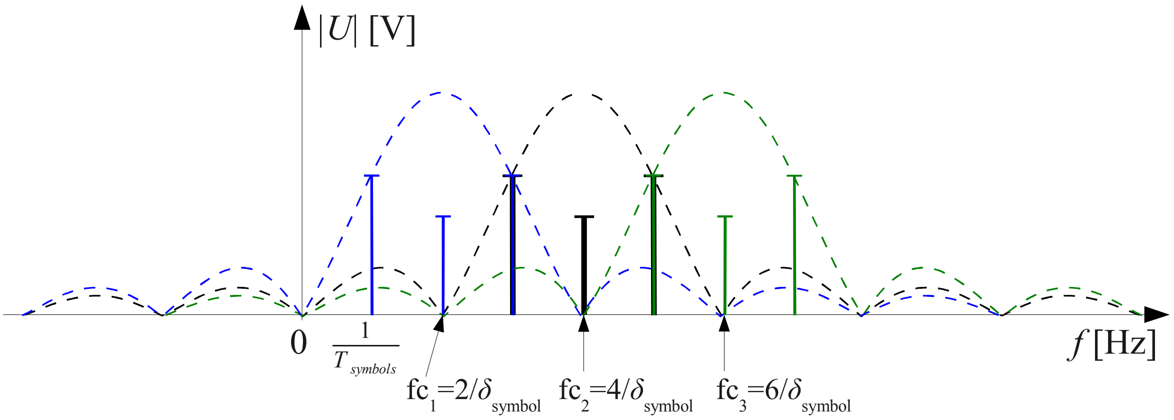

|U| [V]
0
$\frac{1}{T_{symbols}}$
f [Hz]
$fc_1=2/\delta_{\text{symbol}}$
$fc_2=4/\delta_{\text{symbol}}$
$fc_3=6/\delta_{\text{symbol}}$

Figure 12: Individual main spectral lobes of 2 OFDM symbols (T= 2 δ_{symbol}) and 3 carriers

Note: the figure is not complete – only the discrete spectral pulses in the main lobes are drawn, the pulses in side lobes are omitted in an attempt to keep the picture simple.

In the picture we can see that the only non-interfering spectral components are the original DC components of the square signals. All the other spectral pulses will receive interferences from other carriers. So the orthogonality is broken now!

To illustrate this situation more completely, a Matlab simulation was made with the following example: let us have an OFDM symbol with 5 subcarriers, whose distance is 10.f$_{\text{carrier1}}$ (the greater distance was chosen just for the purpose of better visualization). Each of the carriers is modulated by the same 4 consecutive BPSK symbols [0, π, 0, π]. The individual spectra of these 5 distant carriers holding 4 symbols is drawn by different colours into figure 13.

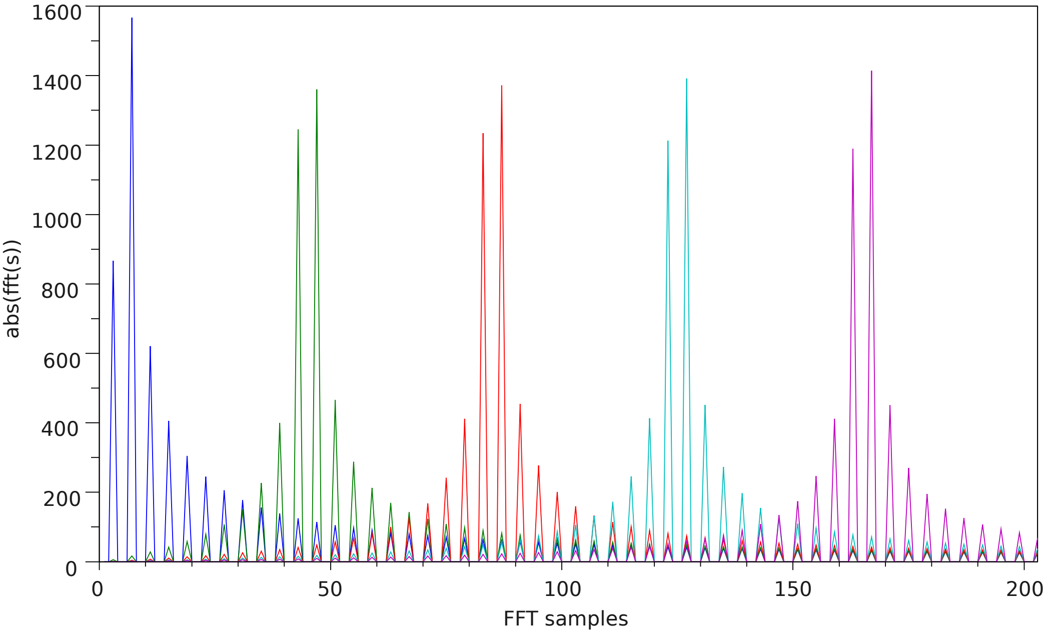


Figure 13: Inter-carrier interference for $T_{symbols}$= 2 $\delta_{symbol.}$
The inter-carrier spacing is 10 $f_{carrier1}$

Note: the graphical representation used on this picture looks like the spectral pulses are not really discrete, but it is the default Matlab plotting algorithms which is nice here because we can better see the contributions of the individual sub-carriers to the mutual interference.

The proper way for displaying the discrete spectral components would be the one below:

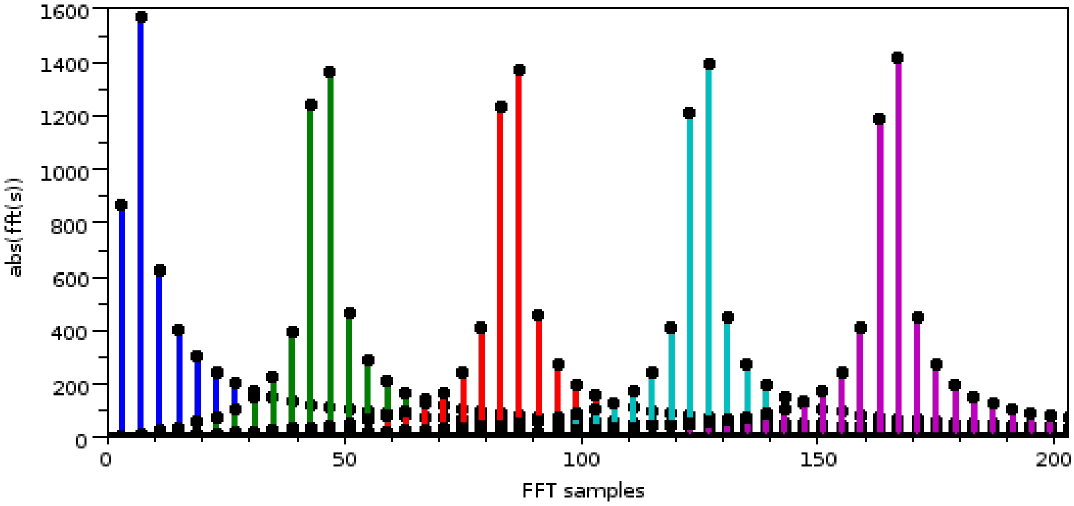


Figure 14: Re-styled figure 13 to emphasize the discreteness of spectral components

As we can see, the (square) symbol signals have no DC components, so there are no spectral impulses at the carrier frequencies.

4.5.1 Introducing the cyclic prefix

To deal with the multi-path propagation channels, which are typically found inside buildings, it is necessary to extend the symbol duration by appending a guard interval. The guard interval duration must be longer than the time difference between the least and the most delayed signal arrivals, but short enough not to decrease the symbol rate. Generally, the symbol duration can be expressed as:

$$\delta_{symbol} = \frac{T_{symbols}}{2} + t_{guard} \tag{7}$$

In 802.11a/g, the guard interval is 800 ns and symbol duration is 3200 ns, so the period $T_{symbols}$ is 4000 ns. The construction of the cyclic prefix is shown on figure 15.

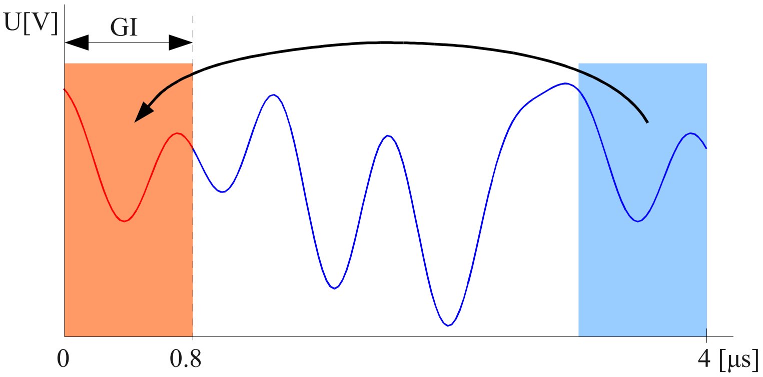


Figure 15: Construction of cyclic prefix inside the GI

When such symbols are transmitted in a packet, as shown on figure 16, the sharp transitions between symbols are present similarly to figure 10. However, as the guard interval is useless for the receiver, the sharp transitions can be smoothed without any risks, see chapter 8.4.1 - Time domain windowing in baseband according to 802.11 standard.

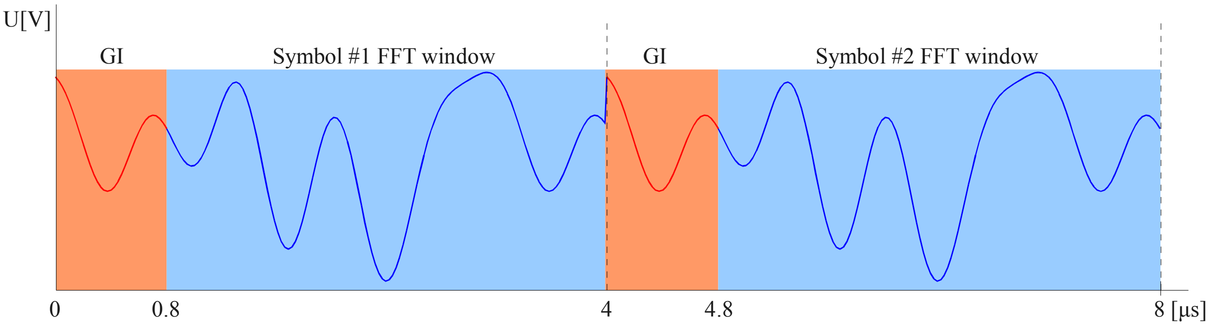


Figure 16: Two OFDM symbols with cyclic prefixes

The spectrum of an unfiltered, ideal discrete OFDM signal with 5 carriers and 802.11a symbol timing is shown of figure 17.

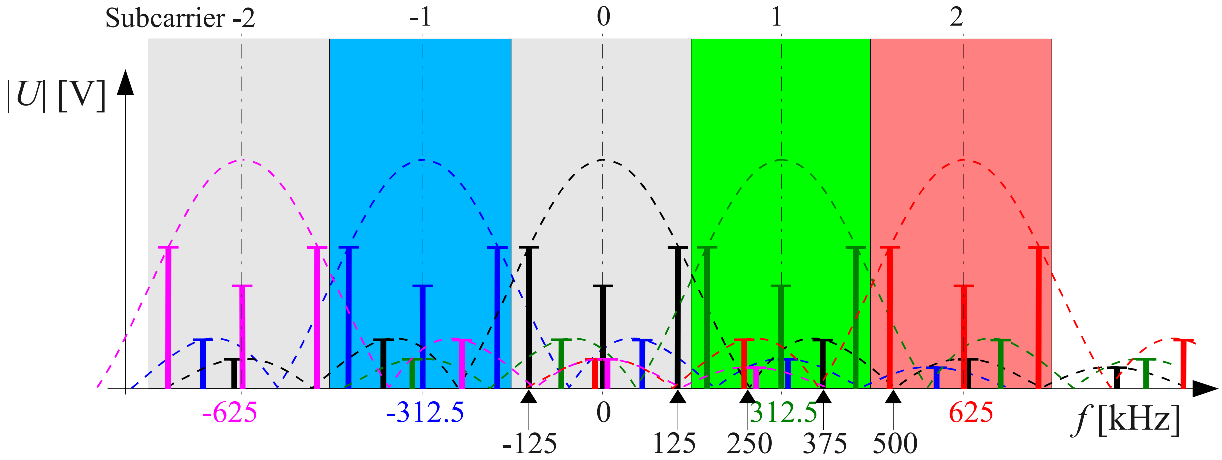


Figure 17: Spectrum of several OFDM symbols with 5-subcarriers and guard interval

Compared to figure 12, the inter-carrier interference is not so strong because the ratio $T_{symbols} / \delta_{symbol}$ is not integer. Therefore, each sub-carrier receives interference from only every 2nd sub-carrier. Moreover, these interferences are lowered by the baseband smoothing windows and DAC output filters.

At last, the figure 18 shows Matlab simulation of multipath propagation with only two signals, one of them delayed by maximum the maximum allowed delay 800ns.

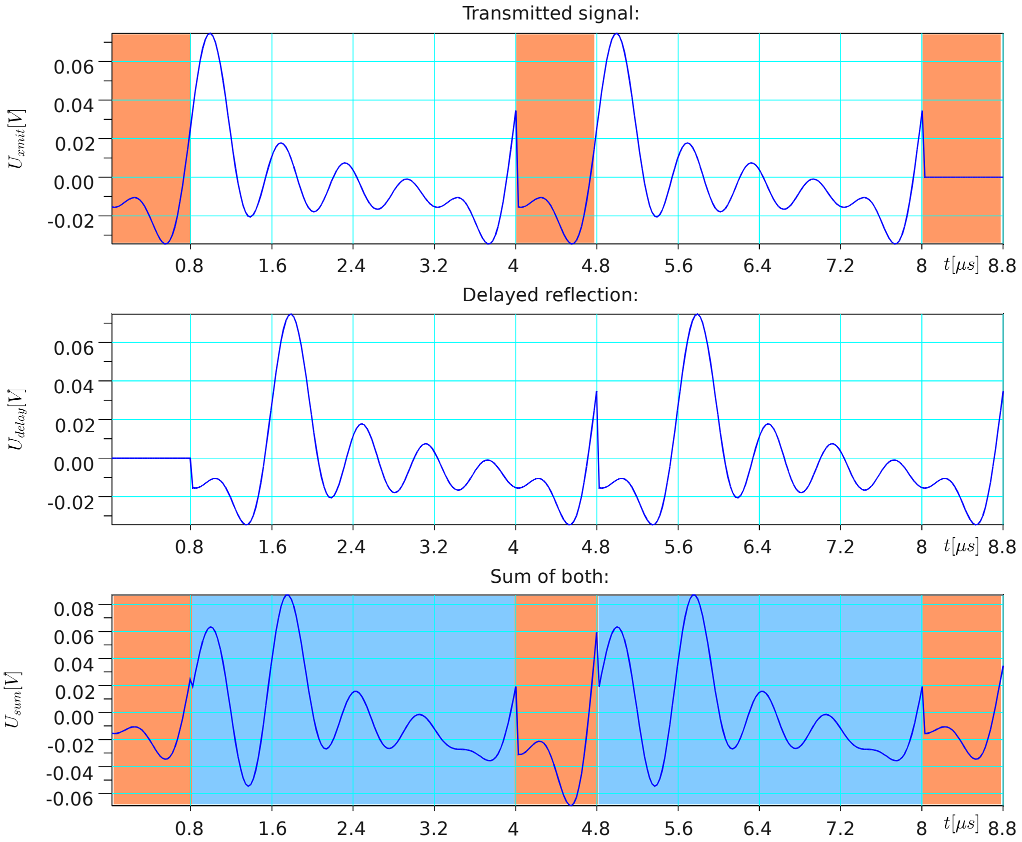


Figure 18: The effect of multipath propagation on signals with cyclic prefix

4.5.2 OFDM and DFT spectral leakage

The most efficient way for decoding and generating the OFDM symbol signals is using the FFT and inverse FFT.

The problem with real-world implementations is that the transmitter and remote receiver local oscillators are not ideally precise and their frequencies can even drift in time.

The 802.11 standard specifies center frequency clock precision which allows frequency offsets as large as almost whole sub-carrier bandwidth (see chapter 4.5.3), therefore it is important to know the effects of frequency shifts between the 802.11 transmitters and receivers.

An OFDM receiver decodes the baseband signal by computing 64-point FFT from I/Q samples buffered from the A/D converters. For this purpose, the symbol start position must be found by a symbol synchronization algorithm [13]. The symbol start position is then used as the FFT window start position.

The influences of FFT windowing on the transform results is covered in great detail in [42] and [44], shorter summary can be found (e.g.) in [16]. The 802.11 standard receiver specifications do not mention FFT window functions at all – in general, the standard leaves most of receiver implementation decisions up to the designers who should consider windows effect on the FFT results.

As noted in [42], "*the window behaves as a filter, gathering contributions for its estimate over its bandwidth*". And this is true for any signal processing using windows, not only the DFT. The window bandwidth is often expressed as the Equivalent Noise Bandwidth (ENBW), which is the bandwidth of a filter with ideal rectangular frequency response which has the same peak power gain. The greater ENBW is, the smaller noise bandwidth the window collects.

Aside of the broadband noise collected by the FFT window, another important windowing effect is the spectral leakage. Spectral leakage is caused by the fact that with windowing, the DFT behaves as bank of filters, each having the same bandwidth and spectral characteristics. For example, rectangular window has a spectrum in the shape of sinc(...) function, which is indefinitely long. Now, if the DFT with rectangular window is used for analyzing a single harmonic tone whose period is not integer multiple of the window length, each DFT bin will receive a contribution from the tone – see the figure 19 which illustrates such a situation for a single DFT bin #1. Although the tone resides in the bin #2, it is not placed at the bin centre frequency where the window spectral images of all the other bins are null, therefore part of the tone energy will be detected by (or will "leak" into) the other bins.

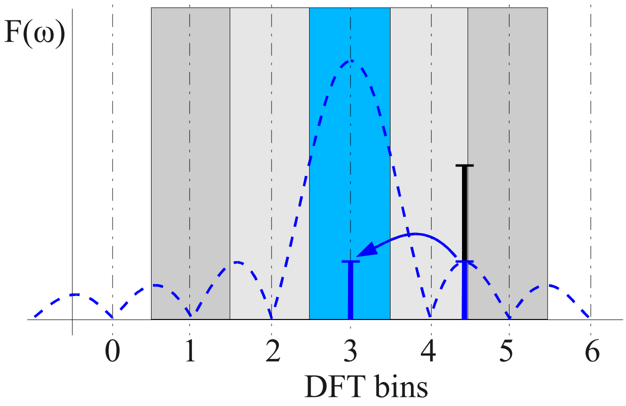


Figure 19: Spectral leakage: DFT of one tone, rectangular window used.

The FFT spectral leakage is the reason why even frequency offset as small as a fraction of one FFT bin must be synchronized very precisely in the receiver, as discussed in the next subchapter.

4.5.3 Frequency offsets and their synchronization

In previous chapters, it was assumed that the signals are sampled and modulated with ideal precise frequencies. In reality, this is not possible.

The 802.11 standard defines maximum frequency offset for the 802.11 basic bit-rates and 802.11b/g modes as ±25ppm (pulse per million = 10^{-6}) for both the center frequency and the chip clock, see clauses 15.4.7.5, 15.4.7.6, 18.4.7.4, 18.4.7.5, 19.4.7.2 and 19.4.7.3.

With 802.11a, the clauses 17.3.9.4 and 17.3.9.5 specify the center frequency and symbol clock frequency offsets as ±20ppm for 20MHz and 10MHz channels and ±10ppm for 5MHz channel.

In 802.11n, clauses 20.3.21.4 and 20.3.21.6 specify the center frequency and symbol clock frequency offsets as ±20ppm for 5GHz bands and ±25ppm for 2.4GHz band.

The range ±10ppm to ±25ppm was obviously chosen with regard to the components price. Such a precision is provided by ordinary “upper-middle class” crystal oscillators, so the more precise temperature-compensated or even expensive oven-controlled oscillators are not needed.

The 802.11standard also specifies that the center frequency and symbol clocks must be derived from the same reference oscillator.

As discussed in [54], the symbol (sample) clock phase noise and frequency variations are not critical for 802.11a, where symbols have only 64 samples and sampling clock with ±20ppm accuracy can drift by $20.10^{-6} * 64 = 0.00128$ samples per symbol. The symbol clock offset can be problem in OFDM systems with large number of subcarriers: for example, the DVB-T in 8k mode has 8192 samples. A sampling clock with ±20ppm accuracy can drift by $20.10^{-6} * 8192 =$

0.16 samples per symbol and if transmitter and receiver oscillators offsets have opposite signs, their maximum offset would be 0.32 samples per symbol. This can become problem with the highest sub-carriers which have only few (e.g., 3) samples per cycle.

However, the center frequency offset (CFO) must be treated and mitigated with greatest care. For example, 802.11a device set to 5.7 GHz channel could have maximum CFO = $20.10^{-6} * 5.7*10^{9}$= 114 kHz. If transmitter and receiver oscillators offsets have opposite signs, their maximum frequency offset would be 228 kHz. This is shown on figure 20: the black dashed lobe depicts the spectral envelope of ideal OFDM sub-carriers without any CFO, the blue and green lobes impersonate subcarriers with the +114 kHz and -114 kHz frequency offsets for transmitter and receiver respectively.

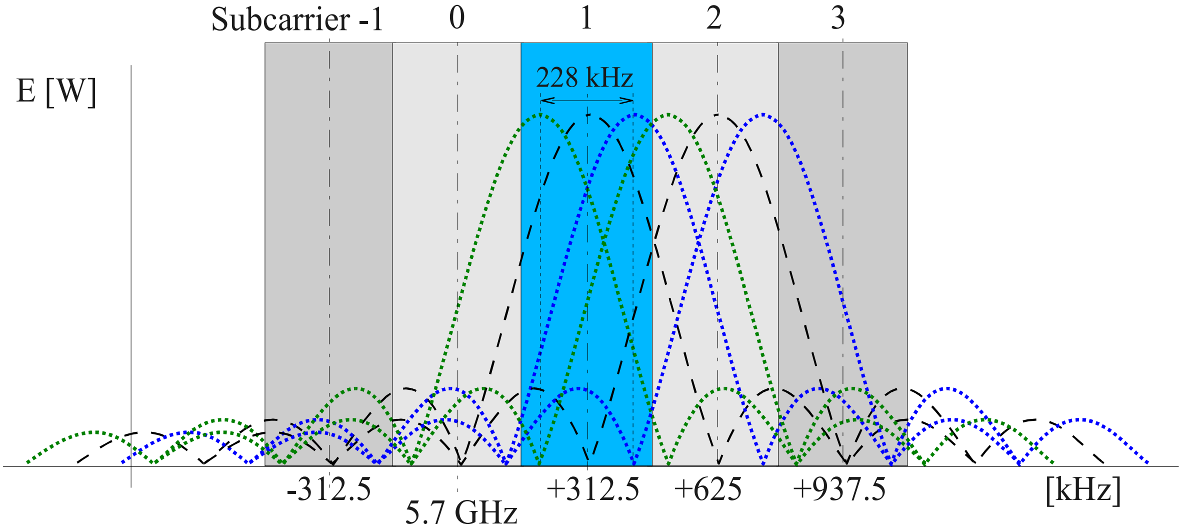


Figure 20: 802.11 OFDM and center frequency offset

Note: the subcarrier #0, which is at the carrier frequency, is always null – see the figure 17-3 in the 802.11 standard clause 17.3.2.5 of figure 17-11 in clause 17.3.5.9.

The figure illustrates how the CFO disturbs the transmission:

- instead of detecting the highest energy levels from the subcarrier main lobe, the receiver detects only the side lobe(s) of the corresponding transmitter's subcarrier.
- The receiver detects most energy from the transmiter's adjacent sub-carrier.

This situation is called inter-carrier interference (ICI) and it must be solved by implementing a CFO synchronization method. There are many efficient methods available for this task and many publications describing them, for e.g. [20] or [37], which includes the analysis of FPGA implementations for multi-standard devices.

4.5.4 The oversimplification of OFDM in textbooks

In the previous sub-chapters, a spectral analysis of OFDM was introduced as the necessary background for chapter 4.7 - Co-location interferences in WLAN networks.

Another reason for writing these sub-chapters was an oversimplification of OFDM, which can be found in many textbooks. There are statements and pictures which are confusing and not comprehensible – a typical example is shown on figure 21.

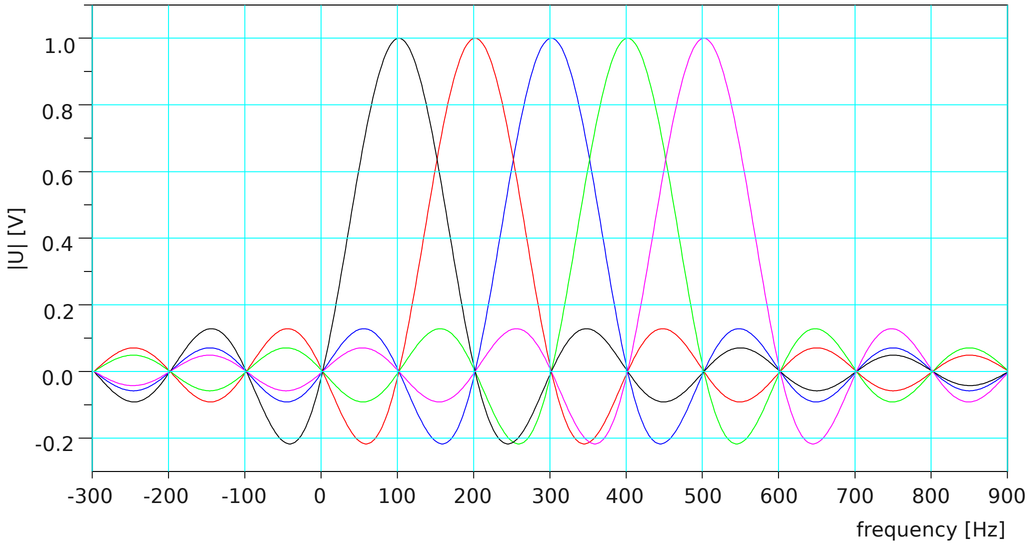


Figure 21: What is wrong with the most popular depiction of OFDM ?

A typical comment with such a picture is e.g. in the tutorial [18], which says:

"*OFDM works because the frequencies of the subcarriers are selected so that at each subcarrier frequency, all other subcarriers do not contribute to overall waveform. In this example, subcarriers are overlapped but do not interfere with each other. Notice that only the peaks of each subcarrier carry data. At the peak of each of the subcarriers, the other two subcarriers have zero amplitude*".

The problem with the comment is oversimplification – it mixes the OFDM and DFT spectral leakage concepts without properly referencing or naming them.

The problem with the picture is its ambiguity. Does it represent the DFT spectral leakage or the spectrum of single OFDM symbol or the spectrum of several packets with random data, integrated over time? If the figure shows spectrum of a single symbol (which means symbol period T=∞, which is unrealistic case not used in any real communication system), then the accompanying explanation is wrong – in such case, all sub-carrier lobes interfere with each other, except their DC components. If it shows a spectrum of many symbols integrated over time, then this should be clearly said in the comment.

4.6 RF part of the transmit/receive chain

It is clear that for the most accurate representation of the WLAN interferences, we need to explore also the RF parts of the transmit/receive chains. We start with the most important part - IQ modulator.

4.6.1 IQ modulation

The IQ modulation is simple to implement and allows producing virtually any kind of modulation. Its principles are introduced as a basic topic in many textbooks. However, the comprehensibility of the reading greatly depends on scientific ambitions of its authors. For the golden rule of keeping things as simple as possible, we recommend to use approach similar to [77]. Here, the basic I/Q modulation principle is derived for the phase and amplitude modulated signal *s*(*t*):

$$s(t)=A\cos(\omega_c t+\phi) \qquad (8)$$

where *A* and *Φ* is the wanted amplitude and phase of the resulting modulated signal.

Because modifying the phase directly and precisely is not easy at all, we should look for an alternative approach and we find it in this high-school-math trigonometric identity [113]:

$$\cos(\alpha+\beta)=\cos(\alpha)\cos(\beta)-\sin(\alpha)\sin(\beta) \qquad (9)$$

By substituting $\alpha=\omega_c t\,,\beta=\phi$ we get this equation:

$$s(t)=A\cos(\omega_c t+\phi)=A\cos(\omega_c t)\cos(\phi)-A\sin(\omega_c t)\sin(\phi) \qquad (10)$$

where

- cos(ω_c *t*) and sin(ω_c *t*) represent pure harmonic carrier signal, whose only parameter ω_c - the radial frequency - is (ideally) constant.
- the terms $A\cos(\phi)$ and $A\sin(\phi)$ contain the amplitude and phase variables that we want to modify.

Based on equations above, we may express the I and Q signals as:

$$\begin{aligned} I&=A\cos(\phi)\\ Q&=A\sin(\phi) \end{aligned} \qquad (11)$$

And the modulated signal as:

$$s(t)=A\cos(\omega_c t+\phi)=I\cos(\omega_c t)-Q\sin(\omega_c t) \qquad (12)$$

Because the cos($\omega_c\, t$) and sin($\omega_c\, t$) functions are mutually orthogonal, they can be considered as base vectors of a (2D) Euler space and the *I*, *Q* values become the coordinates in such a space. The geometric representation of this concept can be seen in figure 22. In fact, the names of *I*, *Q* signals came from here: *I* means "In-phase" and *Q* means "Quadrature".

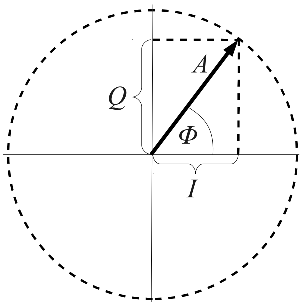


Figure 22: I/Q signals relation to phase and amplitude of the modulated signal

From the equations above, it might seem as if the I and Q signals are not dependent on time, but this is not generally true. I/Q signals can be constant inside a symbol period, but this is only the case with simple phase and amplitude modulation methods.

When used with more advanced modulation schemes like OFDM, I/Q values are varied during whole symbol period. Therefore, they are plain ordinary signals and we can rewrite the I/Q modulation formula as:

$$s(t)=A(t)\cos(\omega_c t+\phi(t))=s_i(t)\cos(\omega_c t)-s_q(t)\sin(\omega_c t) \qquad (13)$$

where $s_i(t)$, $s_q(t)$ are I and Q signals

$$\begin{aligned} s_i(t)&=A(t)\cos(\phi(t)) \\ s_q(t)&=A(t)\sin(\phi(t)) \end{aligned} \qquad (14)$$

whose instant values determine the instant amplitude $A(\delta)$ and phase $\Phi(\delta)$ of the modulated signal:

$$\begin{aligned} A(t)&=\sqrt{(s_i^2+s_q^2)} \\ \phi(t)&=atan(\frac{s_i}{s_q}) \end{aligned} \qquad (15)$$

The block schema of the modulator is just a graphical form of the equation (14):

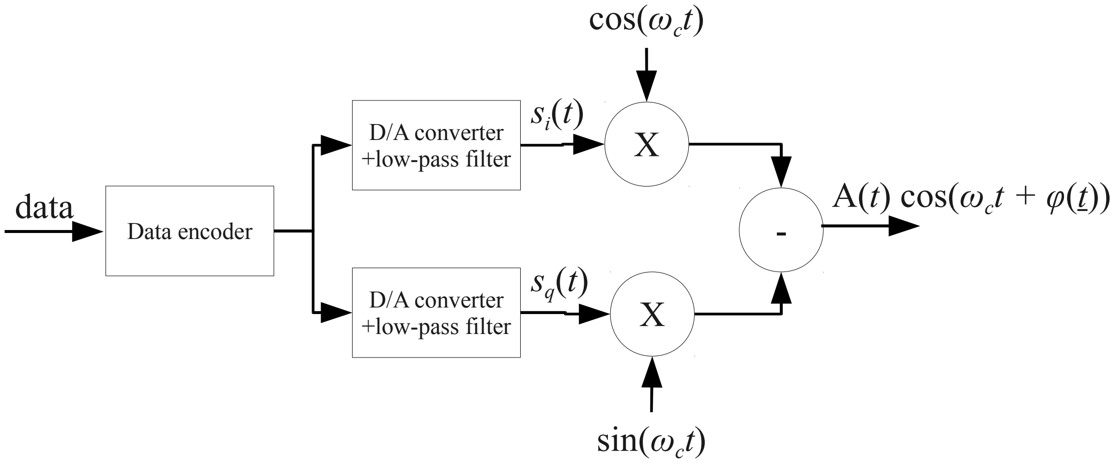


Figure 23: I/Q modulator block schema

I/Q modulation in the complex domain

Leonhard Euler came with the astounding formula[1]:

$$e^{jx}=\cos(x)+j\sin(x) \qquad (16)$$

Which can be easily adjusted to express modulation in complex domain:

$$\hat{s}(t)=A(t)e^{j\omega_c t+\phi(t)}=A(t)\cos(\omega_c t+\phi(t))+jA(t)\sin(\omega_c t+\phi(t)) \qquad (17)$$

For deriving the the I/Q signals, we could use the trigonometric identities as we did in equation (10), but it is much easier to separate phase from the exponent like this:

$$A(t)e^{j\omega_c t+\phi(t)}=A(t)e^{j\omega_c t}e^{j\phi(t)} \qquad (18)$$

Using the Euler formula for unfolding the amplitude and phase term $A(t)\ e^{\ j\Phi(t)}$ we get:

$$\hat{s}_e(t)=A(t)e^{j\phi(t)}=A(t)\cos(\phi(t))+A(t)j\sin(\phi(t)) \qquad (19)$$

Using (14) we can simplify this to:

$$\hat{s}_e(t)=s_i(t)+js_q(t) \qquad (20)$$

where $s_i(t)$, $s_q(t)$ are the real domain I/Q signals, same as in the classic I/Q modulator case.

The complex signal $\hat{s}_e(t)$ is called **complex envelope**.

The complex I/Q modulation equation can then by rewritten as this short formula:

$$\hat{s}(t)=\hat{s}_e(t)e^{j\omega_c t} \qquad (21)$$

[1] The original Euler formula uses *i* instead of *j* We use *j* because it is commonly used in electrical engineering where *i* is reserved for current.

I/Q modulator as complex multiplier or complex spectrum shifter

At the modulator output, we are only interested in the real part of the complex signal. How do we obtain the real output from the complex I/Q modulation?

We can expand the equation (21) using Euler formula and make the multiplication:

$$\begin{aligned}\hat{s}(t)&=\hat{s_e}(t)e^{j\omega_c t}=(s_i(t)+j\,s_q(t))(\cos(\omega_c t)+j\sin(\omega_c t))\\ \hat{s}(t)&=s_i(t)\cos(\omega_c t)+i^2\,s_q(t)\sin(\omega_c t)+\\ &\quad j\,s_i(t)\sin(\omega_c t)+j\,s_q(t)\cos(\omega_c t)\end{aligned} \tag{22}$$

The last line can be simplified because i^2=-1:

$$\begin{aligned}\hat{s}(t)&=s_i(t)\cos(\omega_c t)-s_q(t)\sin(\omega_c t)+\\ &\quad j\left(s_i(t)\sin(\omega_c t)+s_q(t)\cos(\omega_c t)\right)\end{aligned} \tag{23}$$

The real part of this complex signal is therefore

$$s(t)=\Re(\hat{s}(t))=s_i(t)\cos(\omega_c t)-s_q(t)\sin(\omega_c t) \tag{24}$$

Which is the same result as we got in (13) - this is the output of real domain I/Q modulator.

But if the final modulator output is a real function anyway, why did we bother with the complex representation at all?

As can be seen in (21)-(24), the I/Q modulator in fact implements the real part of complex multiplication of two complex signals.

Compared to multiplication of two real harmonic signals in real domain, complex multiplication of two complex harmonic signals is easier and yields a single frequency output – if a complex harmonic tone is sent to I/Q modulator input, then $\hat{s}_e(t)=e^{j\omega t}$, which means:
$s_i(t)$ = cos(ωt), $s_q(t)$ = sin(ωt), the modulator output would be

$$s(t)=\Re(e^{j\omega t}e^{j\omega_c t})=\Re(e^{j(\omega+\omega_c)t}) \tag{25}$$

While the multiplication of real harmonic signals always produces two distinct frequencies:

$$\cos(\omega_c t)\cos(\omega t)=\frac{\cos(\omega_c t-\omega t)+\cos(\omega_c t-\omega t)}{2} \tag{26}$$

Thus, the complex nature of the I/Q modulator gives us an easy way to implement the single side band spectrum shifter or single side band modulation.

4.6.2 Sources of error in I/Q based RF signal processing

A nice introduction to problems in I/Q based RF signal processing is in [107]

The good understanding of the real-world problems in the RF domain is fundamental for implementing any interference mitigation methods. This is because in baseband we can use DSP methods and as far as we have enough processing power and good algorithms, almost anything is possible. However, in the RF domain we must do all the processing in continuous time (analog) domain by electronic components which have their physical limitations that we must know in order to be able to propose methods which would work in the real world.

This chapter will be greatly expanded in the final dissertation thesis. Currently this merely a list of topics with some references.

4.6.3 I/Q Signal Impairments

In the real world, the transceiver components are always worse than ideal. This variances of the components parameters and their non-linearities are the sources of errors which must be counted in any analysis.

The I/Q signal impairments are caused by differences in I and Q signal paths.

The differences can be caused by any of the discrete components included in the path and can be

1. permanent, typically due to tolerances of components parameters. These can be minimized either by careful selection of the components, which is hardly possible in a mass production scenario or by integrating the complete I/Q path on a single chip where the differences are minimized principally.
2. temporal, due to different temperatures in the I/Q paths caused by flawed thermal management of the printed circuit board (PCB). These can minimized by careful thermal management and symmetric placement of the PCB components or by integrating the complete I/O path on a single chip.

The I/O signal impairments are manifesting themselves by

- unwanted spectral components at the negative signal frequencies
- deformation of the constellation diagram

As shown in [118], the unwanted spectral components can be inferred easily by introducing amplitude and phase unbalance into one of the I/Q signals – e.g., into the Q:

$$y(t)=A(t)(\cos(\omega t+\phi(t))+j\alpha\sin(\omega t+\phi(t)+\varepsilon)) \qquad (27)$$

Using Euler formula we can expand this into:

$$y(t)=\frac{(1+\alpha e^{j\varepsilon})Ae^{j(\omega t+\phi(t))}+(1-\alpha e^{-j\varepsilon})Ae^{-j(\omega t+\phi(t))}}{2} \qquad (28)$$

And after some more steps, we can get the ratio of unwanted and wanted spectral components:

$$\left|\frac{y_{unwanted}}{y_{wanted}}\right|=\sqrt{\frac{1+\alpha^2-2\alpha\cos(\varepsilon)}{1+\alpha^2-2\alpha\cos(\varepsilon)}} \qquad (29)$$

4.6.4 Local oscillator considerations

At the first sight, the local oscillator seems to be the simplest component in the whole RF chain, but the contrary is true. The most important local oscillator parameters are:

- frequency accuracy and stability – these topics are more thoroughly discussed in chapters 4.5.2 and 4.5.3.
- phase noise - this topic is covered in great details in [8], including the theoretic/analytic models. Another good reference is [26], which includes nice pictures of phase noise effects on receiver sensitivity in selectivity (adjacent channel rejection).
- Harmonics suppression: local oscillator produces higher harmonic signals. When these higher harmonics get into the poly-phase bridge, it does not generate the accurate 90° phase shift for the I/Q modulator LO inputs. A very nice analyse of this is described in [5] including a solution – a good filter of the harmonics at the LO output.

4.6.5 Peak to Average Power Ratio - PAPR

Enhancing the PAPR is an important topic because the PAPR directly relates to the linearity and efficiency parameters of the RF path components, especially the power amplifier. Because non-linearities increase the EVM and worsen adjacent channel protection ratio (ACPR), WLAN chipset manufacturers must take great care to PAPR reduction. The PAPR reduction is discussed more in chapter 8.4.5.

4.7 Co-location interferences in WLAN networks

This chapter deals with the basic principles of co-location interferences in WLAN networks.

It may seem that the term “co-location interference” refers to the well-known “near-far problem” [50] of CDMA networks, but it is a bit different thing.

Compared to the CDMA, WLAN has different media access method, where simultaneous transmission on the same WLAN channel is not possible and must be avoided by the MAC protocol layer. Contrary to this, the “near-far problem” in CDMA networks refers to the situations where nearby “aliens” transmitting on the same channel but with different chipping sequence are able to overwhelm the faraway legitimate transmitter (which has the right chipping sequence).

The near-far problem in CDMA is caused by the residual cross-correlation product whose energy is albeit small, but in case of near alien transmitters, the energy from their transmissions can be strong enough to drown out the cross-correlated signal from the weak distant legitimate transmitters.

This problem does not happen in WLAN networks simply because there are no multiple orthogonal chipping sequences and therefore no multiple access to single frequency channel.

In WLAN, the biggest interference problems are caused by co-located radios working on adjacent or alternate channels, whose transmissions can not be avoided by MAC layer, because from the viewpoint of our channel, they are like random strong interferences.

With DSSS modulation (802.11b), the adjacent channel interferences are processed by the receiver correlator (see figure 4 in [126]) and in case of strong interfering signal, the same situation can be experienced as with the CDMA near-far problem: the interference residual cross-correlation products can drown-out the faraway legitimate transmitters. Thus, we should analyse this problem in the dissertation thesis more thoroughly.

With OFDM modulation (802.11a/g/n), the adjacent channel interferences are more complicated topic – there is the OFDM versus interfering DSSS variant, which is very hard to solve, and the OFDM versus interfering OFDM variant, which can be solved several ways.

4.7.1 Adjacent/alternate channel interferences power levels

In order to estimate the power levels a receiver can get from co-located adjacent or alternate channel transmitters, we can check the transmit spectrum mask in the 802.11 [47] standard:

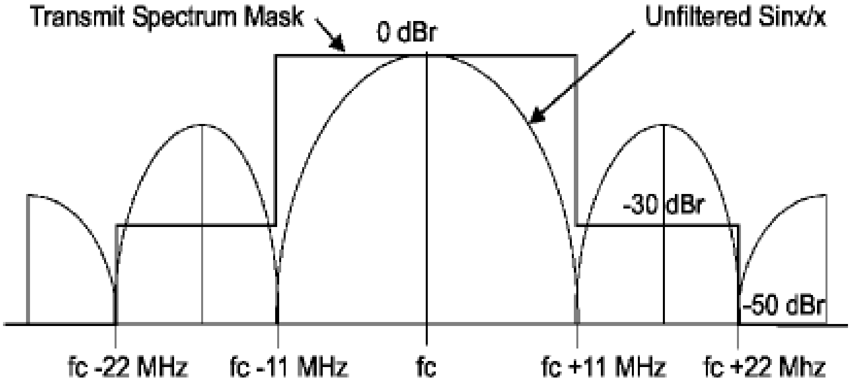


Figure 24: IEEE 802.11 transmit spectrum mask for DSSS (source: Figure 18-19 in IEEE 802.11)

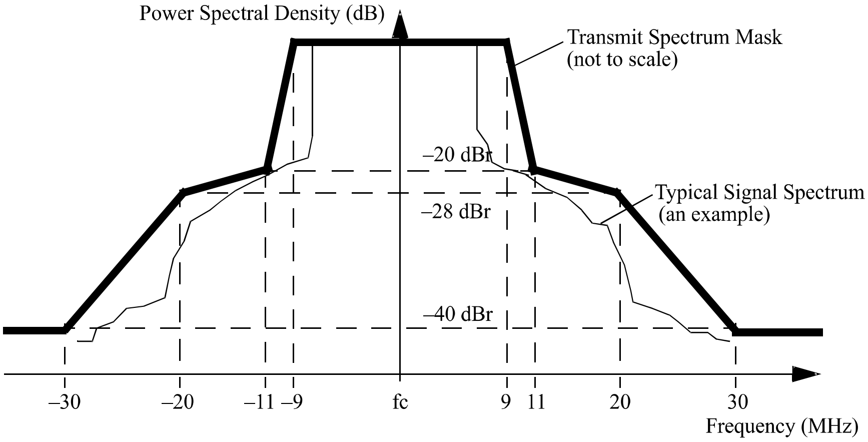


Figure 25: IEEE 802.11 transmit spectrum mask for OFDM (source: Figure I.1 in IEEE 802.11)

As we can see, the power levels in adjacent channels defined by these spectral masks are very high – considering that most commonly, the remote transmitters power levels experienced by the receivers are between -60 and -70 dBm, it seems impossible to use the adjacent channels at a co-location at the first sight.

Fortunately, the interferences between the co-located radios are attenuated by the free space path loss (FSPL), which can be expressed [32] as:

$$FSPL[dB] = -27.55 + 20\log(f_{MHz}) + 20\log(d) \qquad (30)$$

Where *f* is the frequency in MHz, *d* is distance in meters.

Note: the equation (30) is only accurate in the far field where spherical wave propagation can be assumed. Close to the transmitter, the EMG field will not be uniform and will have various spatial lobes and “deaf corners”. We use the equation here just for the coarse reference of what power levels we can expect at a co-location site. The losses for various distances between co-located radios are in the table below:

distance [m]	loss [dB]
0,1	20,05
0,2	26,07
0,5	34,03
1	40,05
2	46,07
5	54,03
10	60,05
20	66,07

Table 1: Free path loss

As we can see, the theoretical attenuation due to distance is still not enough to guarantee good isolation from adjacent channel operating even 20m far from a radio. Fortunately, the antenna characteristics play in our favour – on radio towers, antennas are never directed towards each other so they are picking only weaker side lobes of the other co-located antennas.

In reality, we routinely see adjacent power levels between -30 to -40 dBm from the co-located radios so we must consider such power levels in the rest of this work.

4.7.2 Adjacent/alternate channel interferences with DSSS

Based on the figure 2 and chapter 4.4-”DSSS in 802.11 “, we can draw the amplitude spectrum of 802.11 DSSS signal:

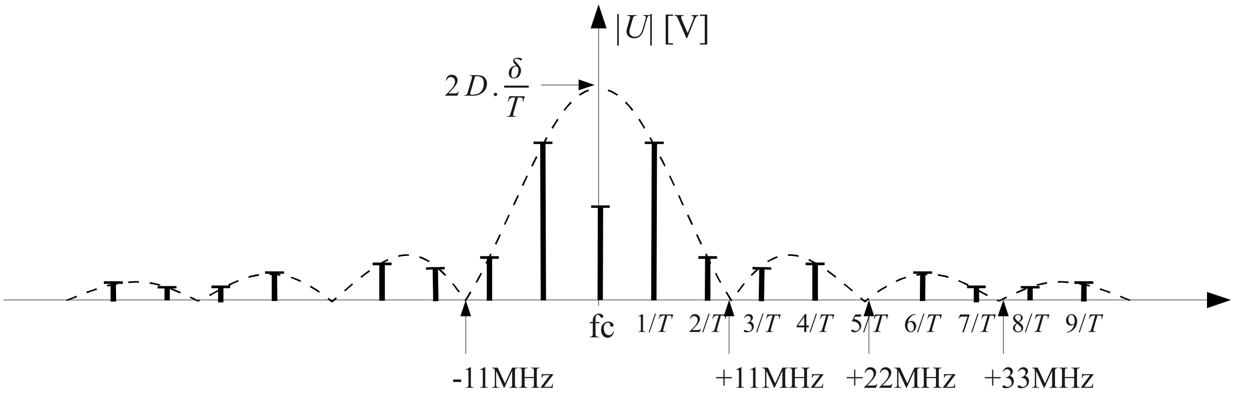


Figure 26: Amplitude spectrum of 802.11 DSSS signal

Because the duration of one data symbol is 1μs and it also determines the duration of the whole chipping sequence which has 11 bits, the δ=1/11 μs and thus the main spectrum lobe spans from centre frequency to ±11 MHz.

The period varies from 2/11 μs to 5/11 μs because there is no periodic subsequence inside the Barker chipping sequence.

So the figure 26 captures a specific moment when T=3/11 μs.

Now we can draw the same picture with two transmitters in adjacent channels:

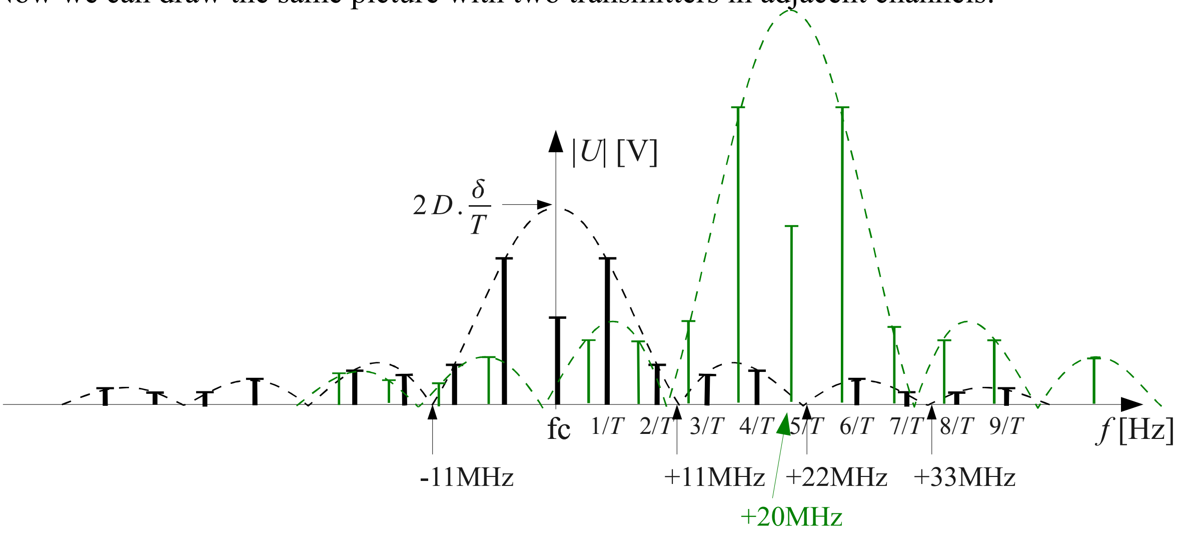


Figure 27: 802.11 DSSS on adjacent channels

The green lobe is signal from adjacent channel co-located transmitter, while the black lobe is a signal received from a remote radio.

In this specific example, the spectrum components do not interfere at all.

Moreover, the green and black lobes centre frequencies distance is 20 MHz, which is not multiple of the chipping sequence frequency. Therefore the green signal after being demodulated by carrier f_c, will differ heavily from the original Barker sequence in time domain and thus would be eliminated by the receiver correlator.

However, this picture is not realistic, because in a co-location scenario, the green signal power would be typically -40 dBm (10^{-4} mW) while the black signal would be -60 dBm (10^{-6} mW). Also, a real transmitter would filter out the adjacent side lobes to reduce them by at least 30dB and the alternate channel lobes by 50dB (see the 802.11 transmitter masks on figure 24).

Another factor to consider for the analysis is the *receiver adjacent channel rejection* parameter which is specified in the 802.11 clause 15.4.8.3:

"Adjacent channel rejection is defined between any two channels with ≥ 30 MHz separation in each channel group defined in 15.4.6.2. The adjacent channel rejection shall be ≥ 35 dB with an FER of 8×10–2 using 2 Mb/s DQPSK modulation described in 15.4.6.4 and an MPDU length of 1024 octets."

And of course, the type of the transceiver must be considered as well: super heterodyne (as were most of the original 802.11b devices) or direct conversion (zero IF, as are most or all WLAN designs today) will each suffer different set of problems [94], and the quality of the filters used will be different as well.

4.7.3 Adjacent/alternate channel interferences with OFDM

From its principles, the OFDM is more sensitive to adjacent channel interferences than DSSS – this is because DSSS receivers are based on signal correlator which can mitigate the interfering adjacent channel signals ideally by the processing gain ratio, while the OFDM receivers must deal with the adjacent channel side-lobes leakage into the working channel subcarriers.

Our experimental results (see chapter 6) show a quantitative comparison of adjacent channel rejection in 802.11 OFDM and DSSS modes.

The other part of the adjacent channel “power wrestling” game are the CS/CCA functions, which inhibit the transmission when a power level threshold is detected.

4.7.4 CS/CCA in the 802.11a/b/g/n receivers

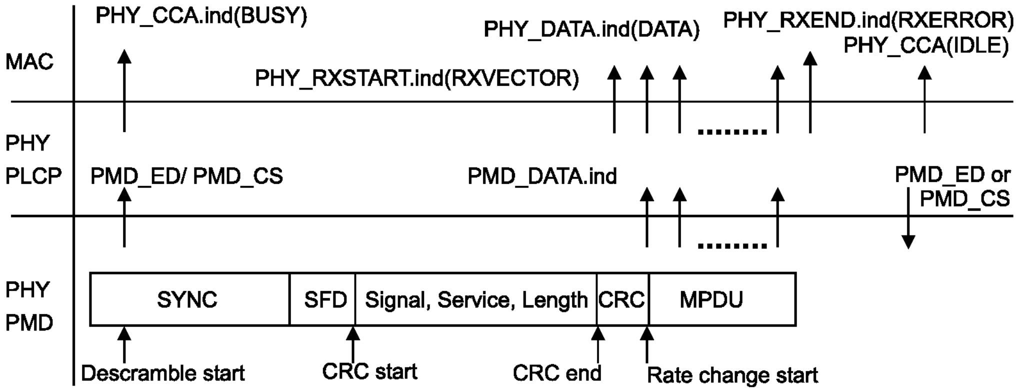


Figure 28: Receive PLCP (picture source: 802.11 standard, Figure 15-8)

The Carrier-sense (CS) and Clear-Channel-Assessment (CCA) functions implementation is critical for 802.11 device performance in presence of adjacent channel interferences. These functions are defined in 802.11 standard as integral part of the MAC and PHY layers implementation. A reader interested in the performance requirements for these functions must search relevant data scattered all over the 1230 pages of the 802.11 standard and 536 pages of the 802.11n standard. To shorten the quest, the extracted essences from both the standards are included in the next sub-chapter. The relation between CS and CCA for DSSS PHY is shown on figure 28. A similar figure 18-9 of the 802.11b HR (High Rate)/DSSS receive PLCP can be found in clause 18.2.6.

The DSSS (802.11-only) receiver can detect an ongoing transmission by 3 methods as defined in clause 15.4.8.4:

- CCA Mode 1: Energy above threshold: "*The ED* (energy detection) *threshold shall be ≤ –80 dBm for TX power > 100 mW, –76 dBm for 50 mW < TX power ≤ 100 mW, and -70 dBm for TX power ≤ 50 mW.*". Note: in the figure 28, the energy detection signal is labelled as PMD_ED.
- CCA Mode 2: Carrier sense (CS) only: "*With a valid signal (according to the CCA mode of operation) present at the receiver antenna within 5 µs of the start of a MAC slot boundary, the CCA indicator shall report channel busy before the end of the slot time.*". In the figure 28, the carrier sense signal is labelled as PMD_CS.
- CCA Mode 3: CS with ED above threshold.

The HR/DSSS (802.11b) receivers use different set of 3 modes defined in clause 18.4.8.4:

- CCA Mode 1: Energy above threshold. "*If a valid High Rate signal is detected during its preamble within the CCA window, the ED threshold shall be less than or equal to -76 dBm for TX power > 100 mW; –73 dBm for 50 mW < TX power ≤ 100 mW; and -70 dBm for TX power ≤ 50 mW.*"
- CCA Mode 4: CS with timer. "*CCA shall start a timer whose duration is 3.65 ms and report a busymedium only upon the detection of a High Rate PHY signal. CCA shall report an IDLE medium after the timer expires and no High Rate PHY signal is detected. The 3.65 ms timeout is the duration of the longest possible 5.5 Mb/s PSDU*".
- CCA Mode 5: A combination of CS and energy above threshold. "*CCA shall report busy at least while a High Rate PPDU with energy above the ED threshold is being received at the antenna.*"

The output signals of CCA modes 1-5, which are implemented in the PHY PMD (Physical Medium Dependent) sublayer, are passed to the PHY PLCP (Physical Layer Convergence Procedure) sublayer. The PLCP generates the PHY_CCA.indication signal for the MAC layer – see the clauses 15.4.5.15, 17.3.12, 18.2.6, 18.4.8.4 and 19.3.5.

The ERP (Extended rate PHY, aka 802.11b/g) receivers must handle all the CCA modes listed above and fulfil the "CCA performance" parameters defined by clause 19.4.6: "*When a valid signal with a signal power of –76 dBm or greater at the receiver antenna connector is present at the start of the PHY slot, the receiver's CCA indicator shall report the channel busy with* *probability CCA_Detect_Probabilty within a CCA_Time.*"

The OFDM (802.11a) CCA is defined in clause 17.3.12:

- "*Upon receiving the transmitted PLCP preamble, PMD_RSSI.indicate shall report a significant received signal strength level to the PLCP. This indicates activity to the MAC via PHY_CCA.indicate.*"
- Clause 17.3.10.5 defines the CCA sensitivity: "*The start of a valid OFDM transmission at a receive level equal to or greater than the minimum modulation and coding rate sensitivity (–82 dBm for 20 MHz channel spacing, –85 dBm for 10 MHz channel spacing,and –88 dBm for 5 MHz channel spacing) shall cause CCA to indicate busy with a probability > 90% within 4 µs for 20 MHz channel spacing, 8 µs for 10 MHz channel spacing, and 16 µs for 5 MHz channel spacing. If the preamble portion was missed, the receiver shall hold the CS signal busy for any signal 20 dB above the minimum modulation and coding rate sensitivity (–62 dBm for 20 MHz channel spacing, –65 dBm for 10 MHz channel spacing, and –68 dBm for 5 MHz channel spacing).*"

The 802.11n CCA sensitivity is defined:

- in clause 20.3.22.5.1 for the 20 MHz channel: "*the start of a valid 20 MHz HT signal at a receive level equal to or greater than the minimum modulation and coding rate sensitivity of –82 dBm shall cause the PHY to set PHY-CCA.indicate(BUSY) with a probability > 90% within 4 µs. The receiver shall hold the CCA signal busy for any signal 20 dB or more above the minimum modulation and coding rate sensitivity (–82 + 20 = –62 dBm) in the 20 MHz channel. A receiver that does not support the reception of HT-GF format PPDUs shall hold the CCA signal busy (PHY_CCA.indicate(BUSY)) for any valid HT-GF signal in the 20 MHz channel at a receive level equal to or greater than –72 dBm.*"
- In clause 20.3.22.5.2 for 40 MHz channel: "*The receiver of a 20/40 MHz STA with the operating channel width set to 40 MHz shall provide CCA on both the primary and secondary channels. When the secondary channel is idle, the start of a valid 20 MHz HT signal in the primary channel at a receive level equal to or greater than the minimum modulation and coding rate sensitivity of –82 dBm shall cause the PHY to set PHY-CCA.indicate(BUSY, {primary}) with a probability >90% within 4 µs.*

  The start of a valid 40 MHz HT signal that occupies both the primary and secondary channels at a receive level equal to or greater than the minimum modulation and coding *rate sensitivity of –79 dBm shall cause the PHY to set PHY-CCA.indicate(BUSY, {primary, secondary}) for both the primary and secondary channels with a probability per channel > 90% within 4 μs.*

(...definitions of receivers without HT-GF support are skipped ...)

The receiver shall hold the 20 MHz primary channel CCA signal busy for any signal at or above –62 dBm in the 20 MHz primary channel. This level is 20 dB above the minimum modulation and coding rate sensitivity for a 20 MHz PPDU. When the primary channel is idle, the receiver shall hold the 20 MHz secondary channel CCA signal busy for any signal at or above –62 dBm in the 20 MHz secondary channel. The receiver shall hold both the 20 MHz primary channel CCA and the 20 MHz secondary channel CCA busy for any signal present in both the primary and secondary channels that is at or above –62 dBm in the primary channel and at or above –62 dBm in the secondary channel."

Note that the 802.11 standard defines two different CS mechanisms: "physical CS" and "virtual CS" (see clause 9.2.1). The virtual CS is implemented by the MAC layer Network Allocation Vector (NAV) counter. As the virtual CS is not relevant to our PHY measurements, it is not discussed here.

The standard does not specify the CS/CCA sensitivity to adjacent channel interference. In fact, the CS/CCA sensitivity to generic interfering signals is dependent on:

- the adjacent channel rejection,
- and adjacent channel protection ratio of the interfering transmitter,
- CS/CCA implementation.

In the chapter 7.3, it is shown that if CS/CCA is enabled, the 802.11 DSSS in co-located adjacent channel is likely to inhibit the OFDM transmissions in the tested channel. These findings are supported by both analytic and simulation models published in [45], [56] and also by the experimental results of other teams, e.g. [14].

However, the current 802.11a/g/n OFDM implementations still have a lot of space for improvements. Some of the many methods for mitigating the adjacent channel interferences are discussed or proposed in chapter 8. But before we get there, we need a working model to support these proposals. Thus, the next subchapter is dedicated to the selection of proper wireless network model which would be used for studying the various wireless interferences.

4.8 Analytic, simulation or emulation model?

Analytic and simulation models of adjacent channel interferences have been done already. The paper [56] presents both analytic and simulation model of OFDM (802.11a/g/n) adjacent channel interferences, while analytic and empirical model of 802.11b are published in [45]. A complete OFDM 802.11a simulation model for MATLAB is available in [23]. There are many other publications with analytic and simulation models on the MAC level, like [102].

However, we argue that analytic or simulation models should be always used with experimental validation in real wireless networks, because otherwise they can never reflect the complexity of real devices and their performance in the real-world environment.

4.8.1 The theoretical models versus implementation issues of real 802.11 devices

The previous chapters discussed the basic principles used in 802.11 standards, but not much has been said about their implementation in real devices. The implementation can not be omitted, because it is the area where the reality jumps into the game and it always changes the score into the developer's/researcher's discomfort.

Rough idea of the 802.11 implementation complexity can be get from the 802.11a MATLAB simulation model published in [23] or 802.11p simulation model [103].

But things get much more complex when the design moves from simulation to real silicon and printed circuit boards. There are few publications describing the various implementation problems of real 802.11 devices – a very good example is [65] or the GNU radio implementation of 802.11p [35], although the later is not the “hard-core” single chip implementation which is standard for wireless devices these days.

Finally, we can not omit the issues of SW drivers code, which controls the HW and contributes greatly to the overall wireless device behaviour and performance. The scale and quantity of the SW drivers issues can be estimated by a brief look into (e.g.) the Linux wireless mailing lists, although the proprietary SW drivers are surely not lacking behind – only their bug reports are not visible to the outside world. On the other hand, the SW drivers can also substantially enhance the performance of the HW by using clever methods for mitigating the HW problems and bottlenecks– see e.g. [80], [88], etc.

Another example of the 802.11 theory versus reality clash is the “Ambient noise immunity” (ANI) feature implemented in Atheros chipsets. The ANI was implemented as a measure against the internal silicon digital noise, which could cause false carrier detect events. However, when enabled, it also affects the receiver sensitivity in an unexpected manner, as reported in [97].

And yet another instance of real devices behaviour, which is hard to model analytically and whose symptoms differ heavily between several manufacturers devices, is described in [14] (the paper title "*All Bits Are Not Equal - A Study of IEEE 802.11 Communication Bit Errors*" is quite self-explaining). There are also older papers dealing with similar topics - see [120], [73], [106] etc.

4.8.2 Experimental validation using testbeds

The usual experimental validation of analytic or simulation models used in academic publications uses free-space radio links deployed somewhere in the authors' campus area.

However, any credible scientific experiment must be reliable, repeatable and verifiable, which directs to controlled environment conditions simply impossible to achieve in free-space deployments. For demonstrating this statement, a quick scan of various 802.11 networks power levels in our building is shown on figure 29 below.

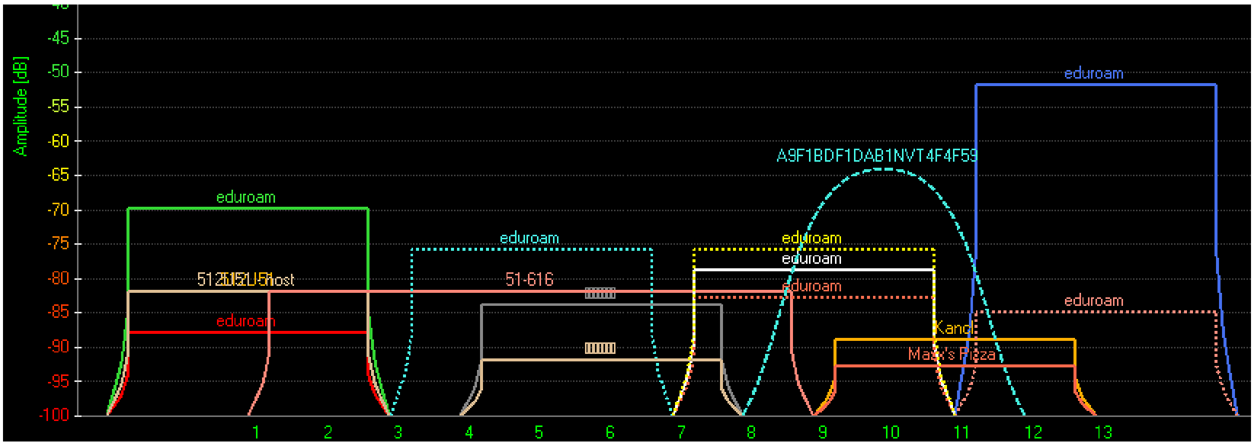


Figure 29: A scan of WiFi networks in the U5 TBU building

As it is impossible to find any area free of uncontrollable radio interferences in our building and its surroundings, the only way towards controlled experiments is using an RF emulation testbed. Two examples of existing emulation testbeds (built for different purposes) are discussed in chapter 2 - see [11] and [22]. For testing the interference properties of radio devices, we need to design a similar testbed fulfilling different set of requirements.

5 EMULATION TESTBED FOR WLAN EXPERIMENTS

In order to support many different experiments and measurements within a fully deterministic environment, an emulation RF testbed must be built for isolating the tested radio links from the stochastic outside world. For this purpose, an interconnection network composed of fixed and programmable attenuators, power splitters/combiners and coaxial cables must be designed.

5.1 Testbed measurement modes and configurations

There are myriad ways of how a RF testbed and its interconnection network could be designed. Therefore we must first discuss all the measurements we want to carry out, their corresponding interconnection network configurations and only then design the complete testbed supporting all required measurements. This is done in the next sub-chapter.

5.1.1 Adjacent and non-adjacent channel rejection

The adjacent/alternate channel rejection parameters and test/measurements procedures are defined by the clauses 15.4.8.3, 17.3.10.2, 18.4.8.3, 19.5.2 and 19.6.2 of the 802.11 standard [47]. The non-adjacent channel rejection is defined only for OFDM in clause 17.3.10.3.

The testbed needed for this measurement is the most basic one as shown on the block schematics below:

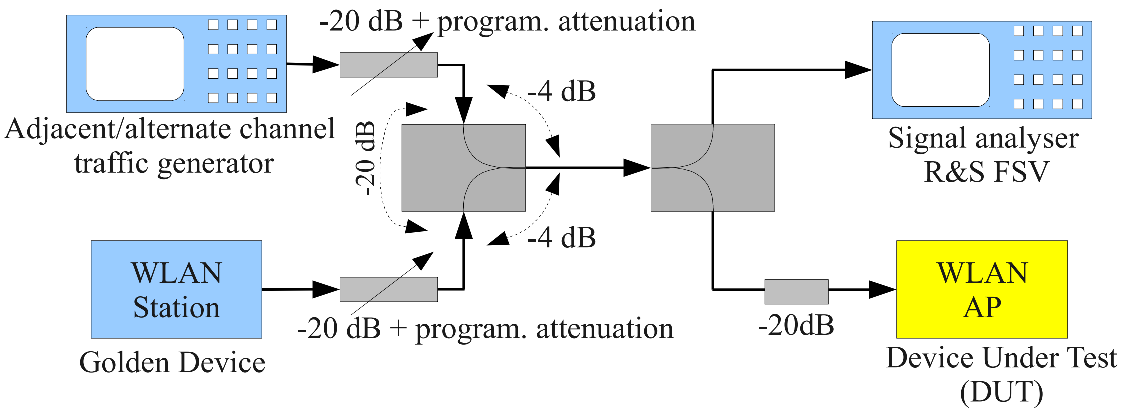


Figure 30: RF testbed for simple tests of adjacent/alternate channel rejection

This simple schema models situation where the WLAN AP is co-located with the adjacent channel AP which is emulated by the traffic generator. To emulate a real world situation, the signal received by the AP from the connected WLAN Station ("Golden Device") should be weaker than the signal from generator. The programmable attenuators in both signal paths allow setting the power levels to arbitrary values.

5.1.2 Testing hidden nodes

Hidden nodes are intrinsic phenomenon in external wireless networking. Unfortunately, there are no exact test procedures defined for hidden nodes handling in the 802.11 family of standards that were designed primarily for interior networks. However, hidden nodes affect the network performance so substantially that their inclusion or emulation should be considered as a must for every testbed.

For testing a wireless network segment coordination and synchronization, we need to be able to emulate hidden nodes. We can do this easily by connecting the splitters/combiners into a tree as shown on the figure below.

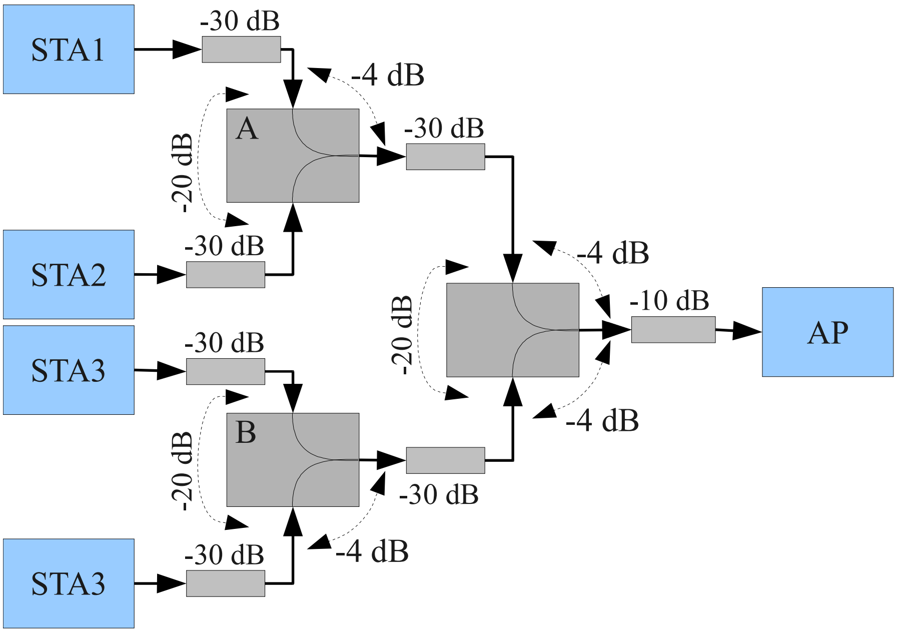


Figure 31: Splitter/combiner tree emulating hidden nodes

On this picture, the power levels by the individual nodes are following:

- all the stations can “hear” the AP with power level N-78 dBm (where N is AP transmit power level, e.g. 18 dBm),
- both stations connected to the same splitter/combiner hear each other with power level N-80 dBm,
- but the stations connected to the splitter/combiner “A” can not hear stations connected to splitter/combiner “B” – the power levels received from the other group are N-140 dBm. For transmit power level N=20 we get -120 dBm at the other end, which is at least 30dB below receiver minimum sensitivity level.

5.1.3 Using multi-port splitters/combiners

For testing wireless networks with more nodes, we can add more ports by using multi-port splitters/combiners. They give us more flexibility than dual-port splitters/combiners because they

- allow emulation of bigger mutually non-hidden groups,
- are more cost effective than dual port splitters/combiners.

An example of the an testbed with multi-port splitter/combiners is shown on the figure below.

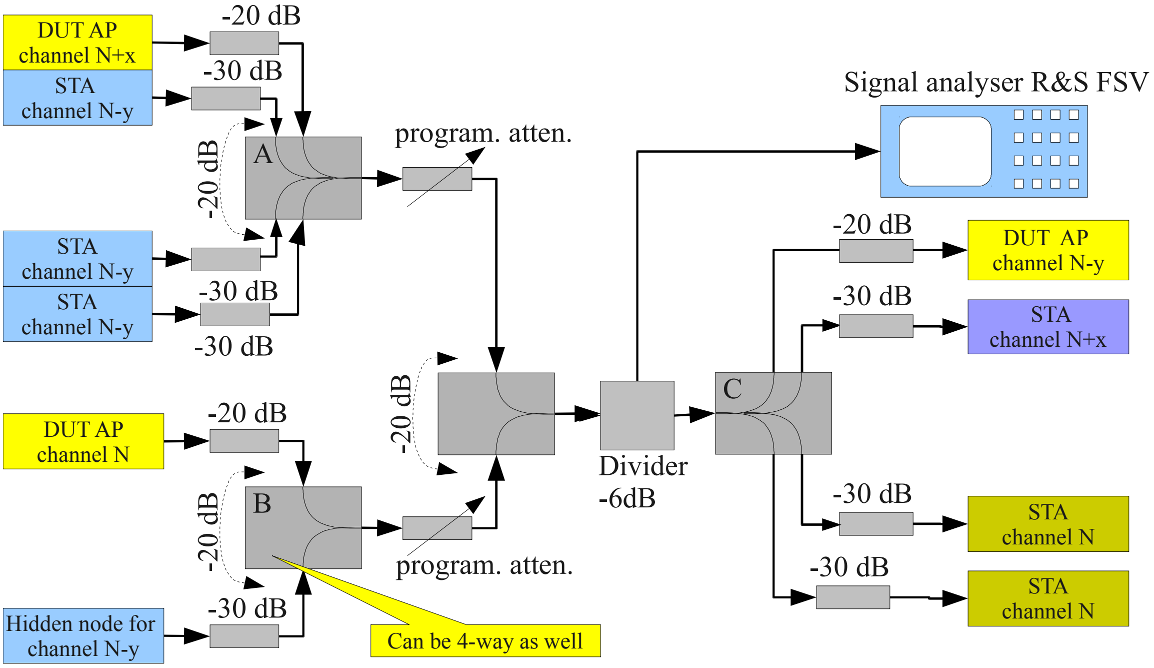


Figure 32: RF testbed for adjacent channel rejection with multiple networks

Here we have 3 wireless networks, working on 3 channels N, N+x, N-x. We will test the behaviour of the APs whose co-location interference signal strengths can be emulated using the programmable attenuators the same way as on figure 30. In the same testbed, we can also test the hidden nodes – here, the stations connected to splitter/combiner A are hidden to stations connected to splitter/combiner B.

In fact, this is the schema we will use in the dissertation experiments, the only modification is that the splitter/combiner “B” will have four ports.

5.1.4 802.11n MIMO 2x2

The 802.11n standard most important enhancement is the ability to transmit and receive independent data streams on multiple independent antennas (MIMO means Multiple In, Multiple Out). In fact, this means that the 802.11n devices contain multiple radio transmitters and multiple receivers, in the case of MIMO 2x2 configuration it is 2 pairs of transmitters and receivers.

The streams are separated by polarization or spatially. Thus, the if we want to connect a 802.11n devices into our testbed, we need to separate the streams into independent path and we need to emulate the cross-talk between these paths.

The schema of simple testbed for testing the cross-talk influence is below:

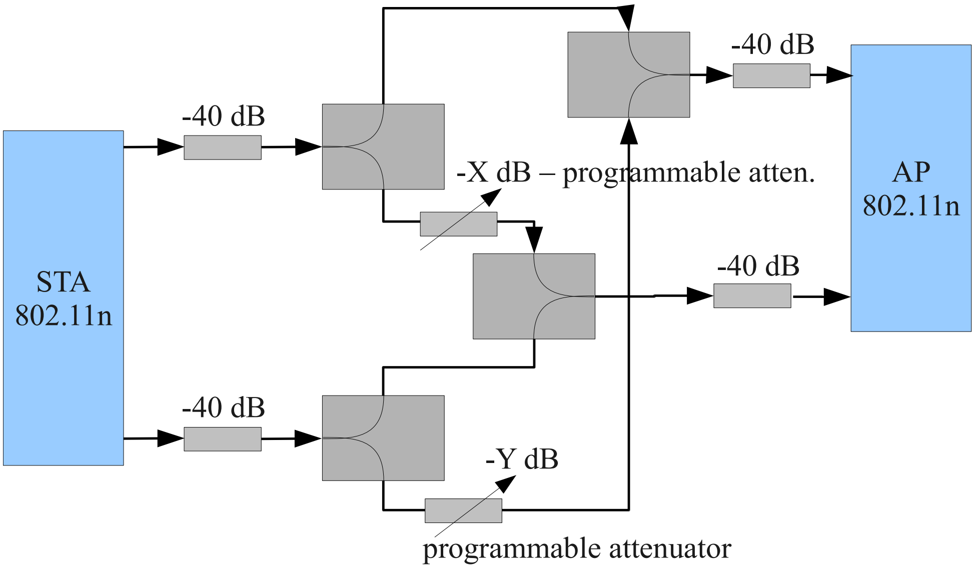


Figure 33: testbed for emulating cross-talks between MIMO2x2 streams

Based on the schema above, the complete testbed for testing the 802.11n devices could look like this:

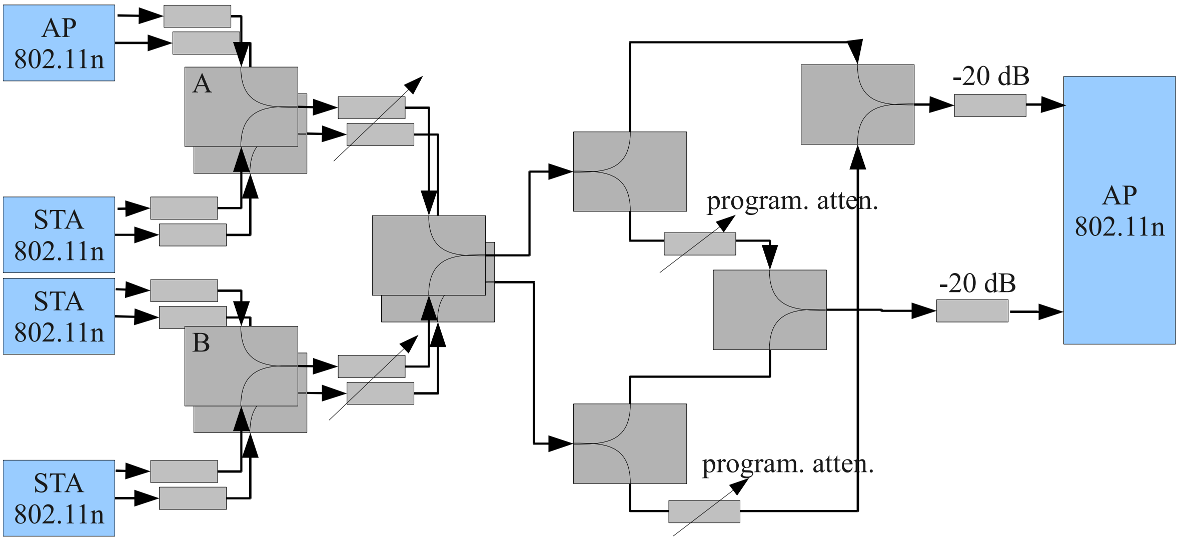


Figure 34: testbed for testing the 802.11n

In the left part of the figure, there is the usual tree-like structure of splitters/combiners, but this time it is doubled to accommodate the second radio path. On right side, there is the cross-talk emulation schema from figure 33.

5.1.5 The complete testbed

The complete testbed was created as a combination of the previous configurations with one extension – the signal analyzer is now connected using resistive couplers and RF switch, see figure 35:

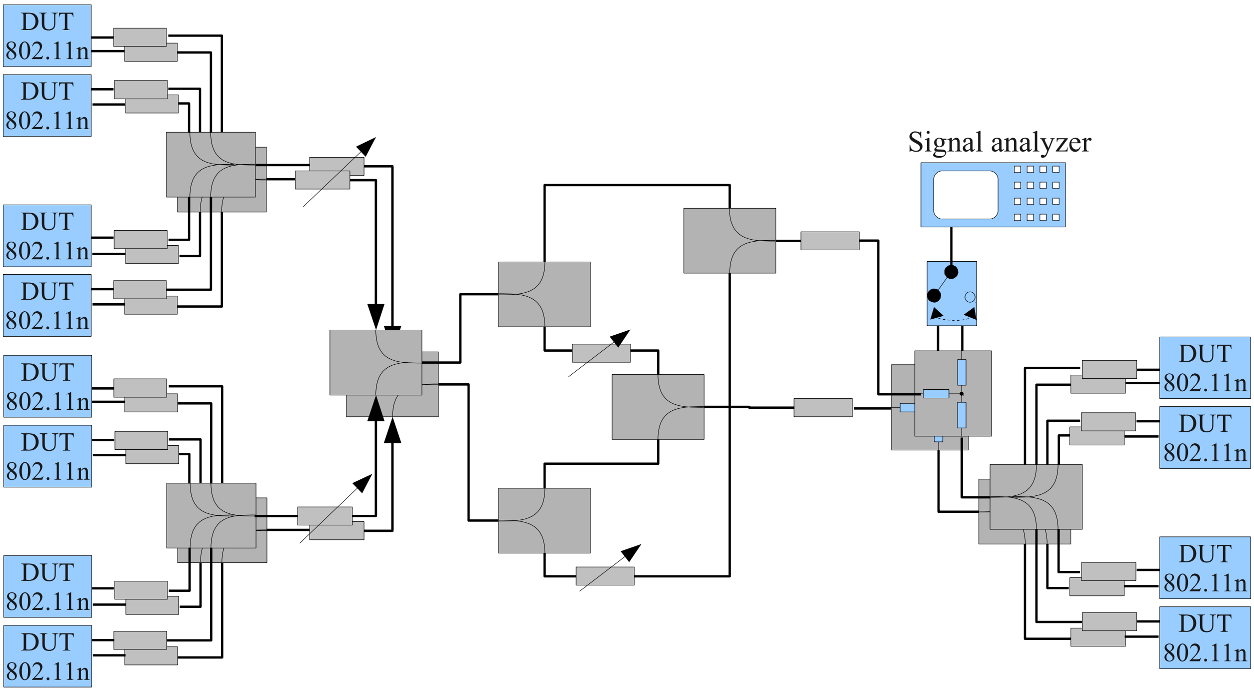


Figure 35: The complete test- bed

The testbed allows connecting twelve 802.11n devices MIMO 2x2 or 24 devices MIMO 1x1. The leftmost programmable attenuators can be used to emulate various signal fading and hidden nodes conditions, while the programmable attenuators in the middle can model cross-talk between the two 802.11n streams.

The configuration is a compromise between the emulated parameters granularity and the cost of of the testbed, which was the most decisive factor in our case. It would be nice to have programmable attenuators connected to each tested device, but as the attenuators are the most expensive components, we simply could not afford them. If needed, we can still attenuate the individual devices using their transmit power settings.

5.2 Testbed implementation

The actual RF emulation interconnection network design is shown in the next sub-chapter (5.2.1).

However, the testbed is a not just the RF network. It is a complex system composed of many programmable components: signal analyzer and generators, programmable attenuators and switches, power supply distribution unit, LAN switches, tested devices etc.

All these devices are orchestrated by the testbed servers accessible from the internet by SSH and/or the web interface.

The logical schema of the testbed management infrastructure is depicted on figure 36.

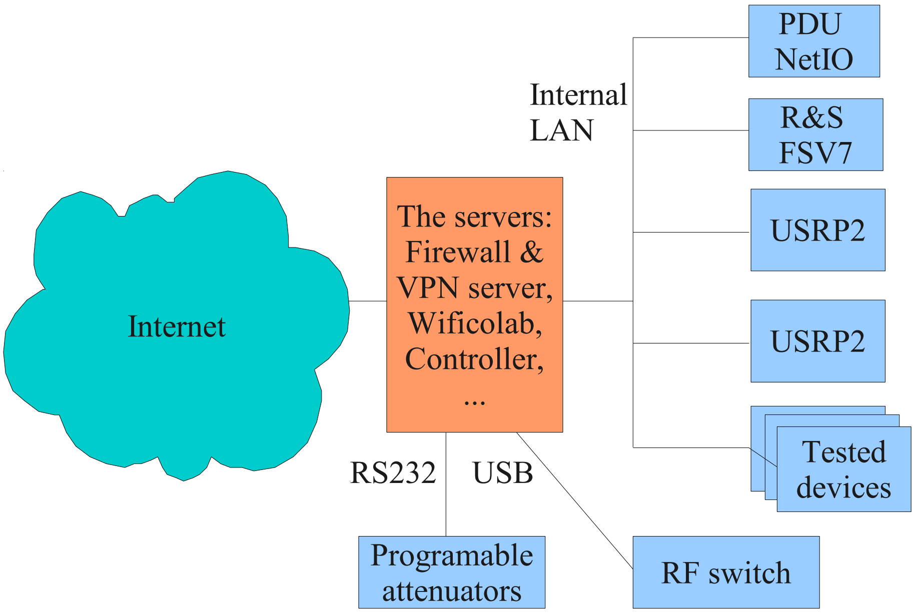


Figure 36: testbed management infrastructure

The testbed management infrastructure details are described in chapters 5.2.2-5.2.5.

5.2.1 The RF interconnection network

The interconnection of the RF components follows the schema in figure 34. For achieving compact dimensions, we have used semi-flexible, hand-formable coaxial cables Microcoax UT-141C-Form with SMA connectors. As the minimum bending radius specified in the cable datasheet is 9.5mm, the lengths of all the cables had to be designed a priori in a CAD software. The design is shown on figure 37 below:

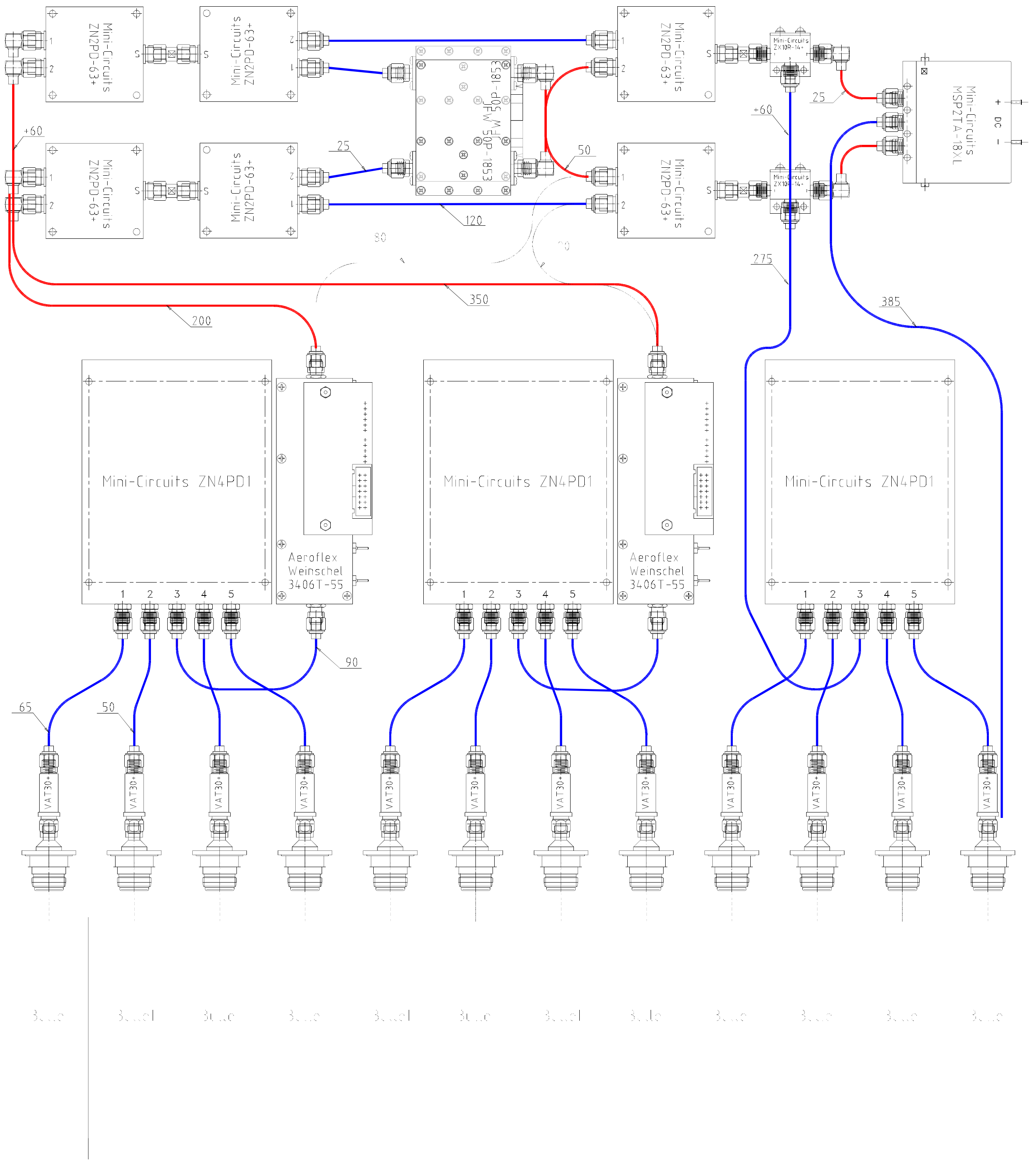


Figure 37: Mechanical design of the RF interconnection network

All the components are specified at least for the 2GHz-6GHz band. The dividers/combiners are Minicircuits ZN4PD1-63+ and ZN2PD-63-S+. The resistive divider/combiner used for connecting the spectrum analyzer through the RF switch is ZX10R-14+ (. The fixed 20dB attenuators protecting the combiner and the WiFi devices input stages against damage are Minicircuits VAT-30+. The programmable attenuators are JFW 50P-1853-SMA (solid state), Aeroflex Wienschel 3406T-55 and 3408T-103 (electro-mechanical relay switched).

As the whole testbed was supposed to be placed in a standard 19” server rack, the dimensions of the testbed match a 19” server rack drawer. After the mechanical design was done, all the cables were custom-crafted according to the lengths specified in figure 37.

The testbed installation is shown on figure 38 and figure 39 below.

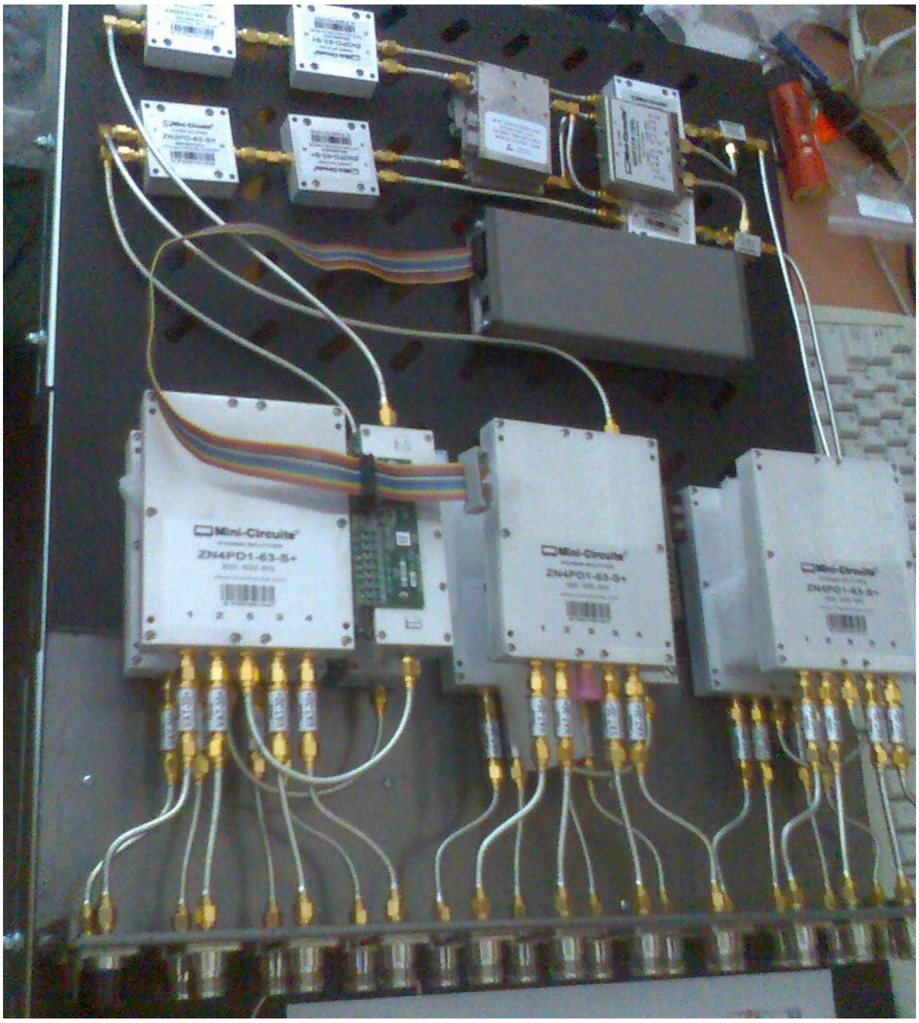


Figure 38: RF interconnection network implementation

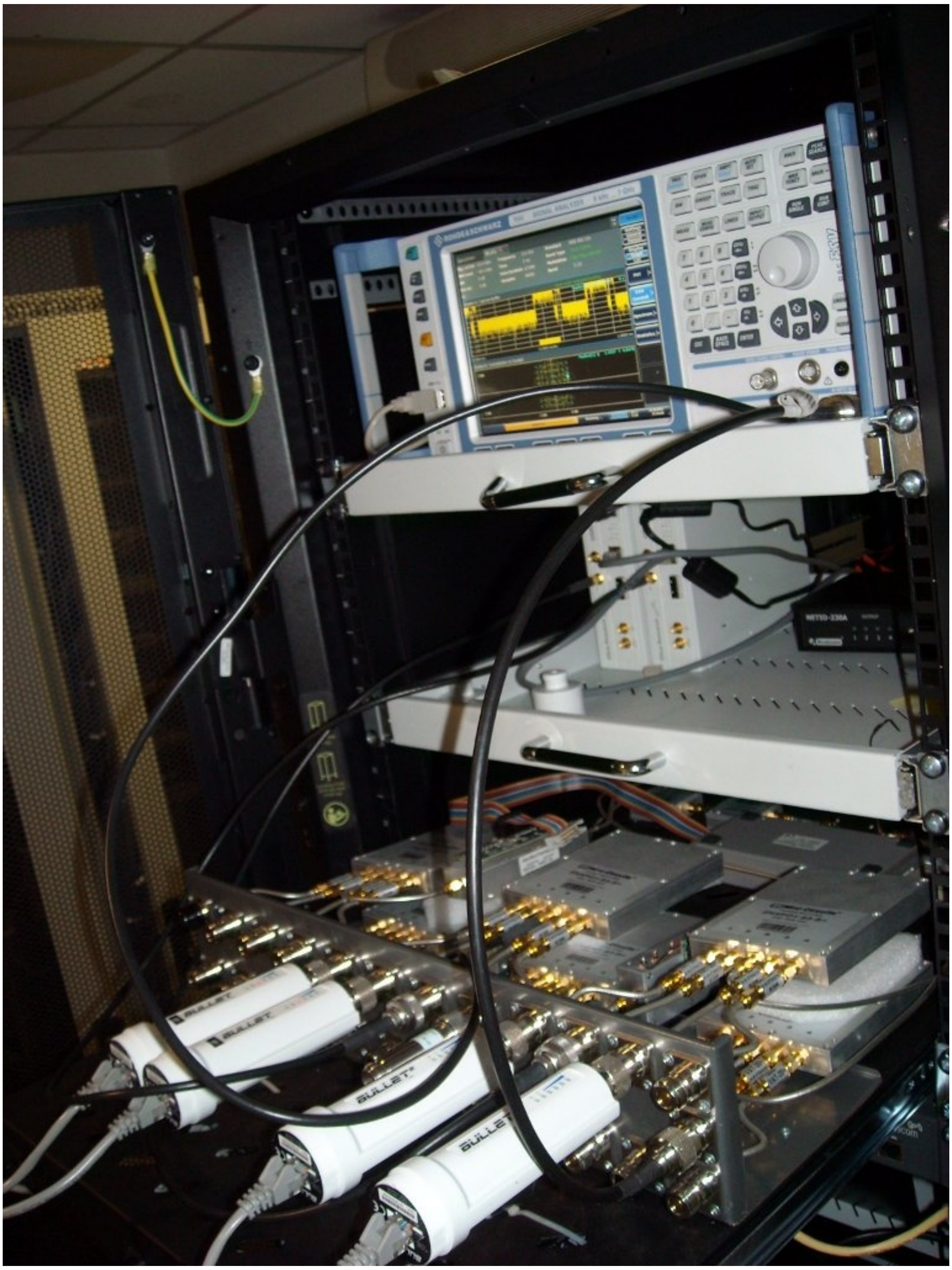


Figure 39: RF network installed in rack

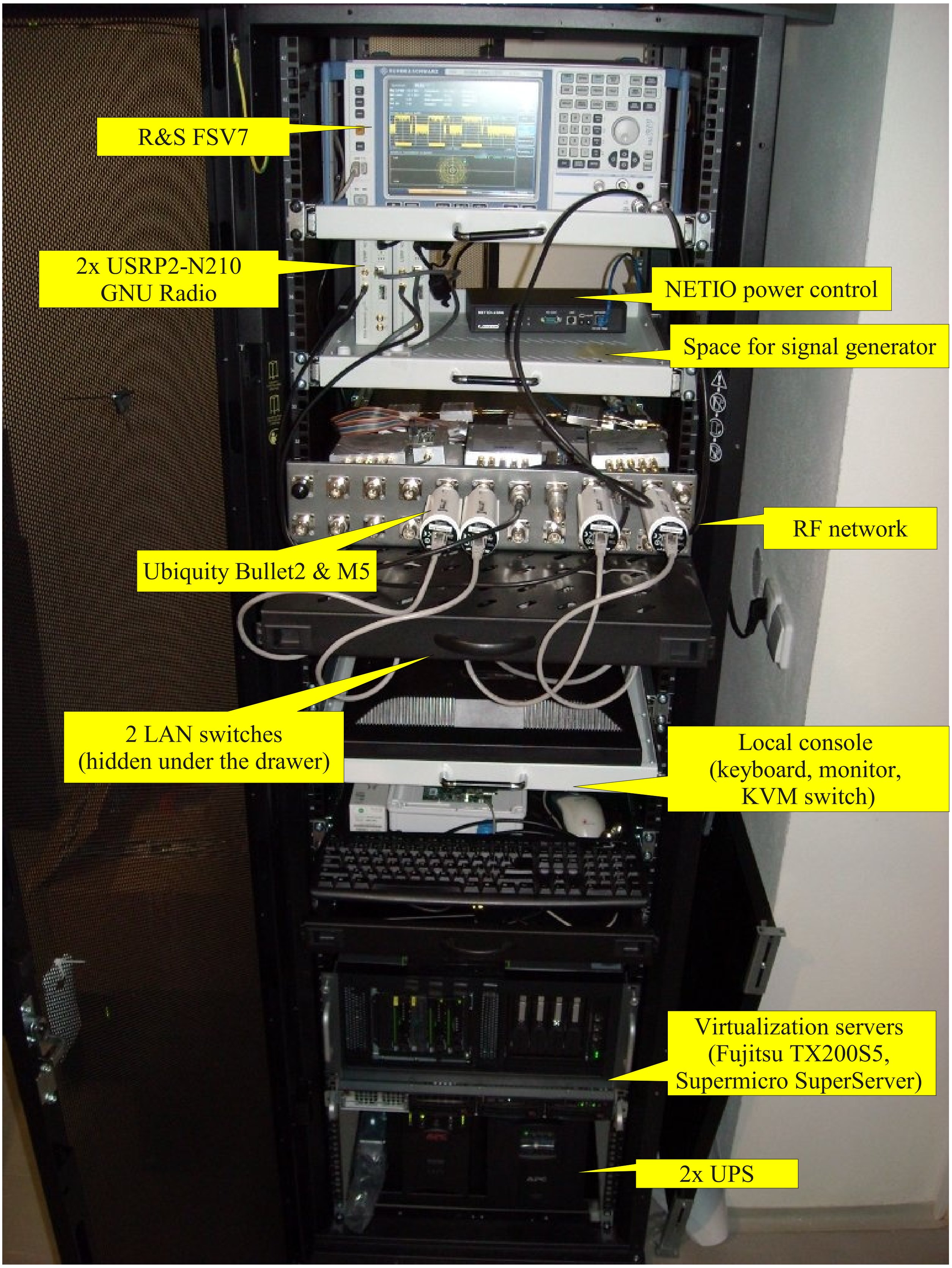


Figure 40: testbed installation in 42U server rack

5.2.2 Virtualization servers

The servers are placed in a server room in the same rack as the test-bed, so they can be connected to the programmable attenuators and RF switch by USB and RS232 interfaces.

The testbed is using virtualization as much as possible. Without virtualization, several physical computers would be needed – at least the following services should not share single computer for security or performance reasons:

- the firewall + VPN server
- the web server with the Wificolab web application
- the GNU radio virtual machines
- the testbed controller ("the orchestrator"), TCP and UDP traffic server, ...

Thanks to virtualization, the testbed is much more compact – only single virtualization server is needed (or two servers in case redundancy and high-availability is required).

We are actually using two servers with online data replication based on DRBD, so the virtual machines can be freely migrated between the two servers. For historical reasons, the first server (Fujitsu TX200S5) runs the Xen 3.2 hypervisor in Debian 5 (Lenny), while the second server has KVM on Debian 6 (Squeeze).

It was not desirable to replace Xen by KVM, because KVM does not support PCI or USB pass-through on machines without the VT-d feature, which is only available on newer chipsets like Intel i5500 (see [60]), while Xen allows PCI and USB pass-through on any HW (see [122]).

Instead of upgrade, I developed a method for migrating the virtual machines back and forth between Xen and KVM hypervisors. The method is published in [29] and is also attached in Appendix 11.3.

5.2.3 Firewall and VPN server virtual machine

The main purpose of the firewall is protect the testbed against the attacks from the wild internet. For example, the spectrum analyzer is not designed as a bullet-proof internet device – in fact, its Windows XP Embedded OS does not even have the standard Windows firewall and its LXI web service and VXI protocol do not even have any authentication protection. Thus it is critically vulnerable against all the possible internet threats.

The same holds for the tested devices. Usually, they are connected to the testbed with their default configuration including the user names and passwords which could be easily guessed by any dictionary attack.

Also, many of the tested devices do not support IPv6 for their management interfaces. As the IPv4 public addresses are now precious resource even in our university network, we had to resort to the internal IPv4 address ranges for them. Therefore, aside of the unwanted traffic filtering, the firewall must do also the network address translation (NAT) for the testbed.

The VPN server then allows the users to access the internal network of the testbed. Currently, the PPTPD server is used, but OpenVPN could be installed as well upon an user request.

5.2.4 Web application – wificolab.utb.cz

The Wificolab application was created as a master thesis project [15]. The application provides the following features:

- User management: users can self-register and administrator can approve them.
- Measurement reservations: the testbed should be always used by a single user at time. Therefore, the users must make reservations for time slots they want to measure.
- Measurements: wificolab communicates with the FSV7 analyzer using the VXI11 protocol. It can setup the FSV7 using a setup template. An user can then run the measurement directly from wificolab web and store the resulting data in the database. The data are presented in the form of graphs.
- Measurement results management: the user can share the results with other users and can compare the results graphically – it is possible to display several different results in a single graph, as shown on figure 41:

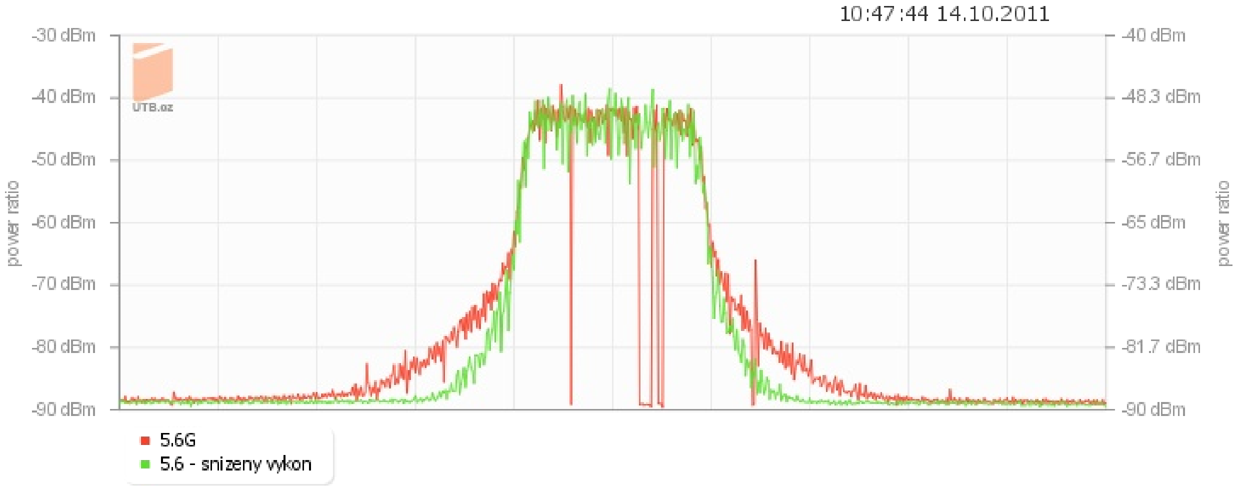


Figure 41: A comparison graph generated by Wificolab web application

Wificolab is now published with open source license at http://sourceforge.net/projects/wificolab/ and will be further developed also in future.

5.2.5 Testbed controller virtual machine

The testbed controller virtual machine is used to run the measurement automation scripts. Currently, it drives also the USB-RS232 interface connected to the Wienschel 8210A controller. This is implemented by the XEN PCI passthrough – an USB PCI card was added to the physical server and this whole PCI device is passed-through to the VM, so any USB device connected to the PCI card then appears in the virtual machine.. The passthrough was accomplished by steps published in appendix 11.4.

Note: for achieving full independence of the testbed on a virtualization server, it is planned to connect the Wienschel 8210A controller to the Alix miniPC, which will then serve as a bridge between RS232&USB and ethernet. Then the virtualization servers will not have to be in the same rack, they can be even in completely different server room.

After the Wienschel programmable attenuators were connected to the Wienschel 8210A controller, it was necessary to configure this controller by a terminal application – standard Linux "Minicom" terminal with "9600, 8N1, no handshake" configuration was used. The following commands were run in the terminal:

```
*IDN?
Weinschel, 8210A-2, 1493, V2.98
LIST? DEVICE CONFIG
3, NONAME1, 3408-103, 1350, 2, NONAME2, 3406T-55, 1405, 4,  NONAME3, 3406T-55, 1420, 5
ASSIGN AT103 3408-103 1350
ASSIGN AT55 3406T-55 1405
ASSIGN AT55-2 3406T-55 1420
LIST? DEVICE
2, AT103, AT55
SAVE ASSIGN
```

From this point, the three programmable attenuators have been assigned names AT103, AT55, and AT55-2. From this point, it is possible to set the attenuators values by sending the following strings to the serial port:

- Setting the 3408T-103 attenuator to 20dB:
  `ATTN AT103 20`
- Setting the 3406T-55 attenuator to 30dB:
  ATTN AT55 30
- Getting the attenuator value:
  `ATTN? AT55`

For easy command line control of the attenuators, the “attnctrl” application was implemented – see chapter 11.2. The testbed controller VM can also run applications for orchestrating the measurements - the code URL is also in the chapter 11.2.

6 TESTBED MEASUREMENT METHODS AND PROCEDURES

The measurement methods are basically predetermined by the 802.11 standards. However, the standards do not address the actual measurement procedures. A very good introduction into 802.11 standard-compliance measurement methods is published in application note [119]. For receiver chain testing, this document refers to another application note [27] describing a method for measuring the packet/frame error rates (PER/FER) using a vector signal generator. As both documents rely on reader's sufficient budget for acquiring or renting the required signal generators and analyzers, their methods can be used only as an inspiration.

Therefore, for making measurements in these thesis, the complete set of measurement methods and tools had to be developed.

6.1 Measuring the transmitter parameters

Transmitter measurements require a professional, calibrated signal analyzer, ideally with 802.11 WLAN plugins. Thanks to the CESNET grant 351/2009, we own one – R&S FSV7.

The tested device antenna output should be connected directly to the analyzer RF input. For all transmitter measurements, there must be some traffic on the device WLAN output, therefore we need an application which will generate that traffic.

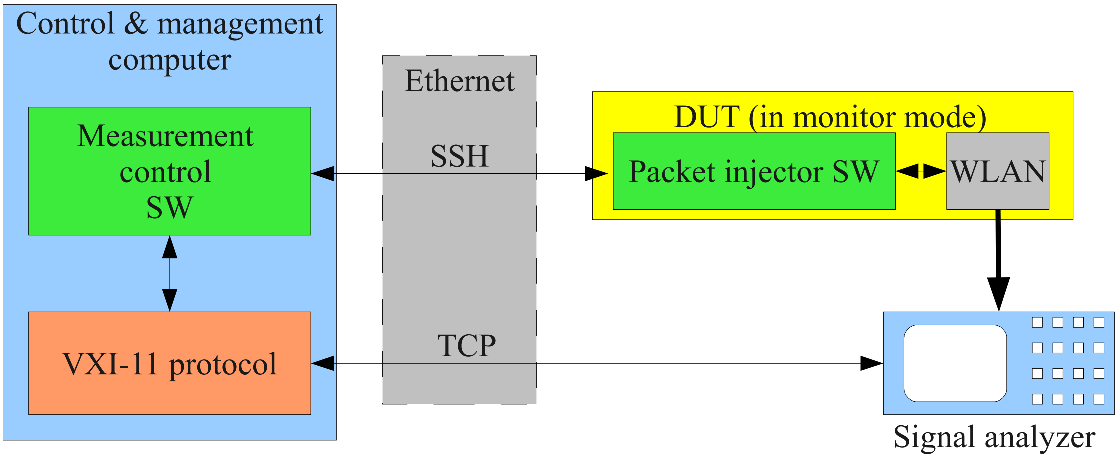


Figure 42: Measuring transmitter parameters using WLAN monitor mode

A measurement method using a DUT in monitor mode with packet injection support is shown on figure 42. This is the most simple and most flexible way, because:

- only one WLAN device is needed. As the analyzer captures traffic from only one device, there is no uncertainty about which signal comes from which device.
- The packets can be injected in arbitrary bit rates.
- The DUT can be connected directly to the analyzer, no power dividers/combiners, which could influence the measurement accuracy, are necessary.

If the device has closed firmware and/or does not support monitoring mode with packet injection, then it is possible to try using the workaround solution shown on figure 43.

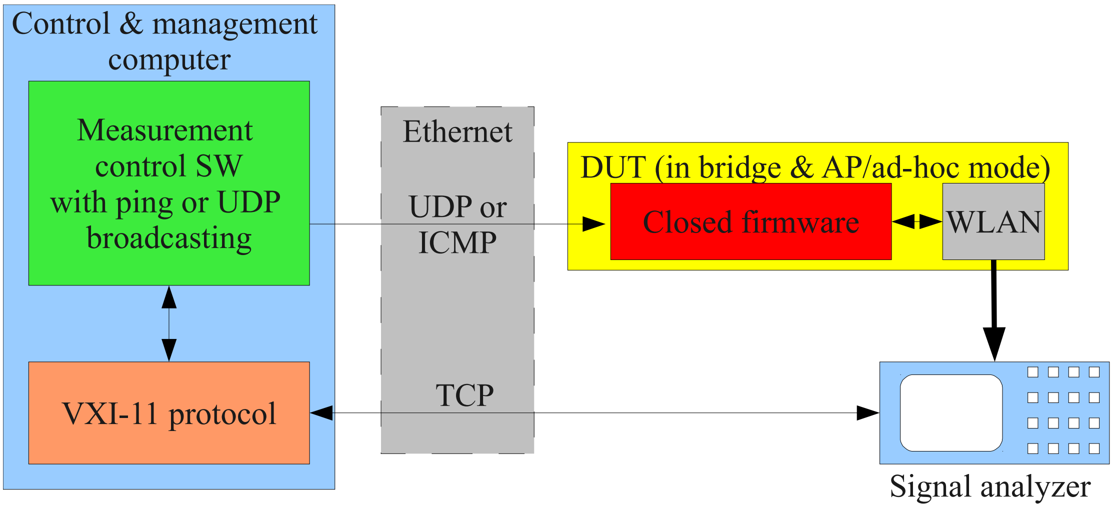

Control & management computer
Measurement control SW with ping or UDP broadcasting
VXI-11 protocol
Ethernet
UDP or ICMP
TCP
DUT (in bridge & AP/ad-hoc mode)
Closed firmware
WLAN
Signal analyzer

Figure 43: Measuring transmitter parameters with device in bridge and ad-hoc mode

The AP or ad-hoc mode allows using only a single device, because if broadcast ICMP (ping) or UDP packets are sent to it, they get transmitted even if there are no other devices associated to the DUT. For testing the transmitter parameters, a simple UDP broadcaster tool was developed – see chapter 11.2.

When experimenting with different devices, the ideal example of a device which can be used in this mode was found: the Ubiquity Bullet, as its broadcast performance in the AP mode without association was always perfectly stable and it also allows changing the bit rates for the multicasts and broadcasts by issuing the following command:

```
iwpriv "ath0" mcast_rate <bit rate in kbit/s>
```

The problem with this measurement mode is that some devices do not have the AP nor ad-hoc mode. With other devices, the broadcasting in AP mode without association does not work reliably or at all. With some devices the broadcasting works, but only with the basic bit rates.

In such cases, we must use a suboptimal solution of two-devices configuration shown on figure 44. Here the traffic is generated by a udp-sender tool installed on the control & management computer. As the unicast UDP socket needs a server to talk to, another computer must be used for running the udp-receiver.

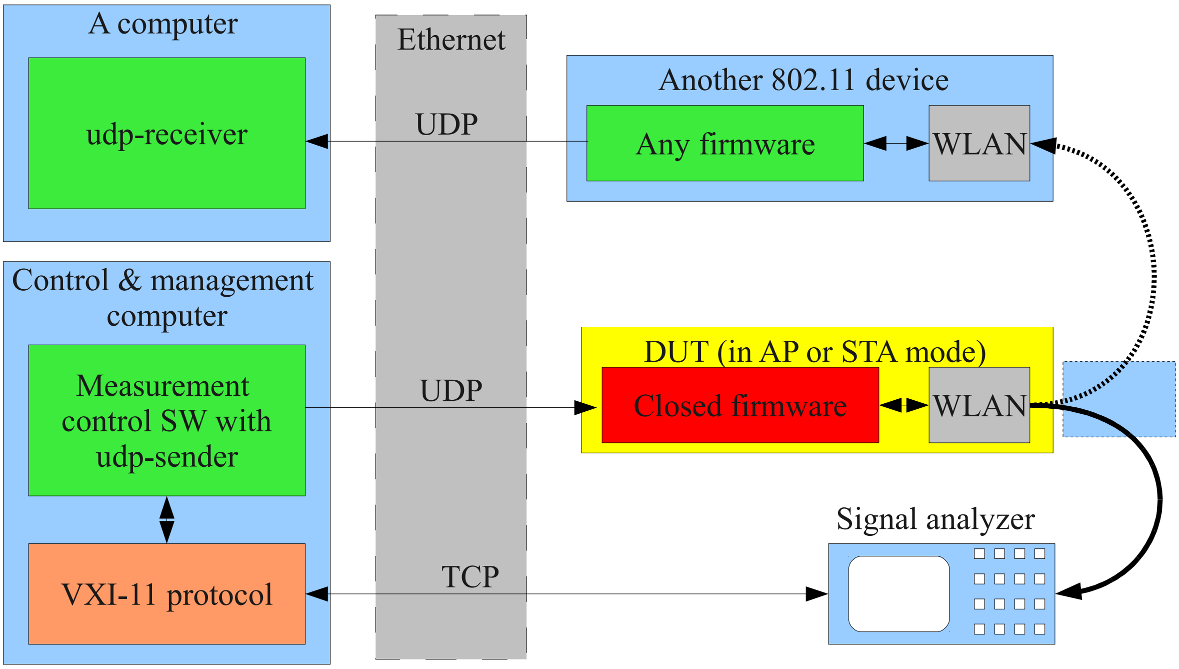


Figure 44: Measuring transmitter parameters without WLAN monitor mode

This configuration has one disadvantage: the second 802.11 device responses by 802.11 ACK packets to every UDP packet transmitted by the DUT. This confuses the signal analyzer, which can't distinguish between the two devices. To get around this problem, the second 802.11 device should not be connected to the DUT and analyzer by a cable and a coupler – instead, a free-space coupling should be used, if possible.

6.2 Measuring receiver minimum sensitivity

The receiver minimum sensitivity parameter needs to be measured as a base for adjacent and alternate channel rejection parameters.

The measurement procedures must respect the following clauses in the 802.11 standard [47]:

- 15.4.8.1: "*The FER shall be less than 8×10–2 at an MPDU length of 1024 octets for an input level of –80 dBm measured at the antenna connector. This FER shall be specified for 2 Mb/s DQPSK modulation. The test for the minimum input level sensitivity shall be conducted with the ED threshold set ≤ –80 dBm.*"
- 18.4.8.1: "*The FER shall be less than 8×10–2 at a PSDU length of 1024 octets for an input level of –76 dBm measured at the antenna connector. This FER shall be specified for 11 Mb/s CCK modulation. The test for the minimum input level sensitivity shall be conducted with the ED threshold set less than or equal to –76 dBm.*"
- 17.3.10.1: "*The packet error rate (PER) shall be less than 10% at a PSDU length of 1000 octets for rate-dependent input levels shall be the numbers listed in Table 17-13 or less. The minimum input levels are measured at the antenna connector (noise factor of 10 dB and 5 dB implementation margins are assumed).*"

For quick reference, the table 17-13 is cited below.

Table 2: 802.11 receiver performance requirements (source: table 17-13 in 802.11 standard)

Modulation (bit rate for 20MHz channel)	FEC coding rate	Adjacent channel rejection [dB]	Non-adjacent channel rejection [dB]	Minimum sensitivity [dBm] (20 MHz channel)	Minimum sensitivity [dBm] (10 MHz channel)	Minimum sensitivity [dBm] (5 MHz channel)
BPSK (6Mbits)	1/2	16	32	-82	-85	-88
BPSK (9Mbits)	3/4	15	31	-81	-84	-87
QPSK (12Mbits)	1/2	13	29	-79	-82	-85
QPSK (18Mbits)	3/4	11	27	-77	-80	-83
16QAM (24Mbits)	1/2	8	24	-74	-77	-80
16QAM (36Mbits)	3/4	4	20	-70	-73	-76
64QAM (48Mbits)	2/3	0	16	-66	-69	-72
64QAM (54Mbits)	3/4	-1	15	-65	-68	-71

For 802.11n standard [46], its clause 20.3.22.1 must be respected:

"*The packet error rate (PER) shall be less than 10% for a PSDU length of 4096 octets with the rate-dependent input levels listed in Table 20-22 or less. The minimum input levels are measured at the antenna connectors and are referenced as the average power per receive antenna. The number of spatial streams under test shall be equal to the number of utilized transmitting STA antenna (output) ports and also equal to the number of utilized device under test input ports. Each output port of the transmitting STA shall be connected through a cable to one input port of the device under test. The test in this subclause and the minimum sensitivity levels specified in Table 20-22 apply only to non-STBC modes, MCSs 0–31, 800 ns GI, and BCC.*"

For quick reference, the table 20-22 is also cited below.

Note that while the 802.11b clauses specify frame error rates (FER), the 802.11a/g/n clauses use packet error rates (PER). Because frame errors are measured at the the MAC (link) layer where automatic retransmission of corrupted packets takes place, while packet errors are measured on the IP layer, it is clear that FER >= PER.

Also, the PER is much easier to measure, because IP layer can be easily accessed from application level, while sending and receiving link layer frames is problematic. A more detailed discussion about this is in chapter 6.4.

Table 3: 802.11n receiver performance requirements (source: table 20-22 in 802.11n standard)

Modulation (bit rate for 20MHz channel, MIMO 1x1)	FEC coding rate	Adjacent channel rejection [dB]	Non-adjacent channel rejection [dB]	Minimum sensitivity [dBm] (20 MHz channel)	Minimum sensitivity [dBm] (40 MHz channel)
BPSK (6.5Mbits)	1/2	16	32	-82	-79
QPSK (13Mbits)	1/2	13	29	-79	-76
QPSK (19Mbits)	3/4	11	27	-77	-74
16QAM (26Mbits)	1/2	8	24	-74	-71
16QAM (39Mbits)	3/4	4	20	-70	-67
64QAM (52Mbits)	2/3	0	16	-66	-63
64QAM (58.5Mbits)	3/4	-1	15	-65	-62
64QAM (65Mbits)	5/6	-2	14	-64	-61

6.3 Measuring the adjacent channel rejection

In 802.11 standard, the adjacent and alternate channel rejection parameters are defined by the clauses 15.4.8.3, 17.3.10.2, 18.4.8.3, 19.5.2 and 19.6.2 [47]. The non-adjacent channel rejection is defined only for OFDM in clause 17.3.10.3. The 802.11n defines adjacent channel rejection in clause 20.3.22.2 and non-adjacent channel rejection in 20.3.22.3.

The most important parts of these definitions are cited below:

- For basic DSSS bit rates, the parameter "*shall be measured using the following method: Input a 2 Mb/s DQPSK modulated signal at a level 6 dB greater than* (the receiver minimum sensitivity) *specified in 15.4.8.1. In an adjacent channel (≥ 30 MHz separation as defined by the channel numbering), input a signal modulated in a similar fashion that adheres to the transmit mask specified in 15.4.7.4 to a level 41 dB above the* (receiver minimum sensitivity) *level specified in 15.4.8.1. The adjacent channel signal shall be derived from a separate signal source. It cannot be a frequency shifted version of the reference channel. Under these conditions, the FER shall be no worse than* 8×10^{-2}."
- For HR/DSSS bit rates (802.11b): "*Adjacent channel rejection is defined between any two channels with > 25 MHz separation in each channel group, as defined in 18.4.6.2. The adjacent channel rejection shall be equal to or better than 35 dB, with an FER of 8×10–2 using 11 Mbit/s CCK modulation described in 18.4.6.3 and a PSDU length of 1024 octets. The adjacent channel rejection shall be measured using the following method. Input an 11 Mb/s CCK modulated signal at a level 6 dB greater than* (receiver min. sensitivity) *specified in 18.4.8.1. In an adjacent channel (≥ 25 MHz separation as defined by the channel numbering), input a signal modulated in a similar fashion, which adheres to the transmit mask specified in 18.4.7.3, to a level 41 dB above the* (receiver min. sensitivity) *level specified in 18.4.8.1. The adjacent channel signal shall be derived from a separate signal source. It cannot be a frequency shifted version of the reference channel. Under these conditions, the FER shall be no worse than* 8×10^{-2}."
- For OFDM modulation, the parameter "*shall be measured by setting the desired signal's strength 3 dB above the rate-dependent sensitivity specified in Table 17-13/20-22 and raising the power of the interfering signal until 10% PER is caused for a PSDU length of ... octets. The power difference between the interfering channel and the desired channel is the corresponding adjacent channel rejection.*

  (...)

> *The interfering signal in the adjacent channel shall be a conformant OFDM signal, unsynchronized with the signal in the channel under test. For a conforming OFDM PHY, the corresponding rejection shall be no less than specified in Table 20-22. The interference signal shall have a minimum duty cycle of 50%.*"

As both minimum receiver sensitivity and adjacent channel rejection require the packet or frame error rates (PER/FER) to be measured, we need to look for the methods for such measurements.

6.4 Measuring the PER, FER and BER

This chapter is dedicated to the methods for measuring the packet, frame and bit error rates. Only the basic principles are described here, the actual measurement configuration must be designed by using some of these principles according to the devices and equipment available.

6.4.1 Measuring the PER/FER/BER using vector signal generator(s)

The method is proposed by Rohde&Schwarz and is described by the application note [27].

The principle is shown on the figure 45. The method requires a vector signal generator able to produce 802.11 signals, or two signal generators in case of adjacent channel rejection measurements. The generators provide low-noise baseband signal generation, frequency stability and RF modulator overall quality, which would be hardly achieved by any golden device.

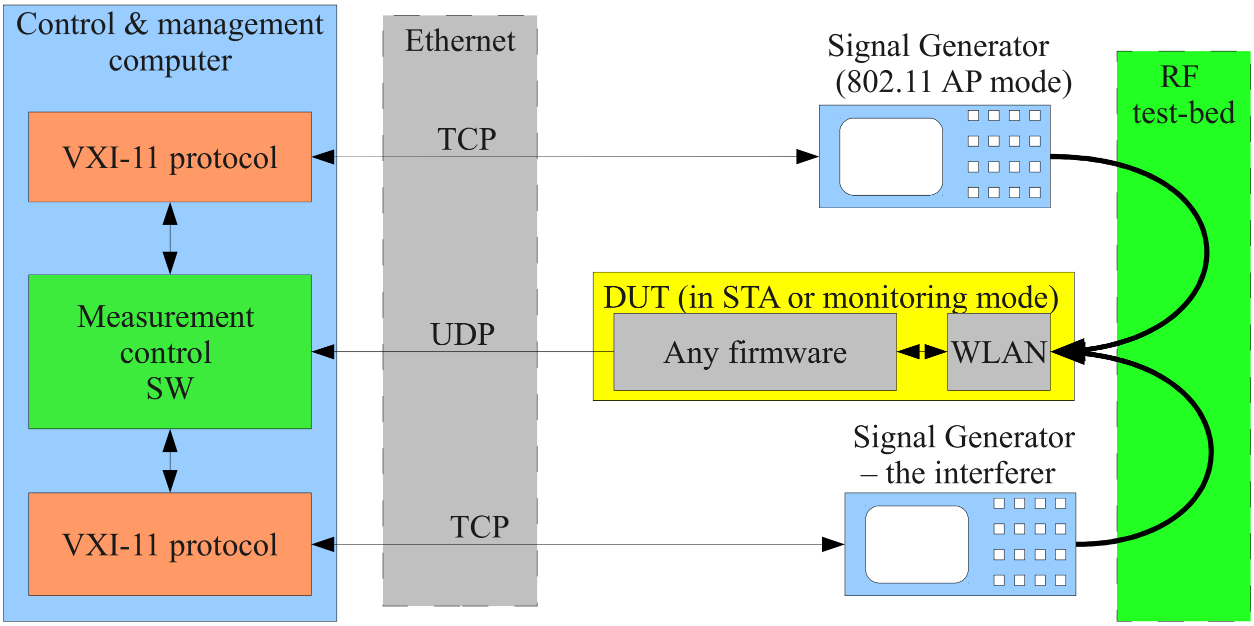

Control & management computer
VXI-11 protocol
Measurement control SW
VXI-11 protocol
Ethernet
TCP
UDP
TCP
Signal Generator (802.11 AP mode)
DUT (in STA or monitoring mode)
Any firmware
WLAN
Signal Generator – the interferer
RF test-bed

Figure 45: PER/FER measurment using vector signal generators

The first generator emulates the 802.11 AP by generating periodic sequence of 802.11 positive probe responses, beacons and data frames with the DUT MAC destination address, so the DUT in station mode can associate with the emulated AP, although the generator actually can not hear the station responses. For measuring PER/FER, the generator inserts sequential numbers into the data frames/packets. The DUT station receives data frames at the application level and as the generator does not do any MAC-layer frame retransmissions, the application can easily calculate PER=FER by counting the missing frame sequence numbers.

The original method does not allow measuring BER. However, BER can be possibly measured using DUT in monitor mode, as explained in the following chapter(s).

6.4.2 Analyzing and generating 802.11 traffic using WLAN devices in monitor mode

The problem with FER measurement tools is that they must work with the 802.11 MAC layer, where automatic packet retransmissions and also the FEC (Forward Error Checking) take place. While MAC layer packet retransmissions can be disabled in some 802.11 devices, the FEC can not. The FEC influences the receiver chain measurements heavily, because it masks the bit errors in a way that e.g. with devices based on Atheros chipsets, it is hard to see the 8% FER or 10% PER threshold defined by the 802.11 standard requirements for minimum receiver sensitivity and adjacent channel rejection testing. With FEC, the behaviour of the link can be as extreme as when it quickly jumps from <5% to the 100% PER when the signal strength reaches certain threshold and is further attenuated just by 1dB. The station then de-associates from the AP, which complicates the measurement and slows it down drastically.

Fortunately, most WLAN device drivers in Linux provide a special monitor mode, which allows receiving and injecting packets directly on/to the 802.11 MAC layer. This special mode is very popular among the wireless research groups – it is used in [14], [39], [79], [101] etc.

The WLAN monitor mode allows:

- measuring FER,
- measuring bit error rate (BER, see [14]), if the device driver monitor mode supports the „fcsfail“ a „plcpfail“ flags [9] [91],
- injecting (transmitting) and capturing (receiving) the 802.11 packets even without association with an AP,
- sending 802.11 packets without needing the ACK reply and without automatic retransmissions in case of missing ACK,
- setting arbitrary bit rates and control fields in the injected packets.

As the monitor mode is used mainly by the wireless-hacking community and few researchers, it still does not have a standard API. Therefore, commands to invoke and use the monitor mode differ between different wireless cards. Although there are efforts to standardize the wireless API in the Linux-wireless kernel developers group, the monitor mode is still one of the most cumbersome Linux-wireless functionalities – among other things, for many drivers it requires patching driver code in order to allow the packet injection.

An unified way to initiate the monitor mode on different device drivers is using the airmon-ng script, which is included in the aircrack-ng package. To inject and capture the 802.11 packets without having to write a lot of C code, it is possible to use the Scapy framework [98] [99]. For an example of ANSI C packet injector code, see the inject.c source file in Jouni Malinen's hostpad wlantest tool [69].

6.4.3 Measuring FER and BER with devices in monitor mode

With the tools mentioned in the previous chapter, the PER/FER/BER testing method/architecture can be designed as shown on figure 46. The prerequisites are:

- DUT and golden device support the monitor mode.
- Golden device monitor mode supports packet injection.
- If possible, the DUT monitor mode should support the the „fcsfail“ a „plcpfail“ flags.

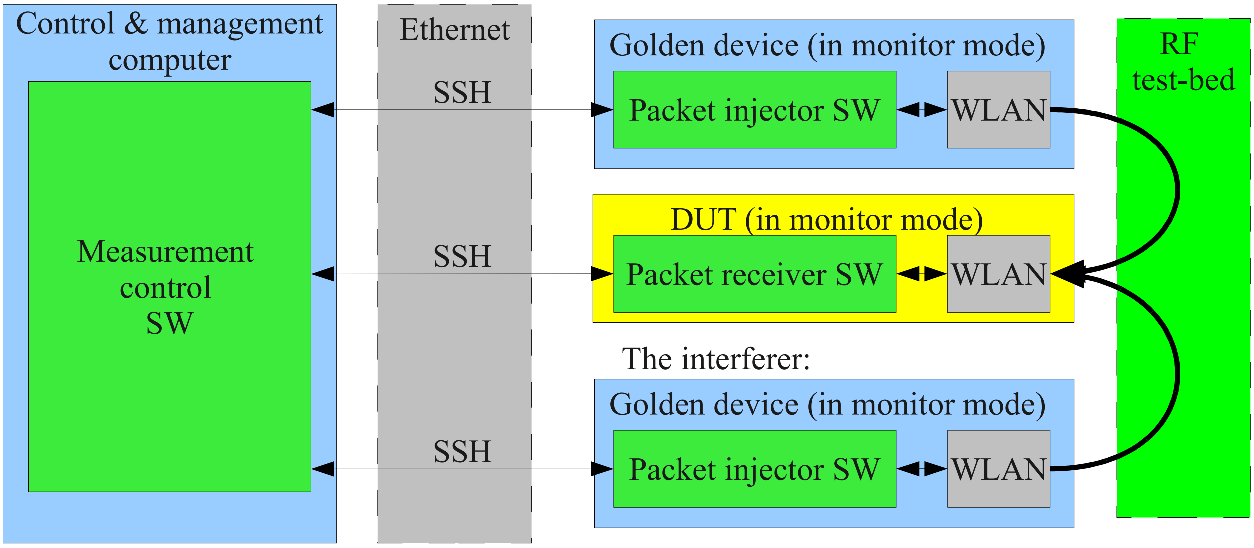


Figure 46: Measuring PER/FER/BER with devices in monitor mode

For measuring the receiver sensitivity, only two devices are needed. For measuring adjacent channel rejection parameter, an interfering device must be added – either a signal generator, or another golden device, ideally in monitor mode with support for injection.

Generating (injecting) the 802.11 packets

Although a signal generator would make the measurement results mode credible, we still do not own one, so we must generate the 802.11 traffic by using WLAN devices in monitor mode.

Note that there are already SW tools for such measurements – e.g., the Atheros radio test (ART) utility [128] used by Anritsu MT8860B WLAN test-set [24]. However, this tool is only available to Atheros partners and is closely bound to a specific hardware board. Therefore it is not usable for measuring generic devices.

The packet injector for measuring FER/BER has functionality similar to the Broadcaster tool (see chapter 11.2), which sends packets filled with a sequential number of the packet. The sequential packet numbers are then used by the packet receiver for computing the FER/BER.

Receiving (sniffing) the 802.11 packets

The packet receiver SW must capture the 802.11 packets and:

- compute the FER by counting the missing packets – this is the only possibility for cases when the „fcsfail" a „plcpfail" flags are **not** supported by the WLAN device driver.
- Compute BER (when the „fcsfail" a „plcpfail" flags are available [9]).

For calculating BER, the receiver application must evaluate whether the received packet is corrupted and if it is, it must guess its sequential number. This is an easy task, because every packet is filled by the same sequential number, so the majority function can be used. An additional check for the "sequentiality" of the resulting number must be made for the case when BER is much greater than 50% (and the majority function may give an incorrect result). After finding the correct sequential number, the BER can be computed by counting the corrupted bits in every byte.

6.4.4 PER measurement using devices without monitor mode

If the DUT does not support monitor mode, then PER can be measured using a configuration similar to figure 47.

Note that the interferer must be either a signal generator, a golden device in monitoring mode with injection or a golden device with reliable broadcasting in AP or ad-hoc mode without association to fulfil the 802.11 standard requirement that "*The interference signal shall have a minimum duty cycle of 50%*".

The UDP protocol should be used, TCP is unusable as it does packet retransmissions with exponential backoff. The udp-sender software is basically the same application as the Broadcaster (see chapter 11.2) but with standard unicast socket options. The standard iperf application in UDP client mode (`iperf -uc <HOST_IP>`) can be used as well – it does not even need the server on the other side to be started.

If the golden device allows disabling the MAC layer retransmissions, then this method can be used also for measuring the FER.

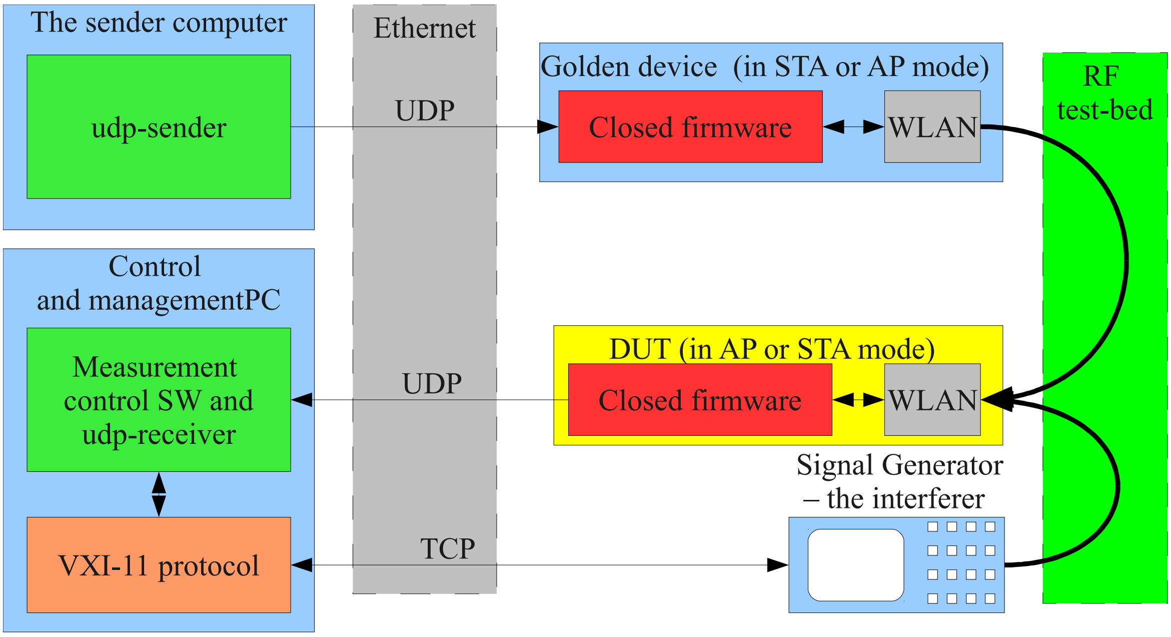


Figure 47: Measuring PER with devices in standard mode

A variant of this configuration without the generator and with two golden devices is shown on figure 48. For isolation of the broadcasting domain, a VLAN network is used. This configuration was actually used for adjacent channel rejection measurements in chapter 7.2.2.

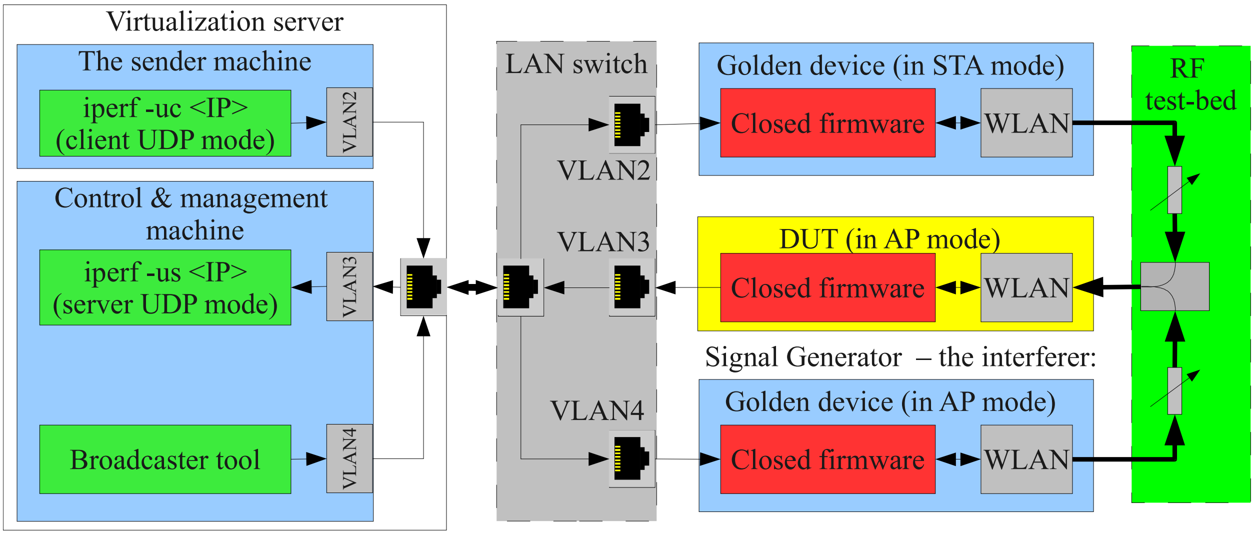


Figure 48: Measuring adjacent channel rejection with 3 devices in standard modes

6.5 Measuring CS/CCA

The chapter 4.7.4 contains CCA parameters for in-channel operation, however, we are more interested in the CCA behaviour in case of adjacent channel interferences from co-located transmitters. The 802.11 standards do not specify this case, but the practical measurement results in chapter 7.3 show that the CCA performance in such conditions is often more critical than the adjacent channel rejection.

To emulate the adjacent channel interference, an interferer signal generator must be used. If it is not available, a golden device can be used if its CCA can be disabled completely (e.g., the Mikrotik nstreme) or its CCA is much less sensitive to adjacent channel interference than the DUT CCA.

The CCA performance can be then measured by several methods:

- Sending a special 802.11 MAC frame to the DUT with request to undertake the CCA measurement. Sending the request and displaying the measurement results must be supported by the DUT and golden devices firmware. Details can be found in 802.11 clauses 7.3.2.21 “Measurement Request element”, 7.3.2.21.2 “CCA request” and 7.3.2.22.2 “CCA report”.
- Invoke the DUT transmission in a standard mode (AP or STA). As the interference level increases beyond the CCA detection threshold, the DUT CCA will be postponing or completely stopping the transmissions, which can be detected as reduction of throughput.
  This method is the most generic one. The testbed configuration is similar to figure 48, only the DUT traffic direction is reversed: the DUT is now transmitting the packets using the iperf tool in UDP client mode. The first golden device is receiving the UDP packets with iperf in UDP server mode and displays the throughput, packet loss and jitter.
  The second golden device generates the interfering traffic in adjacent channel by using the broadcaster tool. In practical measurements, we used the Ubiquity Bullet device, because it has the best CCA performance in adjacent channel interference conditions.

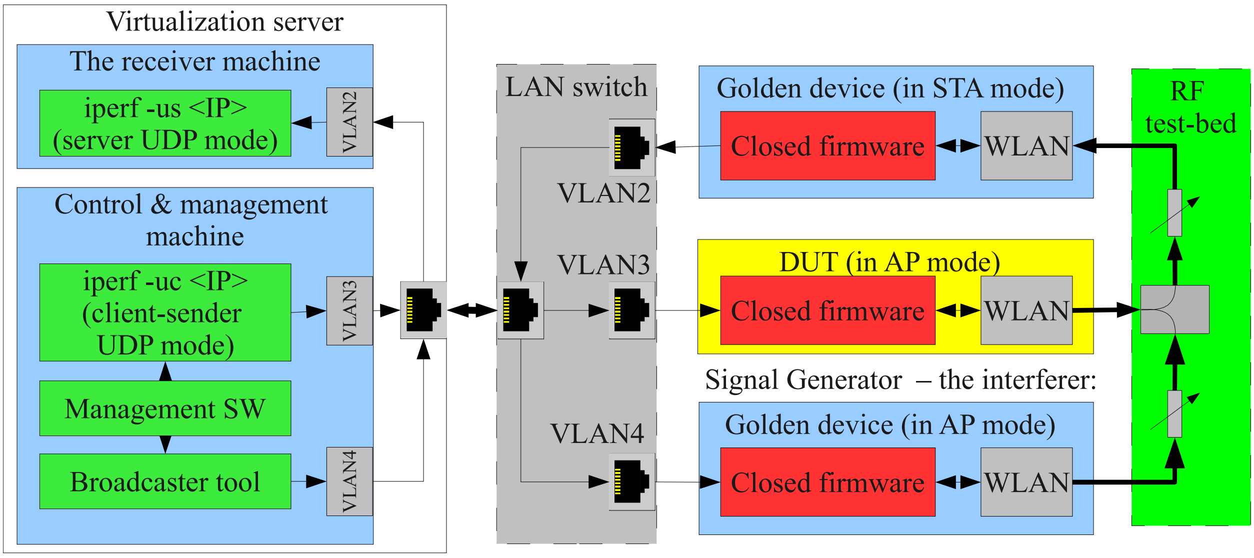


Figure 49: Measuring CCA using 2 golden devices

7 PRACTICAL MEASUREMENTS EXAMPLES

7.1 Transmit chain measurement

In this chapter, standard-compliance measurements of three 802.11 transmitters will be shown.

7.1.1 AzureWave AR5BXB63 AW-GE780

This miniPCI-express 802.11b/g card is used in Asus EEE netbooks. For identification, the card top and bottom side photo is included below:

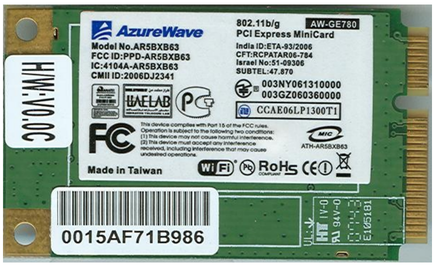


Figure 50: AzureWave AR5BXB63 (AW-GE780) miniPCIe card

In Linux, the card is controlled by the ath5k driver version is 0.6.0 with antenna selection patch:

```
# modinfo ath5k
filename:       /lib/modules/2.6.30-1-
686/updates/drivers/net/wireless/ath/ath5k/ath5k.ko
version:        0.6.0 (EXPERIMENTAL)
```

Another driver version from Debian Squeeze 2.6.39 backports kernel version was tried too, but its antenna selection algorithm was set to "antenna diversity" without any possibility to change it, therefore this version was not used.

The AR2425/AR5007 chipset is the last and final generation of the Atheros 5000 chipsets series. Unlike its predecessors, the AR5007 is a single-chip solution with PCIexpress bus.

The card was inserted into the miniPCIe slot of Intel 945GSEJT miniITX mainboard and connected to the testbed by uFL-to-RSMA and RSMA-to-N pigtails.

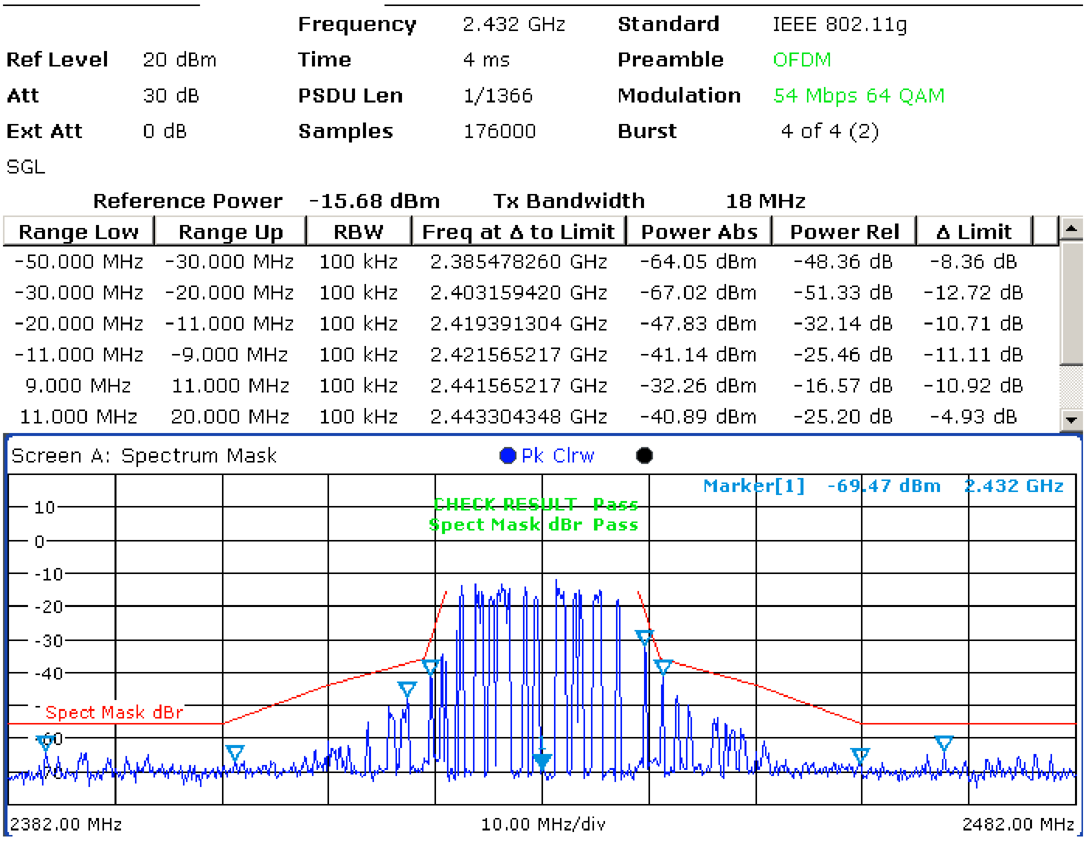
The ath5k driver had problems with broadcasts performance, so the Broadcaster tool (see chapter 11.2) was not used. Therefore the card was set into the station mode, associated with the Ubiquity Bullet2 access point and then iperf tools was used to generate traffic which was analysed by the FSV7.

In this configuration, the card passed all EVM and timing transmitter compliance tests for all bit-rates. However, in the 54Mbit/s (64QAM) mode, the measured parameters were on the edge of the EVM limits and some packets even failed the EVM measurement. Therefore, the FSV7 was set to analyze 4 packets bursts so the measurements results are shown with the minimum, mean and maximum measured values.

For the 54 Mbit/s rate, setting TX power manually below 13 dBm seemed to improve the results a bit, but did not completely avoid the EVM compliance failures.

When compared to other compliant devices, it might seem that 2 or 3 dB difference in EVM values is not that much, but the problem is that the attenuation of real wireless link adds more noise which increases the EVM even more. This is demonstrated on figures 55 - 62: with the (very minimum) 50dB attenuation added for protecting the other receiver, the EVM for higher bit rates 36-54 Mbit/s is out of bounds. This does not happen with the Ubiquity, Cisco and Mikrotik R52 devices we have measured under the same conditions.

In real (free space) wireless links, the card simply refuses to use these higher bit rates.



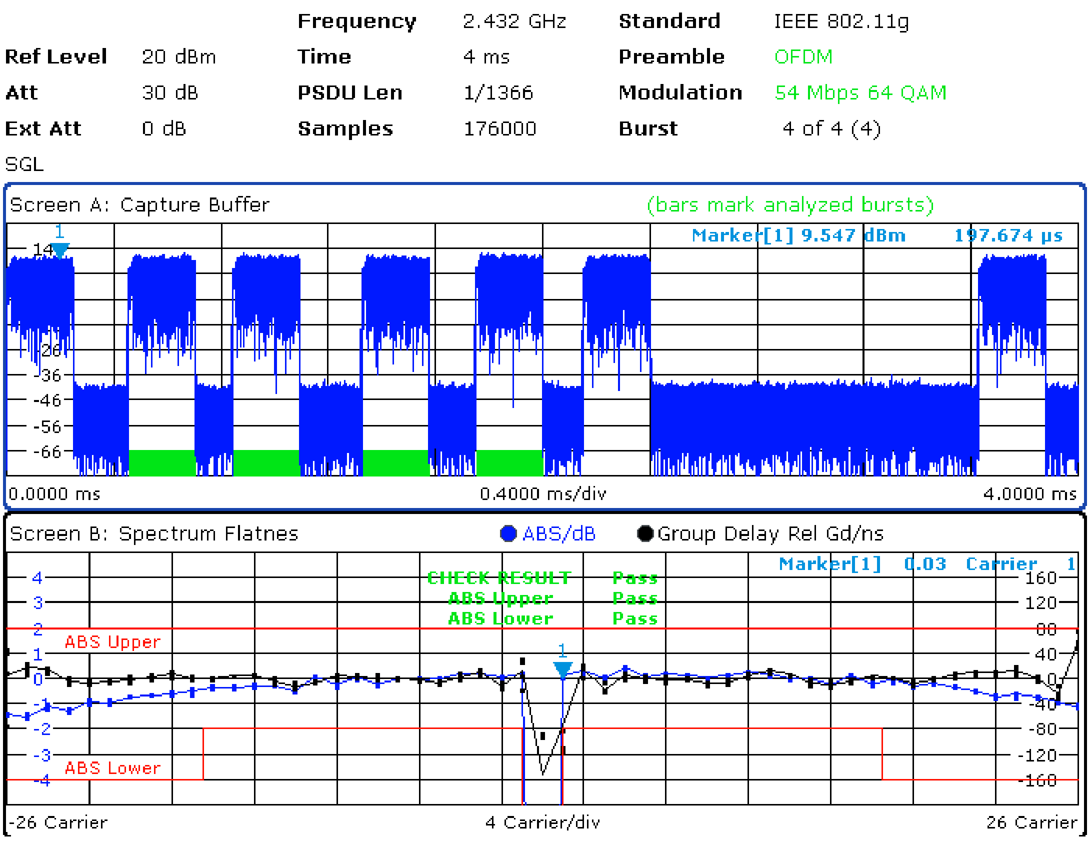
		Frequency	2.432 GHz	Standard	IEEE 802.11g
Ref Level	20 dBm	Time	4 ms	Preamble	OFDM
Att	30 dB	PSDU Len	1/1366	Modulation	54 Mbps 64 QAM
Ext Att	0 dB	Samples	176000	Burst	4 of 4 (2)

SGL

Reference Power -15.68 dBm Tx Bandwidth 18 MHz

Range Low	Range Up	RBW	Freq at Δ to Limit	Power Abs	Power Rel	Δ Limit
-50.000 MHz	-30.000 MHz	100 kHz	2.385478260 GHz	-64.05 dBm	-48.36 dB	-8.36 dB
-30.000 MHz	-20.000 MHz	100 kHz	2.403159420 GHz	-67.02 dBm	-51.33 dB	-12.72 dB
-20.000 MHz	-11.000 MHz	100 kHz	2.419391304 GHz	-47.83 dBm	-32.14 dB	-10.71 dB
-11.000 MHz	-9.000 MHz	100 kHz	2.421565217 GHz	-41.14 dBm	-25.46 dB	-11.11 dB
9.000 MHz	11.000 MHz	100 kHz	2.441565217 GHz	-32.26 dBm	-16.57 dB	-10.92 dB
11.000 MHz	20.000 MHz	100 kHz	2.443304348 GHz	-40.89 dBm	-25.20 dB	-4.93 dB

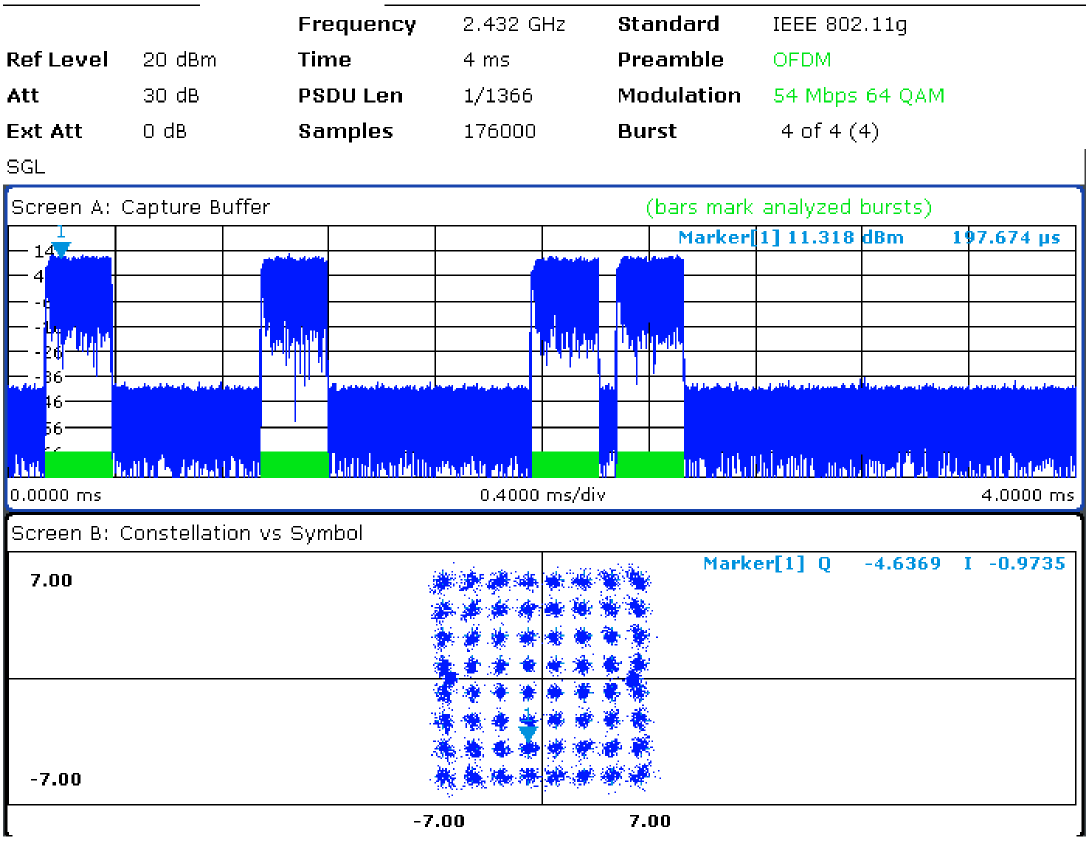
Figure 51: AzureWave @ 54Mbit/s: Spectrum mask



		Frequency	2.432 GHz	Standard	IEEE 802.11g
Ref Level	20 dBm	Time	4 ms	Preamble	OFDM
Att	30 dB	PSDU Len	1/1366	Modulation	54 Mbps 64 QAM
Ext Att	0 dB	Samples	176000	Burst	4 of 4 (4)

SGL

Figure 52: AzureWave @ 54Mbit/s: Spectrum flatness

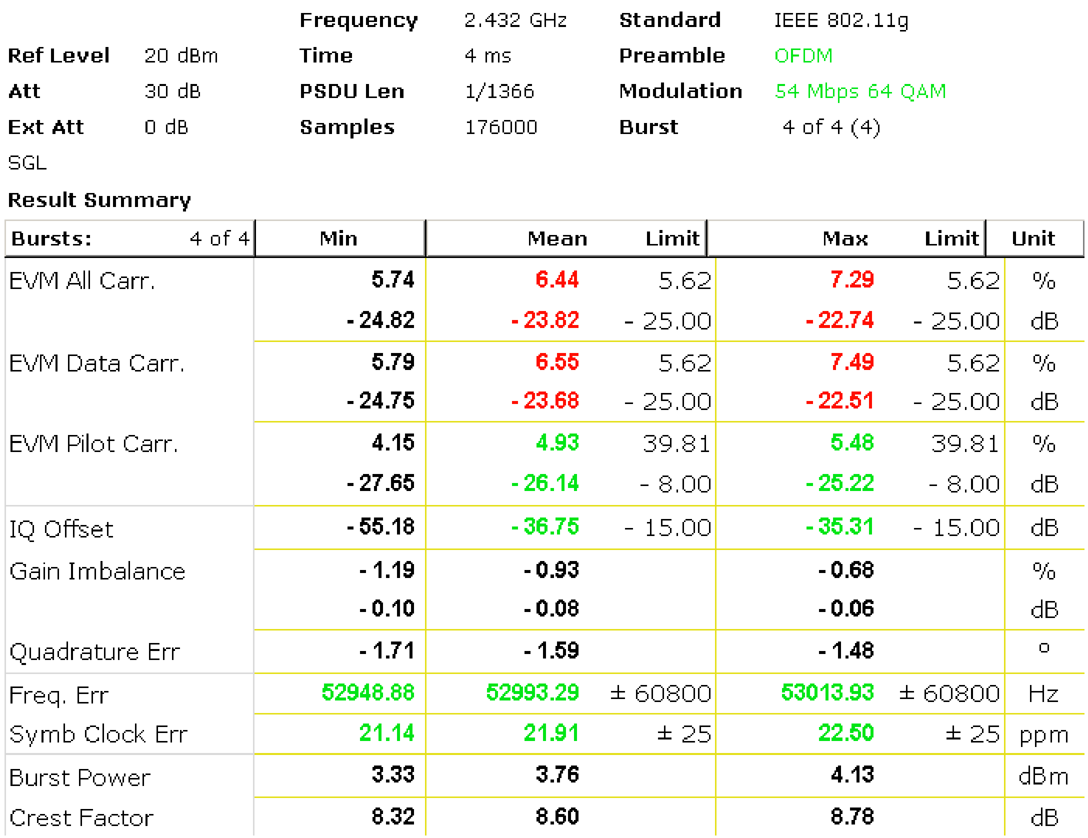


		Frequency	2.432 GHz	**Standard**	IEEE 802.11g
Ref Level	20 dBm	**Time**	4 ms	**Preamble**	OFDM
Att	30 dB	**PSDU Len**	1/1366	**Modulation**	54 Mbps 64 QAM
Ext Att	0 dB	**Samples**	176000	**Burst**	4 of 4 (4)

SGL

Screen A: Capture Buffer
(bars mark analyzed bursts)
Marker[1] 11.318 dBm 197.674 µs
14
4
-6
-16
-26
-36
-46
-56
0.0000 ms
0.4000 ms/div
4.0000 ms
Screen B: Constellation vs Symbol
Marker[1] Q -4.6369 I -0.9735
7.00
-7.00
-7.00
7.00

Figure 53: AzureWave @ 54Mbit/s: PVT with constellation diagram



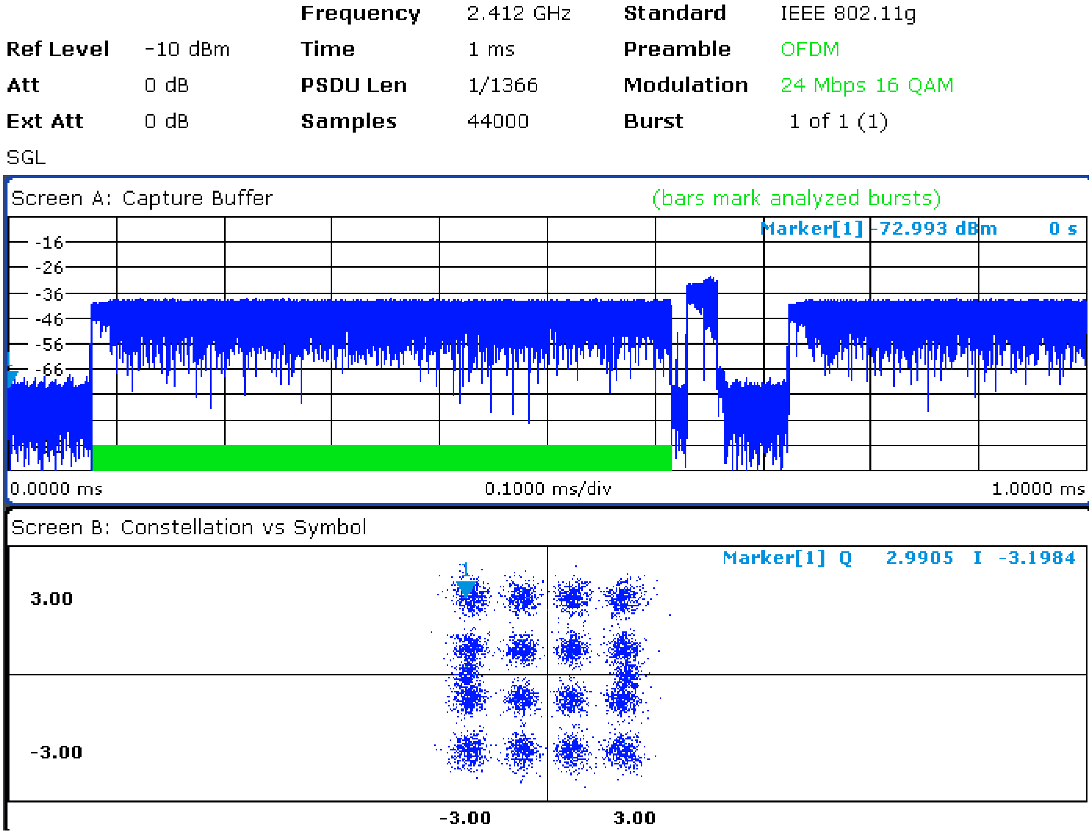
		Frequency	2.432 GHz	**Standard**	IEEE 802.11g
Ref Level	20 dBm	**Time**	4 ms	**Preamble**	OFDM
Att	30 dB	**PSDU Len**	1/1366	**Modulation**	54 Mbps 64 QAM
Ext Att	0 dB	**Samples**	176000	**Burst**	4 of 4 (4)

SGL

Result Summary

Bursts: 4 of 4	**Min**	**Mean**	**Limit**	**Max**	**Limit**	**Unit**
EVM All Carr.	5.74	6.44	5.62	7.29	5.62	%
	- 24.82	- 23.82	- 25.00	- 22.74	- 25.00	dB
EVM Data Carr.	5.79	6.55	5.62	7.49	5.62	%
	- 24.75	- 23.68	- 25.00	- 22.51	- 25.00	dB
EVM Pilot Carr.	4.15	4.93	39.81	5.48	39.81	%
	- 27.65	- 26.14	- 8.00	- 25.22	- 8.00	dB
IQ Offset	- 55.18	- 36.75	- 15.00	- 35.31	- 15.00	dB
Gain Imbalance	- 1.19	- 0.93		- 0.68		%
	- 0.10	- 0.08		- 0.06		dB
Quadrature Err	- 1.71	- 1.59		- 1.48		°
Freq. Err	52948.88	52993.29	± 60800	53013.93	± 60800	Hz
Symb Clock Err	21.14	21.91	± 25	22.50	± 25	ppm
Burst Power	3.33	3.76		4.13		dBm
Crest Factor	8.32	8.60		8.78		dB

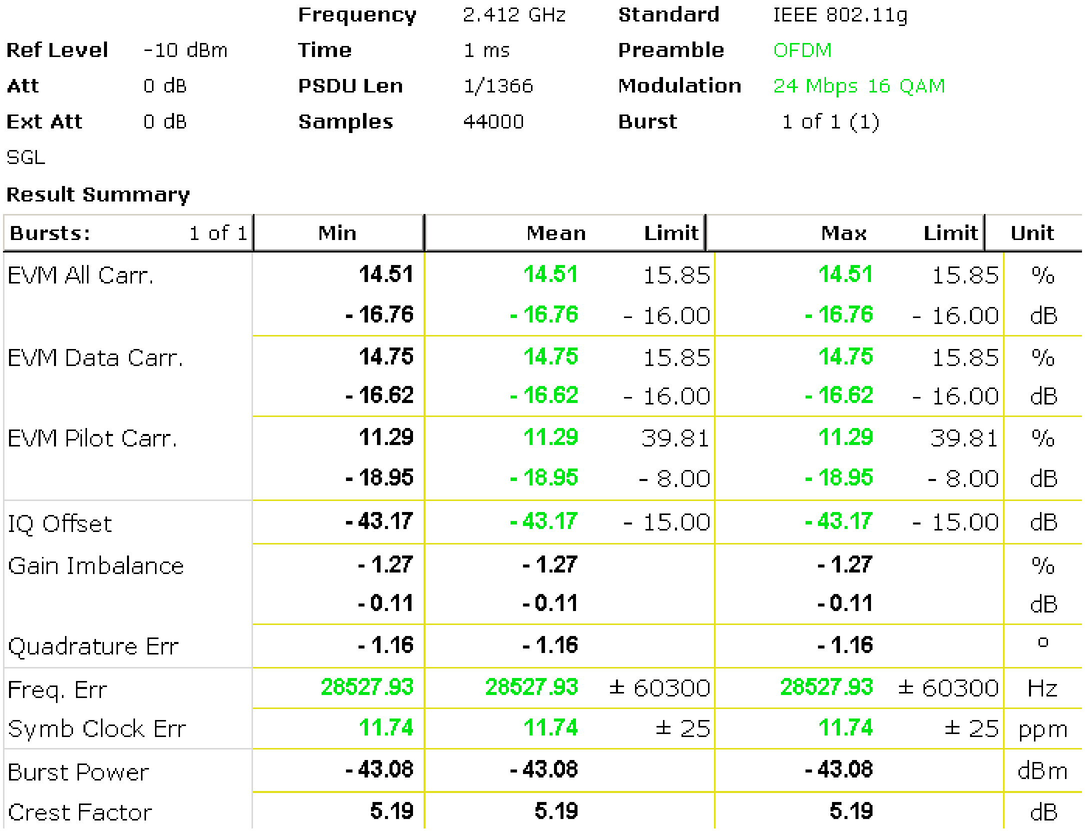
Figure 54: Figure 49: AzureWave @ 54Mbit/s: transmitter compliance failure



		Frequency	2.412 GHz	Standard	IEEE 802.11g
Ref Level	-10 dBm	Time	1 ms	Preamble	OFDM
Att	0 dB	PSDU Len	1/1366	Modulation	24 Mbps 16 QAM
Ext Att	0 dB	Samples	44000	Burst	1 of 1 (1)

SGL

Figure 55: AzureWave @ 24Mbit/s, 50dB attenuation: PVT with constellation



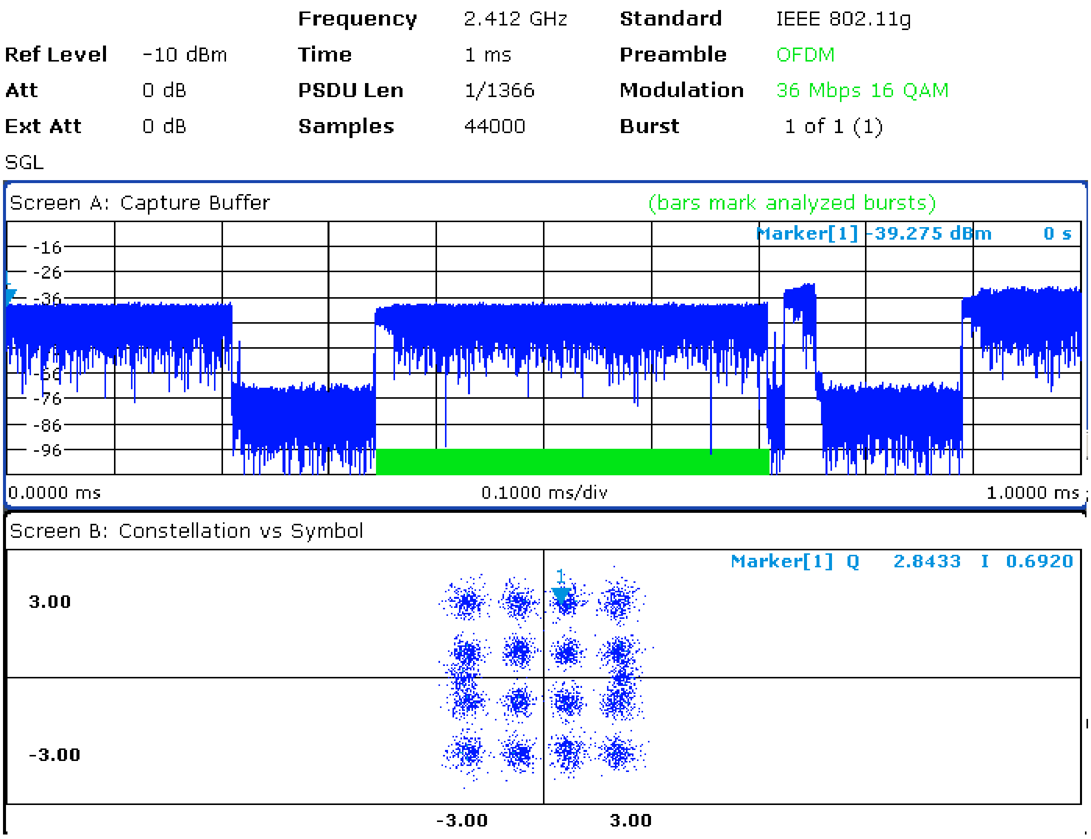
		Frequency	2.412 GHz	Standard	IEEE 802.11g
Ref Level	-10 dBm	Time	1 ms	Preamble	OFDM
Att	0 dB	PSDU Len	1/1366	Modulation	24 Mbps 16 QAM
Ext Att	0 dB	Samples	44000	Burst	1 of 1 (1)

SGL

Result Summary

Bursts: 1 of 1	Min	Mean	Limit	Max	Limit	Unit
EVM All Carr.	14.51	14.51	15.85	14.51	15.85	%
	- 16.76	- 16.76	- 16.00	- 16.76	- 16.00	dB
EVM Data Carr.	14.75	14.75	15.85	14.75	15.85	%
	- 16.62	- 16.62	- 16.00	- 16.62	- 16.00	dB
EVM Pilot Carr.	11.29	11.29	39.81	11.29	39.81	%
	- 18.95	- 18.95	- 8.00	- 18.95	- 8.00	dB
IQ Offset	- 43.17	- 43.17	- 15.00	- 43.17	- 15.00	dB
Gain Imbalance	- 1.27	- 1.27		- 1.27		%
	- 0.11	- 0.11		- 0.11		dB
Quadrature Err	- 1.16	- 1.16		- 1.16		°
Freq. Err	28527.93	28527.93	± 60300	28527.93	± 60300	Hz
Symb Clock Err	11.74	11.74	± 25	11.74	± 25	ppm
Burst Power	- 43.08	- 43.08		- 43.08		dBm
Crest Factor	5.19	5.19		5.19		dB

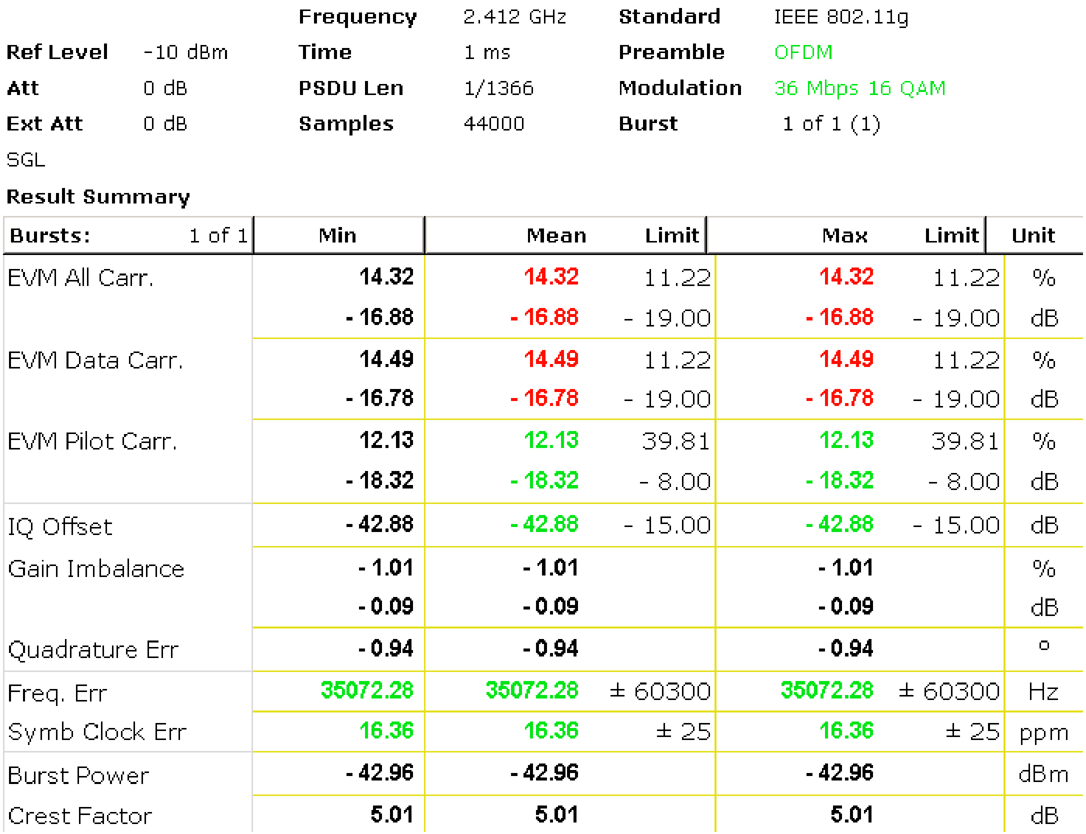
Figure 56: AzureWave @ 24Mbit/s, , 50dB attenuation: transmitter paragraphs



		Frequency	2.412 GHz	Standard	IEEE 802.11g
Ref Level	-10 dBm	Time	1 ms	Preamble	OFDM
Att	0 dB	PSDU Len	1/1366	Modulation	36 Mbps 16 QAM
Ext Att	0 dB	Samples	44000	Burst	1 of 1 (1)
SGL					

Screen A: Capture Buffer
(bars mark analyzed bursts)
Marker[1] -39.275 dBm 0 s
-16
-26
-36
-66
-76
-86
-96
0.0000 ms
0.1000 ms/div
1.0000 ms
Screen B: Constellation vs Symbol
Marker[1] Q 2.8433 I 0.6920
1
3.00
-3.00
-3.00
3.00

Figure 57: AzureWave @ 36Mbit/s, 50dB attenuation: PVT with constellation

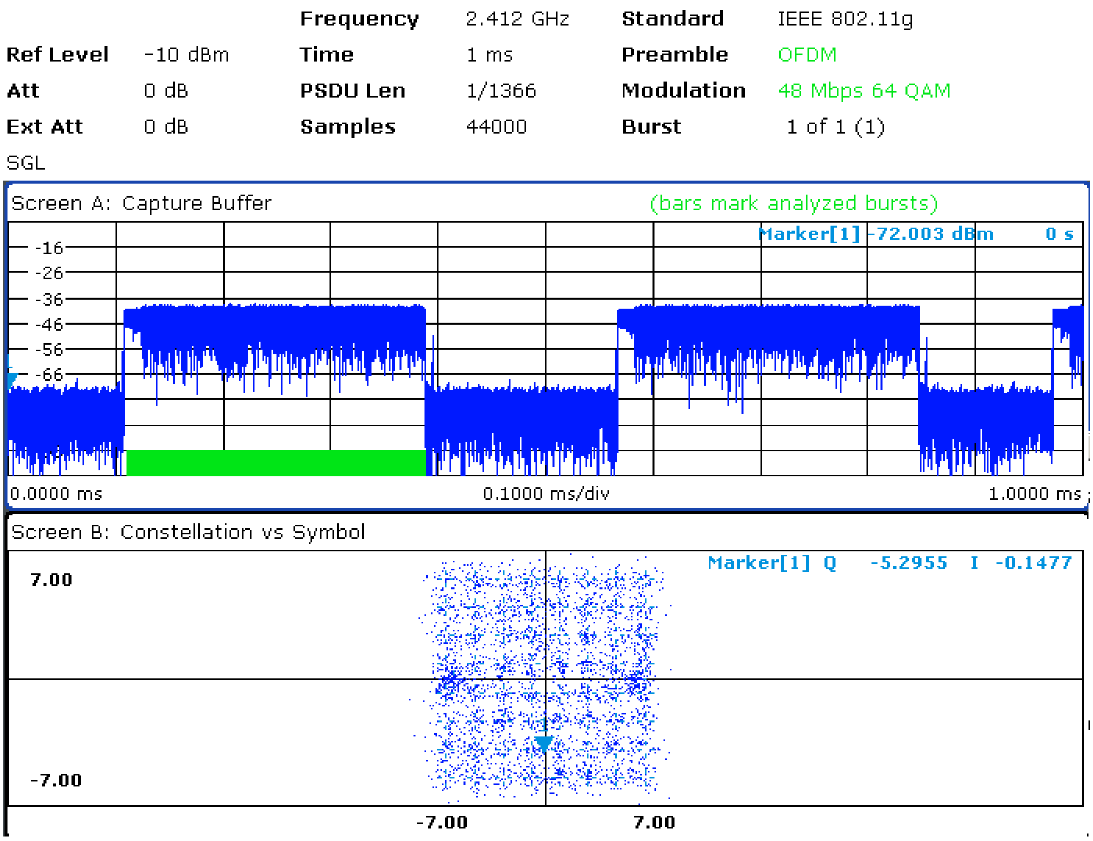


		Frequency	2.412 GHz	Standard	IEEE 802.11g
Ref Level	-10 dBm	Time	1 ms	Preamble	OFDM
Att	0 dB	PSDU Len	1/1366	Modulation	36 Mbps 16 QAM
Ext Att	0 dB	Samples	44000	Burst	1 of 1 (1)
SGL					

Result Summary

Bursts: 1 of 1	Min	Mean	Limit	Max	Limit	Unit
EVM All Carr.	14.32	14.32	11.22	14.32	11.22	%
	- 16.88	- 16.88	- 19.00	- 16.88	- 19.00	dB
EVM Data Carr.	14.49	14.49	11.22	14.49	11.22	%
	- 16.78	- 16.78	- 19.00	- 16.78	- 19.00	dB
EVM Pilot Carr.	12.13	12.13	39.81	12.13	39.81	%
	- 18.32	- 18.32	- 8.00	- 18.32	- 8.00	dB
IQ Offset	- 42.88	- 42.88	- 15.00	- 42.88	- 15.00	dB
Gain Imbalance	- 1.01	- 1.01		- 1.01		%
	- 0.09	- 0.09		- 0.09		dB
Quadrature Err	- 0.94	- 0.94		- 0.94		°
Freq. Err	35072.28	35072.28	± 60300	35072.28	± 60300	Hz
Symb Clock Err	16.36	16.36	± 25	16.36	± 25	ppm
Burst Power	- 42.96	- 42.96		- 42.96		dBm
Crest Factor	5.01	5.01		5.01		dB

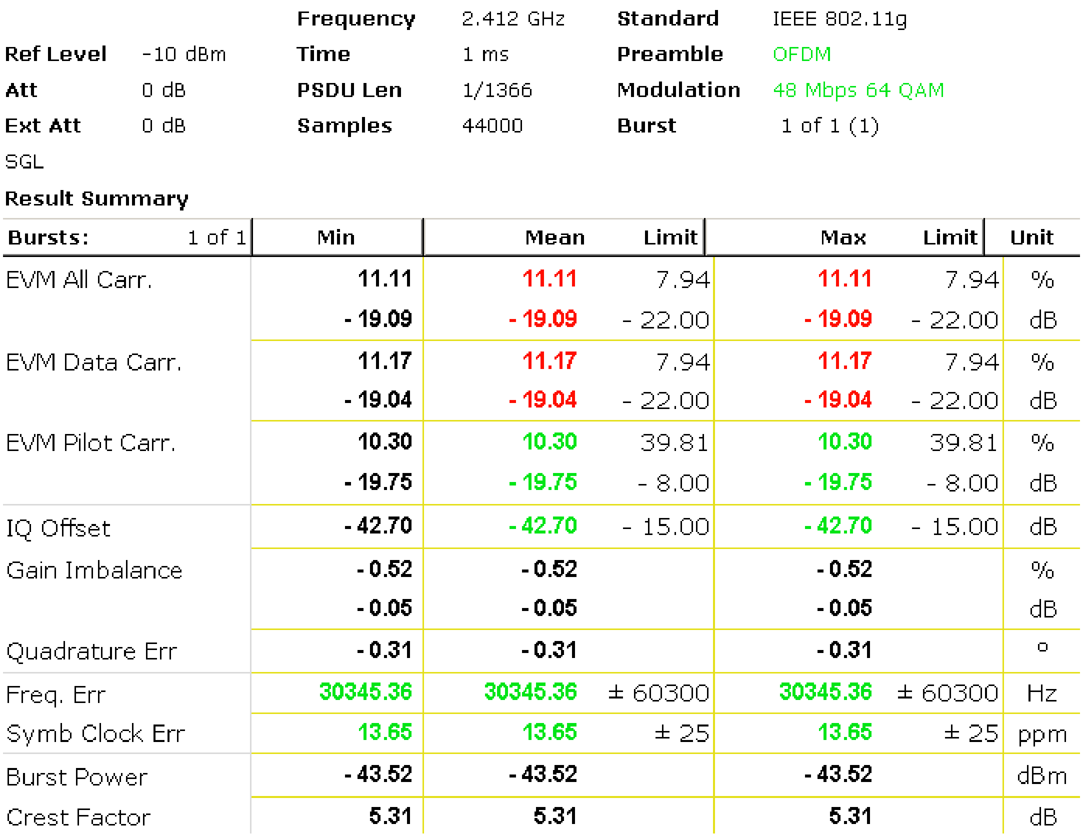
Figure 58: AzureWave @ 36Mbit/s, 50dB attenuation: signal parameters



		Frequency	2.412 GHz	Standard	IEEE 802.11g		
Ref Level	-10 dBm	Time	1 ms	Preamble	OFDM		
Att	0 dB	PSDU Len	1/1366	Modulation	48 Mbps 64 QAM		
Ext Att	0 dB	Samples	44000	Burst	1 of 1 (1)		
SGL							

Screen A: Capture Buffer
(bars mark analyzed bursts)
Marker[1] -72.003 dBm 0 s
-16
-26
-36
-46
-56
-66
0.0000 ms
0.1000 ms/div
1.0000 ms
Screen B: Constellation vs Symbol
Marker[1] Q -5.2955 I -0.1477
7.00
-7.00
-7.00
7.00

Figure 59: AzureWave @ 48Mbit/s, 50dB attenuation: PVT with constellation

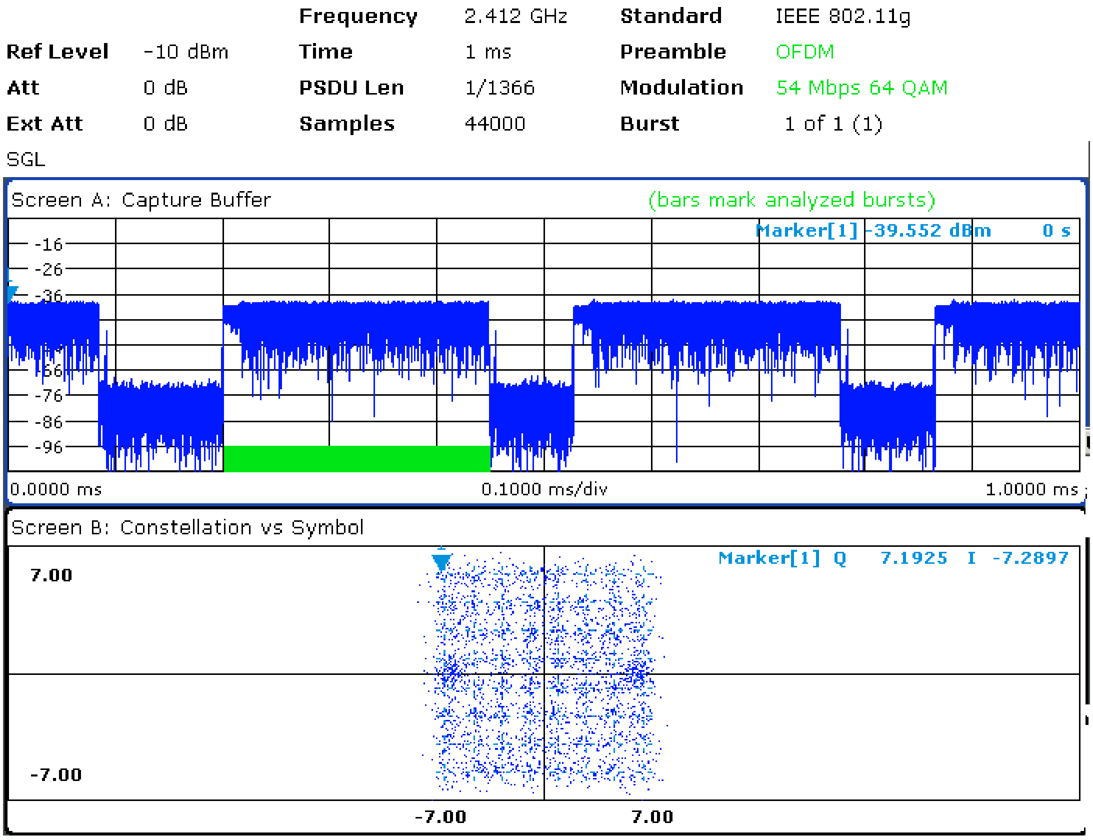


		Frequency	2.412 GHz	Standard	IEEE 802.11g
Ref Level	-10 dBm	Time	1 ms	Preamble	OFDM
Att	0 dB	PSDU Len	1/1366	Modulation	48 Mbps 64 QAM
Ext Att	0 dB	Samples	44000	Burst	1 of 1 (1)
SGL					

Result Summary

Bursts: 1 of 1	Min	Mean	Limit	Max	Limit	Unit
EVM All Carr.	11.11	11.11	7.94	11.11	7.94	%
	- 19.09	- 19.09	- 22.00	- 19.09	- 22.00	dB
EVM Data Carr.	11.17	11.17	7.94	11.17	7.94	%
	- 19.04	- 19.04	- 22.00	- 19.04	- 22.00	dB
EVM Pilot Carr.	10.30	10.30	39.81	10.30	39.81	%
	- 19.75	- 19.75	- 8.00	- 19.75	- 8.00	dB
IQ Offset	- 42.70	- 42.70	- 15.00	- 42.70	- 15.00	dB
Gain Imbalance	- 0.52	- 0.52		- 0.52		%
	- 0.05	- 0.05		- 0.05		dB
Quadrature Err	- 0.31	- 0.31		- 0.31		°
Freq. Err	30345.36	30345.36	± 60300	30345.36	± 60300	Hz
Symb Clock Err	13.65	13.65	± 25	13.65	± 25	ppm
Burst Power	- 43.52	- 43.52		- 43.52		dBm
Crest Factor	5.31	5.31		5.31		dB

Figure 60: AzureWave @ 48Mbit/s, 50dB attenuation: signal parameters

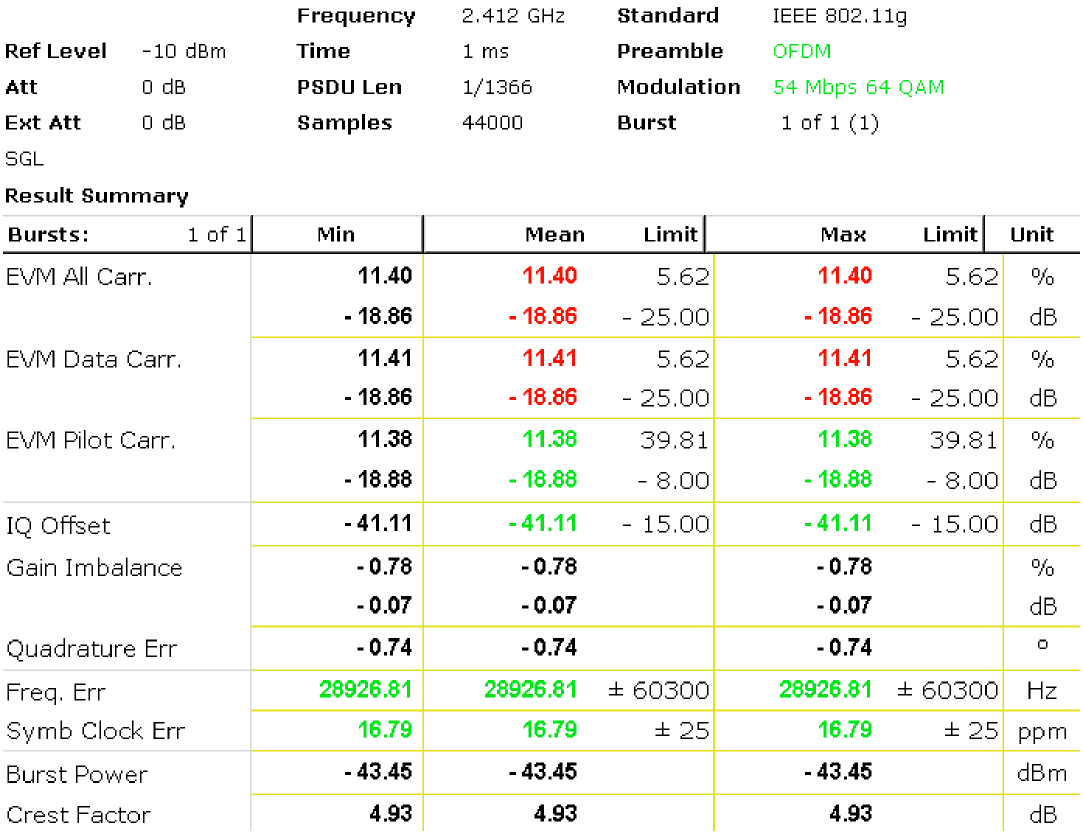


		Frequency	2.412 GHz	**Standard**	IEEE 802.11g
Ref Level	-10 dBm	**Time**	1 ms	**Preamble**	OFDM
Att	0 dB	**PSDU Len**	1/1366	**Modulation**	54 Mbps 64 QAM
Ext Att	0 dB	**Samples**	44000	**Burst**	1 of 1 (1)

SGL

Screen A: Capture Buffer
(bars mark analyzed bursts)
Marker[1] -39.552 dBm 0 s
-16
-26
-36
-66
-76
-86
-96
0.0000 ms
0.1000 ms/div
1.0000 ms
Screen B: Constellation vs Symbol
Marker[1] Q 7.1925 I -7.2897
7.00
-7.00
-7.00
7.00

Figure 61: AzureWave @ 54Mbit/s, 50dB attenuation: PVT with constellation



		Frequency	2.412 GHz	**Standard**	IEEE 802.11g
Ref Level	-10 dBm	**Time**	1 ms	**Preamble**	OFDM
Att	0 dB	**PSDU Len**	1/1366	**Modulation**	54 Mbps 64 QAM
Ext Att	0 dB	**Samples**	44000	**Burst**	1 of 1 (1)

SGL

Result Summary

Bursts: 1 of 1	**Min**	**Mean**	**Limit**	**Max**	**Limit**	**Unit**
EVM All Carr.	11.40	11.40	5.62	11.40	5.62	%
	- 18.86	- 18.86	- 25.00	- 18.86	- 25.00	dB
EVM Data Carr.	11.41	11.41	5.62	11.41	5.62	%
	- 18.86	- 18.86	- 25.00	- 18.86	- 25.00	dB
EVM Pilot Carr.	11.38	11.38	39.81	11.38	39.81	%
	- 18.88	- 18.88	- 8.00	- 18.88	- 8.00	dB
IQ Offset	- 41.11	- 41.11	- 15.00	- 41.11	- 15.00	dB
Gain Imbalance	- 0.78	- 0.78		- 0.78		%
	- 0.07	- 0.07		- 0.07		dB
Quadrature Err	- 0.74	- 0.74		- 0.74		°
Freq. Err	28926.81	28926.81	± 60300	28926.81	± 60300	Hz
Symb Clock Err	16.79	16.79	± 25	16.79	± 25	ppm
Burst Power	- 43.45	- 43.45		- 43.45		dBm
Crest Factor	4.93	4.93		4.93		dB

Figure 62: AzureWave @ 54Mbit/s, 50dB attenuation: signal parameters

7.1.2 Mikrotik R52

The miniPCI card has the Atheros AR5006XS/AR5414 chipset. For identification, the card photo is included below:

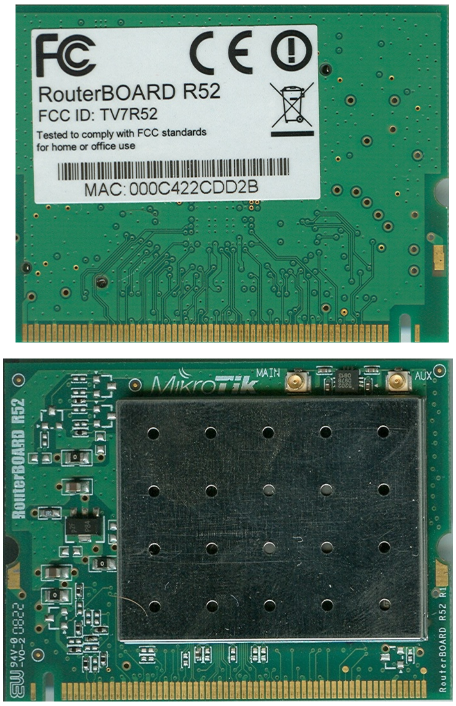


Figure 63: Mikrotik R52 card

The card was connected to the analyzer using the uFL to RSMA pigtail and RSMA to N-connector cable RF240. The Aeroflex 3-port 18 GHz resistive coupler at the spectrum analyzer input was used to combine the signal from card output with the signals from the testbed, so the card could e.g. associate with an AP. This was not necessary at the end, because the “broadcast in ad-hoc mode without association” measurement method could be used as the card allowed to set fixed bit rates in this mode (for explanation, see chapter 6.1).

The card was measured in two platforms - Intel D945GSEJT mainboard with RB14 PCI expander and the Routerboard RB411AR but the transmitter measurements were the same. The card fulfilled all the transmitter compliance tests in both 2.4 GHz and 5 GHz bands with a good reserve margin. Therefore, only the highest 54 Mbit/s bit rate is shown on the figures below.

Aside of the compliance measurements, the EVM dependence on TX power setting was measured. This measurement was done solely on RB411AR with RouterOS 5.2. The results are summarized in Table 4. The TX power column contains values used in the command line, e.g.:

```
[admin@RB411] > interface wireless set wlan3 tx-power-mode=all-rates-fixed tx-power=5
```

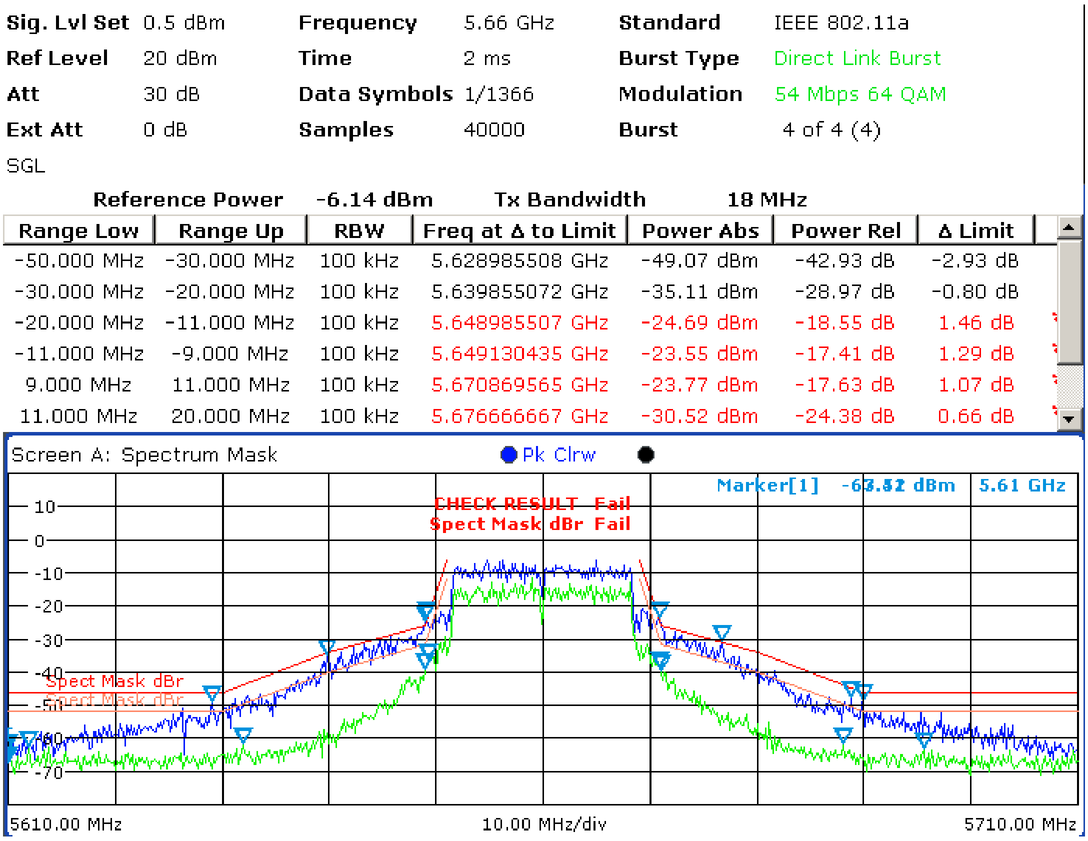
The measurement was done on 2452 MHz and 5660 MHz, the results are in columns 2.4GHz and 5GHz. Note that the EVM limit for bit rate 54 Mbit/s in both frequency bands is 5.62%.

Table 4: Mikrotik R52: the measured EVM values

tx-power	2.4GHz EVM	5 GHz EVM
5	1,97%	N/A
13	2,75%	3,49%
16	4,23%	5,34%
18	6,50%	8,12%
20	7,13%	11,04%

The moral is following:

1. users should never set the TX power to values higher than the defaults, which are stored in the card EEPROM, because
   - the transmitter is then unable to use the higher bit rates due to high EVM,
   - with TX power increased over the limit, the ratio of power levels in adjacent channel side-lobes to in-channel power level grows too and may cause failure of transmitter spectrum mask compliance. This is shown on figure 64 below.
2. Decreasing TX power decreases also the EVM.
3. Therefore, the best solution for achieving better range is not to over-drive the TX amplifier, but use better antennas with higher gain.



Sig. Lvl Set	0.5 dBm	**Frequency**	5.66 GHz	**Standard**	IEEE 802.11a
Ref Level	20 dBm	**Time**	2 ms	**Burst Type**	Direct Link Burst
Att	30 dB	**Data Symbols**	1/1366	**Modulation**	54 Mbps 64 QAM
Ext Att	0 dB	**Samples**	40000	**Burst**	4 of 4 (4)

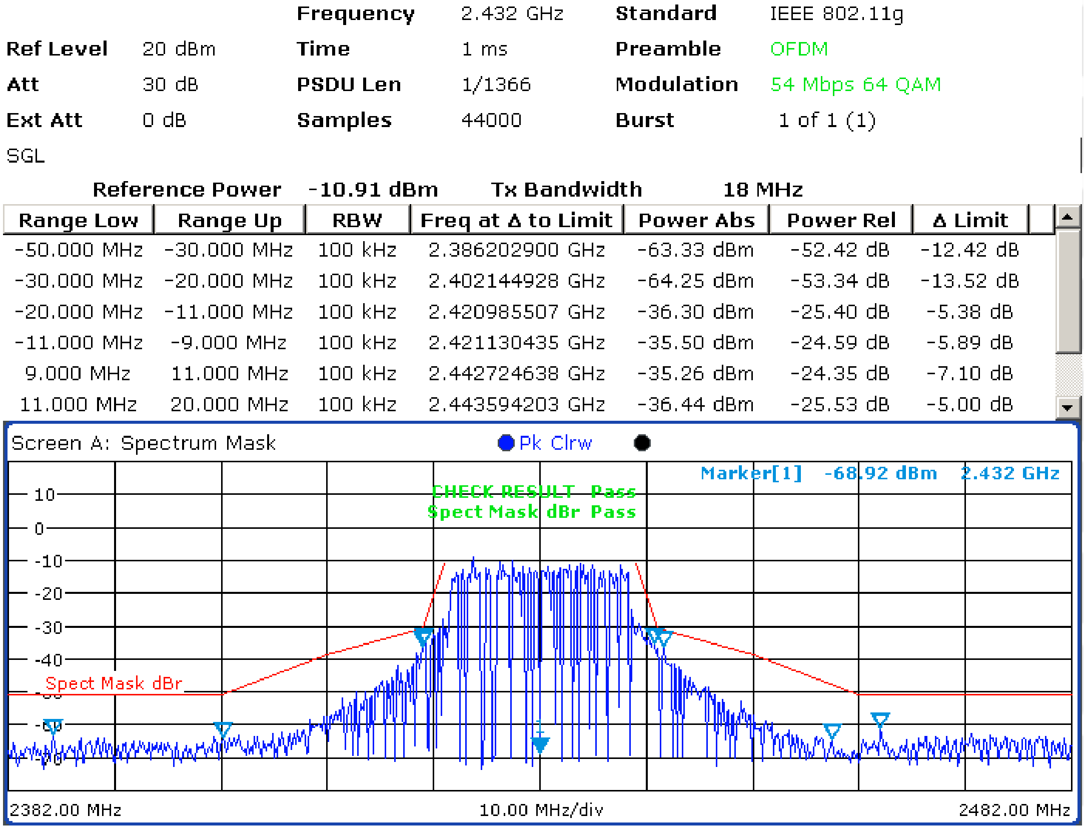
SGL

Reference Power **-6.14 dBm** **Tx Bandwidth** **18 MHz**

Range Low	Range Up	RBW	Freq at Δ to Limit	Power Abs	Power Rel	Δ Limit
-50.000 MHz	-30.000 MHz	100 kHz	5.628985508 GHz	-49.07 dBm	-42.93 dB	-2.93 dB
-30.000 MHz	-20.000 MHz	100 kHz	5.639855072 GHz	-35.11 dBm	-28.97 dB	-0.80 dB
-20.000 MHz	-11.000 MHz	100 kHz	5.648985507 GHz	-24.69 dBm	-18.55 dB	1.46 dB
-11.000 MHz	-9.000 MHz	100 kHz	5.649130435 GHz	-23.55 dBm	-17.41 dB	1.29 dB
9.000 MHz	11.000 MHz	100 kHz	5.670869565 GHz	-23.77 dBm	-17.63 dB	1.07 dB
11.000 MHz	20.000 MHz	100 kHz	5.676666667 GHz	-30.52 dBm	-24.38 dB	0.66 dB

Screen A: Spectrum Mask

Figure 64: R52 spectrum mask for tx-power=20 (blue) and tx-power=13 (green)



		Frequency	2.432 GHz	Standard	IEEE 802.11g
Ref Level	20 dBm	Time	1 ms	Preamble	OFDM
Att	30 dB	PSDU Len	1/1366	Modulation	54 Mbps 64 QAM
Ext Att	0 dB	Samples	44000	Burst	1 of 1 (1)
SGL					

Reference Power -10.91 dBm Tx Bandwidth 18 MHz

Range Low	Range Up	RBW	Freq at Δ to Limit	Power Abs	Power Rel	Δ Limit
-50.000 MHz	-30.000 MHz	100 kHz	2.386202900 GHz	-63.33 dBm	-52.42 dB	-12.42 dB
-30.000 MHz	-20.000 MHz	100 kHz	2.402144928 GHz	-64.25 dBm	-53.34 dB	-13.52 dB
-20.000 MHz	-11.000 MHz	100 kHz	2.420985507 GHz	-36.30 dBm	-25.40 dB	-5.38 dB
-11.000 MHz	-9.000 MHz	100 kHz	2.421130435 GHz	-35.50 dBm	-24.59 dB	-5.89 dB
9.000 MHz	11.000 MHz	100 kHz	2.442724638 GHz	-35.26 dBm	-24.35 dB	-7.10 dB
11.000 MHz	20.000 MHz	100 kHz	2.443594203 GHz	-36.44 dBm	-25.53 dB	-5.00 dB

Figure 65: R52 @ 54Mbit/s, tx-power=16 (default): spectrum mask

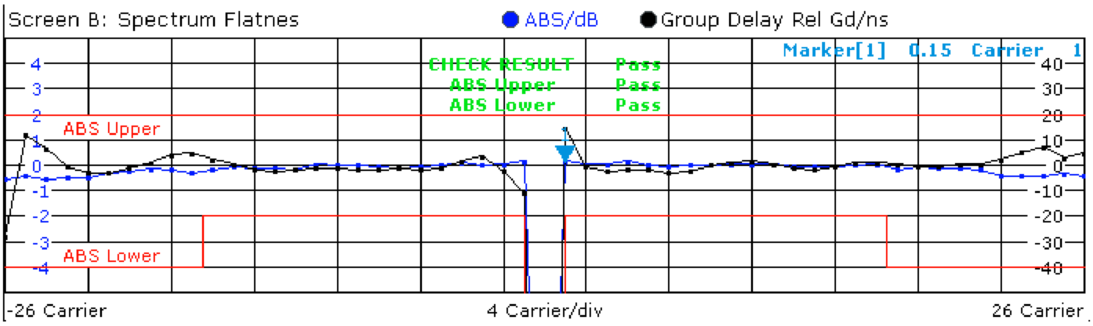
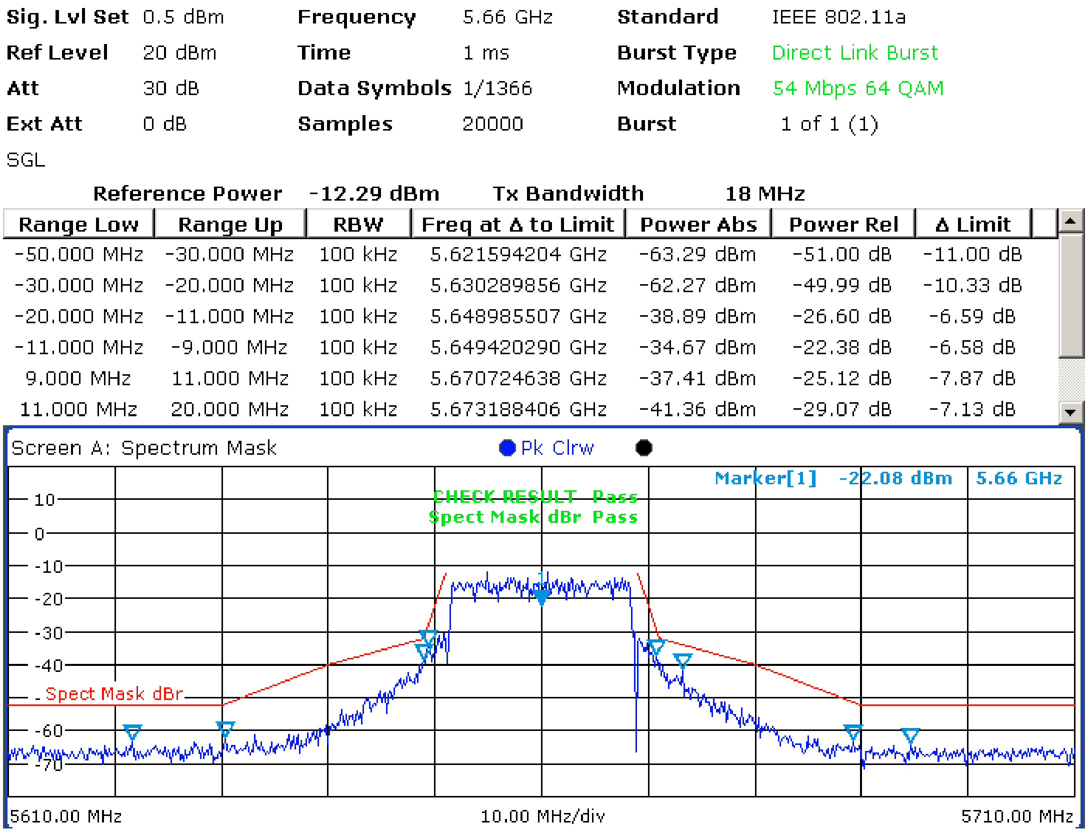


Figure 66: R52 @ 54Mbit/s, tx-power=16 (default): spectrum flatness



Sig. Lvl Set	0.5 dBm	**Frequency**	5.66 GHz	**Standard**	IEEE 802.11a
Ref Level	20 dBm	**Time**	1 ms	**Burst Type**	Direct Link Burst
Att	30 dB	**Data Symbols**	1/1366	**Modulation**	54 Mbps 64 QAM
Ext Att	0 dB	**Samples**	20000	**Burst**	1 of 1 (1)

SGL

Reference Power -12.29 dBm **Tx Bandwidth** 18 MHz

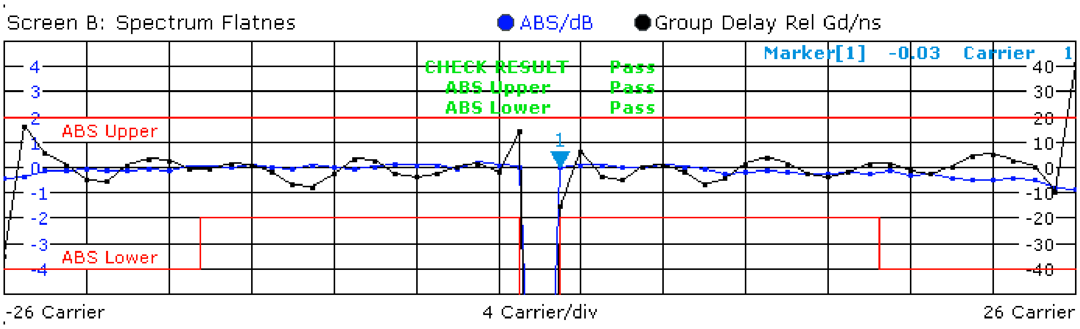
Range Low	Range Up	RBW	Freq at Δ to Limit	Power Abs	Power Rel	Δ Limit
-50.000 MHz	-30.000 MHz	100 kHz	5.621594204 GHz	-63.29 dBm	-51.00 dB	-11.00 dB
-30.000 MHz	-20.000 MHz	100 kHz	5.630289856 GHz	-62.27 dBm	-49.99 dB	-10.33 dB
-20.000 MHz	-11.000 MHz	100 kHz	5.648985507 GHz	-38.89 dBm	-26.60 dB	-6.59 dB
-11.000 MHz	-9.000 MHz	100 kHz	5.649420290 GHz	-34.67 dBm	-22.38 dB	-6.58 dB
9.000 MHz	11.000 MHz	100 kHz	5.670724638 GHz	-37.41 dBm	-25.12 dB	-7.87 dB
11.000 MHz	20.000 MHz	100 kHz	5.673188406 GHz	-41.36 dBm	-29.07 dB	-7.13 dB

Screen A: Spectrum Mask ● Pk Clrw ●

Marker[1] -22.08 dBm 5.66 GHz

CHECK RESULT Pass
Spect Mask dBr Pass

Figure 67: R52 @ 54Mbit/s, 5660 MHz, tx-power=16 (default): spectrum mask

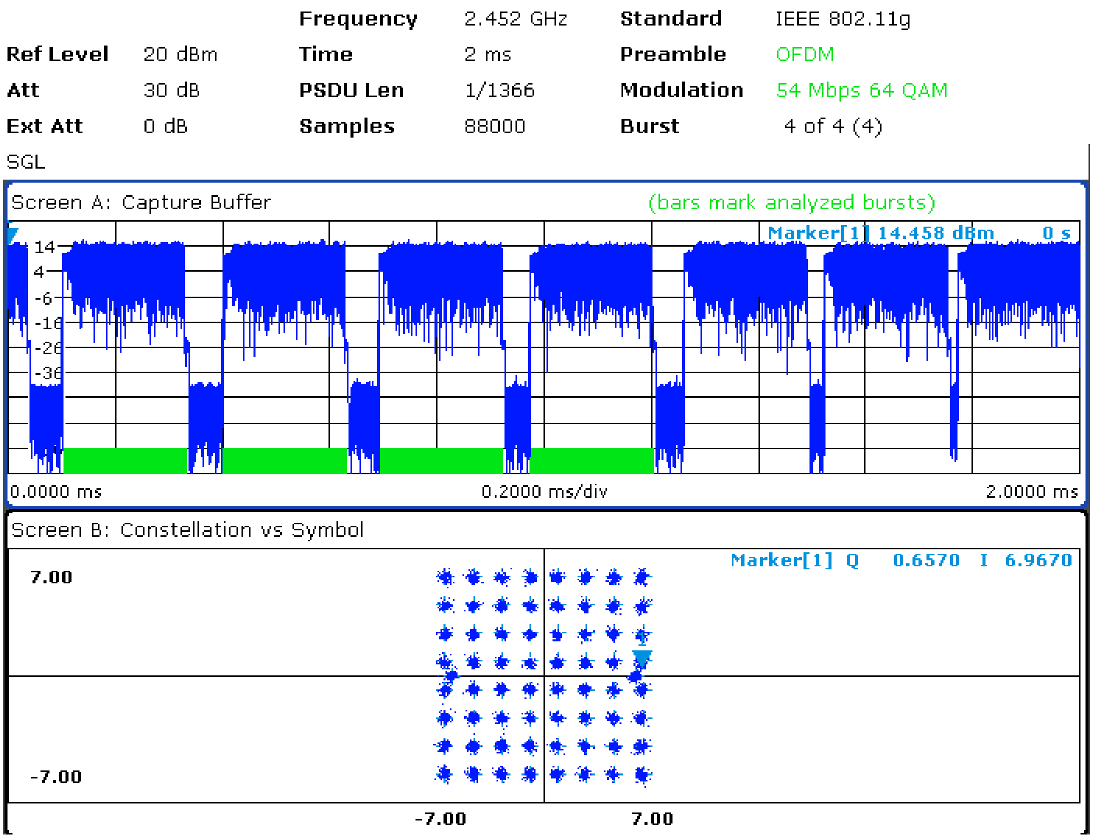


Screen B: Spectrum Flatnes ● ABS/dB ● Group Delay Rel Gd/ns

Marker[1] -0.03 Carrier 1

CHECK RESULT Pass
ABS Upper Pass
ABS Lower Pass

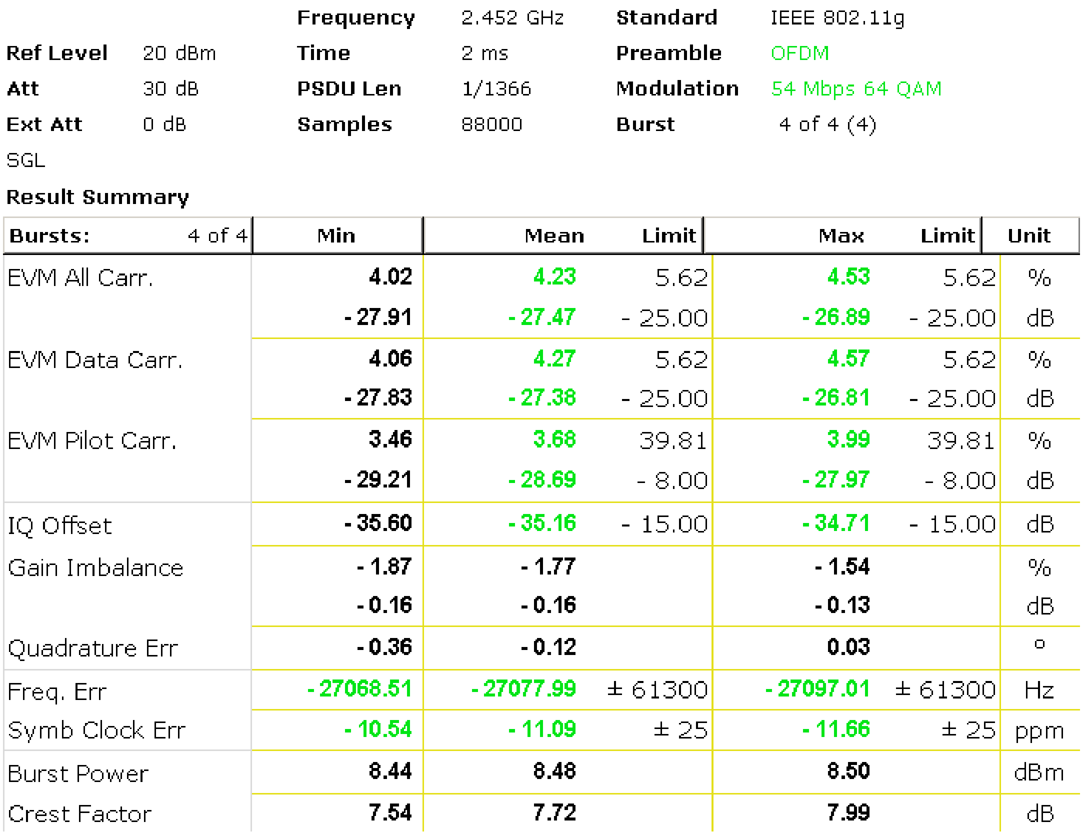
Figure 68: R52 @ 54 Mbit/s, 5660 MHz, tx-power=16 (default): spectrum flatness



		Frequency	2.452 GHz	Standard	IEEE 802.11g
Ref Level	20 dBm	Time	2 ms	Preamble	OFDM
Att	30 dB	PSDU Len	1/1366	Modulation	54 Mbps 64 QAM
Ext Att	0 dB	Samples	88000	Burst	4 of 4 (4)

SGL

Figure 69: R52 @ 54 Mbit/s, tx-power=16 (default): PVT with constellation



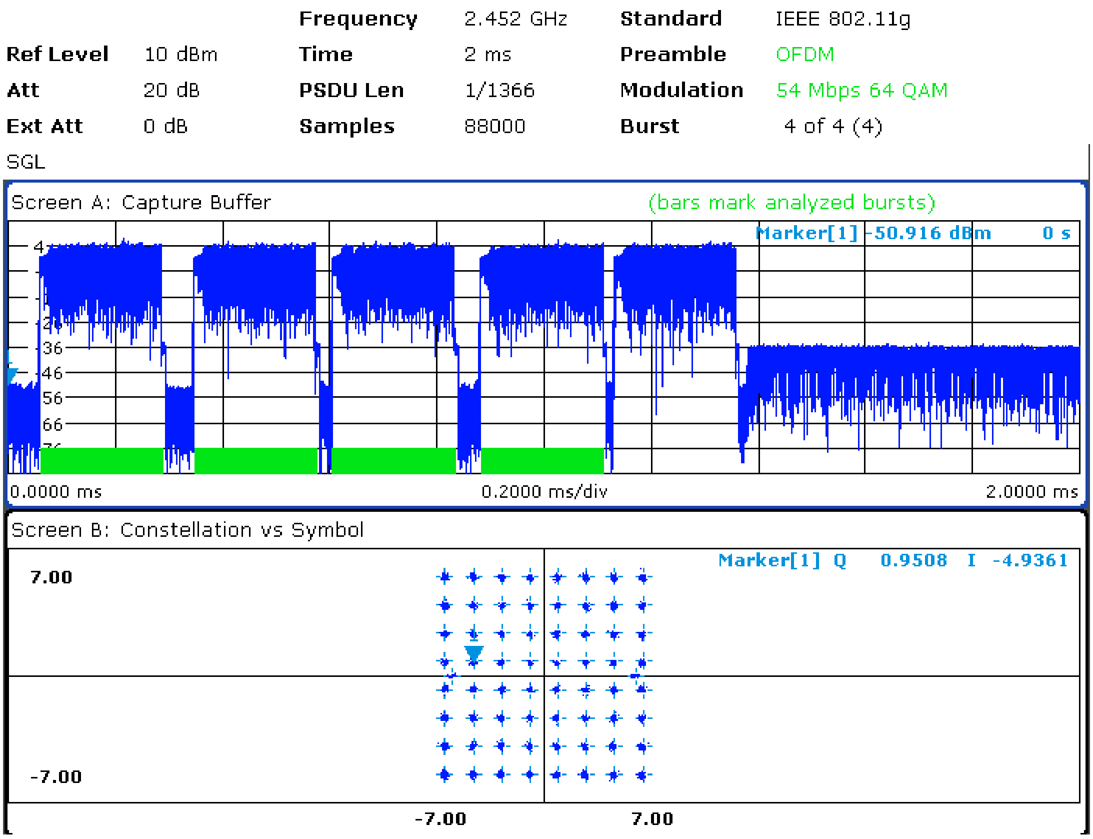
		Frequency	2.452 GHz	Standard	IEEE 802.11g
Ref Level	20 dBm	Time	2 ms	Preamble	OFDM
Att	30 dB	PSDU Len	1/1366	Modulation	54 Mbps 64 QAM
Ext Att	0 dB	Samples	88000	Burst	4 of 4 (4)

SGL

Result Summary

Bursts: 4 of 4	Min	Mean	Limit	Max	Limit	Unit
EVM All Carr.	4.02	4.23	5.62	4.53	5.62	%
	- 27.91	- 27.47	- 25.00	- 26.89	- 25.00	dB
EVM Data Carr.	4.06	4.27	5.62	4.57	5.62	%
	- 27.83	- 27.38	- 25.00	- 26.81	- 25.00	dB
EVM Pilot Carr.	3.46	3.68	39.81	3.99	39.81	%
	- 29.21	- 28.69	- 8.00	- 27.97	- 8.00	dB
IQ Offset	- 35.60	- 35.16	- 15.00	- 34.71	- 15.00	dB
Gain Imbalance	- 1.87	- 1.77		- 1.54		%
	- 0.16	- 0.16		- 0.13		dB
Quadrature Err	- 0.36	- 0.12		0.03		°
Freq. Err	- 27068.51	- 27077.99	± 61300	- 27097.01	± 61300	Hz
Symb Clock Err	- 10.54	- 11.09	± 25	- 11.66	± 25	ppm
Burst Power	8.44	8.48		8.50		dBm
Crest Factor	7.54	7.72		7.99		dB

Figure 70: R52 @ 54 Mbit/s, tx-power=16 (default): transmitter compliance

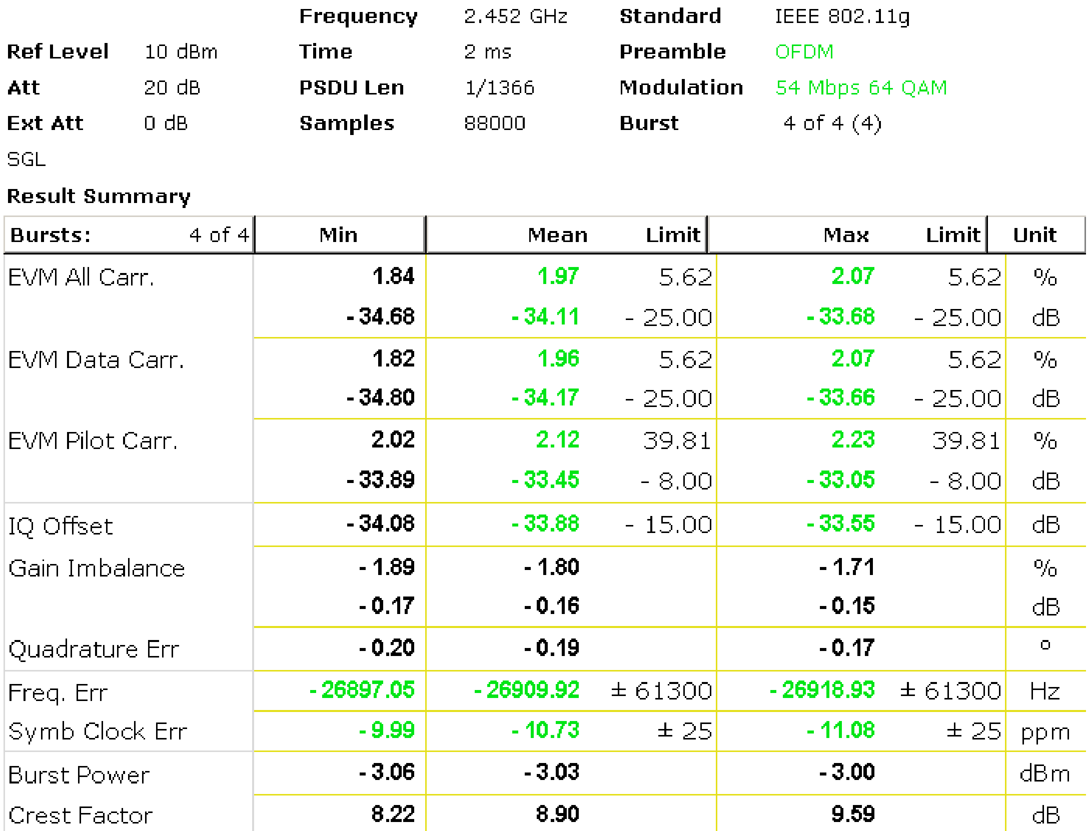


		Frequency	2.452 GHz	Standard	IEEE 802.11g
Ref Level	10 dBm	Time	2 ms	Preamble	OFDM
Att	20 dB	PSDU Len	1/1366	Modulation	54 Mbps 64 QAM
Ext Att	0 dB	Samples	88000	Burst	4 of 4 (4)

SGL

Screen A: Capture Buffer
(bars mark analyzed bursts)
Marker[1] -50.916 dBm 0 s
4
36
46
56
66
0.0000 ms
0.2000 ms/div
2.0000 ms
Screen B: Constellation vs Symbol
Marker[1] Q 0.9508 I -4.9361
7.00
-7.00
-7.00
7.00

Figure 71: R52 @ 54 Mbit/s, tx-power=5 (default): PVT with constellation



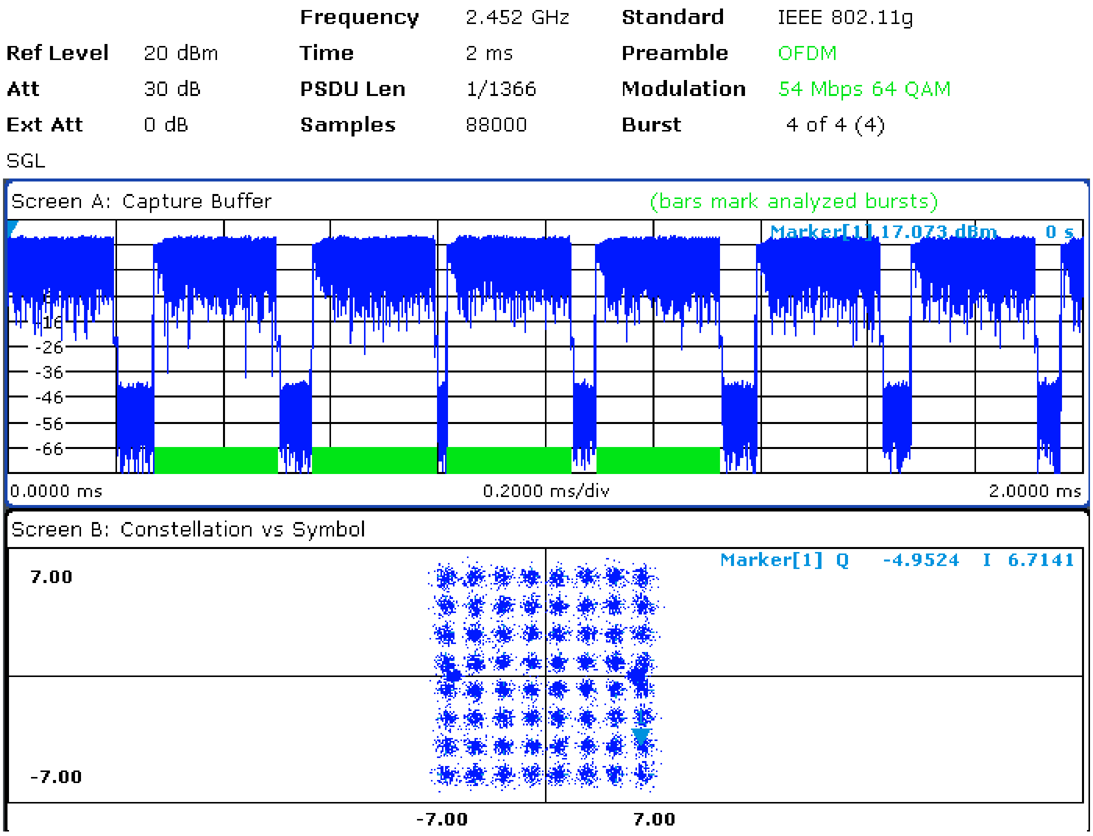
		Frequency	2.452 GHz	Standard	IEEE 802.11g
Ref Level	10 dBm	Time	2 ms	Preamble	OFDM
Att	20 dB	PSDU Len	1/1366	Modulation	54 Mbps 64 QAM
Ext Att	0 dB	Samples	88000	Burst	4 of 4 (4)

SGL

Result Summary

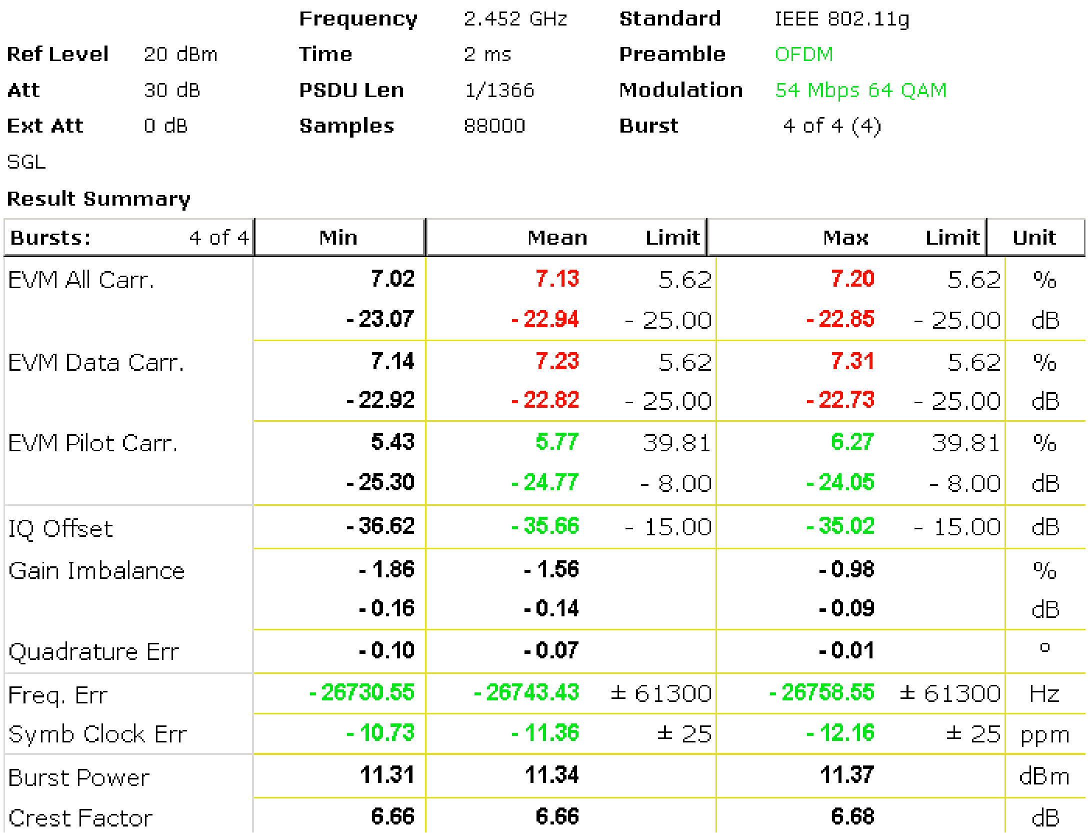
Bursts: 4 of 4	Min	Mean	Limit	Max	Limit	Unit
EVM All Carr.	1.84	1.97	5.62	2.07	5.62	%
	-34.68	-34.11	-25.00	-33.68	-25.00	dB
EVM Data Carr.	1.82	1.96	5.62	2.07	5.62	%
	-34.80	-34.17	-25.00	-33.66	-25.00	dB
EVM Pilot Carr.	2.02	2.12	39.81	2.23	39.81	%
	-33.89	-33.45	-8.00	-33.05	-8.00	dB
IQ Offset	-34.08	-33.88	-15.00	-33.55	-15.00	dB
Gain Imbalance	-1.89	-1.80		-1.71		%
	-0.17	-0.16		-0.15		dB
Quadrature Err	-0.20	-0.19		-0.17		°
Freq. Err	-26897.05	-26909.92	± 61300	-26918.93	± 61300	Hz
Symb Clock Err	-9.99	-10.73	± 25	-11.08	± 25	ppm
Burst Power	-3.06	-3.03		-3.00		dBm
Crest Factor	8.22	8.90		9.59		dB

Figure 72: R52 @ 54 Mbit/s, tx-power=5 (default): transmitter compliance



		Frequency	2.452 GHz	**Standard**	IEEE 802.11g
Ref Level	20 dBm	**Time**	2 ms	**Preamble**	OFDM
Att	30 dB	**PSDU Len**	1/1366	**Modulation**	54 Mbps 64 QAM
Ext Att	0 dB	**Samples**	88000	**Burst**	4 of 4 (4)
SGL					

Figure 73: R52 @ 54 Mbit/s, tx-power=20: PVT with constellation diagram

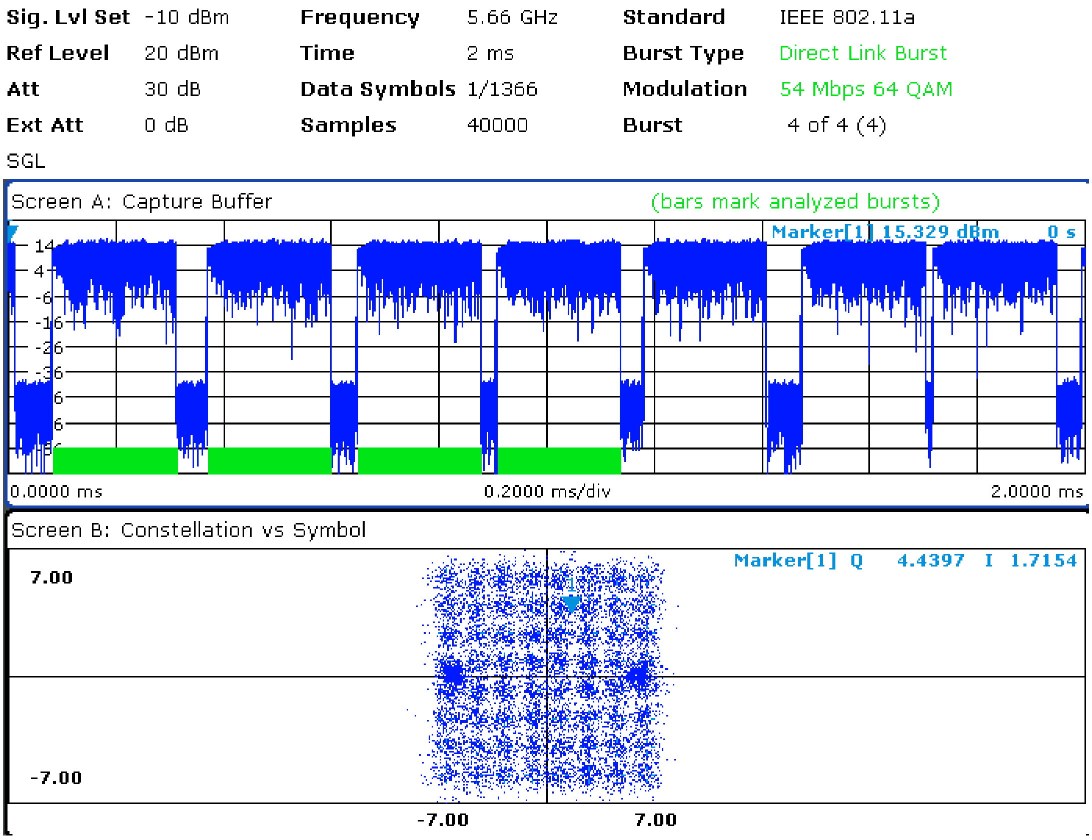


		Frequency	2.452 GHz	**Standard**	IEEE 802.11g
Ref Level	20 dBm	**Time**	2 ms	**Preamble**	OFDM
Att	30 dB	**PSDU Len**	1/1366	**Modulation**	54 Mbps 64 QAM
Ext Att	0 dB	**Samples**	88000	**Burst**	4 of 4 (4)
SGL					

Result Summary

Bursts: 4 of 4	Min	Mean	Limit	Max	Limit	Unit
EVM All Carr.	7.02	7.13	5.62	7.20	5.62	%
	- 23.07	- 22.94	- 25.00	- 22.85	- 25.00	dB
EVM Data Carr.	7.14	7.23	5.62	7.31	5.62	%
	- 22.92	- 22.82	- 25.00	- 22.73	- 25.00	dB
EVM Pilot Carr.	5.43	5.77	39.81	6.27	39.81	%
	- 25.30	- 24.77	- 8.00	- 24.05	- 8.00	dB
IQ Offset	- 36.62	- 35.66	- 15.00	- 35.02	- 15.00	dB
Gain Imbalance	- 1.86	- 1.56		- 0.98		%
	- 0.16	- 0.14		- 0.09		dB
Quadrature Err	- 0.10	- 0.07		- 0.01		°
Freq. Err	- 26730.55	- 26743.43	± 61300	- 26758.55	± 61300	Hz
Symb Clock Err	- 10.73	- 11.36	± 25	- 12.16	± 25	ppm
Burst Power	11.31	11.34		11.37		dBm
Crest Factor	6.66	6.66		6.68		dB

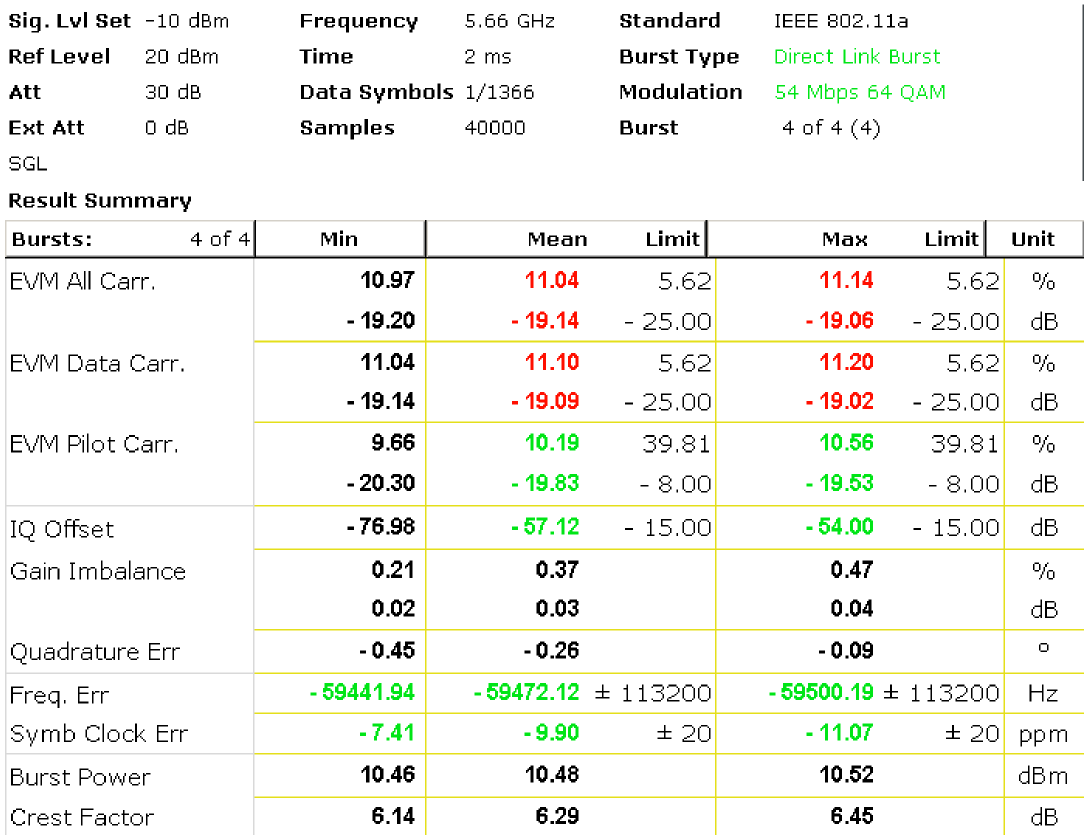
Figure 74: R52 @ 54 Mbit/s, tx-power=20: transmitter parameters



Sig. Lvl Set	-10 dBm	**Frequency**	5.66 GHz	**Standard**	IEEE 802.11a
Ref Level	20 dBm	**Time**	2 ms	**Burst Type**	Direct Link Burst
Att	30 dB	**Data Symbols**	1/1366	**Modulation**	54 Mbps 64 QAM
Ext Att	0 dB	**Samples**	40000	**Burst**	4 of 4 (4)

SGL

Figure 75: R52 @ 54 Mbit/s, 5660 MHz, tx-power=20: PVT with constellation



Sig. Lvl Set	-10 dBm	**Frequency**	5.66 GHz	**Standard**	IEEE 802.11a
Ref Level	20 dBm	**Time**	2 ms	**Burst Type**	Direct Link Burst
Att	30 dB	**Data Symbols**	1/1366	**Modulation**	54 Mbps 64 QAM
Ext Att	0 dB	**Samples**	40000	**Burst**	4 of 4 (4)

SGL

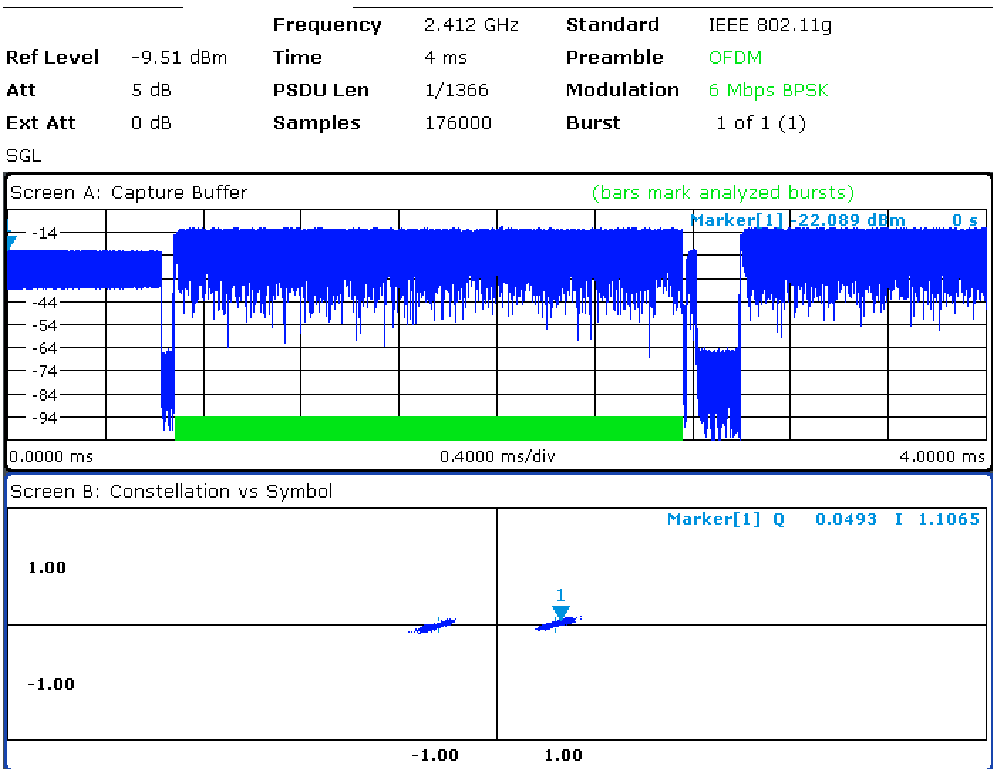
Result Summary

Bursts: 4 of 4	**Min**	**Mean**	**Limit**	**Max**	**Limit**	**Unit**
EVM All Carr.	10.97	11.04	5.62	11.14	5.62	%
	-19.20	-19.14	-25.00	-19.06	-25.00	dB
EVM Data Carr.	11.04	11.10	5.62	11.20	5.62	%
	-19.14	-19.09	-25.00	-19.02	-25.00	dB
EVM Pilot Carr.	9.66	10.19	39.81	10.56	39.81	%
	-20.30	-19.83	-8.00	-19.53	-8.00	dB
IQ Offset	-76.98	-57.12	-15.00	-54.00	-15.00	dB
Gain Imbalance	0.21	0.37		0.47		%
	0.02	0.03		0.04		dB
Quadrature Err	-0.45	-0.26		-0.09		°
Freq. Err	-59441.94	-59472.12	± 113200	-59500.19	± 113200	Hz
Symb Clock Err	-7.41	-9.90	± 20	-11.07	± 20	ppm
Burst Power	10.46	10.48		10.52		dBm
Crest Factor	6.14	6.29		6.45		dB

Figure 76: R52 @ 54 Mbit/s, 5660 MHz, tx-power=20: transmitter parameters

7.1.3 Ubiquity Bullet2

This device also complies with all transmitter requirements. Only few pictures are included here for comparison with the other devices.

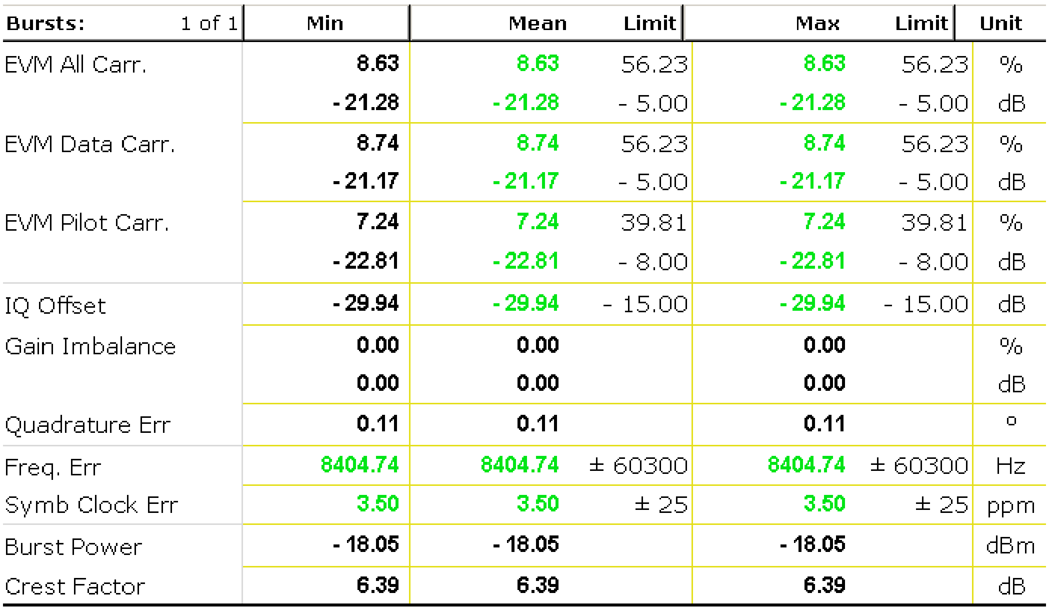


		Frequency	2.412 GHz	Standard	IEEE 802.11g
Ref Level	-9.51 dBm	Time	4 ms	Preamble	OFDM
Att	5 dB	PSDU Len	1/1366	Modulation	6 Mbps BPSK
Ext Att	0 dB	Samples	176000	Burst	1 of 1 (1)

SGL

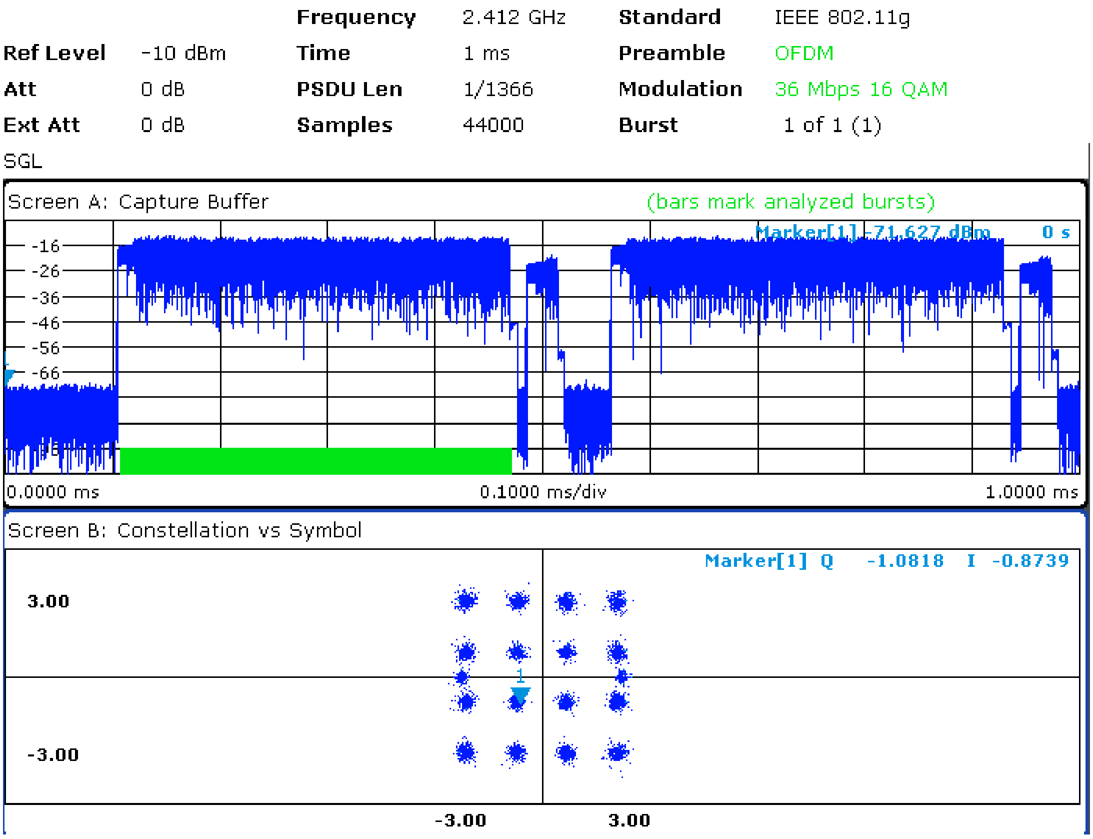
Screen A: Capture Buffer
(bars mark analyzed bursts)
Marker[1] -22.089 dBm 0 s
-14
-44
-54
-64
-74
-84
-94
0.0000 ms
0.4000 ms/div
4.0000 ms
Screen B: Constellation vs Symbol
Marker[1] Q 0.0493 I 1.1065
1.00
-1.00
1
-1.00
1.00

Figure 77: Ubiquity Bullet2 @ 6Mbit/s: PVT with constellation diagram



Bursts: 1 of 1	Min	Mean	Limit	Max	Limit	Unit
EVM All Carr.	8.63	8.63	56.23	8.63	56.23	%
	-21.28	-21.28	- 5.00	-21.28	- 5.00	dB
EVM Data Carr.	8.74	8.74	56.23	8.74	56.23	%
	-21.17	-21.17	- 5.00	-21.17	- 5.00	dB
EVM Pilot Carr.	7.24	7.24	39.81	7.24	39.81	%
	-22.81	-22.81	- 8.00	-22.81	- 8.00	dB
IQ Offset	-29.94	-29.94	- 15.00	-29.94	- 15.00	dB
Gain Imbalance	0.00	0.00		0.00		%
	0.00	0.00		0.00		dB
Quadrature Err	0.11	0.11		0.11		°
Freq. Err	8404.74	8404.74	± 60300	8404.74	± 60300	Hz
Symb Clock Err	3.50	3.50	± 25	3.50	± 25	ppm
Burst Power	-18.05	-18.05		-18.05		dBm
Crest Factor	6.39	6.39		6.39		dB

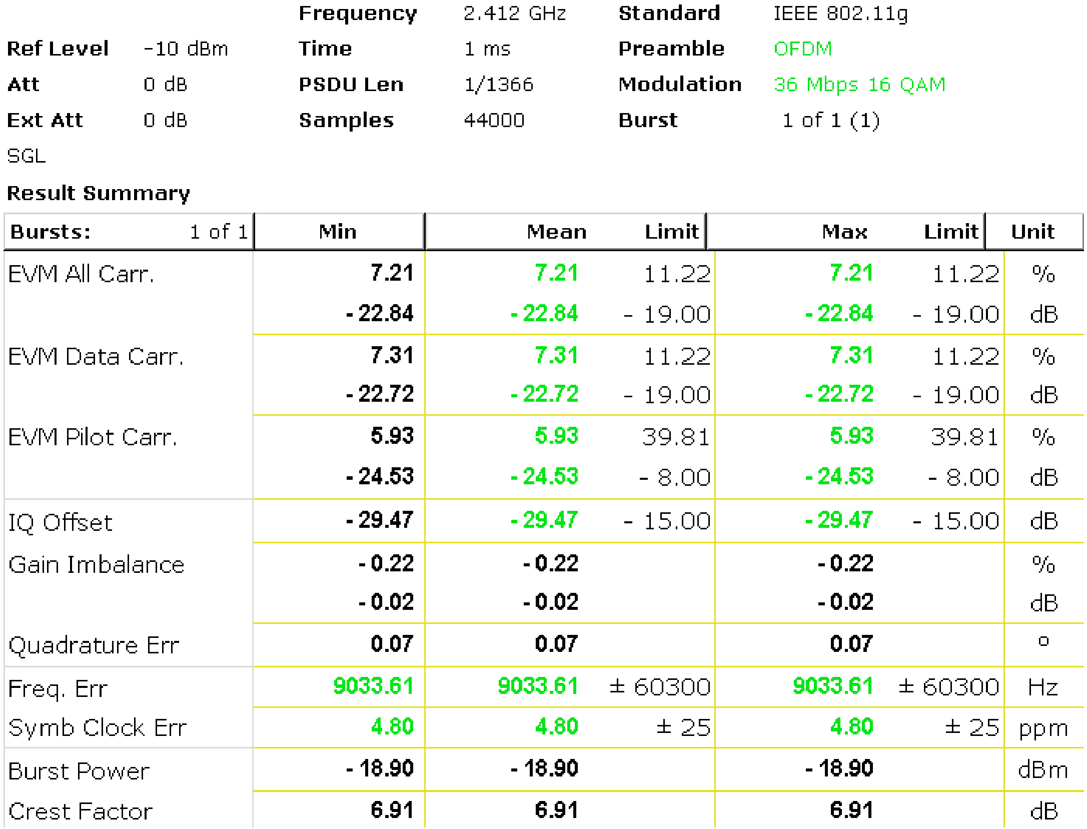
Figure 78: Ubiquity Bullet2 @ 6Mbit/s: transmitter compliance parameters



		Frequency	2.412 GHz	Standard	IEEE 802.11g
Ref Level	-10 dBm	Time	1 ms	Preamble	OFDM
Att	0 dB	PSDU Len	1/1366	Modulation	36 Mbps 16 QAM
Ext Att	0 dB	Samples	44000	Burst	1 of 1 (1)

SGL

Figure 79: Ubiquity Bullet2 @ 36Mbit/s: PVT with constellation diagram



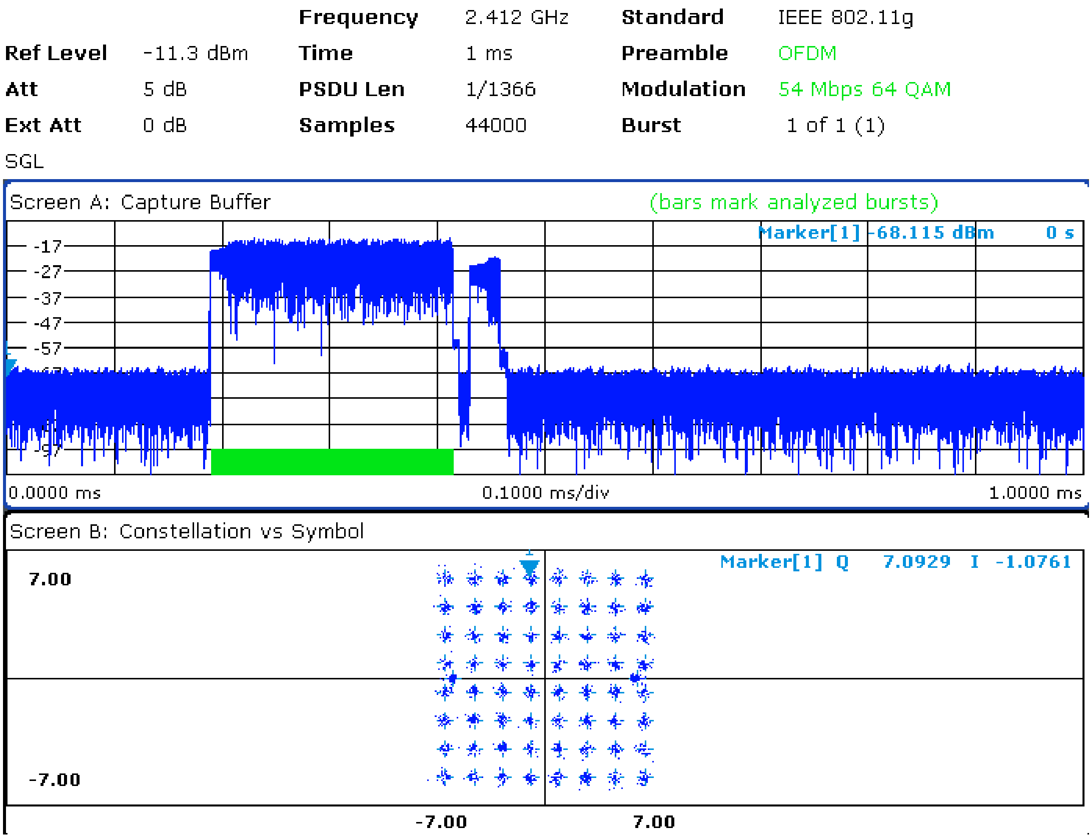
		Frequency	2.412 GHz	Standard	IEEE 802.11g
Ref Level	-10 dBm	Time	1 ms	Preamble	OFDM
Att	0 dB	PSDU Len	1/1366	Modulation	36 Mbps 16 QAM
Ext Att	0 dB	Samples	44000	Burst	1 of 1 (1)

SGL

Result Summary

Bursts: 1 of 1	Min	Mean	Limit	Max	Limit	Unit
EVM All Carr.	7.21	7.21	11.22	7.21	11.22	%
	-22.84	-22.84	- 19.00	-22.84	- 19.00	dB
EVM Data Carr.	7.31	7.31	11.22	7.31	11.22	%
	-22.72	-22.72	- 19.00	-22.72	- 19.00	dB
EVM Pilot Carr.	5.93	5.93	39.81	5.93	39.81	%
	-24.53	-24.53	- 8.00	-24.53	- 8.00	dB
IQ Offset	-29.47	-29.47	- 15.00	-29.47	- 15.00	dB
Gain Imbalance	-0.22	-0.22		-0.22		%
	-0.02	-0.02		-0.02		dB
Quadrature Err	0.07	0.07		0.07		°
Freq. Err	9033.61	9033.61	± 60300	9033.61	± 60300	Hz
Symb Clock Err	4.80	4.80	± 25	4.80	± 25	ppm
Burst Power	-18.90	-18.90		-18.90		dBm
Crest Factor	6.91	6.91		6.91		dB

Figure 80: Ubiquity Bullet2 @ 36Mbit/s: transmitter compliance parameters

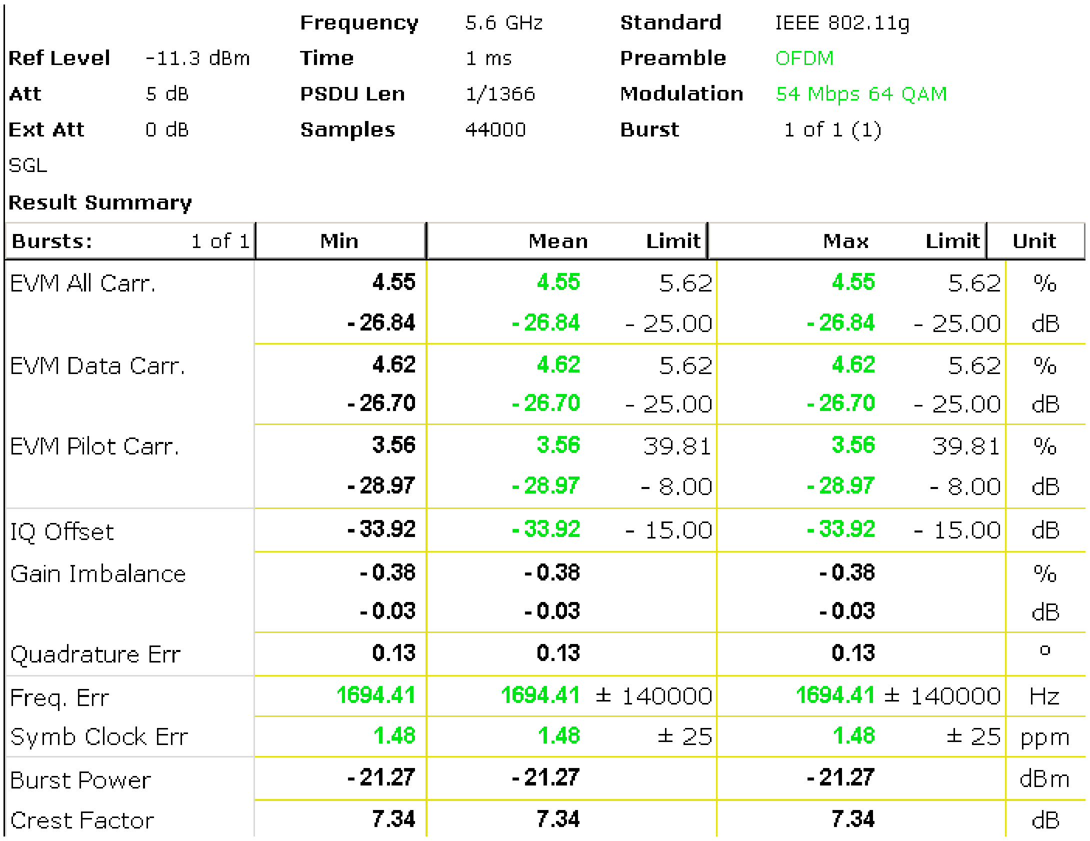


		Frequency	2.412 GHz	Standard	IEEE 802.11g
Ref Level	-11.3 dBm	Time	1 ms	Preamble	OFDM
Att	5 dB	PSDU Len	1/1366	Modulation	54 Mbps 64 QAM
Ext Att	0 dB	Samples	44000	Burst	1 of 1 (1)

SGL

Screen A: Capture Buffer
(bars mark analyzed bursts)
Marker[1] -68.115 dBm 0 s
-17
-27
-37
-47
-57
-67
-97
0.0000 ms
0.1000 ms/div
1.0000 ms
Screen B: Constellation vs Symbol
Marker[1] Q 7.0929 I -1.0761
7.00
-7.00
-7.00
7.00

Figure 81: Ubiquity Bullet2 @ 54Mbit/s: PVT with constellation diagram



		Frequency	5.6 GHz	Standard	IEEE 802.11g
Ref Level	-11.3 dBm	Time	1 ms	Preamble	OFDM
Att	5 dB	PSDU Len	1/1366	Modulation	54 Mbps 64 QAM
Ext Att	0 dB	Samples	44000	Burst	1 of 1 (1)

SGL

Result Summary

Bursts: 1 of 1	Min	Mean	Limit	Max	Limit	Unit
EVM All Carr.	4.55	4.55	5.62	4.55	5.62	%
	-26.84	-26.84	-25.00	-26.84	-25.00	dB
EVM Data Carr.	4.62	4.62	5.62	4.62	5.62	%
	-26.70	-26.70	-25.00	-26.70	-25.00	dB
EVM Pilot Carr.	3.56	3.56	39.81	3.56	39.81	%
	-28.97	-28.97	-8.00	-28.97	-8.00	dB
IQ Offset	-33.92	-33.92	-15.00	-33.92	-15.00	dB
Gain Imbalance	-0.38	-0.38		-0.38		%
	-0.03	-0.03		-0.03		dB
Quadrature Err	0.13	0.13		0.13		°
Freq. Err	1694.41	1694.41	± 140000	1694.41	± 140000	Hz
Symb Clock Err	1.48	1.48	± 25	1.48	± 25	ppm
Burst Power	-21.27	-21.27		-21.27		dBm
Crest Factor	7.34	7.34		7.34		dB

Figure 82: Ubiquity Bullet2 @ 54Mbit/s: transmitter compliance parameters

7.2 Receive chain measurement

For the receiver tests, there was a problem with unwanted RF coupling between the radio units. Without additional arrangements, the tested receivers would always hear the transmitters more through the free space inside the testbed rack than through the RF interconnection network. After few hours of experimenting with the possible shielding strategies, the best results for the Ubiquity M5 devices were achieved when the whole units were wrapped by aluminium foil – see the figure 83. With this arrangement, the M5 receiver is not able to detect any signal if the testbed programmable attenuator is set to a value > 50dB.

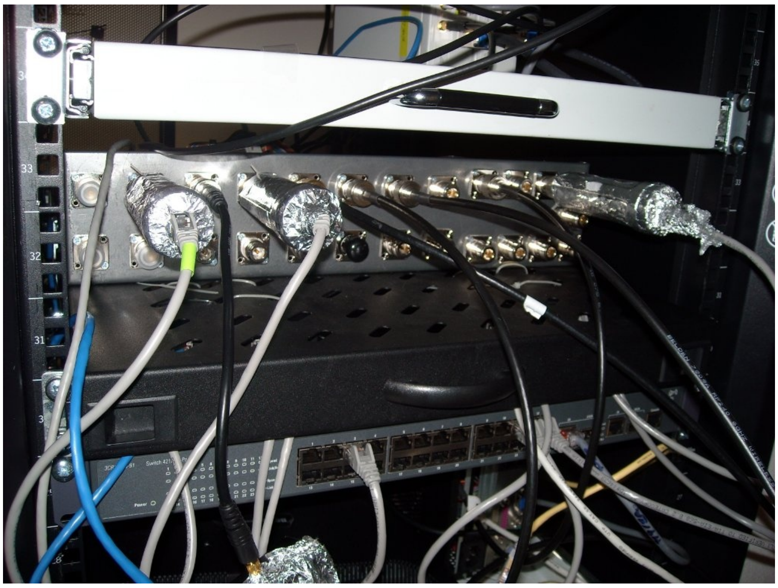


Figure 83: Ubiquity Bullet shielded by aluminium foil

For the 2.4GHz devices – e.g. Ubiquity Bullet2 – the aluminium foil helped a little, but was not able to block the free-space coupling completely. Therefore, the transmitter (or receiver) must get additional shielding. A small metal miniITX case was used and solved the problem successfully.

Also, in order to use the full dynamic range of the programmable attenuators, the originally proposed 30dB fixed attenuators in the testbed had to be replaced by 10dB ones.

7.2.1 Receiver minimum sensitivity

For automating the receiver sensitivity tests, the “Orchestrator” SW tool was written – see chapter 11.2.

The following measurement used two Ubiquity Bullet M5 units with 20MHz channel bandwidth, AirMax disabled and the other parameters listed below:

```
# cat /etc/sysinit/radio.conf
plugin_start() {
       insmod $(find /lib/modules/ -name "ath_[ap][hc][bi].ko") 1>/dev/null 2>&1||true
       iwpriv "wifi0" setCountryID 203
       if [ -e /proc/sys/dev/ubnt_poll/mode ]; then
              echo 0 > /proc/sys/dev/ubnt_poll/mode
       fi
       echo 0 > /proc/sys/dev/ubnt_poll/no_ack
       echo 0 > /proc/sys/dev/ubnt_poll/use_ant_array
       /sbin/wlanconfig "ath0" create wlandev wifi0 wlanmode ap  > /dev/null 2>&1
       if [ $? -eq 0 ]; then
              echo wifi0 > /tmp/.wifi_ath0
       fi
       iwpriv "wifi0" ClkSel 1
       iwpriv "wifi0" RegObey 0
       iwpriv "wifi0" FreqShift 0
       iwpriv "wifi0" AMPDU 1
       iwpriv "wifi0" AMPDUFrames 32
       iwpriv "wifi0" AMPDULim 65535
       iwpriv "wifi0" extant 0
       iwpriv "wifi0" ant_gain 0
       iwpriv "wifi0" txchainmask 1
       iwpriv "wifi0" rxchainmask 1
       iwpriv "wifi0" diversity 1
       iwpriv "wifi0" txant 0
       iwpriv "wifi0" rxant 0
       iwpriv "wifi0" ForBiasAuto 0
       iwpriv "wifi0" ATHDebug 0x0
       iwpriv "ath0" mode 11NAHT20
       iwconfig "ath0" freq 5600M
       iwconfig "ath0" rate 7 auto
       echo 1 7 > /proc/sys/dev/ubnt_poll/no_ack_rate
       iwconfig "ath0" frag off
       iwconfig "ath0" rts off
       iwconfig "ath0" txpower 25
       iwpriv "wifi0" ANIEna 0
       echo 1 > /proc/sys/dev/ath/htvendorieenable
       echo 1 > /proc/sys/dev/ath/htdupieenable
       iwpriv "ath0" shortgi 1
       iwpriv "ath0" cwmenable 0
       iwpriv "ath0" cwmmode 0
       iwpriv "wifi0" distance 0
       iwpriv "wifi0" dacount 25
       iwpriv "ath0" puren 0
       true
}
```

The results for the receiver in AP mode with are shown on the graph below:

Because of the free-space signal coupling inside the rack, the measured signal levels have some uncertainty. To assess the uncertainty, a more complex arrangement would need to be done to

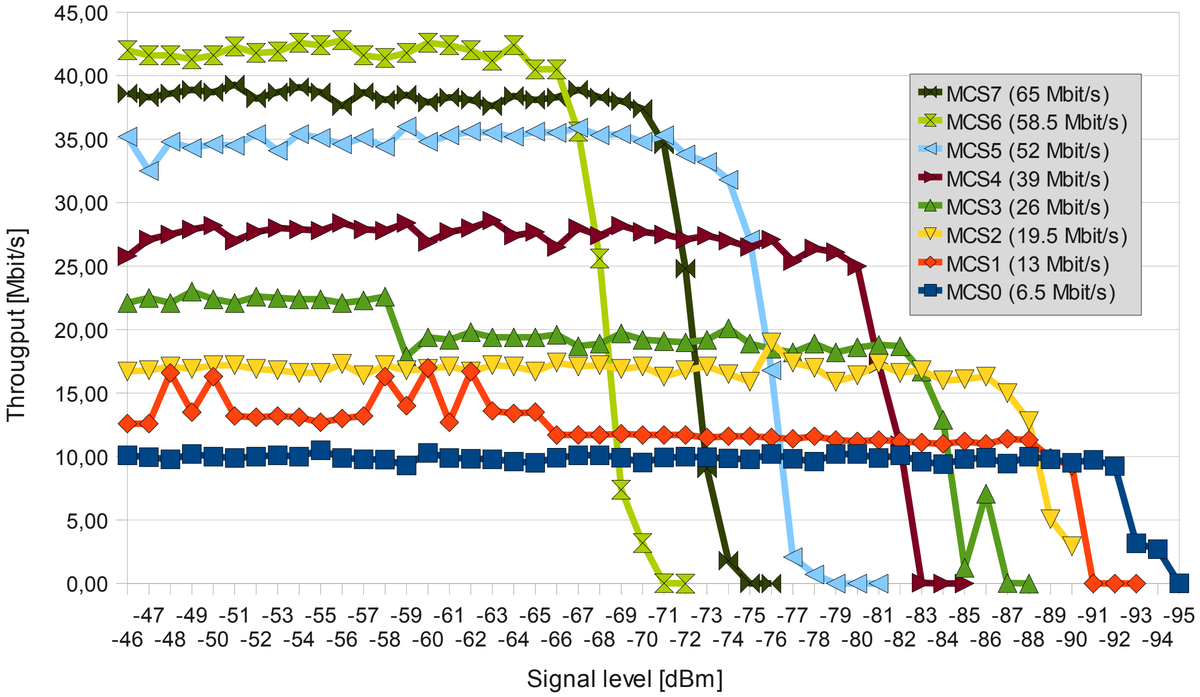


Figure 84: Ubiquity Bullet M5: TCP throughput versus signal level

completely isolate the units from the free-space signal coupling, measure the signal power levels at the receiver connector for various programmable attenuator settings and then adjust the testbed measurement results accordingly.

Also, note that the measured TCP throughputs for MCS0 and MCS1 rates are higher than the actual bit rate set on the transmitter by the command “iwconfig ath0 rate X fixed” (where X is [0..7] in this case, 0 for MCS0 etc). After checking the communication on FSV7 analyzer, the reason was found quickly: the transmitter does not always obey the configured bit rate and occasionally tries to transmit on higher rates. This feature must be counted with when using this radio, therefore the measurements results are published as-they-are, without any additional tweaking.

The sensitivity limits for the specific bit rates are shown in Table 5.

Note that the Ubiquity M5-HP datasheet [116] declares different values with +- 2dB tolerances, however, we used a different model (without the “HP”) whose datasheet is not available. Also, the M5-HP datasheet does not specify the configuration (operating channel, channel bandwidth) used for achieving the declared values.

Compared to the 802.11 minimum sensitivity requirements (see chapter 6.2), we can see that the device complies with the values required by the 802.11 standard (see Table 2).

Table 5: Ubiquity Bullet M5 minimum sensitivity levels

Bit rate	Attenuator setting limit [dB]	Signal level limit [dBm]
MCS0 (6.5 Mbit/s)	46	-92
MCS1 (13 Mbit/s)	44	-90
MCS2 (19.5 Mbit/s)	41	-87
MCS3 (26 Mbit/s)	36	-82
MCS4 (39 Mbit/s)	34	-80
MCS5 (52 Mbit/s)	29	-75
MCS6 (58.5 Mbit/s)	25	-71
MCS7 (65 Mbit/s)	21	-67

7.2.2 Adjacent channel rejection

For measuring adjacent channel rejection, the configuration shown on figure 48 was used. The tested device was Cisco AIR-AP1231G-E-K9, the two golden devices were Ubiquity Bullet2 units because of their excellent ability to generate arbitrary 802.11 traffic (see chapter 6.1 for explanation). The iperf tool with UDP was used to generate the traffic and to measure the packet loss and jitter between the DUT and the golden device in station mode. The broadcaster tool was used for generating the interfering traffic in adjacent channel. The spectrum analyzer was used only for constant “reality checks” of the transmitted signals levels.

The measured values are shown on figure 85, the summary is in Table 6.

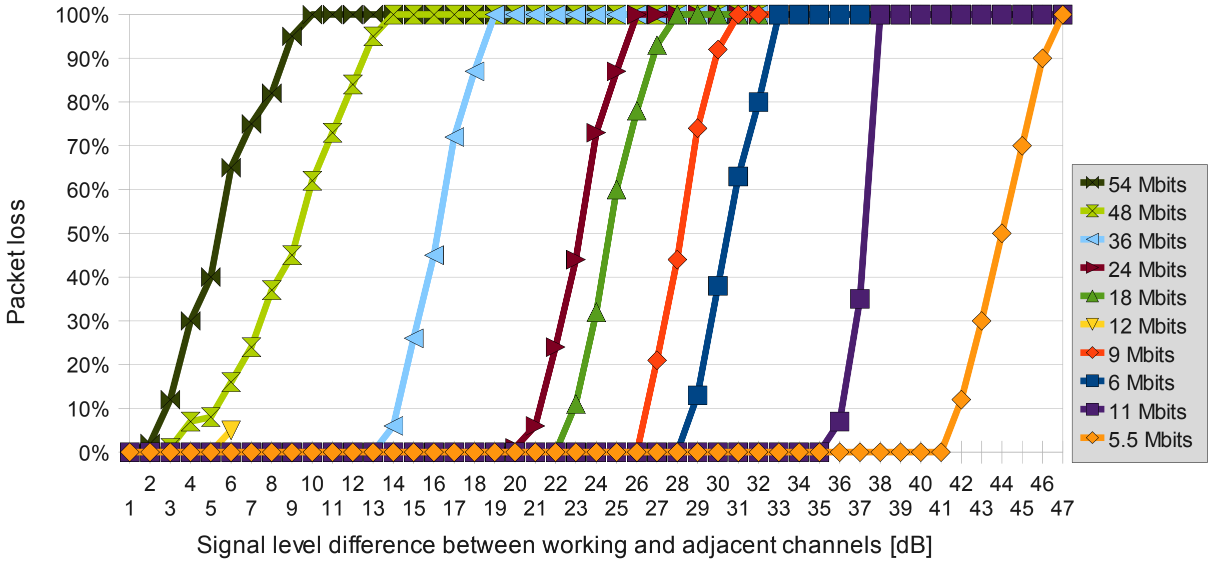


Figure 85: Cisco AIR-AP1231G-E-K9 adjacent channel rejection- measured data

Table 6: Cisco AIR-AP1231G minimum sensitivity and adjacent channel rejection

Bit rate	Minimum sensitivity [dBm]	Adjacent channel rejection [dB]
54 Mbit/s	-67	2
48 Mbit/s	-71	5
36 Mbit/s	-74	14
24 Mbit/s	-75	21
18 Mbit/s	-80	23
12 Mbit/s	-88	27
9 Mbit/s	-90	26
6 Mbit/s	-90	28
11 Mbit/s	-84	35
5.5 Mbit/s	-94	41

Note: the golden and interferer devices were using the same bit rates to compensate the automatic TX power level regulation, which would distort the measurement results if different bit rates were used for the two signals.

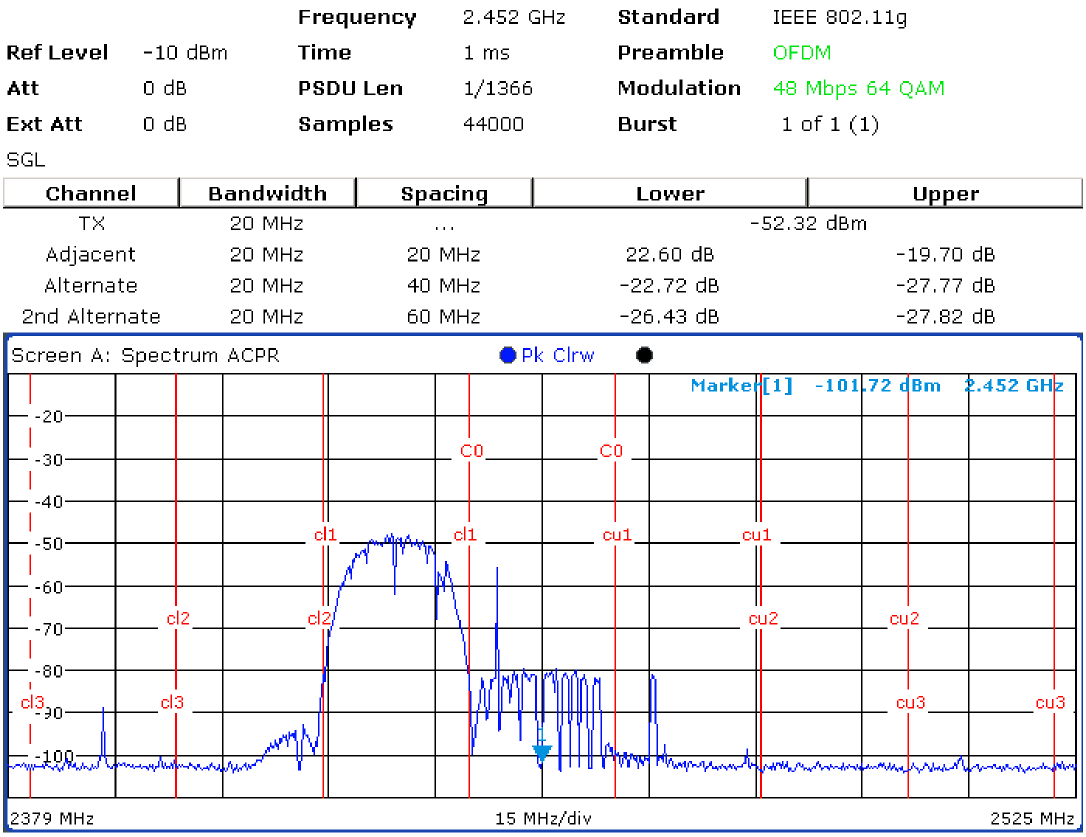
When compared with the 802.11 standard requirements (see Table 2 in chapter 6.2), it is clear that the DUT passes the compliance test with huge margins and this holds even for the 11 Mbit/s and 5.5 Mbit/s rates, which were tested with only 20 MHz channel separation, not the ≥ 25 MHz or even 30 MHz separation required by the standard.

Now, we can make another measurement to answer an interesting question: what is the adjacent channel rejection in case when OFDM is received by the DUT and the interfering signal is DSSS and vice versa?

With Cisco AIR-AP1231G-E-K9, the following rejection values were measured:

- 33 dB for DUT receiving 6 Mbit/s OFDM signal with 5.5 Mbit/s DSSS interference.
- 39 dB for DUT receiving 5.5 Mbit/s DSSS signal with 6 Mbit/s OFDM interference.

Considering the (now historical) results of our past experiments with adjacent channel interferences, this result is surprising – we expected that the OFDM transmissions would be blocked by much smaller DSSS adjacent channel signals. However, the actual rejection is better than any combination of pure OFDM modulations in both channels and this was double-checked by the spectrum analyzer during the whole measurement - see figure 86. The reason of this effect is explained in the next chapter.



		Frequency	2.452 GHz	Standard	IEEE 802.11g
Ref Level	-10 dBm	Time	1 ms	Preamble	OFDM
Att	0 dB	PSDU Len	1/1366	Modulation	48 Mbps 64 QAM
Ext Att	0 dB	Samples	44000	Burst	1 of 1 (1)

SGL

Channel	Bandwidth	Spacing	Lower	Upper
TX	20 MHz	...	-52.32 dBm	
Adjacent	20 MHz	20 MHz	22.60 dB	-19.70 dB
Alternate	20 MHz	40 MHz	-22.72 dB	-27.77 dB
2nd Alternate	20 MHz	60 MHz	-26.43 dB	-27.82 dB

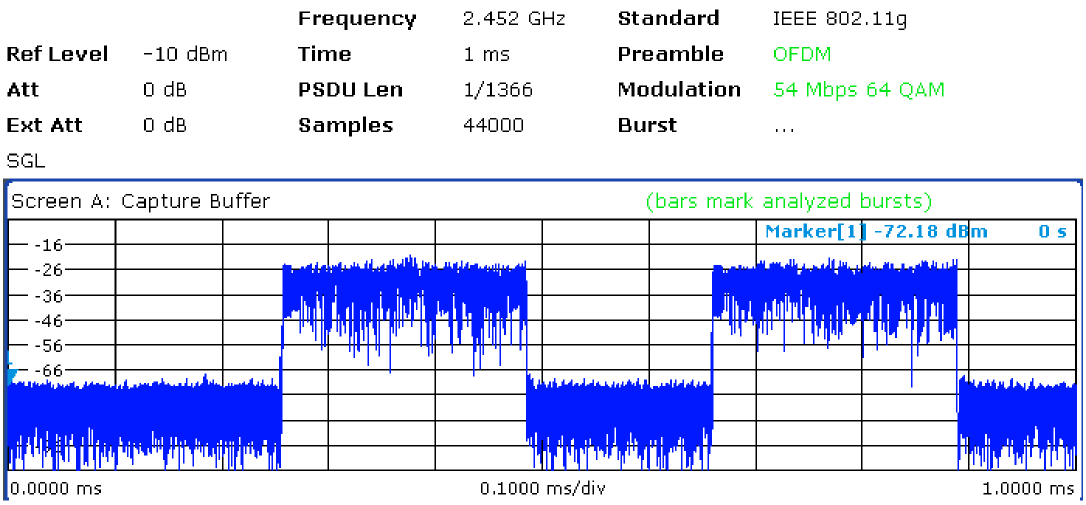
Figure 86: adjacent channel DSSS signal vs. Cisco AIR-AP1231G in OFDM mode

7.3 CCA and adjacent channel interferences

During experiments we have carried out in the past with Atheros devices, we noticed that adjacent channel DSSS interference can stop the in-channel OFDM transmissions easily. Because this happens even in the case of broadcasts sent from an AP without associated stations, where the transmitted packets are not acknowledged, it was clear that the transmissions are not stopped by receiver errors but the culprit must be the transmitter itself, more specifically its CCA implementation. Similar problem is mentioned in [3], with additional reference to the Atheros “Ambient Noise Immunity” (ANI) function.

The 802.11 standard specification of CCA functions is discussed in chapter 4.7.4, but the problem of adjacent channel interference and CCA was left up to this chapter. To shed a more light on this issue, let's have a look on figure 87 showing the power-versus-time (PVT) graph of an interference signal received from adjacent channel.

Although the PVT looks very much the same like a legitimate in-channel signal, it is not - it is mostly composed of harmonic artefacts caused by RF stage non-linearities. These artefacts are obviously not decodable by a DSSS/OFDM receiver.



		Frequency	2.452 GHz	Standard	IEEE 802.11g
Ref Level	-10 dBm	Time	1 ms	Preamble	OFDM
Att	0 dB	PSDU Len	1/1366	Modulation	54 Mbps 64 QAM
Ext Att	0 dB	Samples	44000	Burst	...
SGL					

Screen A: Capture Buffer

(bars mark analyzed bursts)

Marker[1] -72.18 dBm 0 s

Figure 87: PVT graph of interference signal received from adjacent channel

The false "channel busy" condition caused by adjacent channel interference can be most probably attributed to the OFDM CCA, which (aside of the preamble detection) uses a simple signal power threshold scheme – see 802.11 standard clause 17.3.10.5: "*If the preamble portion was missed, the receiver shall hold the CS signal busy for any signal 20 dB above the minimum modulation and coding rate sensitivity (–62 dBm for 20 MHz channel spacing, ...)*". Unfortunately, the adjacent channel interference signal power levels in the co-location scenarios are often higher than -62 dBm and in such a case, any standard compliant 802.11a/g/n device should wait until the interfering signal disappears and only then retry the transmission procedure.

Compared to OFDM, the DSSS CCA provides additional detection method - the physical carrier sense (CS), which is based on the correlation with the Barker code and therefore it is able to distinguish the useful in-channel signals from the adjacent channel interferences. That is why DSSS PHY can be more immune against the false "channel busy" condition.

To verify this assumption, the measurement configuration from chapter 6.5 was used and 3 devices tested for their CCA performance. The results are discussed in the following paragraphs.

On figure 88, measurement results for Cisco AIR-AP1231G are shown. This DUT was set to 2452 MHz frequency and the interfering Ubiquity Bullet2 was set to 2432 MHz. Three combinations of useful and interfering bit rates were tried, as shown on the graph legend. The transmitter throughput drop was measured by setting the iperf UDP bandwidth to the maximum value allowable by zero interference and then by measuring the packet loss when the interference level was increased.

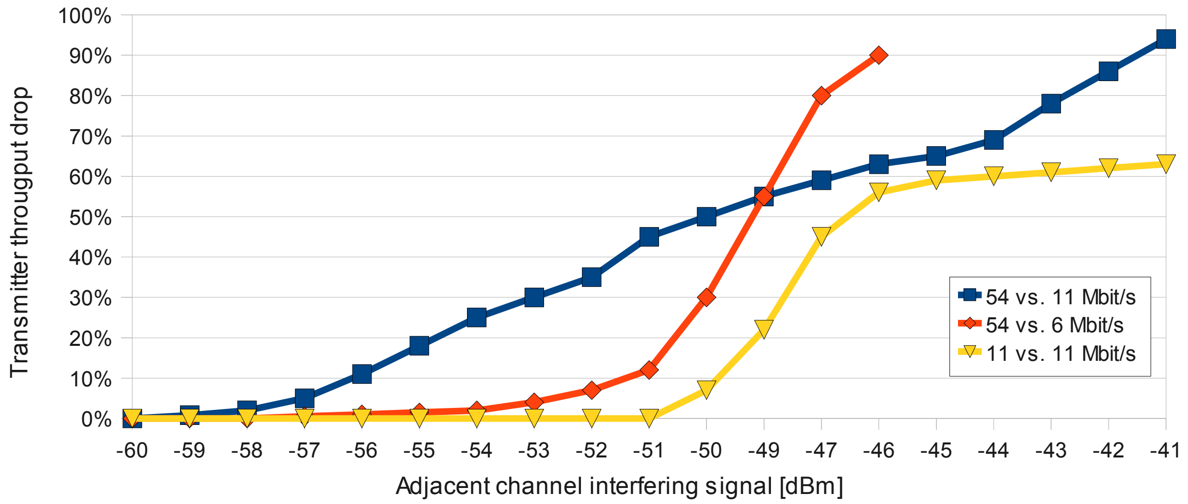


Figure 88: Cisco AIR-AP1231G: CCA performance in presence of adj. channel interference

The measured values correspond to the device own function "Carrier Busy Test", which is present in the device web management – see Table 7. Note that the device detects the carrier busy in interferer's channel and both its adjacent channels, which means that at this signal level, the adjacent channels must not be used. For better understanding the situation, consult the figure 91, because the Table 7 is not so graphic.

Table 7: Cisco Carrier Busy Test
for 6Mbit/s interference @ -46 dBm

Frequency [MHz]	**Carrier Busy**
2412	19%
2417	96%
2422	95%
2427	94%
2432	94%
2437	89%
2442	99%
2447	92%
2452	86%
2457	3%
2462	1%
2467	0%
2472	0%

Similar measurements were made with Mikrotik R52 and AzureWave AR5BXB63 cards.

The AzureWave was measured only in the Intel D945GSEJT mainboard with Linux ath5k driver as no other platforms with miniPCI express slot were avaialable.

Mikrotik R52 was measured in two platforms:

- Intel D945GSEJT mainboard with the RB14 PCI expander, with Linux and ath5k driver,
- Routerboard RB411AR with Mikrotik RouterOS 5.2

The results for both cards were very similar so only single graph can be used:

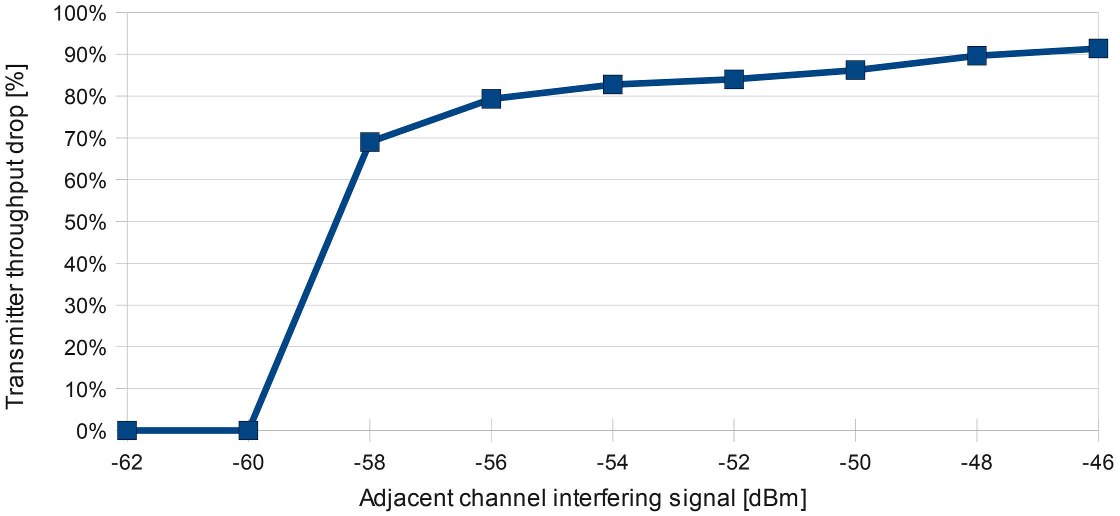


Figure 89: R52 and AzureWave miniPCI cards: CCA performance in presence of adjacent channel interference

It seems that the Atheros cards CCA sensitivity for adjacent channel signals is similar (around -62 dBm) as the in-channel CCA sensitivity defined by 802.11 standard clause 17.3.10.5, which is not good. So compared to Cisco AIR-AP1231G, the Atheros cards CCA implementation is more sensitive to adjacent channel interference by 8 – 10dB (please note, that the exact power level of the packets is hard to measure even with FSV7, so the power levels in our measurements have ±2 dBm uncertainty).

The only difference between R52 and AzureWave cards was that AzureWave could keep transmitting 1-2 Mbit/s througput even in highest interference levels, while R52 would stay completely quiet when interference level reached -46 dBm.

The results from these measurements are important base for the “Know your enemy” (and know yourself)and “Coordinated TPC and DFS“ methods in the next chapter.

8 METHODS FOR INTERFERENCE MITIGATION

As was already announced in chapter 2, we are not trying to “save the world”. Therefore, we only concentrate on methods for mitigating the interferences between co-located devices. Our typical use case is shown on figure 90:

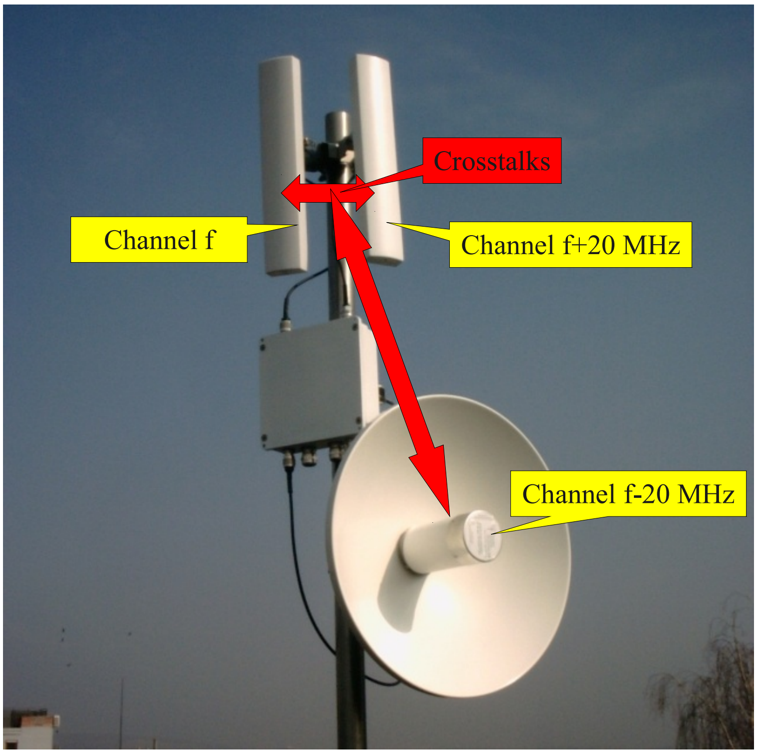


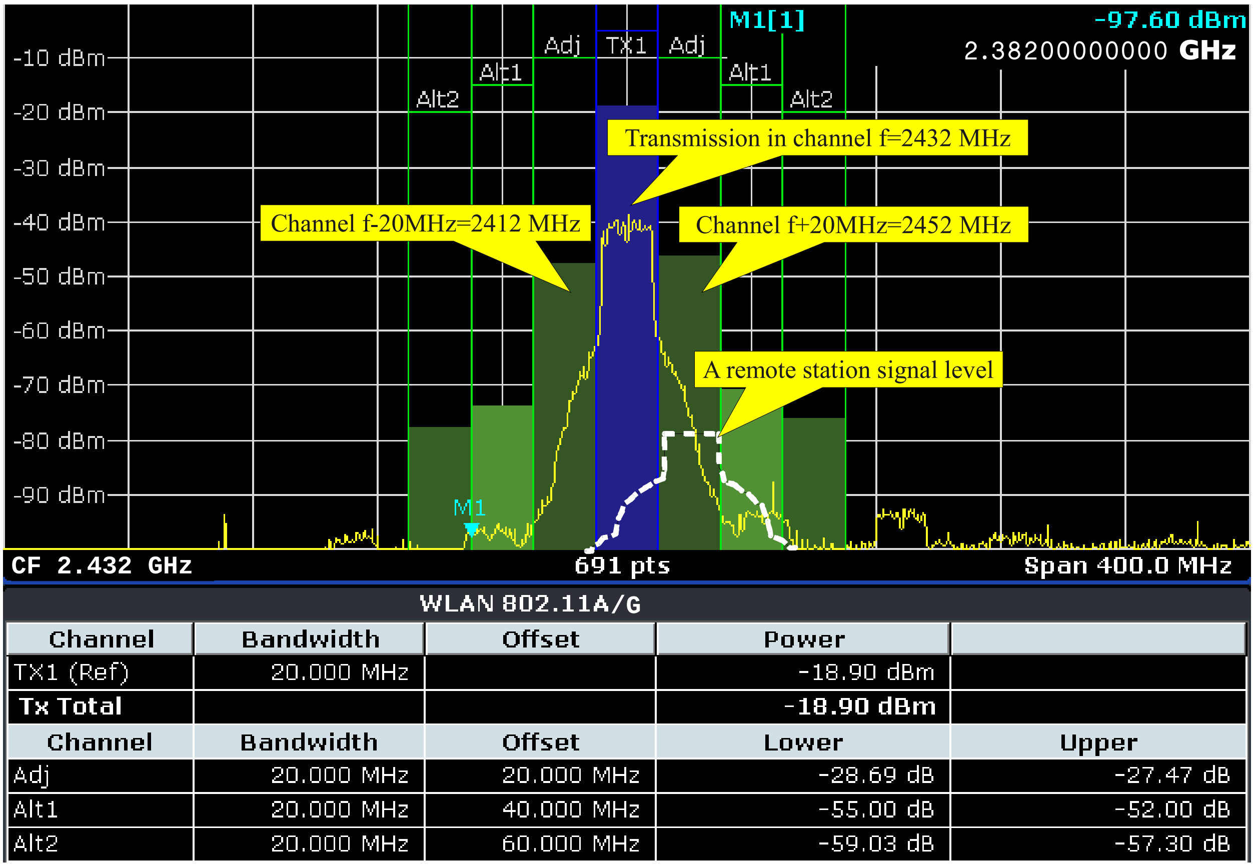
Figure 90: Typical colocation of several 802.11 links

The figure shows a RIA24 product [74] connected to two sectors and one parabolic antenna. The problem is the crosstalk between the antennas, so that a receiver on one sector antenna can “hear” transmitter on the second antenna with signal levels typically -30dBm to -40dBm.

The problem occurs when the network operator must use adjacent channels as shown on the figure, which is typical for 2.4GHz band with its 2-4 non-overlapping channels.

The problem is depicted on figure 91 showing spectrum captured by FSV7 analyzer during the “Channel N” transmission. If in the same timeslot the other co-located devices should receive data in adjacent channels N-1, N+1, the reception will not be possible as their channels are masked by channel N side lobes.

This whole chapter is dedicated to the possible solutions of such interferences – the adjacent channel interferences mitigation methods.



Transmission in channel f=2432 MHz

Channel f-20MHz=2412 MHz

Channel f+20MHz=2452 MHz

A remote station signal level

WLAN 802.11A/G

Channel	Bandwidth	Offset	Power	
TX1 (Ref)	20.000 MHz		-18.90 dBm	
Tx Total			**-18.90 dBm**	
Channel	**Bandwidth**	**Offset**	**Lower**	**Upper**
Adj	20.000 MHz	20.000 MHz	-28.69 dB	-27.47 dB
Alt1	20.000 MHz	40.000 MHz	-55.00 dB	-52.00 dB
Alt2	20.000 MHz	60.000 MHz	-59.03 dB	-57.30 dB

Figure 91: Typical signal levels of co-located devices on a spectrum analyzer

8.1 “Know your enemy” (and know yourself)

This method can be used instantly within all current wireless networks. It is not new at all - it was first published around 500 BC in [115], where the author says in the last verse in Chapter 3: “*So it is said that if you know your enemies and know yourself, you can win a hundred battles without a single loss.*

If you only know yourself, but not your opponent, you may win or may lose.

If you know neither yourself nor your enemy, you will always endanger yourself.”

Translated to the wireless networking domain:

- “knowing ourselves” means knowing the parameters of the devices we are using,
- “knowing our enemy” means knowing the interferences we experience, including their principles.

At this point, we already know the parameters of our devices – we have measured them and results are published in chapter 7. Also, we should already know the basic principles of co-location interferences, which were introduced in chapter 4.7. So we know the enemy and we know ourselves, now we only need the right strategy or method to “win the battle”.

The first method is a simple algorithm, which must be carried out for every co-located device:

1. By using the scan function, measure signal levels from those co-located devices which operate in adjacent channels. Let the strongest measured signal level be $|S_{max\_colocated}|$.
2. For the transmitter to work correctly, the following condition must hold:

$$|S_{max\_colocated}| < CCA\_threshold \qquad (31)$$

   where $CCA\_threshold$ is the maximum signal level in adjacent channel for which the CCA still returns “channel free” result.

3. Find the weakest power level from remote stations – let us call it $|S_{min\_remote}|$.
4. Note: while measuring $|\mathrm{S}_{\mathrm{max\_colocated}}|$ and $|S_{min\_remote}|$, some outgoing traffic should be generated on the other devices to measure the correct levels, e.g. by using the flood ping.
5. For the receiver to by able to detect transmissions from remote stations, the following condition must hold:

$$|S_{max\_colocated}| < |S_{min\_remote}| + Adjacent\_channel\_Rejection \qquad (32)$$

6. If the conditions (31) and (32) can not be fulfilled for the desired bit rates, which means that the signal levels from co-located transmitters in adjacent channels are greater than what the receiver can tolerate, the co-location setup must be changed by committing some of these adjustments:

   a) if (31) is the major problem (see the R52 with ath5k measurement in 7.3), a possible solution is using a driver with software-defined MAC layer with TDMA access method (e.g., Ubiquity AirMax, Mikrotik nstreme or nv2), which disables the CSMA/CA and CS/CCA HW completely and controls the medium access purely by using a centralized time-division based protocols. This solution may require to change the hardware of the AP and probably also the remote stations.

   b) If $CCA\_threshold$ is big enough, the easiest solution is reducing transmit power of the other co-located devices in order to get under the threshold defined by $min(CCA\_threshold,\ |S_{min\_remote}|+Adjacent\_channel\_Rejection)$.

c) Reduce the OFDM channel bandwidth of the AP and **all remote stations**. This is possible because aside the 20 MHz mode, the 802.11 standard defines additional 10 MHz and 5 MHz modes – see its clause 16.1. The 10 MHz channel is the most probable choice as it yields good adjacent channel performance and still reasonable throughput. As for the 5 MHz channel, please note that it still uses the same number of sub-carriers as 20 MHz, so it requires 4x better oscillator stability. In a quick adjacent channel interference immunity test, the 5 MHz behaviour was strange and even worse than 10 MHz.

d) Switching the co-located devices and **all the remote** stations to 802.11b-only mode, because the 802.11b adjacent channel rejection and *CCA_threshold* should be always better than 802.11g.

e) Disabling the higher transmission bit rates (in **all the remote stations**) which are affected by the adjacent channels transmissions levels.

f) Increase the spatial isolation of the co-located devices. This could be done by placing the devices further away from each other, as far as possible, or/and hiding them behind an obstacle (a wall, a chimney...).

7. Test the new setup using the client stations performance tool listed in chapter 11.2.
8. If these adjustments fail to improve the problem, then

a) try to set one of the devices (typically, the uplink) to a non-adjacent channel. In the 2.4 GHz band, the non-adjacent channels are usually not available, so transition to the 5 GHz band must be done. If the 5 GHz band is unusable due to high-density co-locations (like radio towers occupied by many operators), another frequency band must be used.

b) Replace the devices by something with better adjacent channel interference immunity.

Obviously, solutions 6.b) – 6.e) sacrifice the performance. The solution 6.b) may not be possible at all, because part of the remote stations may stop hearing the AP with reduced TX power. The step 6.e) may not be possible from the same reason – some of the remote stations would not hear the AP devices if they are moved to different place.

For these cases, it is good to know that there are also other methods available.

8.2 Coordinated TPC and DFS

This method can be used with any existing 802.11 network whose APs allow gathering statistics and controlling the transmitting power and frequency.

The basic use case is displayed on figure 92 below:

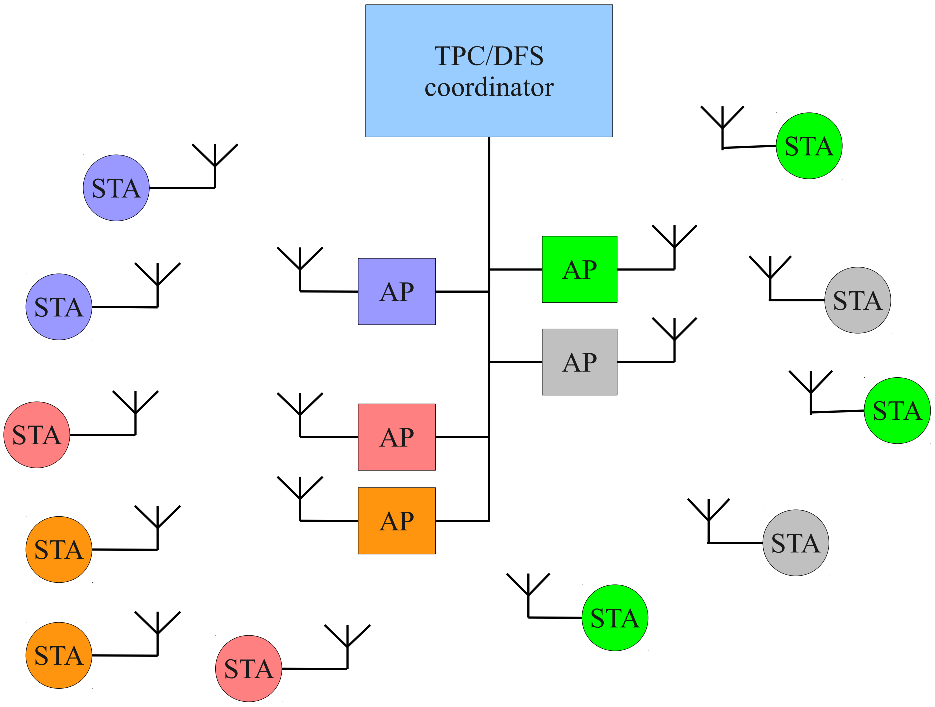


Figure 92: Cross-device adaptive TPC and DFS

The different colors are used to depict the current association between stations and access points. It is clear that in such a spatial configuration, some of the many stations and access points will experience adjacent channels interferences. Without coordination, the association of the stations is left to their own decision, which they made according to SSID, signal levels or RSSI. As the adjacent channels interferences may not be present during the association, it does not play any role in the process.

The adjacent channels interference may appear at any time between stations as they appear or move in the space dynamically.

But if an intelligent coordinator device is added to the network, some of the interferences can be avoided if the coordinator changes channels in a way that stations in one area will not use adjacent channels. The coordinator can change channels also for solving temporary interferences caused e.g. by microwave ovens or several stations using the ad-hoc mode. It can also reduce the access points transmit power in order to limit the adjacent channel interferences between them – the transmit power can be increased only for transmissions to the most distant stations. The coordinator can also commit hand-overs of stations to other access points in range in case it could improve the current spatial/frequency configuration.

These solutions are included in the coordinated dynamic frequency selection (DFS) and transmit power control (TPC) methods. This method is in fact not new at all, there are many academic papers describing its variants, e.g. [7], [38], [68], [72] and even many commercial products for a while now – see [21], [92] or [104]. But there is still a lot of space for improvements.

For example, the currently known implementations of DFS suffer from delays caused by frequency hops. These delays are inherent to the currently used voltage-controlled local oscillator designs, because the PLL needs some time to stabilize the frequency after every frequency change. Thus, all the associated stations must re-associate after the AP changes the channel. Re-association also invokes new DHCP leases for all the stations and all this is costly operation and user-unfriendly experience. That is why the wireless controllers try to avoid the channel hopping at the AP side and use only TPC and stations handover techniques.

A way for mitigating the channel hopping delays at the most critical AP side is to deploy redundant (backup) AP units and orchestrate them in a way that the current AP is active until the backup AP is fully functional on the new frequency. Then, the backup AP can take over the SSID and MAC address of the first AP which then becomes the backup at the same moment.

The redundant AP could be used for plethora of purposes while not active – it can scan the unused channels for activity, it can cooperate on the real-time-location service estimating the stations positions etc.

There is currently single free open source implementation of wireless controller – the OpenCAPWAP project [81]. It requires Atheros WLAN devices with an Linux OS which allows installing the OpenCAPWAP WTP application, although support for other AP types could be implemented in the future. Currently, the CAPWAP protocol is used only by Cisco in all its wireless products.

The OpenCAPWAP uses a simple frequency planning method [10] which associates stations to the closest AP in range. It does not do any advanced frequency allocation based e.g. on adjacent channels interference properties or spatial isolation between the associated stations.

Thus, the OpenCAPWAP project should be considered as a good starting point with wide range of possibilities for developing the controller algorithms, which can improve the behaviour of wireless networks. It can be also combined with real-time location and/or time-synchronization systems to achieve even better results.

8.3 Active noise cancellation

"This is so simple that it will not work", Philip Levis quotes a colleague as having warned [58]. Yet, it works – their paper [53] describes a functional prototype implementing full-duplex single channel wireless network device based on the active noise cancellation principle. The same principle is successfully used in headphones to mitigate the outside noise. The same principle should work also for mitigating the adjacent channel interferences between two co-located devices.

The principle is simple: we need to subtract the signal received from the antenna from the inverted and attenuated RF signal led from the other transmitter by a special coax cable - see the picture below:

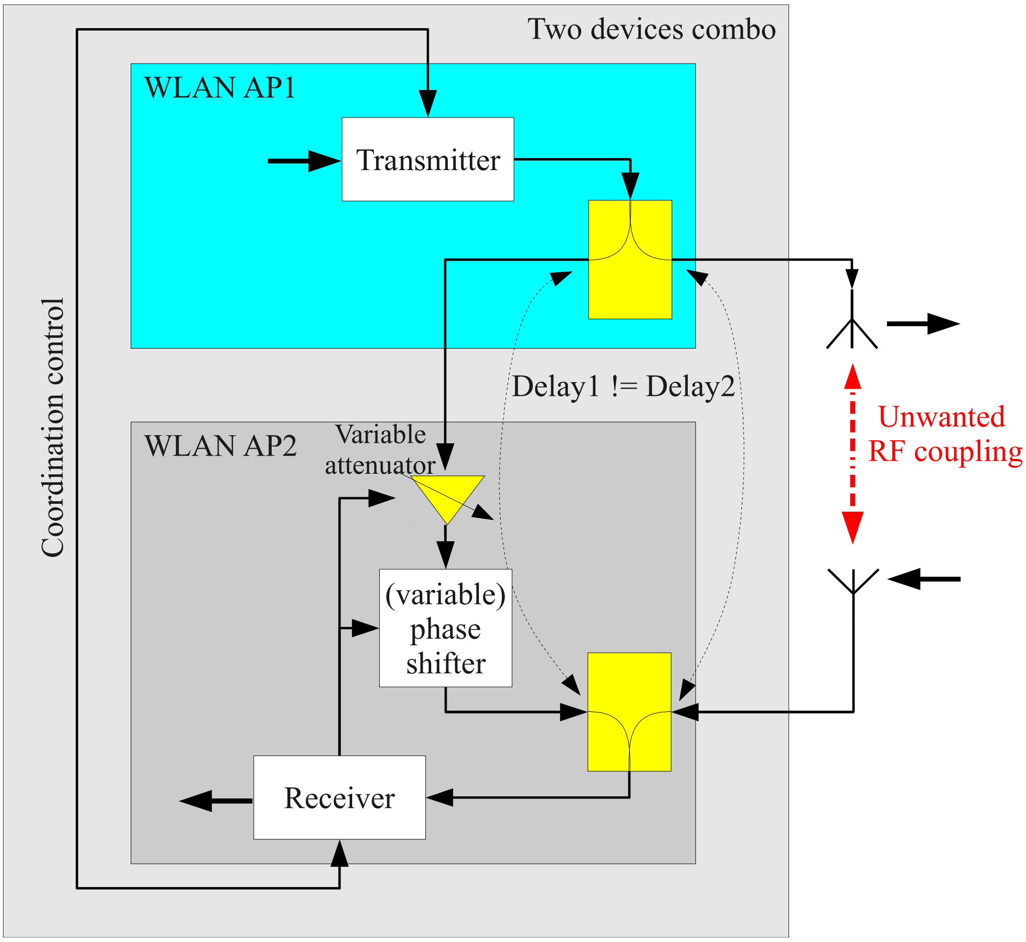


Figure 93: Active noise cancellation principle

The cancellation needs to be done before any other receiver circuitry to remove the strong interfering RF signal before it reaches the VGC (Voltage-Controlled Gain) input amplifier, which adjusts the signal to the level appropriate for the demodulator mixer.

The biggest challenge is the signal propagation delay. For the cancellation to work, we must get the phase alignment in the order of the fraction of the wavelength, and for 2.4GHz this means a fraction of 12.5 cm. We can achieve this dynamically by using a digitally-controlled RF phase shifter, e.g. [28] – this way we can compensate any delay without a need for time-consuming techniques like manual cable-length balancing etc.

After the phase is set to an optimal value, the variable attenuator must be set to a value which would yield the best cancellation. The whole process can be implemented as dynamic and self-adaptive: once after a time, when the receiver is idle, it can requests the other transmitter to transmit some data (using the coordination control communication channel, which would be typically a UDP request through the wire Ethernet). The transmitter acknowledges the request and transmits series of training WLAN packets, each preceded by wired Ethernet UDP message. After each wireless packet, the receiver would measure the received power and adjust the attenuator and phase shifter to a new value to achieve the best cancellation.

Up to this point, the method is exactly the same as the one proposed by the Stanford team [53]. The RF interference cancellation in their prototype yields interference attenuation of 20dB, which would improve the adjacent channel rejection greatly.

The Stanford team pushed the cancellation even further by implementing the "Digital Interference Cancellation" baseband block, which is not that easy to implement in our two-devices case, because it requires a high-throughput data link to be present between the devices.

8.4 Improving the ACPR

In general, improving the adjacent channel protection ratio (ACPR) is possible by shaping the transmitted spectrum in baseband, in RF or IF band or both. Although modifying the baseband signal processing is generally not possible in current 802.11 devices, it can be done within custom radio implementations, e.g. on software-defined radio. The RF or IF filtering is the more traditional technique - it was used heavily in super-heterodyne radios, but it is done in analog domain with all its disadvantages.

The baseband signal shaping can be done by several methods:

1. using analog filters between the DAC and IQ modulators. This is the most traditional method, but it is not easy to implement an analog filter with the required characteristics.
2. Using oversampling with interpolating and/or passband filters. A nice implementation of 802.11 interpolating/passband filter in GNURadio is published in [64]. The relation

between interpolation filtering and PAPR is shown in [31].

3. Signal shaping in frequency-domain, in time domain using FIR/IIR filters or by using windows for smoothing the transitions between the symbols.
4. Combination of all the above mentioned methods.

8.4.1 Time domain windowing in baseband according to 802.11 standard

802.11 standard mentions the signal shaping in clause 17.3.2.4: "*Smoothing the transition* (between symbols) *is required in order to reduce the spectral sidelobes of the transmitted waveform. However, the binding requirements are the spectral mask and modulation accuracy requirements, as detailed in 17.3.9.2 and 17.3.9.6. Time domain windowing, as described here, is just one way to achieve those objectives. The implementer may use other methods to achieve the same goal, such as frequency domain filtering. Therefore, the transition shape and duration of the transition are informative parameters.*"

Therefore, the standard specifies only two examples of window functions:

- the Equation (17-4) and figure 17-2 in clause 17.3.2.4 defines the squared sine windowing function which could/should be utilized for smoothing the inter-symbol transitions in the preamble:

$$w_T[t]=\left\{\begin{array}{ll} \sin^2\left(\frac{\pi}{2}(0.5+t/T_{TR})\right) & -T_{TR}/2<t<T_{TR}/2 \\ 1 & T_{TR}/2<t<T-T_{TR}/2 \\ \sin^2\left(\frac{\pi}{2}(0.5-(t-T)/T_{TR})\right) & T-T_{TR}/2\le t<T+T_{TR}/2 \end{array}\right\} \tag{33}$$

Where

- T is the symbol duration (period)
- T_{TR} is the parameter – the chosen transition time. The shorter T_{TR} is chosen, the more rectangular shape the window has.

- the Equation (17-5) defines simple rectangular window, which could/should be used elsewhere:

$$w_T[n]=w_T[n\,T_s]=\left\{\begin{array}{cc} 1 & 1\le n\le 79 \\ 0.5 & n=0\vee 80 \\ 0 & otherwise \end{array}\right\} \tag{34}$$

A similar window was used the experiments described in the chapter 8.4.3.

8.4.2 Time domain filtering in baseband with R&S SMBV100A signal generator

In the final phase of this thesis, the R&S SMBV100A generator arrived to our department so it was possible to make several experiments with OFDM signal shaping with high quality baseband signal source and RF modulator. The generator can be controlled remotely either via SCPI commands or the R&S WinIQSIM2 software. Both ways allow generating arbitrary signals. For saving the time spent on the experiments, WinIQSIM2 was used so SCPI programming was not necessary. For generating arbitrary OFDM signals, WinIQSIM2 offers the “Custom digital modulation” and “Multi carrier” baseband blocks. However, the “Multi carrier” block use a specific signal flow which influences the experimental setup in a way that it is necessary to discuss it here. The signal flow is shown on figure 94.

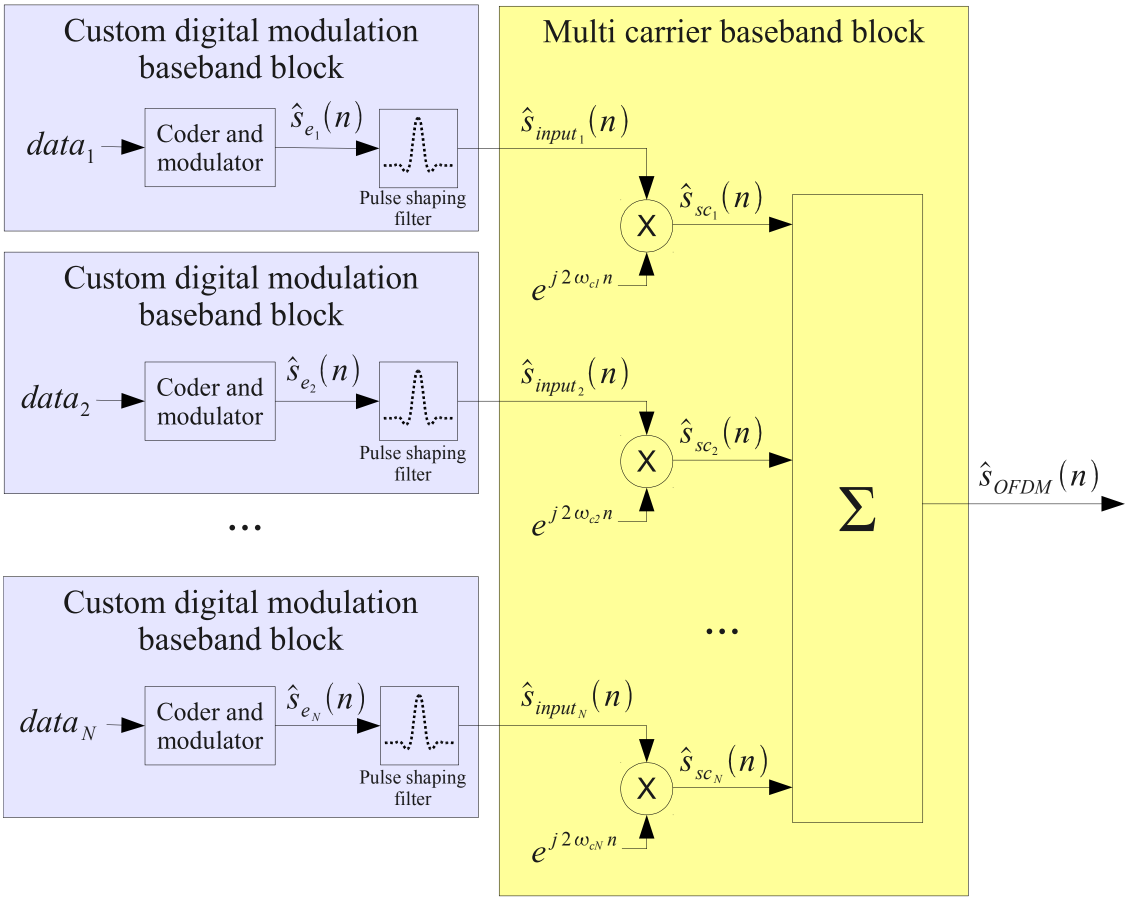


Figure 94: WinIQSIM2 "Multi carrier" baseband block signal flow

As shown on the figure, the Multi carrier block is in fact a bank of modulators/equidistant frequency shifters, whose inputs are the $\hat{s}_{INPUT1}(n)$ ... $\hat{s}_{INPUTN}(n)$ complex envelope signals. These signals must be generated first, typically by the “Custom digital modulation” baseband block, which allows selecting arbitrary data pattern, a data coder, a modulation and (optionally) a filter.

Note that the guard interval is not supported by neither Multi carrier block nor Custom digital modulation block, but it is not needed for our experiments.

The output of the Multi carrier baseband block is the complex baseband OFDM signal, which can be loaded into the generator's ARB (arbitrary baseband signal generator memory) and up-converted to an RF band.

The difference between a classic IFFT OFDM implementation and the WinIQSIM2 Multi carrier baseband block is that the WinIQSIM2 does the filtering on each individual sub-carrier complex envelope, while a typical IFFT OFDM implementation would filter only the IFFT output signal.

8.4.3 The sub-carrier signals and baseband filters

For evaluating the impact of various baseband filtering methods, a single carrier QPSK baseband signal was generated by the WinIQSIM2 Custom digital modulation block and three different baseband filters were applied:

- rectangular filter, which is very similar to the WLAN 802.11 windowing functions described by equation (34) or (33),
- root cosine filter with roll off factor = 0.35,
- pure Gauss filter with BxT = 0.30.

The filtered sub-carrier I/Q signals and their spectra are shown on figures 95-99.

Note: in case of rectangular and Gauss filter, WinIQSIM2 automatically uses oversampling rate which is 8x higher than with root cosine filter.

These I/Q signals were used in the WinIQSIM2 Multi carrier modulation block for generating the OFDM signals, whose spectra are shown on figures 101-103. It is clearly visible how the baseband filtering helps to cut down the adjacent channels emissions – while the rectangular filter reduces the side lobes only very slightly, the root cosine and pure Gauss filters provide almost ideal rectangular spectrum.

Figures 104-108 show the output of the ACPR measurement made with the FSV7 spectrum analyzer for these baseband signals modulated 2.432 GHz frequency by the SMBV100A generator.

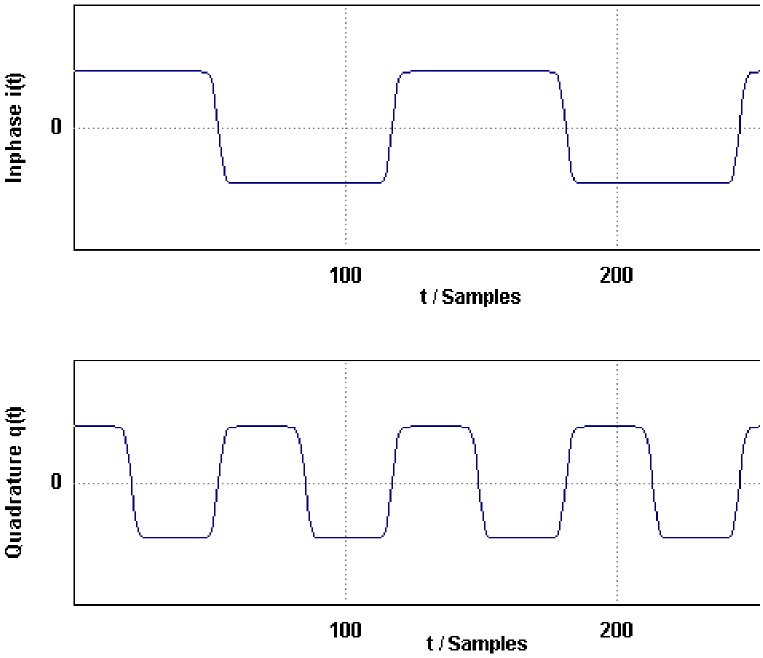


Figure 95: QPSK with rectangular filter

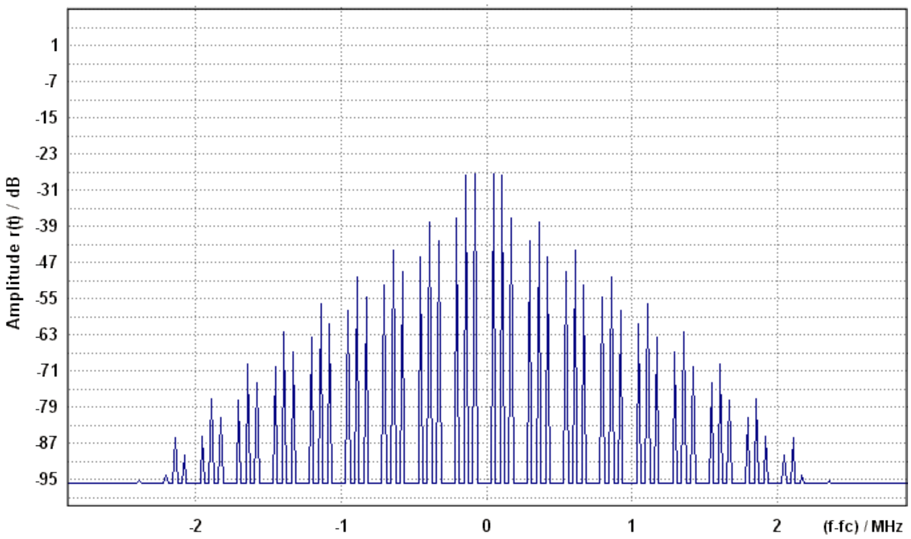


Figure 96: Spectrum of QPSK with rectangular filter

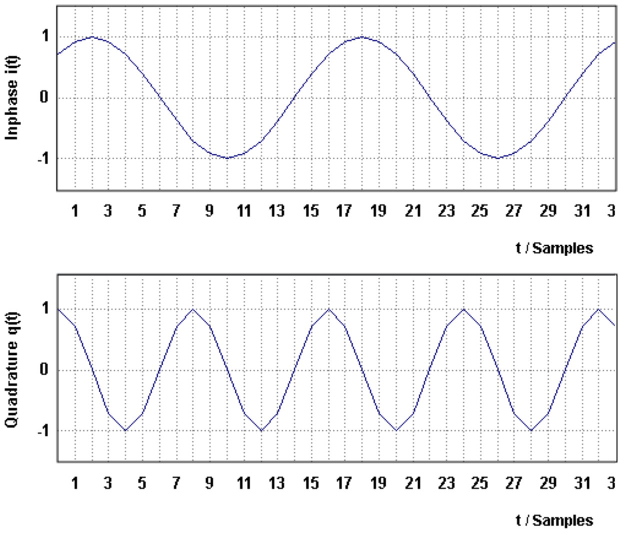


Figure 97: QPSK with root cosine filter

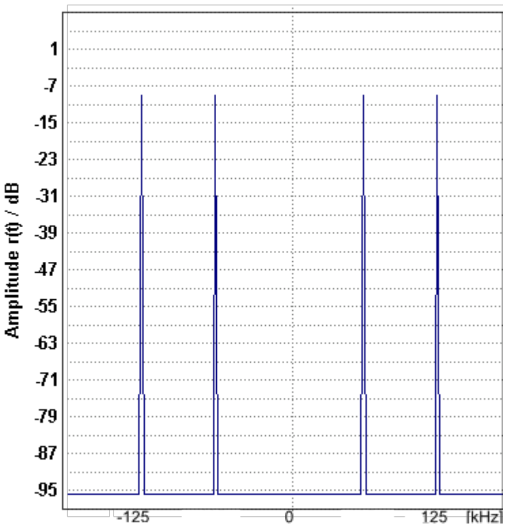


Figure 98: Spectrum of QPSK with root cosine filter

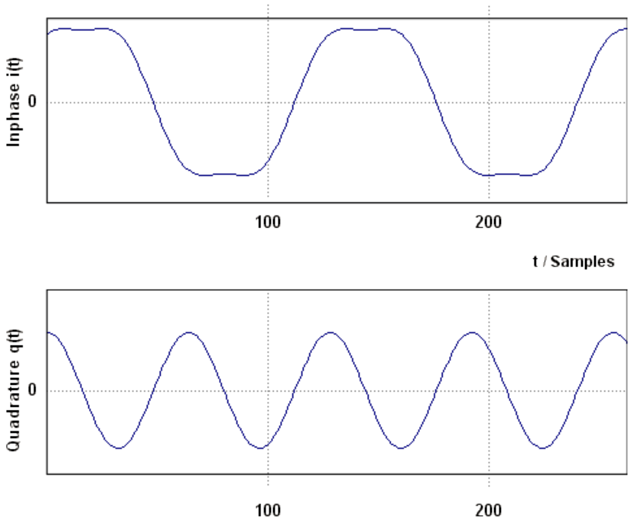


Figure 100: QPSK with Gauss filter

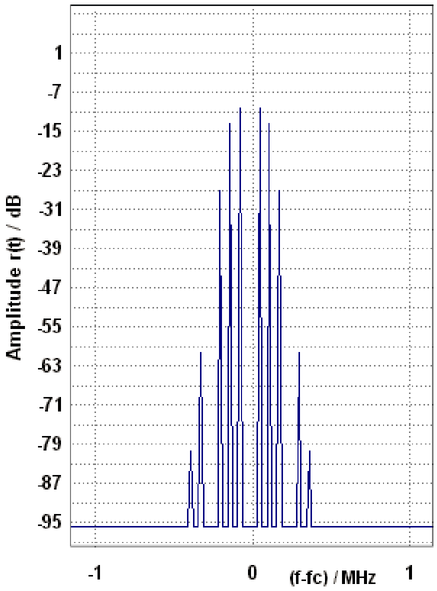


Figure 99: Spectrum of QPSK with Gauss filter

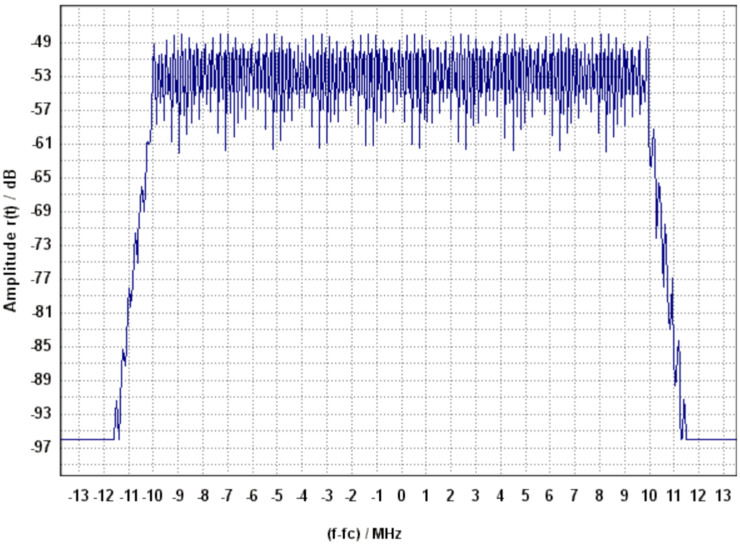


Figure 101: OFDM signal with rectangular filter

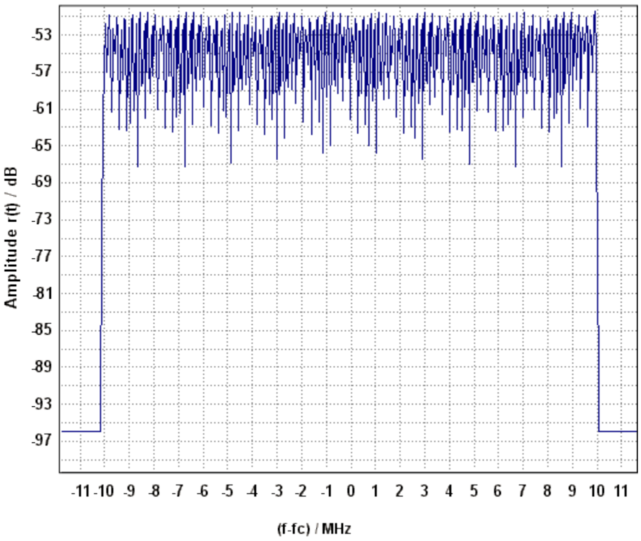


Figure 102: OFDM signal with gauss filter

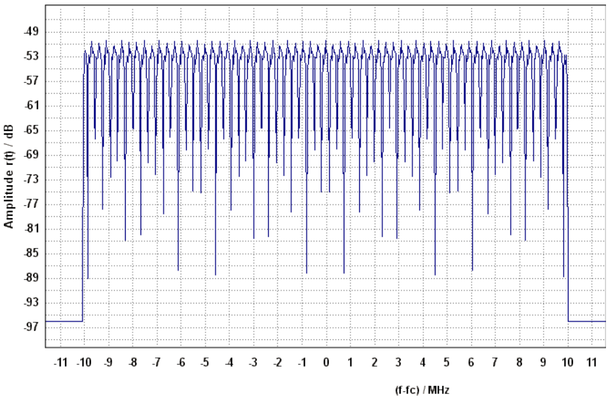


Figure 103: OFDM with root cosine filter

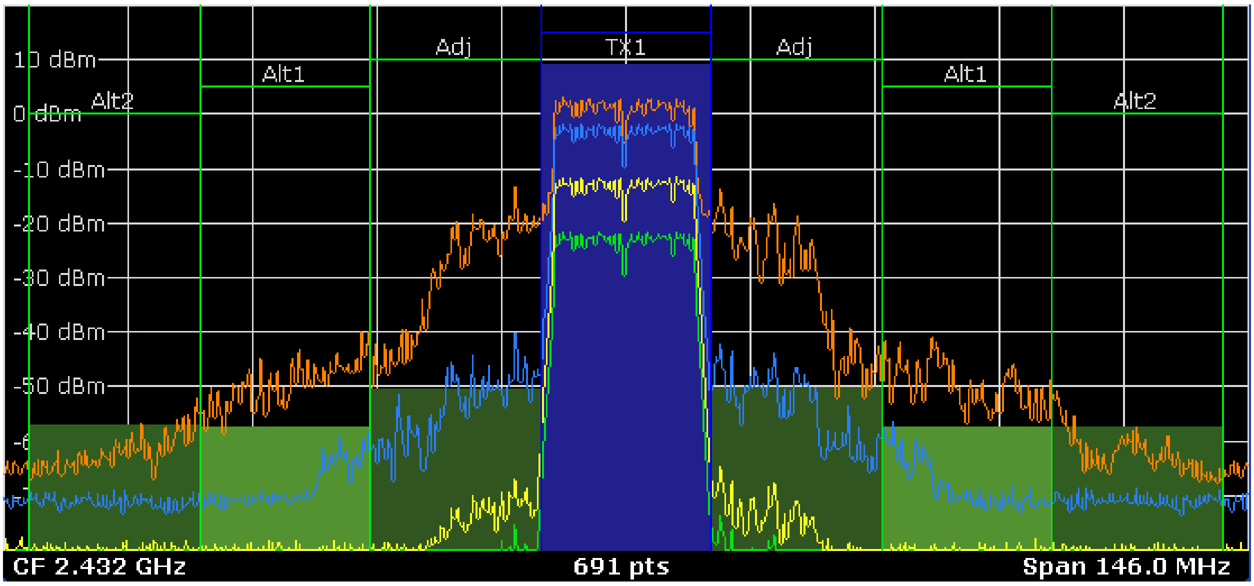
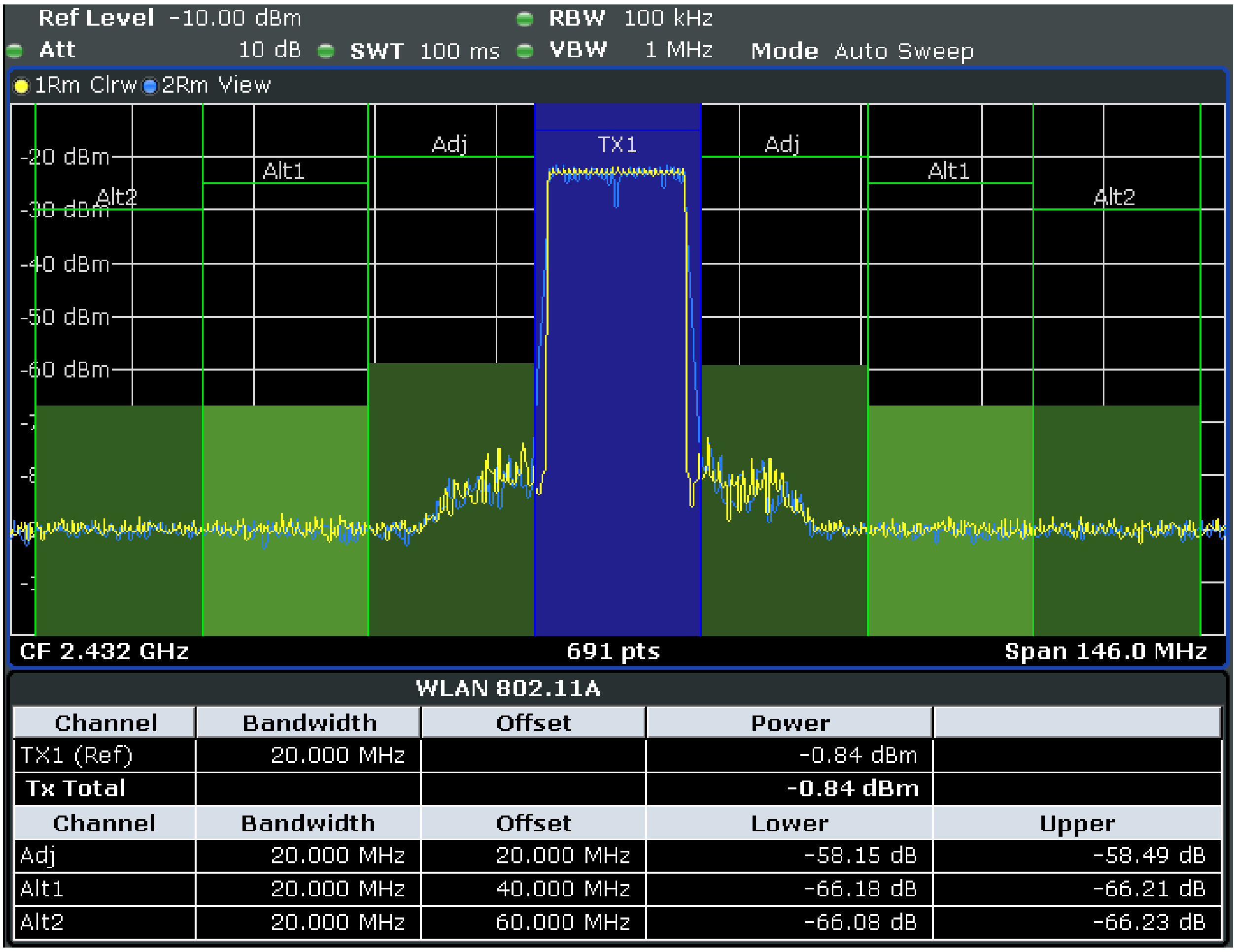


Figure 104: OFDM with rectangular windowing filter and 0, 10, 20, 30 dBm RF power.

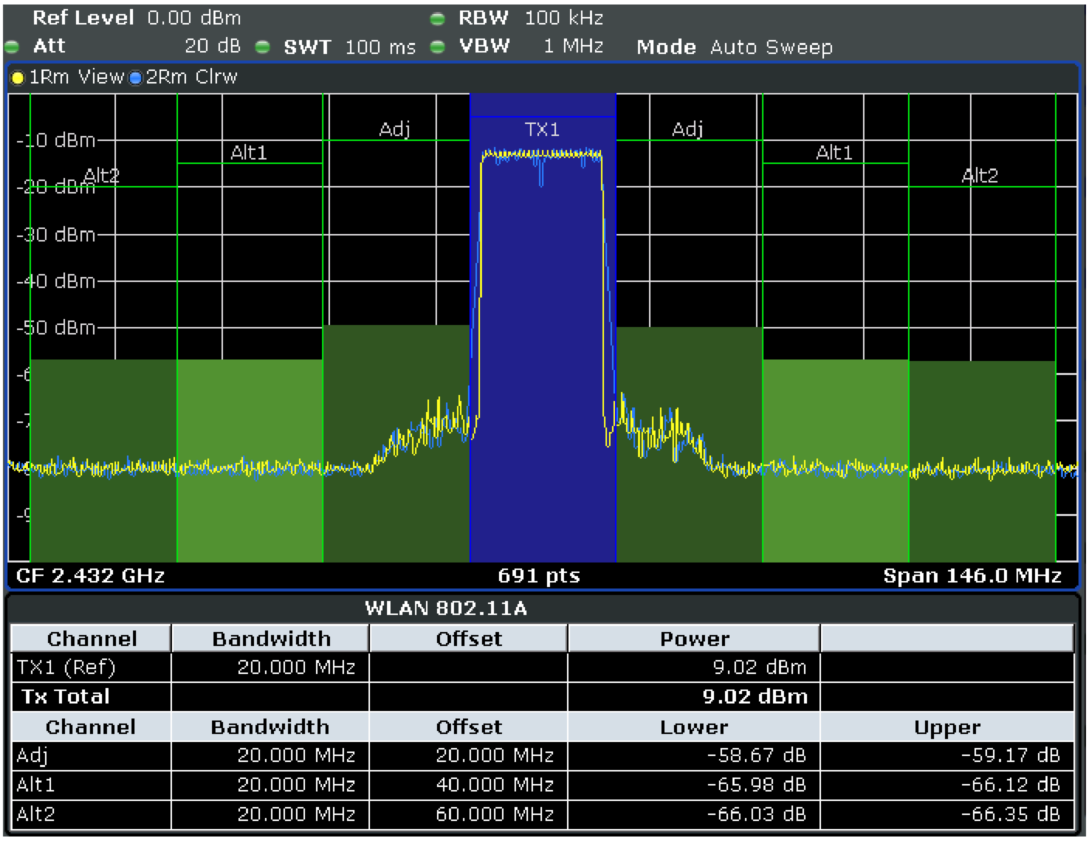


Ref Level -10.00 dBm RBW 100 kHz
Att 10 dB SWT 100 ms VBW 1 MHz Mode Auto Sweep

WLAN 802.11A

Channel	Bandwidth	Offset	Power	
TX1 (Ref)	20.000 MHz		-0.84 dBm	
Tx Total			**-0.84 dBm**	
Channel	**Bandwidth**	**Offset**	**Lower**	**Upper**
Adj	20.000 MHz	20.000 MHz	-58.15 dB	-58.49 dB
Alt1	20.000 MHz	40.000 MHz	-66.18 dB	-66.21 dB
Alt2	20.000 MHz	60.000 MHz	-66.08 dB	-66.23 dB

Figure 105: OFDM with rectangular (blue) vs. root cosine (yellow) filters: 0 dBm RF power.



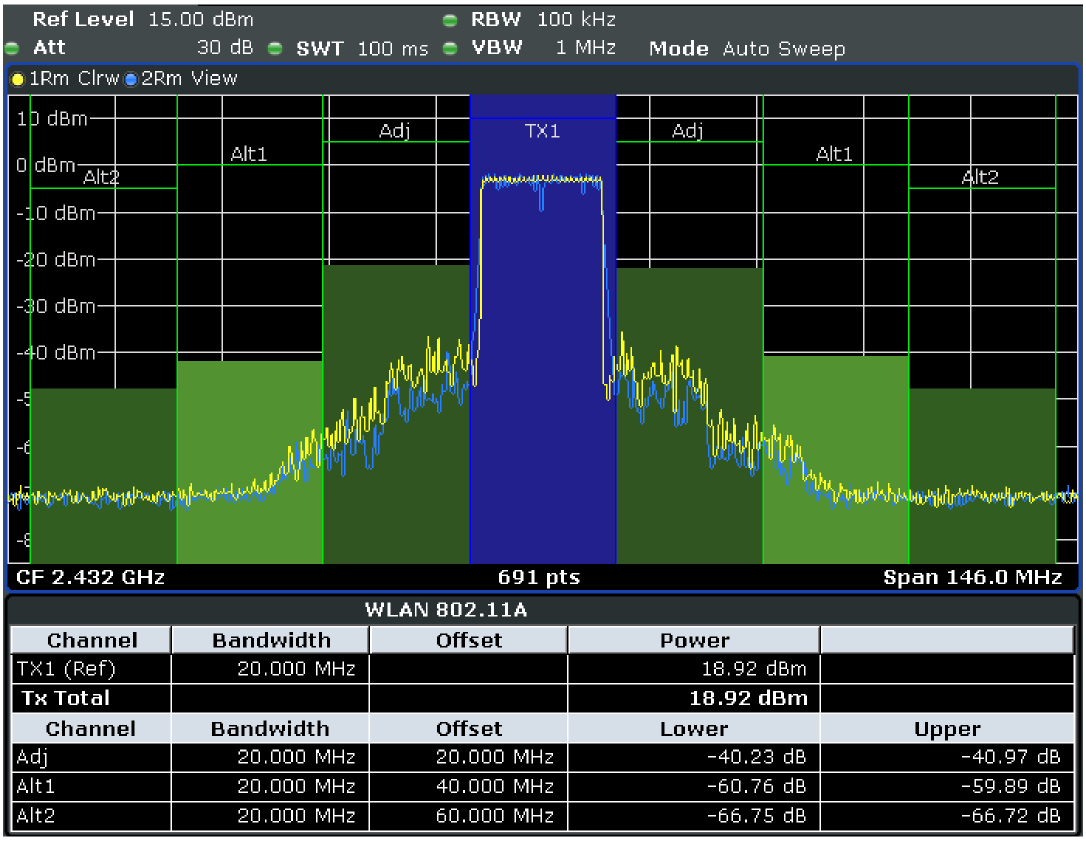
Ref Level 0.00 dBm RBW 100 kHz
Att 20 dB SWT 100 ms VBW 1 MHz Mode Auto Sweep
1Rm View 2Rm Clrw

CF 2.432 GHz 691 pts Span 146.0 MHz

WLAN 802.11A

Channel	Bandwidth	Offset	Power	
TX1 (Ref)	20.000 MHz		9.02 dBm	
Tx Total			**9.02 dBm**	
Channel	**Bandwidth**	**Offset**	**Lower**	**Upper**
Adj	20.000 MHz	20.000 MHz	-58.67 dB	-59.17 dB
Alt1	20.000 MHz	40.000 MHz	-65.98 dB	-66.12 dB
Alt2	20.000 MHz	60.000 MHz	-66.03 dB	-66.35 dB

Figure 106: OFDM with rectangular vs. root cosne filters: 10 dBm RF power.



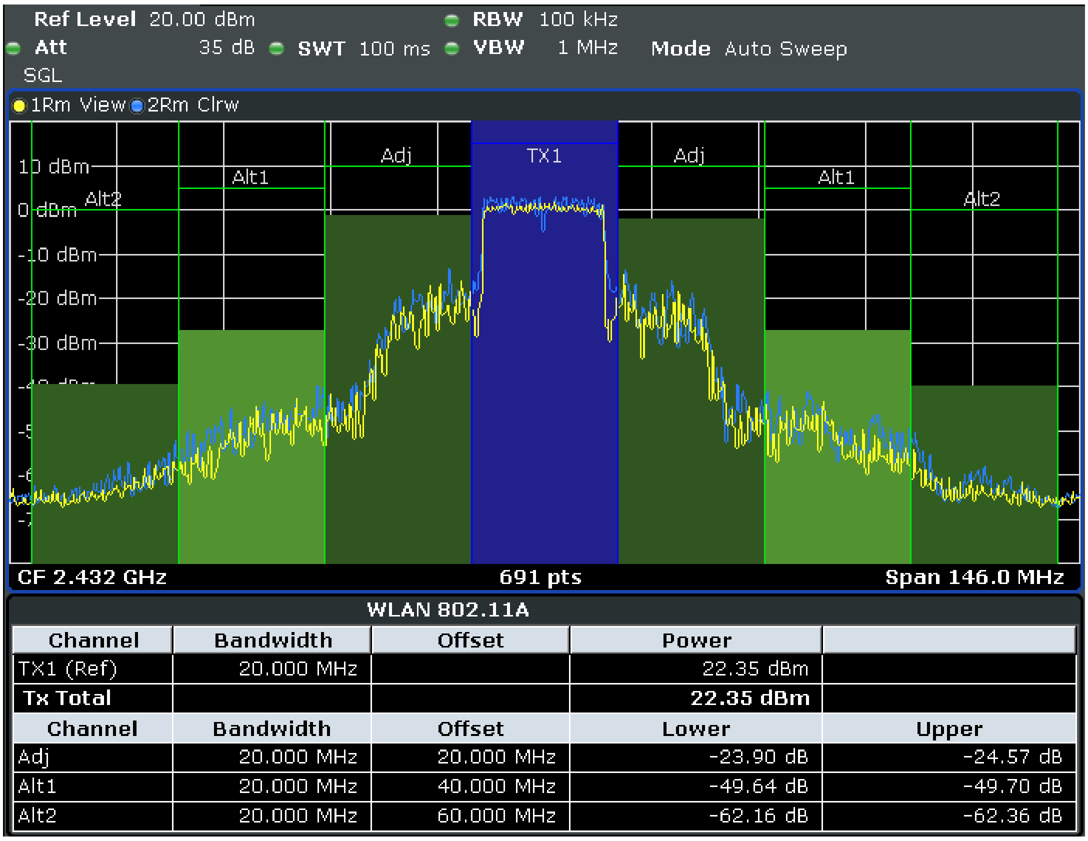
Ref Level 15.00 dBm RBW 100 kHz
Att 30 dB SWT 100 ms VBW 1 MHz Mode Auto Sweep
1Rm Clrw 2Rm View

CF 2.432 GHz 691 pts Span 146.0 MHz

WLAN 802.11A

Channel	Bandwidth	Offset	Power	
TX1 (Ref)	20.000 MHz		18.92 dBm	
Tx Total			**18.92 dBm**	
Channel	**Bandwidth**	**Offset**	**Lower**	**Upper**
Adj	20.000 MHz	20.000 MHz	-40.23 dB	-40.97 dB
Alt1	20.000 MHz	40.000 MHz	-60.76 dB	-59.89 dB
Alt2	20.000 MHz	60.000 MHz	-66.75 dB	-66.72 dB

Figure 107: Rectangular (blue) vs. root cosine (yellow) filters: 20 dBm RF power.


Ref Level 20.00 dBm RBW 100 kHz
Att 35 dB SWT 100 ms VBW 1 MHz Mode Auto Sweep
SGL
1Rm View 2Rm Clrw

WLAN 802.11A

Channel	Bandwidth	Offset	Power	
TX1 (Ref)	20.000 MHz		22.35 dBm	
Tx Total			**22.35 dBm**	
Channel	**Bandwidth**	**Offset**	**Lower**	**Upper**
Adj	20.000 MHz	20.000 MHz	-23.90 dB	-24.57 dB
Alt1	20.000 MHz	40.000 MHz	-49.64 dB	-49.70 dB
Alt2	20.000 MHz	60.000 MHz	-62.16 dB	-62.36 dB

Figure 108: Rectangular (blue) vs. root cosine (yellow) filters: 30 dBm RF power.

These figures demonstrate the impact of the RF stage on the transmitted spectrum: while the baseband OFDM signal generated with root cosine filter yields almost ideal rectangular main lobe in the RF spectrum, its adjacent channel spectrum is very similar to the rectangular filter for RF power up to 10 dBm. However, for RF power set to 20 dBm, the ACPR is about 6dB worse for the root cosine filtered baseband signal. This is caused by the higher PAPR of the root cosine filtered signal – see [19] or [52] for discussion about the root cosine filters and PAPR.

The 30dBm RF power is shown only as a demonstration of the signal distortion caused by operating the RF amplifier largely in the saturation mode – in such conditions, the baseband filtering does not seem to have any influence on the RF spectrum. The measured ACPR is summarized in the Table 8.

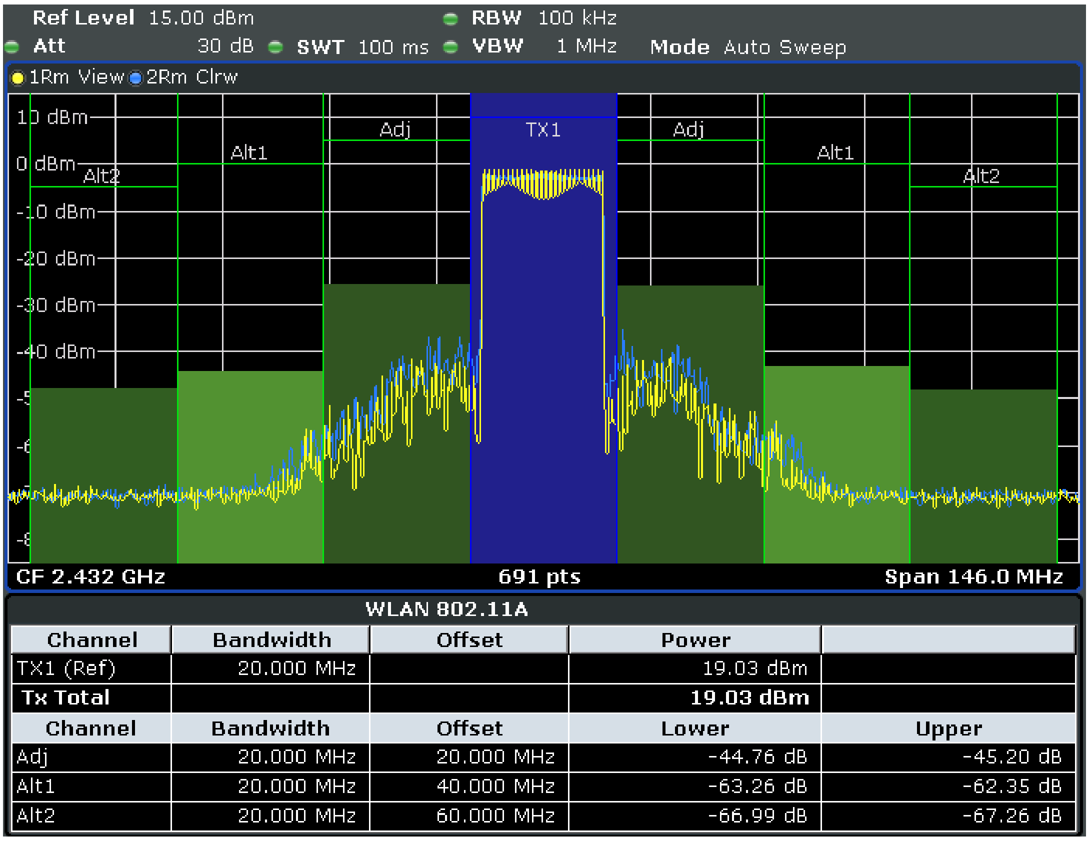
Table 8: ACPR for rectangular and root cosine filters

RF power [dBm]	ACPR	
	Rectangular filter	**Root cosine filter**
0	-59 dB	-58 dB
10	-59 dB	-59 dB
20	-46 dB	-40 dB
30	-24 dB	-24 dB

8.4.4 OFDM signal with baseband filtering versus single OFDM symbol spectrum

Just as a simple “sanity test” regarding the PAPR, a comparison of the OFDM baseband filtered signal with the single repeated OFDM symbol was made.

The baseband spectrum of (repeated) single OFDM symbol is an ideal rectangle. However, when such signal is modulated to an RF band and amplified, some energy will appear in the adjacent channels as a consequence of the RF amplifier non-linearity. This is shown on figures 109-110, where the ACPR was measured on FSV7 for both single OFDM symbol and for root cosine filtered baseband signal – the same which was used on figure 107. On figure 109, the 20dBm RF output power was applied and the ACPR of the single OFDM symbol was -45 dB, which was 5 dB better than with the root cosine filtered signal and 1 dB worse than the rectangular filter signal – see Table 8.



Ref Level 15.00 dBm   RBW 100 kHz

Att 30 dB   SWT 100 ms   VBW 1 MHz   Mode Auto Sweep

1Rm View   2Rm Clrw

CF 2.432 GHz   691 pts   Span 146.0 MHz

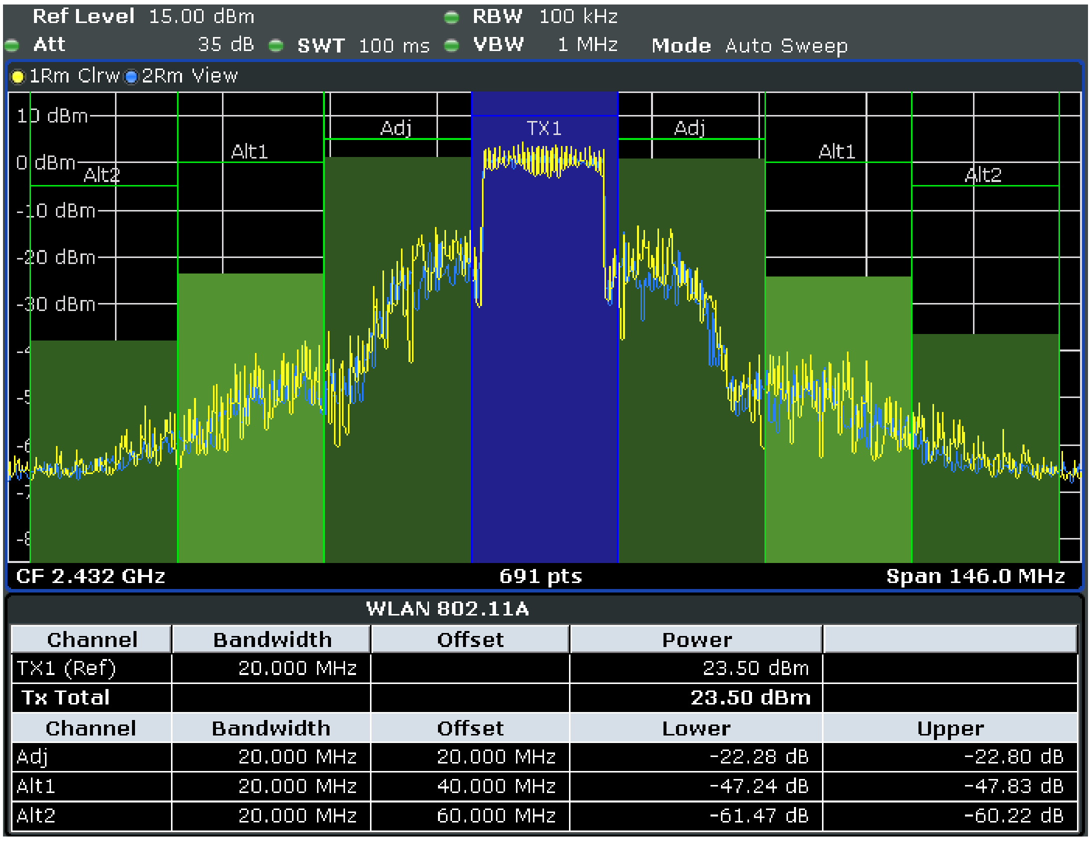
WLAN 802.11A

Channel	Bandwidth	Offset	Power	
TX1 (Ref)	20.000 MHz		19.03 dBm	
Tx Total			**19.03 dBm**	
Channel	**Bandwidth**	**Offset**	**Lower**	**Upper**
Adj	20.000 MHz	20.000 MHz	-44.76 dB	-45.20 dB
Alt1	20.000 MHz	40.000 MHz	-63.26 dB	-62.35 dB
Alt2	20.000 MHz	60.000 MHz	-66.99 dB	-67.26 dB

Figure 109: OFDM with root cosine filter (blue) vs. single OFDM symbol (yellow)

On figure 110, the 30 dBm RF power was applied and this time the ACPR of the single OFDM symbol was even worse than any of the filtered signals discussed previously.

The morale is that it does not make sense to spend too much efforts with the baseband signal filtering without taking care about resulting PAPR – otherwise a simple rectangular window filter will result in a better RF ACPR than using filters with higher roll-off.



Ref Level 15.00 dBm RBW 100 kHz
Att 35 dB SWT 100 ms VBW 1 MHz Mode Auto Sweep

CF 2.432 GHz 691 pts Span 146.0 MHz

WLAN 802.11A

Channel	Bandwidth	Offset	Power	
TX1 (Ref)	20.000 MHz		23.50 dBm	
Tx Total			**23.50 dBm**	
Channel	**Bandwidth**	**Offset**	**Lower**	**Upper**
Adj	20.000 MHz	20.000 MHz	-22.28 dB	-22.80 dB
Alt1	20.000 MHz	40.000 MHz	-47.24 dB	-47.83 dB
Alt2	20.000 MHz	60.000 MHz	-61.47 dB	-60.22 dB

Figure 110: OFDM with root cosine filter vs. single OFDM symbol, 30 dBm

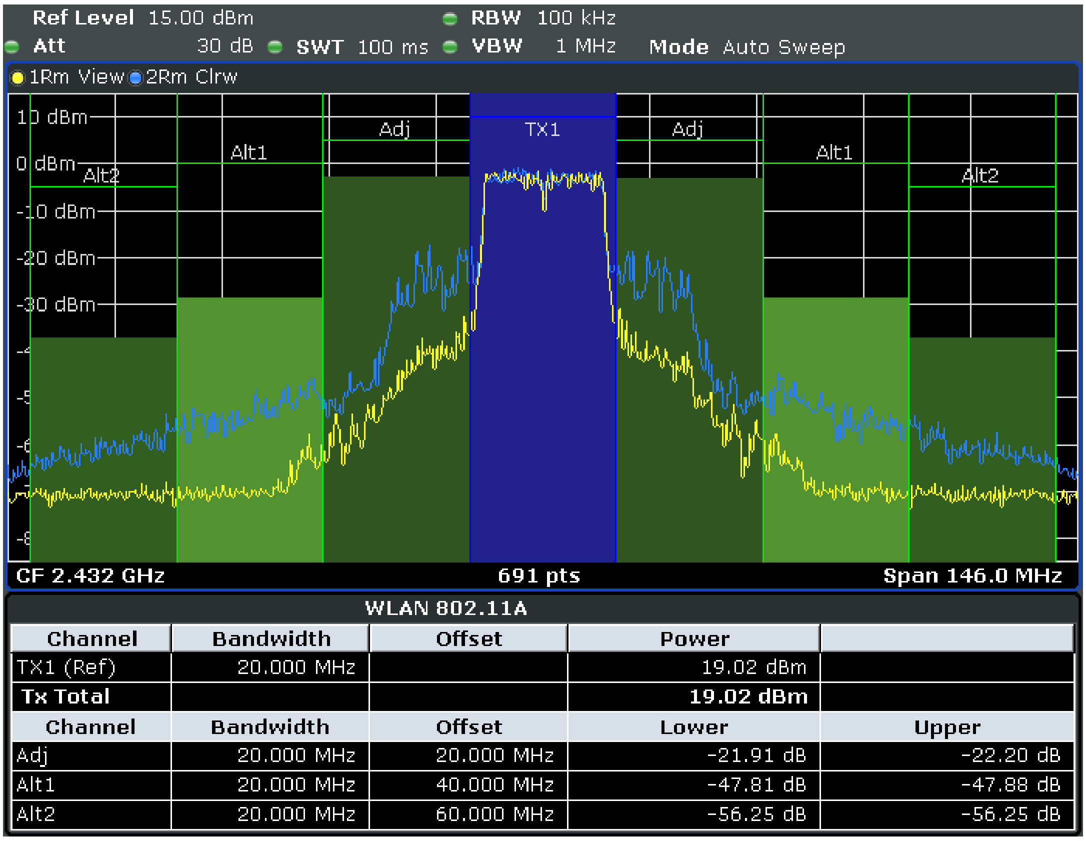
8.4.5 PAPR reduction

To demonstrate the influence of PAPR on the ACPR, the PAPR was reduced by clipping the signal peaks on the output of the WinIQSIM2 Multi carrier signal block. The rectangular window filtered QPSK signal was used as the input of the Multi carrier signal block to see the ACPR improvement by the signal “clipping and filtering” PAPR reduction method.

The figure 111 shows the comparison of the non-clipped OFDM signal spectra with the OFDM signal with peaks clipped to the signal RMS level. The clipping creates sharp edges in the signal which degrades the ACPR by 23 dB.

The sharp edges can be filtered by the WinIQSIM2 Multi carrier signal block output low pass filter (LPF). This is shown on figure 112, where the LPF was set to (±)9 MHz. The filtering improves the ACPR substantially to -48 dB, but to confirm the improvement, EVM measurement should be done at the transmitter.

There are many other PAPR reduction techniques (see e.g. [70], [78] resp. [82]), but only few of them are applicable/compatible with the 802.11a/g/n coding and modulation schemes. Aside of clipping and filtering, signal peak reduction by windowing can be used, as well as subcarriers pulse-shaping methods proposed in [41], [43] or [105].



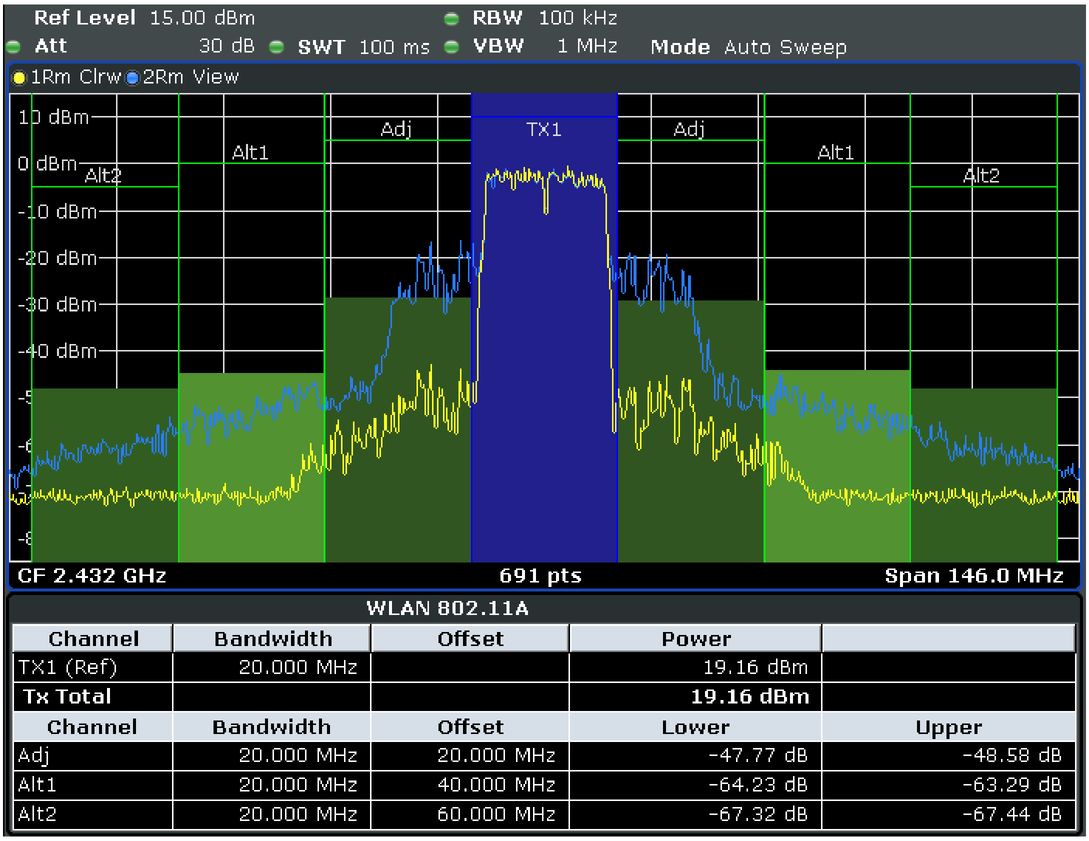
Ref Level 15.00 dBm RBW 100 kHz
Att 30 dB SWT 100 ms VBW 1 MHz Mode Auto Sweep
1Rm View 2Rm Clrw

CF 2.432 GHz 691 pts Span 146.0 MHz

WLAN 802.11A

Channel	Bandwidth	Offset	Power	
TX1 (Ref)	20.000 MHz		19.02 dBm	
Tx Total			**19.02 dBm**	
Channel	**Bandwidth**	**Offset**	**Lower**	**Upper**
Adj	20.000 MHz	20.000 MHz	-21.91 dB	-22.20 dB
Alt1	20.000 MHz	40.000 MHz	-47.81 dB	-47.88 dB
Alt2	20.000 MHz	60.000 MHz	-56.25 dB	-56.25 dB

Figure 111: OFDM with no clipping (yellow) vs. clipping to RMS, LPF=20MHz



Ref Level 15.00 dBm RBW 100 kHz
Att 30 dB SWT 100 ms VBW 1 MHz Mode Auto Sweep
1Rm Clrw 2Rm View

CF 2.432 GHz 691 pts Span 146.0 MHz

WLAN 802.11A

Channel	Bandwidth	Offset	Power	
TX1 (Ref)	20.000 MHz		19.16 dBm	
Tx Total			**19.16 dBm**	
Channel	**Bandwidth**	**Offset**	**Lower**	**Upper**
Adj	20.000 MHz	20.000 MHz	-47.77 dB	-48.58 dB
Alt1	20.000 MHz	40.000 MHz	-64.23 dB	-63.29 dB
Alt2	20.000 MHz	60.000 MHz	-67.32 dB	-67.44 dB

Figure 112: Clipping to RMS, LPF=20 MHz (blue) vs. 9 MHz (yellow)

The pulse shaping method published in [105] uses the same signal generation scheme as the WinIQSIM2 (see figure 94).

Its core idea is to use the the root cosine pulse shape in the pulse shaping filters, but with different time shifts for different subcarriers. The author declares the resulting PAPR can be close to the level of single-carrier signals, which sounds very attractive. However, no further information is provided about verification of the author's claims, e.g. by implementing the whole transmit-receive chain. So far the only obvious disadvantage of this method seems in the higher computational complexity. The computational time complexity of this method per 1 OFDM symbol is:

$$T(N)=O(M.N+M) \tag{35}$$

where

- N is the number of subcarriers,
- M is the symbol length in samples. Note that for 802.11 with 52 subcarriers, M would be >= 2 N , because the root cosine pulse and the input I/Q signals should be oversampled for better results.
- $O(M)$ is the number of complex multiplies needed for modulating the raised cosine pulse with I/Q signals of one sub-carrier,
- $O(M.N)$ is the complexity of the direct OFDM implementation. It is the number of complex multiplies needed for modulating N subcarriers by signals with M samples.

So it is clear that the classic IFFT OFDM implementation is much better in the time complexity aspect, because its time complexity with simple rectangle windowing at the output is:

$$T(N)=O(N\log_2 N+L) \tag{36}$$

where

- N is the number of subcarriers,
- L is the symbol length in samples, which is N+*guard_interval_samples*
- $O(N\log_2 N)$ is time complexity of the IFFT algorithm
- $O(L)$ is the time complexity of the rectangle windowing function.

8.4.6 RF signal filtering

The last method for improving the PAPR is the RF bandpass filtering. Currently, only the external RF fixed filters are available on the market with reasonable prices like [76] or [84]. A typical use case for such fixed-frequency filters is a corporate interior WLAN, where the network administrators have full control over the frequencies used inside the company building(s) so they have no need for changing the operating channels.

However, for exterior wireless networks and licence-free frequency bands, most operators would feel uncomfortable about loosing the possibility to change the channel by a remote command.

Unfortunately, there are only few tunable RF filters for 2.4 GHz frequency band available on the market today (e.g. [48], [83] or [114]) and probably none for the 5 GHz band.

The current state of the art in the area of tunable RF filters is based on two technologies:

- Ferroelectric variable capacitors and switches, see [2], [62], [125].
- RF MEMS (micro-electro-mechanical systems) switches and variable capacitors, see [87] for a current overview of the technology.

Other tunable RF components like III-V semiconducting varactors, switched capacitors, switched varactors and YIG ferrites have also being used in RF tunable filters, however various drawbacks inhibited their wider acceptance. There are also many efforts to exploit the combined benefits of the technologies mentioned above in hybrid devices, see [95].

Although these technologies has been explored for more than 30 years, only the recent advances in MEMS and ferroelectric manufacturing are allowing their substantial market penetration. The impending success of ferroelectric memory components ([6], [34], [123]) and the massive success of mobile phones containing MEMS accelerometers and MEMS gyroscopes indicate that manufacturing of MEMS and ferroelectric devices has definitely reached its maturity. Now the hopes are high that the RF MEMS and/or RF ferroelectric tunable filters can get a similar attention, e.g. because of to the demand for small scale software-defined radios needed in forthcoming LTE mobile phones.

8.5 Distributed OFDM symbol-level synchronization

The principle of this method is simple: multiple OFDM transmitters are synchronized in a way that from the receiver perspective, they transmit the same OFDM symbols data simultaneously. Thus, the OFDM symbols add up non-destructively and the result is increased RSSI.

This method has already found widespread use in "Single frequency network" (SFN) technology, which is specified for DVB-T in ETSI EN 300 744 standard. A thorough discussion about the principles, design and planning of DVB-T SFN can be found in [67].

Compared to DVB-T, symbol-level synchronization in WLAN networks is more challenging, because DVB-T has much longer guard intervals – depending on the mode, it can be 7-224µs. The guard interval minus free space propagation delay of the transmitted signal presents the margins for symbol-level synchronization errors between multiple transmitters. Therefore, DVB-T transmitters can easily use GPS clocks for synchronization across their remote locations, because the GPS clock precision is around 1µs.

However, the guard interval in 802.11a/g with 20MHz channel is much shorter – only 800ns. In 802.11n, the guard interval can be configured to half of this value – 400ns. With 10MHz and 5MHz channels, these values can be doubled/quadrupled, because the guard interval and symbol duration timings are doubled/quadrupled as well – see clause 17.3.3 in the 802.11 standard.

Of course, it is possible to design a proprietary PHY layer with longer symbols and guard intervals, but:

- longer symbols with more sub-carriers would incur higher latency and overheads at the MAC layer. E.g., with 54 Mbit/s rate in 802.11a/g, each OFDM symbol carries 216 bits=27 bytes of MAC layer data. But ACK and CTS packets carry only 14 bytes, RTS packets 20 bytes.
- Shorter symbols allow quicker frequency offset synchronization at the receiver. This is important for bursty two-way communication interleaved with small management packets typical for WLAN IP networks. The DVB can afford much longer symbols because it transmits continuous stream so the synchronization control loop at the receiver has much more time to settle down.
- Considering a same sampling frequency, shorter symbols mean less sub-carriers and this allows greater oscillator frequency drifts between transmitter and receiver.

These are the reasons why the OFDM symbols in 802.11 are only 3.2us long. Therefore, for symbol-level synchronization with 802.11 timings, a maximum synchronization error should be in the order of ±*N*.10ns – the bigger is the *N*, the smaller is the allowable propagation delay. It is clear that GPS clock can not be used for such synchronization. Fortunately, other synchronization methods exist already.

The problem of distributed symbol-level synchronization in wireless networks was already tackled in the SourceSync project [85], where a distributed synchronization protocol was developed for sender diversity and verified on the WiGLAN SDR [30]. The synchronization is implemented by tight cooperation between the PHY and MAC layers and the authors claim that "*two randomly chosen transmitters using SourceSync have a 95th percentile synchronization error of at most 20 ns*". This seems to be a real breakthrough which could have huge impact in the area of mesh networking and also in implementation of wireless synchronization protocols – e.g., the wireless implementation of the IEEE 1588 standard could work exactly this way and this could allow for many new applications and systems which are currently impossible, because the wireless IEEE 1588 is not even planned by any 802.11 chipset manufacturers; there are only few projects trying to implement this, e.g. [1], [25] or [51].

Another contribution to the multiple sender synchronization topic is described in [57], where GNU radio with USRP2 units were used for similar purposes, but a single shared 10 MHz clock was used for synchronization. This approach could be adopted at the collocation sites, where it is easy to deploy additional cables. The OFDM symbol rate is only 250 kHz, therefore there is no need for special-quality cables for connecting all the device to the source of synchronization timing signal – a single pair in a UTP would suffice.

The problem of these methods is that there is no way to use them within the existing standard 802.11 wireless networks, because they require non-standard modifications of the PHY and MAC layers.

But they represent a new research and development areas and also provide a good indication of the future perspectives in wireless network technologies.

9 CONCLUSION

This thesis targeted the most prevalent interferences experienced by 802.11 networks operators. The results are nowhere near to a single “save the world” solution. Instead, a practical measurements platform – the RF emulation testbed – was created, measurement methods and tools were developed and used for measuring the performance of several 802.11 devices. Also, several viable ideas for future research were proposed.

Up today, there have not been many publications dealing with co-location interferences between different 802.11 families and products, nor 802.11 radio parameters measurements published even for the 802.11a/b/g devices, which are on the market since 1999. Thus, the wireless network operators and architects have been living in an information vacuum.

I hope that this thesis and the money spent for implementing the testbed would help the mankind's progress at least a tiny little bit. I hope that I was not fighting yesterday's war, as the 802.11n standard final revision was published only two years ago and it is still finding its way into the current or new wireless networks.

The testbed has already attracted attention of several wireless networking enthusiasts. In June 2011, the first MSc. thesis project [117] has finished successfully using the RF testbed. In November 2011, another MSc. project is being finished using the test-bed, targeting performance problems of wireless networks at Masaryk university in Brno [127].

At TBU Zlin, several new BSc. and MSc. project topics have been advertised:

- Benchmarks of alternative 802.11 media access algorithms: in this BSc. project, the student should create a network with hidden nodes and test the performance of the commercial TDMA software media access layers – the Ubiquity Airmax, Mikrotik Nstreme polling and NV2.
- Emulation of multipath radio signals propagation in wireless networks: in this MSc. project, the USRP2 SW radio should be used for implementing a RF delay line equivalent to a long-haul wireless link. The delay line should have several variable taps emulating reflections along the link. Moreover, the 802.11n MIMO 2x2 RF emulation network, which is present in the testbed already, should be used for emulating the inter-stream crosstalks.
- Automatic testing of 802.11 receiver and transmitter chain: there are many areas in the 802.11 measurements which were not covered by this doctoral thesis. E.g.:

  - evaluation of algorithms for automatic bit rate selection,
  - the packet injector and packet sniffer for transmitter and receiver tests implemented in ANSI C (instead of Python) for achieving better throughput and cross-platform compatibility, as Python is not available on the resource-limited OpenWRT devices.
- Fully automatic device driver testing: Linux wireless kernel developers group elected this topic for Google SoC 2009 [90], but the project was not finished. The idea is to use a RF testbed for full automation of device driver tests before every public release of a driver.
- Wificolab 2.0: this MSc. project should integrate the receiver automation ANSI C code developed in this doctoral thesis into the Wificolab web application. Thus, most of the measurements could be done easily in the web GUI, without any command line tweaking.

10 REFERENCES

[1] Afshaneh Pakdaman. *IEEE 1588 over IEEE 802.11b for synchronization of Wireless Local Area Network Nodes and DeviceNet*. MSc. thesis, San Francisco State University, 2005. Available from WWW: *http://books.google.com/books/about/IEEE_1588_over_IEEE_802_11b_for_synchron.html?id=1sSiNwAACAAJ.*

[2] Amoss, J.W. et al. A Ferroelectric Microwave Switch. *IEEE Transactions on Microwave Theory and Techniques*. 13, 6 (Nov. 1965), pp. 789- 793, DOI: 10.1109/TMTT.1965.1126107.

[3] Anderson, E. *Disabling carrier sense conflicts with enabling TXOP*. Linux-driver for wireless cards based on Atheros chipsets: for driver developers. Available from WWW: *http://comments.gmane.org/gmane.linux.drivers.madwifi.devel/7106*.

[4] Andren, C. *Short PN Sequences for Direct Sequence Spread Spectrum Radios*. Harris Semiconductor, 1997. Available from WWW: *http://www.sss-mag.com/pdf/shortpn.pdf.*

[5] Arnold, M. et al. *LO Harmonic Effects on I/Q Balance and Sideband Suppression in Complex I/Q Modulators*. Application note SLWA059, Texas Instruments, 2010. Available from WWW: *http://focus.ti.com/lit/an/slwa059/slwa059.pdf.*

[6] Aspen Labs, LLC. *Ferroelectric RAM Microcontroller*. Electrical Engineering Design News. Available from WWW: *http://www.eeweb.com/news/ferroelectric-ram-microcontroller/.*

[7] Bahl, P. et al. *SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks*. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, ACM, New York, NY, USA, 2004, pp. 216–230, ISBN: 1-58113-868-7, DOI: 10.1145/1023720.1023742.

[8] Barrett, C. *Fractional/Integer-N PLL Basics*. Technical Brief SWRA029, Wireless Communication Business Unit, Texas Instruments, 1999. Available from WWW: *http://www.ti.com/litv/pdf/swra029*.

[9] Berg, J. *Capturing packets with bad FCS in monitor mode*. Linux wireless networking development. Available from WWW: *http://permalink.gmane.org/gmane.linux.kernel.wireless.general/54175*.

[10] Bernaschi, M. et al. OpenCAPWAP: An open source CAPWAP implementation for the management and configuration of WiFi hot-spots. *Comput. Netw.* 53, 2 (Feb. 2009), pp. 217–230, DOI: 10.1016/j.comnet.2008.09.016.

[11] Bialkowski, K. and Portmann, M. *Design of testbed for Wireless Mesh Networks*. In *Proceedings of Antennas and Propagation Society International Symposium (APSURSI)*, IEEE, Toronto, ON, July 2010, pp. 1-4, ISBN: 978-1-4244-4967-5, DOI: 10.1109/APS.2010.5562247.

[12] Biolek, D. et al. *Systémy, procesy a signály I : sbírka příkladů*. PC-DIR, ISBN: 9788021408326.

[13] Bo, A. et al. Patents on Synchronization Techniques in Wireless OFDM Systems. *Techniques*. 1, 1 (2008), pp. 14-21, DOI: 10.2174/1874476110801010014.

[14] Bo Han et al. *All Bits Are Not Equal - A Study of IEEE 802.11 Communication Bit Errors*. In *IEEE INFOCOM 2009*, IEEE, April 2009, pp. 1602-1610, ISBN: 978-1-4244-3512-8, DOI: 10.1109/INFCOM.2009.5062078.

[15] Bobek, P. *A remote control and data processing system for test&measure equipment*. MSc. thesis, FAI UTB in Zlin, 2010.

[16] Bores signal processing. *FFT window functions: limits on FFT analysis*. Available from WWW: *http://www.bores.com/courses/advanced/windows/files/windows.pdf.*

[17] Bula, J. *Measuring the RF parameters of WiFi devices*. Bc. thesis, FAI UTB ve Zlíně, 2007.

[18] CCNA Wireless Training » Basic Terminologies. *http://www.wirelesstut.com/ccna-wireless-knowledge/basic-terminologies*. Accessed: 10-30-2011.

[19] Chatelain, B. and Gagnon, F. *Peak-to-average power ratio and intersymbol interference reduction by Nyquist pulse optimization*. In *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, IEEE, September 2004, pp. 954- 958 Vol. 2, ISBN: 0-7803-8521-7, DOI: 10.1109/VETECF.2004.1400162.

[20] Chiueh, T.-D. and Tsai, P.-Y. *OFDM baseband receiver design for wireless communications*. John Wiley and Sons, ISBN: 9780470822340.

[21] Cisco 4400 Series Wireless LAN Controllers. *http://www.cisco.com/en/US/products/ps6366/index.html*. Accessed: 03-04-2011.

[22] Clancy, T.C. and Walker, B.D. *MeshTest: Laboratory-Based Wireless Testbed for Large Topologies*. In *Proceedings of 3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities*, Lake Buena Vista, FL, USA, 2007, pp. 1-6, DOI: 10.1109/TRIDENTCOM.2007.4444659.

[23] Clark, M. *IEEE 802.11a WLAN model*. MATLAB Central - File Exchange. Available from WWW: *http://www.mathworks.com/matlabcentral/fileexchange/3540*.

[24] Control Package for Atheros AR500x Devices. *http://www.anritsu.com/en-US/Downloads/Software/Drivers-Software-Downloads/DWL3298.aspx*. Accessed: 10-18-2011.

[25] Cooklev, T. et al. An Implementation of IEEE 1588 Over IEEE 802.11b for Synchronization of Wireless Local Area Network Nodes. *IEEE Transactions on Instrumentation and Measurement*. 56, 5 (Oct. 2007), pp. 1632-1639, DOI: 10.1109/TIM.2007.903640.

[26] Curtin, M. and O'Brien, P. *Phase-Locked Loops For High-Frequency Receivers And Transmitters*. Analog Devices, 1999. Available from WWW: *http://www.analog.com/library/analogDialogue/archives/33-05/phase_locked/index.html*.

[27] Desquiotz, R. *802.11 Packet Error Rate Testing*. Application note 1GP56, Rohde & Schwarz. Available from WWW: *http://www.rohde-schwarz.com/appnote/1GP56*.

[28] Digital Phase Shifter 2-18 GHz. *http://www.gtmicrowave.com/phase_shifters_feature.php*. Accessed: 03-07-2011.

[29] Dulík, T. *Migration of VMs between Xen and KVM - back and forth - the simple way*. Available from WWW: *http://wiki.debian.org/HowToMigrateBackAndForthBetweenXenAndKvm*.

[30] Edalat, F. *Real-time sub-carrier Adaptive Modulation and Coding in wideband Orthogonal Frequency Division Multiplexing wireless systems*. Doctoral thesis, MIT, 2008. Available from WWW: *http://dspace.mit.edu/handle/1721.1/43031*.

[31] Fotopoulou, E. et al. *Analysis and design of a WLAN OFDM transmitter with digital filters*. In *Proceedings of the 3rd international conference on Mobile multimedia communications*, ICST, Nafpaktos, Greece, 2007, pp. 60:1-60:5, ISBN: 978-963-06-2670-5, DOI: 10.4108/ICST.MOBIMEDIA2007.1814.

[32] Free-space path loss. *http://en.wikipedia.org/wiki/Free-space_path_loss*. Accessed: 03-03-2011.

[33] Fryšták, V. *System for localization of users in a wireless network*. MSc. thesis, FAI UTB in Zlin, 2007.

[34] Fujitsu Semiconductor Pacific Asia Ltd. *Ferroelectric RAM Overview*. Available from WWW: *http://www.fujitsu.com/cn/fsp/services/memory/fram/*.

[35] Fuxjäger, P. et al. IEEE 802.11p Transmission Using GNURadio. *Proceedings of the 6th Karlsruhe Workshop on Software Radios (WSR10)*. (Mar. 2010).

[36] Gerakoulis, D. *Interference suppressing OFDM system for wireless communications - Patent 7630290*. US Patent 7630290, AT&T Corp., 2006. Available from WWW: *http://www.freepatentsonline.com/7630290.html*.

[37] González-Bayón, J. et al. A multistandard frequency offset synchronization scheme for 802.11n, 802.16d, LTE, and DVB-T/H systems. *J. Comp. Sys., Netw., and Comm.* 2010, (Jan. 2010), pp. 5:1–5:9, DOI: 10.1155/2010/628657.

[38] Gummadi, R. et al. *Understanding and mitigating the impact of RF interference on 802.11 networks*. In *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, ACM, New York, NY, USA, 2007, pp. 385–396, ISBN: 978-1-59593-713-1, DOI: 10.1145/1282380.1282424.

[39] Haitao Wu et al. SoftMAC: Layer 2.5 Collaborative MAC for Multimedia Support in Multihop Wireless Networks. *IEEE Transactions on Mobile Computing*. 6, 1 (Jan. 2007), pp. 12-25, DOI: 10.1109/TMC.2007.250668.

[40] Haleš, B. *A wireless network implementation*. Bc. thesis, FAI UTB ve Zlíně, 2005.

[41] Harris, F. and Dick, C. OFDM Modulation Using Periodic Square-Root Nyquist Time Domain Kernels to Obtain Reduced Peak-to-Average Power Ratio. *Proceeding of the SDR 08 Technical Conference and Product Exposition*. 2008, (2008).

[42] Harris, F.J. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*. 66, 1 (1978), pp. 51-83, DOI: 10.1109/PROC.1978.10837.

[43] Harris, F.J. and Dick, C. OFDM modulation using a shaping filter. U.S. Patent #United States Patent 8090037. March 1, 2012.

[44] Heinzel. Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows . *Max Plank Institute*. (2002), pp. 1-84.

[45] Howitt, I. et al. *Empirical study for IEEE 802.11 and Bluetooth interoperability*. In *Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd*, IEEE, 2001, pp. 1109-1113 vol.2, ISBN: 0-7803-6728-6, DOI: 10.1109/VETECS.2001.944550.

[46] IEEE. *IEEE standard 802.11n™-2009*. Institute of Electrical and Electronics Engineers, ISBN: 978-0-7381-6046-7.

[47] IEEE. *IEEE standard 802.11™-2007*. Institute of Electrical and Electronics Engineers, ISBN: 9780738156552.

[48] Integrated Cosite Equipment - ICE PSEL1003 Bandpass filter/preselector 20-3000 MHz. *http://www.polezero.com/PSEL1003*. Accessed: 02-07-2012.

[49] Intersil Corporation. *HFA3861: Direct Sequence Spread Spectrum Baseband Processor*. Datasheet, 1999. Available from WWW: *http://jbnote.free.fr/prism54usb/data/documentation/prism2/datasheet_hfa3861.pdf*.

[50] Introduction to CDMA. *http://www.bee.net/mhendry/vrml/library/cdma/Chapter1.htm#_VPID_20*. Accessed: 03-03-2011.

[51] J. Kevin Rhee et al. *Timestamp jitter consideration for 802.11n*. IEEE 802.1 (avb) Interim report, Information and Communications Univ., 2008. Available from WWW: *http://ieee802.org/1/files/public/docs2008/avb-rhee-802-11n-timestamp-0908.pdf*.

[52] Jacobsen, E. *Pulse Shaping in Single-Carrier Communication Systems*. 2008. Available from WWW: *http://www.dsprelated.com/showarticle/60.php*.

[53] Jain, M. et al. *Practical, real-time, full duplex wireless*. In *MobiCom '11: Proceedings of the 17th annual international conference on Mobile computing and networking*, ACM, New York, NY, USA, 2011, pp. 301–312, ISBN: 978-1-4503-0492-4, DOI: 10.1145/2030613.2030647.

[54] Jan-Jaap van de Beek et al. Orthogonal frequency-division multiplexing (OFDM). *Review of radio science, 1996-1999*. Oxford University Press, ISBN: 9780198565727.

[55] Jaroš, J. *Building WiFi network with internet access in a remote rural area*. Bc. thesis, FAI UTB ve Zlíně, 2005.

[56] Jeongho Park et al. *Effect of partial band jamming on OFDM-based WLAN in 802.11g*. In

IEEE, pp. IV-560-3, ISBN: 0-7803-7663-3, DOI: 10.1109/ICASSP.2003.1202704.

[57] Jin Zhang et al. *Implementation and Evaluation of Cooperative Communication Schemes in Software-Defined Radio Testbed*. In *2010 Proceedings IEEE INFOCOM*, IEEE, March 2010, pp. 1-9, ISBN: 978-1-4244-5836-3, DOI: 10.1109/INFCOM.2010.5461915.

[58] Jonathan Angel. *Full-duplex transmission could double speed of wireless networks*. News - Linux for Devices. Available from WWW: *http://www.linuxfordevices.com/c/a/News/Stanford-antenna-cancellation-technology/*.

[59] Jones IV, V.K. et al. *OFDM channel estimation in the presence of interference*. US Patent 6487253, Cisco Technology, Inc., 1999. Available from WWW: *http://www.freepatentsonline.com/6487253.html*.

[60] KVM Virtualization Guide Chapter 13.: PCI passthrough. *http://docs.fedoraproject.org/en-US/Fedora/13/html/Virtualization_Guide/chap-Virtualization-PCI_passthrough.html*. Accessed: 04-04-2011.

[61] Kamperman, F. *Code Sequences for Direct Sequence CDMA*. Available from WWW: *http://www.wirelesscommunication.nl/reference/chaptr05/cdma/codes/codes.htm*.

[62] Klimov, V.V. et al. Ferroelectric variable capacitors. *Ferroelectrics*. 7, 1 (1974), pp. 337-339, DOI: 10.1080/00150197408238039.

[63] Kropff, M. et al. *A Survey on RealWorld and Emulation Testbeds for Mobile Ad hoc Networks*. In *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006.*, Barcelona, Spain, 2006, pp. 448-453, DOI: 10.1109/TRIDNT.2006.1649182.

[64] Kureck, H. and Vogel, C. *Interpolation Filters for the GNURadio+USRP2 Platform*. Project Report for the Course 442.087 Seminar/Projekt Signal Processing, Signal Processing and Speech Communication Laboratory (SPSC Lab) of Graz University of Technology, 2011. Available from WWW: *http://www.spsc.tugraz.at/student_projects/interpolation-filters-gnuradiousrp2-platform*.

[65] Laks, E. et al. The Technical Challenges of Transitioning Intel® PRO/Wireless Solutions to a Half-Mini Card. *Intel Technology Journal*. 2008, 12 (Oct. 2008), DOI: DOI: 10.1535/itj.1203.05.

[66] Liang, C.-J.M. et al. *Surviving wi-fi interference in low power ZigBee networks*. In *SenSys 2010: Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ACM, New York, NY, USA, 2010, pp. 309–322, ISBN: 978-1-4503-0344-6, DOI: 10.1145/1869983.1870014.

[67] Ligeti, A. *Single frequency network planning*. dissertation thesis, Royal Institute of Technology, 1999. Available from WWW: *http://kth.diva-portal.org/smash/record.jsf?pid=diva2:8583*.

[68] Liu, X. *A Practical Interference-Aware Power Management Protocol for Dense Wireless Networks*. Carnegie Mellon University, 2008. Available from WWW: *http://www.cs.cmu.edu/~xil/pwrmgmt.pdf*.

[69] Malinen, J. *Wlantest tool*. Hostap Git repository. Available from WWW: *http://w1.fi/gitweb/gitweb.cgi?p=hostap.git;a=tree;f=wlantest*.

[70] Maršálek, R. *Multicarrier modulations and PAPR reduction*. Habilitation thesis, 2008. Available from WWW: *http://www.urel.feec.vutbr.cz/web_documents/teze/marsalek_hab.pdf*.

[71] Matúšů, J. *A system for monitoring large area networks*. MSc. thesis, FAI UTB in Zlin, 2008.

[72] Mishra, A. et al. Distributed channel management in uncoordinated wireless environments. *IN ACM MOBICOM*. 2006, (2006), p. 170--181.

[73] Miu, A. et al. Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks. *IN MOBICOM*. (2005), p. 16--30.

[74] Modus RIA 24. *http://www.eliatel.cz/Modus_RIA.php*. Accessed: 11-02-2011.

[75] Motorola Canopy Cluster Management Module 3. *http://www.motorola.com/Business/US-EN/Business+Product+and+Services/Accessories/Wireless+Broadband+Accessories/Point+to+Multi-point+Accessories/Cluster+Management+Module3_US-EN*. Accessed: 02-28-2011.

[76] Mudroch, J. *Bandpass cavity filters for 5-6GHz band*. Available from WWW: *http://www.mudrochlabs.sk/cz_WiFi_Filtr_5GHz.htm*.

[77] National Instruments. *What is I/Q Data?* 2006. Available from WWW: *http://zone.ni.com/devzone/cda/tut/p/id/4805*.

[78] Nee, R. van and Prasad, R. *OFDM for wireless multimedia communications*. Artech House, ISBN: 0890065306 9780890065303.

[79] Neufeld, M. et al. *SoftMAC - Flexible Wireless Research Platform*. In *Fourth Workshop on Hot Topics in Networks (HotNets-IV)*, November 2005.

[80] Nv2 manual. *http://wiki.mikrotik.com/wiki/Manual:Nv2*. Accessed: 11-07-2011.

[81] Open Source CAPWAP. *http://sourceforge.net/projects/capwap/*. Accessed: 11-24-2011.

[82] Prasad, R. *OFDM for Wireless Communications Systems*. Artech House Publishers, ISBN: 1580537960.

[83] RF MEMS Tunable Filters from MEMtronics. *http://www.memtronics.com/page.aspx?page_id=12*. Accessed: 02-07-2012.

[84] RF Signal Bandpass Filters: 4, 8-Pole up to 5.8 GHz. *http://www.l-com.com/category.aspx?id=3016*. Accessed: 02-06-2012.

[85] Rahul, H. et al. SourceSync: a distributed wireless architecture for exploiting sender diversity. *SIGCOMM Comput. Commun. Rev.* 41, 4 (Aug. 2010), pp. 171–182, DOI: 10.1145/2043164.1851204.

[86] Raychaudhuri, D. et al. *Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols*. In *IEEE Wireless Communications and Networking Conference, 2005*, New Orleans, LA, USA, 2005, pp. 1664-1669, DOI: 10.1109/WCNC.2005.1424763.

[87] Reinke, J.R. *CMOS-MEMS Variable Capacitors for Reconfigurable RF Circuits*. PhD thesis, Carnegie Mellon University, 2011.

[88] Rocket GPS and Airsync FAQ - Ubiquiti Networks Forum. *http://www.ubnt.com/forum/showthread.php?t=25576*. Accessed: 02-28-2011.

[89] Rocket M GPS. *http://www.ubnt.com/rocketmgps*. Accessed: 02-28-2011.

[90] Rodriguez, L.R. et al. *Automation of testing of Linux wireless device drivers code*. Linux Wireless group. Available from WWW: *http://linuxwireless.org/en/developers/GSoC/2009/Automation_of_testing*.

[91] Rodriguez, L.R. et al. *Flags possible in monitor mode*. iw command manual at Linux Wireless group web. Available from WWW: *http://linuxwireless.org/en/users/Documentation/iw#Monitor_flags_possible*.

[92] Ruckus Wireless Controllers. *http://www.ruckuswireless.com/products/controllers*. Accessed: 03-04-2011.

[93] SYMMETRICOM, INC. *Timing and Synchronization in WiMAX Networks*. Application brief, 2006.

[94] Saidi, M. and FanVice, R. *Dual-band issue: super-heterodyne v.s. zero IF*. Resonext Communications Inc., 2002. Available from WWW: *http://www.eetimes.com/electronics-news/4164144/Dual-band-issue-super-heterodyne-v-s-zero-IF*.

[95] Sallese, J.-M. and Fazan, P. Switch and rf ferroelectric MEMS: a new concept. *Sensors and Actuators A: Physical*. 109, 3 (Jan. 2004), pp. 186-194, DOI: 10.1016/j.sna.2003.10.037.

[96] Sanghani, S. et al. *EWANT: the emulated wireless ad hoc network testbed*. In *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, New Orleans, LA, USA, 2003, pp. 1844-1849, DOI: 10.1109/WCNC.2003.1200667.

[97] Scalia, L. et al. *Side Effects of Ambient Noise Immunity Techniques on Outdoor IEEE 802.11 Deployments*. In *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, New Orleans, LA, USA, 2008, pp. 1-6, DOI: 10.1109/GLOCOM.2008.ECP.941.

[98] Scapy. *http://www.secdev.org/projects/scapy/demo.html*. Accessed: 11-06-2011.

[99] Scapybase: a 802.11 access point based on scapy. *http://jahrome.free.fr/index.php/scapybase-wifi-access-point-scapy*. Accessed: 11-06-2011.

[100] Serbinenko, V. *Bug #32391: Grub will not install in Blocklist*. GNU GRUB - Bugs. Available from WWW: *https://savannah.gnu.org/bugs/index.php?32391#comment1*.

[101] Sharma, A. et al. *MadMAC: Building a Reconfigurable Radio Testbed Using Commodity 802.11 Hardware*. In *First IEEE Workshop on Networking Technologies for Software Defined Radio (SDR) Networks*, 2006.

[102] Shin, S.Y. et al. Mutual interference analysis of IEEE 802.15.4 and IEEE 802.11b. *Computer Networks*. 51, 12 (Aug. 2007), pp. 3338-3353, DOI: 10.1016/j.comnet.2007.01.034.

[103] Shooshtary, S. *Development of a MATLAB Simulation Environment for Vehicle-to-Vehicle and Infrastructure Communication Based on IEEE 802.11p*. University of Gävle, Department of Technology and Built Environment.

[104] Siemens HiPath Wireless Controllers. *http://wiki.siemens-enterprise.com/index.php/HiPath_Wireless#Controllers_models_and_variants*. Accessed: 03-04-2011.

[105] Slimane, S.B. *Peak-to-average power ratio reduction of OFDM signals using broadband pulse shaping*. In *Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th*, IEEE, 2002, pp. 889- 893 vol.2, ISBN: 0-7803-7467-3, DOI: 10.1109/VETECF.2002.1040728.

[106] So, H.W. et al. Packet Loss Behavior in a Wireless Broadcast Sensor Network.

[107] Sources of Error in IQ Based RF Signal Generation. *http://zone.ni.com/devzone/cda/tut/p/id/5657*. Accessed: 02-27-2011.

[108] Sporek, J. *Server, router and WiFi AP providing access to university network for students and guests*. Bc. thesis, FAI UTB ve Zlíně, 2005.

[109] Strong, P.N. et al. *Interference optimized OFDM*. US Patent 7813701, Piping Hot Networks Limited, 2006. Available from WWW: *http://www.freepatentsonline.com/7813701.html*.

[110] Sugiyama, T. et al. *OFDM signal communication system, OFDM signal transmitting device and OFDM signal receiving device*. US Patent 7242720, Nippon Telegraph and Telephone Corporation, 2002. Available from WWW: *http://www.patentstorm.us/patents/7242720.html*.

[111] Sviták, J. *Modeling, simulation and throughput analysis for the 802.11 protocols family*. MSc. thesis, FAI UTB in Zlin, 2007.

[112] Tinnirello, I. et al. On the side-effects of proprietary solutions for fading and interference mitigation in IEEE 802.11b/g outdoor links. *Computer Networks*. 53, 2 (Feb. 2009), pp. 141-152, DOI: 10.1016/j.comnet.2008.10.006.

[113] Trigonometric Addition Formulas. *http://mathworld.wolfram.com/TrigonometricAdditionFormulas.html*. Accessed: 02-26-2011.

[114] Tunable Filters, RF Tunable Filter. *http://www.agilerf.com/products/filters.html*. Accessed: 02-07-2012.

[115] Tzu, S. *The Art of War*. Oxford University Press, ISBN: 0195014766.

[116] Ubiquiti Networks, Inc. *Bullet M5 HP datasheet*. Available from WWW: *http://www.ubnt.com/downloads/bm5hp_datasheet.pdf*.

[117] Vágner, A. *Real performance of devices operating on 802.11n*. MSc. thesis, Faculty

of Electrical Engineering and Communication, Brno University of Technology, 2011. Available from WWW: *https://www.vutbr.cz/en/studies/final-thesis?zp_id=40262*.

[118] Wad, P. *I/Q Signal Mismatch Theory*. Available from WWW: *http://paul.wad.homepage.dk/iq_mismatch_theory/doc.html*.

[119] Weiss, M. *WLAN Tests According to Standard 802.11a/b/g*. Application note 1MA69, Rohde & Schwarz, 2004.

[120] Willig, A. et al. Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. *IEEE Transactions on Industrial Electronics*. 49, 6 (Dec. 2002), pp. 1265- 1282, DOI: 10.1109/TIE.2002.804974.

[121] Woody Allen quotes. *http://thinkexist.com/quotation/if_you_want_to_make_god_laugh-tell_him_about_your/226952.html*. Accessed: 11-10-2011.

[122] Xen PCI passthrough. *http://wiki.xensource.com/xenwiki/XenPCIpassthrough*. Accessed: 10-29-2011.

[123] Yeonbae Chung et al. A 3.3-V, 4-Mb nonvolatile ferroelectric RAM with selectively driven double-pulsed plate read/write-back scheme. *IEEE Journal of Solid-State Circuits*. 35, 5 (May. 2000), pp. 697-704, DOI: 10.1109/4.841494.

[124] Yiddish proverbs. *http://en.wikiquote.org/wiki/Yiddish_proverbs*. Accessed: 11-10-2011.

[125] Yuliang Zheng et al. Compact Substrate Integrated Waveguide Tunable Filter Based on Ferroelectric Ceramics. *IEEE Microwave and Wireless Components Letters*. 21, 9 (Sep. 2011), pp. 477-479, DOI: 10.1109/LMWC.2011.2162615.

[126] Zyren, J. and Petrick, A. *IEEE 802.11 Tutorial*. Available from WWW: *http://www.terabeam.com/solutions/whitepapers/tutorial_80211.php*.

[127] Šlinz, P. *Issues of highly stressed wireless networks based on 802.11b/g standards*. Master's thesis, Masarykova univerzita v Brně, 2012. Available from WWW: *http://is.muni.cz/th/208329/fi_m/*.

[128] *Atheros radio test reference guide*. Available from WWW: *http://wenku.baidu.com/view/8ae40e78168884868762d6fa.html*.

11 APPENDIX

11.1 Errata

This text surely contains many errors. Finding them and pointing them out is very rewarding for keeping the information entropy in the area of truth. However, correcting these "bugs" directly in the text is not possible as the printed and PDF form of these thesis was submitted to the university library and it is impossible to make any corrections there.

Therefore, the errata to this text are available at the following URL:

http://zamestnanci.fai.utb.cz/~dulik/dissertation/errata.pdf

11.2 Software used for automatic measurements

The complete projects (ANSI C source code, makefiles, Eclipse CDT projects) are available in GIT repository available at https://sourceforge.net/p/wificolab/code/

The “tools” subdirectory contains the following directories:

- broadcaster – a tool for sending broadcast packets with given rate,
- attnctrl – a tool for controlling the Aeroflex-Wienschel attenuators using RS232 or SSH,
- orchestrator – a tool for more complex measurements, where SSH, serial line and VXI can be used for orchestrating the measurement automation,
- bwtest – a web based application for testing TCP/IP performance of a network with device(s) containing a web interface or any other TCP/IP port open for communication. Typically, most 802.11 APs and STAs have a web interface for configuration. Such a web interface can be used as TCP/IP server for the bwtest application.

  The bwtest connects to the TCP servers on any open port using client socket and starts receiving and/or transmitting arbitrary data. It can open many parallel connections to different servers and thus evaluate the whole network behaviour.

All these tools are published with the GPLv3 license.

11.3 Migration of VMs between Xen and KVM

This method was already published by me on Debain wiki [29], but text below is rewritten to be more detailed and more comprehensible.

There are many tutorials how to migrate virtual machines from Xen to KVM, but there is not a single mention about the reverse way.

A method for doing this is described in this chapter. It is based on the Xen ability to customize virtual machines (VM) in a way that they can be started in both Xen and KVM with no intermediate steps, so after the "one-time" modifications described below, no other VM conversions or modifications are necessary for the migrations.

11.3.1 Prerequisites

- The Xen is Debian 5 (Lenny), KVM is Debian 6 (Squeeze) or Ubuntu 10.04.
- The Xen VM guests were created by the xen-create-image tool. We have many of them, some with Debian 4, some with Debian 5, some with Debian 6.
- The DRBD devices, which replicate the VM disks from the Xen to the KVM physical host machines, were already set up and started.
  Note: in Debian 5, it is highly recommended to use the DRBD version 8.3.7 from the Debian backports packages repository. With the 8.0.14-2+lenny1 DRBD version, it might happen that in some cases DRBD resources on Lenny will not be able to resynchronize with Squeeze DRBD resources set to primary mode.

11.3.2 /etc/inittab in all Xen VM guests

For its virtual machines, Xen uses a special Hypervisor Virtual Console (HVC), which is not available in KVM. In Xen virtual machines, the HVC is configured in the /etc/inittab file by a single line created by the xen-create-image tool which directs console to the hvc0 device. This line must be commented out:

```
#1:2345:respawn:/sbin/getty 38400 hvc0
```

For achieving KVM compatibility, the standard tty1 console definition must be used:

```
1:2345:respawn:/sbin/getty 38400 tty1
```

However, now the console would not work in Xen. For enabling Xen console, there are two methods:

1. placing another line into the /etc/inittab file:
   ```
   co:23:respawn:/sbin/getty -L hvc0 9600 vt102
   ```
2. or redirecting the hvc0 console to the tty1 by placing additional line in the VM guest definition file (e.g. /etc/Xen/auto/logger.cfg)
   ```
   extra = "console=hvc0 Xencons=tty"
   ```

The second method is preferable, because then the virtual machine configuration is completely independent on the specific hypervisor.

11.3.3 Modifying VM guest definition files

It is recommended to rename the VM disks to sda/sdb/sdc... in the guest definition files (/etc/xen/auto/*.cfg). Why?

- By default, xen-create-image tool generates partition-less filesystem images with disk names containing partition numbers in the form (hd|sd|xvd)[a-z][0-9], e.g. hda1, sda1 or xvda1.
- KVM presumes that its images are partitioned ("whole-disk images") and although it can work with partition-less filesystem images, its guest definition files can only handle the following disk names **without partition numbers**:
  - vd[a-z] when using the virtio type
  - sda[a-z] when using the "IDE" type

Therefore the only possible way forward is to use the sda/sdb/sdc... disk names in Xen. Although the images are partition-less, Xen happily works with sda/sdb/sdc disk names so we can modify the guest definition files to like this:

```
#
# Configuration file for the Xen instance wificolab, created
# by Xen-tools 3.9 on Wed May 19 02:56:03 2010.
#
#
#  Kernel + memory size
#
kernel      = '/boot/vmlinuz-2.6.26-2-xen-amd64'
ramdisk     = '/boot/initrd.img-2.6.26-2-xen-amd64'
memory      = '2048'

#
#  Disk device(s).
#
root        = '/dev/sda ro'
disk        = [
                  'phy:/dev/vg_main/wificolab-swap,sdc,w',
                  'phy:/dev/drbd5,sda,w',
                  'phy:/dev/drbd6,sdb,w',
```

```
              ]
cpus="0-7"
vcpus=8
#for proper clock and standard /etc/inittab (compatible with KVM):
extra="clocksource=jiffies console=hvc0 Xencons=tty"

#
#  Hostname
#
name        = 'wificolab'
#
#  Networking
#
dhcp        = 'dhcp'
vif         = [ 'mac=00:16:3E:37:D0:70' ]

#
#  Behaviour
#
on_poweroff = 'destroy'
on_reboot   = 'restart'
on_crash    = 'restart'
```

11.3.4 Changing /etc/fstab in each guest VM

Before restarting (re-creating) the guest VM, it is necessary to log into it and modify its /etc/fstab mount points to comply with the new disk names!

11.3.5 Installing grub and kernel in each VM guest

While still running the VM guests in Xen, we must install kernel and grub in each VM guest.

Note:

- As the disk images are partition-less, the grub must work in so called “blocklist mode”, which might be affected by bug #32931 [100]. Anyway, for Lenny and Squeeze virtual machines, their default grub2 versions worked without problems.
- Debian 4 (Etch) should probably use the grub2 1.96+20080626-1~bpo40+2 version from backports repository. However, I was unable to make this work. Instead, I had to create a full-disk partitioned image and copy the the filesystem image content into its first partition.

After installing grub2:

```
apt-get install grub2
```

We can install the proper kernel file - e.g., for Debian Squeeze VM guests:

```
apt-get install linux-image-2.6.32-5-amd64
```

The grub2 boot loader should be happy with the /dev/sda root disk, just make sure that the /boot/grub/device.map contains this line:

```
 (hd0)   /dev/sda
```

and then run

```
 update-grub
```

which should output something like

```
Updating /boot/grub/grub.cfg ...
Found linux image: /boot/vmlinuz-2.6.26-2-amd64
Found initrd image: /boot/initrd.img-2.6.26-2-amd64
done
```

and

```
 grub-install –force /dev/sda
```

11.3.6 Re-create and test the guest VM in Xen

Now, it is necessary to poweroff the virtual machine in Xen, so either type:

```
 poweroff
```

in the guest or

```
 xm shutdown wificolab
```

in host, and then re-create it:

```
 xm create -c /etc/Xen/auto/wificolab.cfg
```

11.3.7 Create new guest defintion file for KVM

On the KVM machine, thanks to DRBD, we already have the guest image ready, so the last step is to create the proper guest definitions, e.g. /root/wificolab.xml :

```
 <domain type='kvm'>
  <name>wificolab</name>
  <!-- generate the uuid by #uuidgen command and insert it here: -->
  <uuid>30ea508e-7ed4-d590-3f63-0ea9d22e2545</uuid>
  <memory>2097152</memory>
  <currentMemory>2097152</currentMemory>
  <vcpu>2</vcpu>
  <os>
    <type arch='x86_64' machine='pc-0.12'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/bin/kvm</emulator>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw'/>
      <source dev='/dev/drbd5'/>
      <target dev='sda' bus='ide'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw'/>
      <source dev='/dev/drbd6'/>
      <target dev='sdb' bus='ide'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw'/>
      <source dev='/dev/vg_main/wificolab-swap'/>
      <target dev='sdc' bus='ide'/>
    </disk>
    <interface type='bridge'>
        <mac address='00:16:3E:37:D0:70'/>
      <source bridge='br0'/>
      <model type='virtio'/>
    </interface>
    <console type='pty'>
      <target port='0'/>
    </console>
    <console type='pty'>
      <target port='0'/>
    </console>
    <input type='tablet' bus='usb'/>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='-1' autoport='yes'/>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
    </video>
  </devices>
</domain>
```

Now test the definition syntax by:

```
virsh define /root/wificolab.xml
```

11.3.8 Testing the VM guests in KVM

Before starting the guest in KVM, it is a good idea to make "dry run" test on image snapshot:

```
lvcreate /dev/vg_main/wificolab-disk --snapshot -n logger-system-snap -L 2G
lvcreate /dev/vg_main/wificolab-var --snapshot -n logger-var-snap -L 2G
```

(Note: /dev/vg_main/wificolab-* are the LVM partitions underlying the /dev/drbd5 and drbd6 devices).

Now, after modifying the KVM guest definition in order to use the snapshot instead of DRBD devices, the VM can be started by

```
virsh create /root/wificolab.xml
```

11.3.9 Switch KVM's DRBD to primary and run

If everything works, then the KVM definition can be modified back to /dev/drbd* devices.

Then, on Xen host, we can shutdown the VM:

```
 xm shutdown wificolab
```

On Xen host, wificolab DRBD devices must be switched to "secondary":

```
 drbdadm secondary wificolab-system wificolab-var
```

And on KVM host to "primary":

```
 drbdadm primary wificolab-system wificolab-var
```

The last step is starting the VM on KVM:

```
 virsh create /root/logger.xml
```

When migrating the VM from KVM to Xen, this sequence must be done in reverse order.

The migration takes only few seconds – mostly the time needed for rebooting the image.

11.4 PCI passthrough of USB serial interface to a XEN virtual machine

First of all, it is necessary to find the PCI ID of the PCI device we want to work with – for example, /dev/ttyUSB0 connected to a PCI card USB controller:

```
# ls /sys/class/tty/ttyUSB0/ -l
celkem 0
-r--r--r-- 1 root root 4096 24. říj 09.53 dev
lrwxrwxrwx 1 root root    0 21. říj 09.39 device ->
../../../devices/pci0000:00/0000:00:1e.0/0000:08:08.1/usb11/11-1/11-1:1.0/ttyUSB0
drwxr-xr-x 2 root root    0 24. říj 09.53 power
lrwxrwxrwx 1 root root    0 24. říj 09.53 subsystem -> ../../tty
-rw-r--r-- 1 root root 4096 24. říj 09.53 uevent
```

We also need to find the driver controlling the PCI device:

```
# find /sys/bus/pci/drivers -name "0000:08:08.1"
/sys/bus/pci/drivers/ohci_hcd/0000:08:08.1
```

Now, a temporary PCI passthrough can be started – it will work until the next reboot. First of all, let's unbind the PCI device from the driver:

```
echo -n "0000:08:08.1" > /sys/bus/pci/drivers/ohci_hcd/unbind
```

Then we need to bind that device to the “pciback” XEN driver:

```
echo -n "0000:08:08.1" > /sys/bus/pci/drivers/pciback/new_slot
echo -n "0000:08:08.1" > /sys/bus/pci/drivers/pciback/bind
```

Now, the following lines must be added to the virtual machine configuration file (/etc/xen/auto/*.cfg):

```
pci = [ '08:08.1' ]
extra="clocksource=jiffies console=hvc0 xencons=tty iommu=soft"
```

After this, the virtual machine must be "powered off" and re-created.

The XEN wiki recommends to pass all PCI IDs present on the card, in our case the PCI USB controller has also 08:08.0 and 08:08.2 IDs attached to the other USB ports, but as the other ports are not used, there was no need. If needed, the whole PCI device can be passed:

```
pci = [ '08:08.0', '08:08.1', '08:08.2' ]
```

For permanent passthrough, the device unbinding must be implemented during the physical server boot by adding the following kernel parameters to the /boot/grub/menu.lst file.

For kernel versions < 2.6.31:

```
pciback.permissive pciback.hide=(08:08.0)(08:08.1)(08:08.2) reassigndev
```

For kernels >= 2.6.31:

```
xen-pciback.permissive xen-pciback.hide=(08:08.0)(08:08.1)(08:08.2)
pci=resource_alignment=08:08.0;08:08.1;08:08.2
```

11.5 Author's Curriculum vitae

Education:

- 1993-1998 – MSc. in technical cybernetics, Brno University of Technology, Faculty of Electrical Engineering, Department of computer science.
    - BSc. thesis: “Data transmission in narrow band channels”
    - MSc. project: “Digital signal processing device using FPGA”

Professional career:

- 1996 - Camea s.r.o., Brno - developing HW and SW pro telecommunications and DSP – camera image processing.
- 1998-2001 - UNIS a.s., Brno - developing HW and SW for Fujitsu Semiconductors
- 2001-2003 – civil service, Tomas Bata University in Zlin
- 2003-2009 – Tomas Bata University in Zlin, Faculty of applied informatics, Department of applied informatics, assistant
- 2010-today – Tomas Bata University in Zlin, Faculty of applied informatics, Department of informatics and artificial intelligence, assistant

Teaching:

- Mobile technologies, Java technologies, WWW technologies, Algorithms and data structures

Research activities:

- **Principal researcher:**
    - 2009-2011: Project CESNET 351/2009 “Laboratory with secure remote access for research and development of WLAN 802.11 applications and protocols”
    - 2010-2011: Project MŠMT 2C06008 “VLAM: virtual laboratory of microprocessor technology applications”
    - 2006-2008: iCamp (FP6 IST-2005-027168) http://www.icamp-project.org/
    - 2004-2005: ERASMUS „ERIC“ - http://www.eric.utb.cz
    - 2003-2005: Mobilife (FP6 IST-2004-511607), https://www.ist-mobilife.org
    - 2002-2004: Full Speed (FP5 IST-2001-32463), http://www.mobivas.cnl.di.uoa.gr
    - 2000-2002: MOBIVAS (FP5, IST-1999-10206), http://www.mobivas.cnl.di.uoa.gr
- **Member of project team:**
    - 1998: „Road traffic monitoring system“, grant GAČR 102/97/1012
    - 1997: Design of flexible architectures, grant GAČR č. 102/95/1334

11.6 Complete list of author's activities at TBU Zlin

11.6.1 Papers in journals with impact factor

1. Turek, L., Dulík, T.: Packet Scheduler for Access Points in 802.11 Wireless Networks, Brno University of Technology FEEC, Radioengineering, Brno, written 11/2010, waiting for review.

11.6.2 Papers in reviewed journals

1. Vojtěšek, J., Bližňák, M., Matušů, R., Dulík, T.: Virtualization as a Teaching Tool for IT-based Courses, WSEAS World Science and Engineering Academy and Science, WSEAS Transactions on Advances in Engineering Education, Atény, 2009, 265-274, ISSN 1790-1979

2. Bližňák, M., Dulík, T., Vašek, V.: Automated Production-Ready Source Code Design for Embedded Systems, Internationalsar, International Journal of Factory Automation, Robotics and Soft Computing, Palermo, 2009, 96-101, ISSN 1828-6984

3. Bližňák, M., Dulík, T., Vašek, V.: A PERSISTENT CROSS-PLATFORM CLASS OBJECTS CONTAINER FOR C++ AND WXWIDGETS, WSEAS World Science and Engineering Academy and Science, WSEAS Transactions on Computers, Atény, 2009, 778-787, ISSN 1109-2750

4. Bližňák, M., Dulík, T., Vašek, V.: WXSHAPEFRAMEWORK: AN EASY WAY FOR DIAGRAMS MANIPULATION IN C++ APPLICATIONS, WSEAS World Science and Engineering Academy and Science, WSEAS Transactions on Computers, Atény, 2010, 268-277, ISSN 1109-2750

11.6.3 Papers in conference proceedings

1. Bližňák, M., Dulík, T., Vašek, V., Janáčová, D.: WCONTROL: A Tool for Control Theory Laboratory Education, International Association for the Development of Advances in Technology (IADAT), International Conference on Education IADAT-e2004, Bilbao, 2004, 377-381, ISBN-ISSN 84-933971-0-5
2. Bližňák, M., Vojtěšek, J., Matušů, R., Dulík, T.: Virtualization as a Teaching Tool, WSEAS Press (IT), Proceedings of the 8th WSEAS International Conference on DISTANCE LEARNING and WEB ENGINEERING, Santander, Spain, 2008, 214-217, ISBN-ISSN 978-960-474-005-5
3. Malaník, D.M.., Dulík, T., Drbálek, Z., Červenka, M.: System for capturing, streaming and sharing video files, WSEAS Press (IT), Proceedings of the 8th WSEAS International Conference on DISTANCE LEARNING and WEB ENGINEERING, Venice, 2008, 86-91, ISBN-ISSN 978-960-474-005-5
4. Doležel, P., Dulík, T.: Data mining service for OAIster digital library, IADIS Press, PROCEEDINGS OF THE IADIS INTERNATIONAL CONFERENCE e-LEARNING 2008, Amsterdam, The Netherlands, 2008, 61-65, ISBN-ISSN 978-972-8924-58-4
5. Drbálek, Z., Dulík, T., Koblischke, R.: Developing components for distributed search engine ObjectSpot, WSEAS Press (IT), Proceedings of the 8th WSEAS International Conference on DISTANCE LEARNING and WEB ENGINEERING, Venice, 2008, 82-85, ISBN-ISSN 978-960-474-005-5
6. Bližňák, M., Dulík, T., Vašek, V.: A PERSISTENT CROSS-PLATFORM XML-BASED CLASS OBJECTS CONTAINER, WSEAS Press (Au), Proceedings of the 10th WSEAS International Conference on Automation & Information, Prague, 2009, 316-321, ISBN-ISSN 978-960-474-064-2
7. Bližňák, M., Dulík, T., Vašek, V.: A CROSS-PLATFORM SOFTWARE LIBRARY FOR DIAGRAMS CREATION AND MANIPULATION, WSEAS Press (GR), Proceedings of the 13th WSEAS International Conference on Computers, Rhodes, 2009, 362-367, ISBN-ISSN 978-960-474-099-4
8. Matušů, R., Vojtěšek, J., Dulík, T.: Technology-Enhanced Learning Tools in European Higher Education, WSEAS Press (IT), Proceedings of the 8th WSEAS International Conference on DISTANCE LEARNING and WEB ENGINEERING, Santander, Spain, 2008, 51-54, ISBN-ISSN 978-960-474-005-5

9. Dulík, T., Bližňák, M.: Security measures in virtual laboratory of microprocessor technology, DAAAM International Vienna, Proceedings of the 21st International DAAAM Symposium "Intelligent Manufacturing & Automation: Focus on Interdisciplinary Solutions", Vienna, 2010, 1203-1204, ISBN-ISSN 978-3-901509-73-5

11.6.4 Book chapters

1. Grodecka, K., Dulík, T.: How to Use Social Software in Higher Education, AGH - University of Science and Technology, Poland, How to Use Social Software in Higher Education, Krakow, Poland, 2008, 13, ISBN-ISMN 978-83-60958-28-5
2. Bližňák, M., Dulík, T., Vašek, V.: Automated Production-Ready Source Code Design for Embedded Systems, Internationalsar, Emerging Technologies, Robotics and Control Systems, Palermo, 2009, 64-69, ISBN-ISMN 978-88-901928-5-2

11.6.5 Software

1. Doležel, P., Dulík, T., Vojtěšek, J.: iCamp Extended calendar module for Moodle, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Zlín, 2009
2. Doležel, P., Dulík, T.: iCamp help center, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Zlín, 2009
3. Malaník, D., Dulík, T.: iCamp Federated File Store, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2009
4. Sláma, P., Dulík, T.: iCamp OpenID authentication module for Scuttle, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2009
5. Drbálek, Z., Dulík, T.: iCamp ObjectSpot search engine components, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2009
6. Doležel, P., Dulík, T.: iCamp ObjectSpot search engine SQI targets, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2009
7. Doležel, P., Dulík, T.: iCamp SIP server including web-based GUI, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Zlín, 2009
8. Dulík, T., Bobek, P.: WifiColab, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Zlín, 2010
9. Trbušek, J., Bližňák, M., Dulík, T.: MeasurementManager, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Zlín, 2010

11.6.6 Conferences organising

1. Dulík, T., Holec, M.: Openmobility 2010 - konference o otevřených mobilních technologiích, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2010
2. Dulík, T.: Seminář/workshop Mobilní a bezdrátové komunikace, FAI UTB ve Zlíně, 2009

11.6.7 Supervised diploma projects

1. BAROŠ, J.: Systém pro automatickou detekci plagiátorství v kurzech programování, Bc. project, 2005
2. BEDNÁŘ, J.: Implementace modulů systému Alfresco, MSc. project, in progress
3. BENOVIČ, M.: Automatické hodnocení úkolů v kurzech programování, Bc. project, in progress
4. BOBEK, P.: Systém pro zpracování dat z měřících přístrojů, MSc. project, 2010
5. BUDÍKOVÁ, V.: Virtuální laboratoř mikropočítačů se vzdáleným přístupem, MSc. project, in progress
6. BULA, J.: Měření rádiových parametrů WiFi karet, Bc. project, 2007
7. COURAL, R.: Generický unpacker pro antivirový systém ZAV, Bc. project, in progress
8. DANĚK, P.: Informační systém pro správu a evidenci uživatelů rozsáhlých sítí , MSc. project, 2008
9. DOBROVOLNÝ, T.: Implementace tenkých klientů v Javě, Bc. project, 2009
10. DOŠEK, R.: Bezpečný systém pro mikroplatby s RFID kartami, Bc. project, in progress
11. DRBÁLEK, Z.: Developing component for distributed search engine ObjectSpot, MSc. project, 2008
12. DRBÁLEK, Z.: Využití internetového streamingu pro přenos zvukových informací, Bc. project, 2006
13. FARNÝ, D.: Optimalizace výkonu webových a databázových serverů, MSc. project, 2008
14. FARNÝ, D.: Position tracker, Bc. project, 2006
15. FOJTŮ, J.: Hledání obrazových duplikátů pomocí metod morfologické analýzy a analýzy histogramu, Bc. project, 2005
16. FRYŠTÁK, V.: Systém pro zaměření pozice uživatele bezdrátové sítě, MSc. project, 2007

17. HALEŠ, B.: Projekt bezdrátové sítě, Bc. project, 2005
18. HUBÁČEK, S.: Bezpečnostní audit RFID systémů, MSc. project, in progress
19. CHOLASTA, J.: GUI aplikace pro správu Linuxových serverů/routerů, Bc. project, 2009
20. IGLO, T.: VoIP ústředna s protokolem SIP, MSc. project, 2009
21. JAROŠ, J.: Budování sítě s připojením do internetu v odlehlých lokalitách, Bc. project, 2005
22. JURA, P.: Moduly eLearning systému Moodle pro potřeby výuky na UTB ve Zlíně, MSc. project, 2006
23. KOBALÍČEK, P.: Multiplatformní (Windows / Linux) GUI knihovna napsaná v jazyce C++, Bc. project, 2007
24. KOČAR, J.: Systém pro správu autorských práv u souborů MP3, Bc. project, 2005
25. KOUŘIL, L.: Porovnání webových technologií ASP, ASP.NET a PHP, Bc. project, 2006
26. KOZEL, T.: Grafický editor schémat sítě pro informační systém , MSc. project, 2010
27. KUČERA, O.: Implementace time-management aplikace pro OS Android, Bc. project, in progress
28. KUCHTÍK, M.: Mobilní kancelář, Bc. project, 2005
29. LUDÍK, M.: Domácí VoIP ústředna s připojením do GSM sítí , Bc. project, 2009
30. MAJTÁN, P.: Web implementace strategické hry, Bc. project, 2009
31. MALANÍK, D.: Systém pro zachytávání videí a screencastů na přednáškách UTB, MSc. project, 2008
32. MALANÍK, D.: Ekonomický modulární systém s architekturou klient-server, Bc. project, 2006
33. MALIŇÁK, J.: GIS komponenta pro informační systém komunitní sítě , Bc. project, in progress
34. MARCANÍK, L.: Internetové a intranetové stránky Studentské unie , Bc. project, 2008
35. MARCANÍK, T.: Trojrozměrná správa přístupových práv pro webové systémy, Bc. project, 2010
36. MATÚŠŮ, J.: Monitorování stavu rozsáhlých sítí, MSc. project, 2008

37. MATÚŠŮ, J.: Informační systém pro správu komunitní sítě, Bc. project, 2006
38. MIHÁL, M.: Java frameworky a vývoj rozsáhlých web aplikací pro podnikovou sféru, MSc. project, in progress
39. MÜNSTER, P.: Testy výkonnosti virtualizérů, Bc. project, 2009
40. ORAVETZ, V.: Automatické hodnocení úkolů v kurzech programování, Bc. project, in progress
41. PASTORČÁK, P.: Vzorová řešení pro automatickou kontrolu správnosti úkolů v kurzech programování, Bc. project, 2006
42. PĚRKA, S.: Vytvoření studijních materiálů pro předmět technologie WWW, Bc. project, 2006
43. PLISKOVÁ, M.: Studijní materiály předmětu Technologie WWW, Bc. project, 2005
44. RODE, M.: Informační systém pro výrobní firmu, MSc. project, 2010
45. ROZEHNAL, M.: Informační systém pro správu rozsáhlých sítí, MSc. project, 2008
46. ROZEHNAL, M.: Systém pro dálkovou správu serverů Linux, implementovaný v jazyce PHP, Bc. project, 2006
47. RUSSEK, T.: Uzpůsobení redakčního systému dle legislativy platné pro použití ve státní správě, MSc. project, 2009
48. SAPÍK, O.: IPTV – kabelová televize po ethernetu, MSc. project, 2007
49. SEKANINA, A.: Multimediální centrum, založené na OS Linux, Bc. project, 2008
50. SMĚTALA, P.: Web application development and development trends, Bc. project, 2005
51. SPOREK, J.: Server, router a WiFi AP pro přístup studentů a hostů školy do sítě, Bc. project, 2005
52. SVITÁK, J.: Modelování, simulace a analýza propustnosti protokolů rodiny 802.11, MSc. project, 2007
53. SVOZILOVÁ, B.: Infrastruktura pro IPTV (přenos TV programů v počítačových sítích), MSc. project, 2009
54. ŠTĚPÁN, L.: Šíření DVB-S přes IPTV, MSc. project, in progress

55. URBAN, J.: Návrh a realizace projektu pro podporu výzkumného grantu: Dopravní obslužnost a technologie ve vztahu k regionálnímu rozvoji, Bc. project, 2006
56. VAJDÍK, P.: Srovnávací analýza web frameworků Django, Ruby on Rail a PHP, MSc. project, in progress
57. VALA, R.: Možnosti využití technologie XForms a XML+XSLT v PHP web aplikacích, MSc. project, 2009
58. VALA, R.: Moduly ekonomického systému s architekturou klient-server, Bc. project, 2007
59. VALEŠ, L.: Návrh a realizace softwarového projektu pro výzkum dopravní obslužnosti ve Zlínském regionu, Bc. project, 2006
60. VAŠKOVÁ, V.: Automatické hodnocení úkolů v kurzech programování, Bc. project, 2005
61. VLK, P.: Moduly pro ekonomický systém s architekturou klient-server:, Bc. project, in progress
62. VYDRA, R.: Simulace fyzikálních jevů, Bc. project, 2008
63. ZAVŘEL, J.: Realizace a zabezpečení telefonního centra s využitím technologie Voice Over Internet Protocol, MSc. project, in progress
64. ZIMÁČEK, T.: Hostingový systém a zabezpečení serveru studentských projektů, MSc. project, 2009
65. ZIMÁČEK, T.: Skladový modulární systém s architekturou klient-server, Bc. project, 2006
66. ŽÁČEK, P.: Měření a optimalizace výkonu internetových routerů, MSc. project, 2007

12 LIST OF SYMBOLS AND ABBREVATIONS

ACPR	Adjacent channel protection ratio
AP	Access point
BER	Bit error rate
BPSK	Binary phase shift keying
CCA	Clear channel assessment
CS	Carrier sense
CTI	Cross-technology interference
DSSS	Direct sequence spread spectrum
DFS	Dynamic frequency selection
DVB	Digital Video Broadcasting
ENBW	Equivalent Noise Bandwidth
FEC	Forward error correction
FER	Frame error rate
LTE	Long term evolution
OFDM	Orthogonal frequency division multiplex
QPSK	Quadrature phase shift keying
KVM	Kernel-based virtual machine
MEMS	Micro-electro-mechanical systems
PAPR	Peak to average power ration
PDU	Protocol data unit
PPDU	PLCP protocol data unit
PER	Packet error rate
PHY	Physical layer
PLCP	Physical layer convergence procedure
PMD	Physical medium dependent
PN	Pseudo-noise
RSSI	Receive signal strength indicator
SDR	Software-defined radio
SSID	Service set identifier
STA	WLAN 802.11 station
TPC	Transmit power control
VM	Virtual machine

LIST OF FIGURES