# The MIMONet Software Defined Radio Testbed

Vanessa Gardellin, Konstantinos Mammasis, Francesca Martelli, Paolo Santi
Institute for Informatics and Telematics
Italian National Research Council
Via G. Moruzzi 1, 56124 Pisa, Italy
Email: {v.gardellin, k.mammasis, f.martelli, p.santi}@iit.cnr.it

***Abstract*—Software Defined Radios (SDRs) are radio communication systems where components that are typically implemented in hardware, such as filters, modulators, demodulators, etc., are instead implemented in software. GNU Radio is the most common and efficient software development toolkit that implements SDRs. It is open-source and provides the basic signal processing blocks that can be exploited to implement a SDR testbed, including usage of multi-antenna transmitters and receivers. GNU Radio can be used in combination with different hardware platforms, such as the Universal Software Radio Peripherals (USRPs) produced by Ettus Research$^{TM}$ and designed *with* GNU Radio, which we use in our testbed.**

**This paper presents the setup and initial results of the MIMONet SDR testbed we are currently developing to test multi-antenna system performance at network level. In particular, in this paper we show the importance of appropriately tuning the physical parameters of a SDR testbed, such as transmitter and receiver power, central frequency, etc. As a test case, we compare a single-input single-output (SISO) OFDM communication link with a single-input multiple-output (SIMO) counterpart in terms of bit rate performance. The SIMO link is designed based on the Maximal Ratio Combining (MRC) of signals at the receiving end. MRC is a diversity technique, where signals coming from different receiver antennas are weighted and added together in order to obtain stronger signals compared to the one received by a single antenna.**

**The experimental results collected using the SDR testbed indicate that: $i$) the link quality improvement of SIMO vs SISO link are in accordance with theoretical predictions (3dB); and $ii$) the improvement in bit rate given by a SIMO link depends on the modulation scheme used, and it can be as high as 400% with Quadrature Phase-Shift Keying (QPSK) modulation.**

***Index Terms*—Universal Software Radio Peripheral; Maximal Ratio Combining; Software Defined Radio, GNU Radio.**

## I. Introduction

Software Defined Radio (SDR) systems are at the forefront of modern communication research thanks to their flexibility in assessing various communication architectures with minimal effort. SDR systems are widely considered to be as one of the most important research topics in the area of mobile and personal communications and the reason is attributable on three key terms: reconfigurability, intelligence and flexibility [1]. Therein, reconfigurability plays a major role, enabling mobile systems and networks to be self-configurable by monitoring their own performance and making the appropriate parameter changes. A SDR system essentially adds extra degrees of freedom in the adaptation process, improving the user experience and quality of service in the network. Subsequently, the need of self-awareness from a higher-level network perspective led to the upbringing of cognitive radios, in which communication systems are aware of their internal state and environment, in order to have spectrum agility. Such concepts bind together to form the next generation of intelligent wireless mesh networks.

Technically, SDRs translates to components that were typically implemented in hardware instead being implemented in software, which dramatically reduces the operational and associated hardware integration costs in the transmitting and receiving architectures. As a result of this additional flexibility, functional modules such as modulation/de-modulation, coding/decoding, interleaving/de-interleaving, as well as other more complicated signal processing tasks related to channel equalization and time-frequency synchronization are now performed in software running in personal computers. Therefore, SDR technology facilitates implementation of reconfigurable radio systems where dynamic selection of parameters for the pre-mentioned modules is possible. In a wireless mesh network architecture, the degrees of freedom added by the dynamic selection of parameters cannot be ignored. Although wireless mesh networks are the reference architecture of the MIMONet testbed presented herein, the analysis presented in this paper focuses on a simpler setting, both in terms of the number of nodes used and the associated digital signal processing (DSP) embedded within the SDR platform.

In this paper, a SDR system is presented accompanied by the radio peripheral responsible for performing the baseband to intermediate and intermediate to radio frequency conversions of the incoming signal. Therefore, in this implementation the SDR system serves the complex baseband signal processing needs and it is left to the Universal Software Radio Peripheral (USRP) to perform the above-mentioned conversions. The main task is to develop and test a software defined architecture that enables the communication between two or more USRP devices and sets the basis of a more complex system for the assessment of a multiple-input multiple-output (MIMO) wireless mesh network architecture that employs multi-antenna elements. In this work, a system performance comparison is presented between the Maximal Ratio Combining (MRC) technique employed in a single input multiple output (SIMO) type of system and a single input single output (SISO) system with no diversity combining [2]. As shown later, the MRC scheme confirms the theoretical 3 dB array gain over the SISO system.

The rest of this paper is organized as follows. Section II introduces existing multi antenna testbeds. Section III describes the testbed implementation and configuration, giving in-depth details about the software and hardware used during the experiments. In Section IV, a more general performance assessment is provided based on various types of modulation schemes. In Section IV-C, a performance analysis is performed between the SIMO and SISO systems demonstrating the benefits in using a multi-antenna system. Finally, Section V summarizes the main contributions of this work.

## II. Related Works

Starting in 1998 with the first narrow-band MIMO prototype [3], several testbeds have been set up in the past years in order to investigate specific aspects of MIMO networks, such as MIMO-WCDMA transmission for 3G telephone systems [4], MIMO 3G prototype chip development for high-speed downlink packet access reception [5], and MIMO OFDM transmission for 4G telephone systems [6]. Due to the very high cost of these testbeds, typically operated by industries, nowadays instead testbeds are mostly used in research and about 60 % of them are operated by universities [7].

Wireless network testbeds ([8], [9], [10]) are used to investigate questions about network interactions and performance. Typical applications include mesh networks, multihop routing, link quality measurements, and mobility experiments. These testbeds are characterized by a limited medium access control layer (MAC) customization and a fixed physical layer (PHY).

For this reason, testbeds that investigate the PHY layer, as well as, point-to-point link performance have been implemented ([11], [12], [13]). Applications of these testbeds include MIMO channel measurements, implementing/prototyping new standards, and real-time implementation of space-time codes and MIMO algorithms. These prototypes are generally designed around high-performance hardware such as FPGAs and DSPs. Their advantage is that state-of-the-art PHY layer techniques are implemented, although these prototypes are not coupled to experimental MAC and network software.

Different from these is the *Hydra* testbed implemented at UT Austin [14]. In fact, Hydra is a MIMO testbed built with the goal of facilitating experimentation with state-of-the-art PHY layer, MACs, and other network protocols and their interactions in multihop configurations.

The testbed described in this paper instead, has as main idea the exploitation of MIMO technologies in order to provide a breakthrough capacity increase in wireless mesh networks exploiting advantages from the *network layer*.

## III. Testbed Implementation and Configuration

Wireless network testbed are commonly used in order to investigate questions about network interactions and performance. In comparison to a network simulator, the advantage of a testbed is that the system is *real*, allowing the treatment of various problems related to actual implementations. Keeping in mind the main goal of our testbed, i.e., the network layer exploitation of MIMO techniques, we start designing and developing a multiple-antenna system in order to exploit more efficiently the available frequency spectrum.

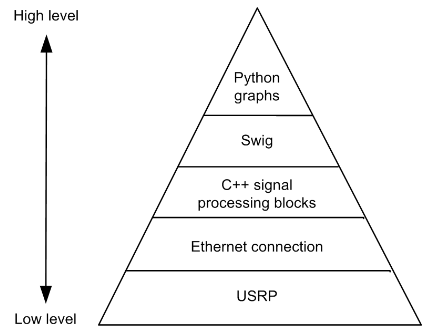



Figure 1. GNU Radio pyramid **(PAOLO: Is this picture taken from [15]? If so, add reference)**.

In Section III-A, GNU Radio is described, which is a software defined radio platform that implements a library of signal processing blocks. In Section III-B, the reconfigurable hardware used in order to manage the signal processing tasks is presented. A description on the testbed implementation is given in Section III-C, while the scenario and associated parameters are listed in Section III-D.

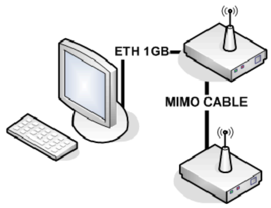
### A. Testbed Software

GNU Radio [15] is a free and open-source software for implementing SDRs. It is easy reconfigurable and hence it adapts to different network architectures and protocols. In fact, the GNU Radio architecture is based on a set of reconfigurable data sources, data sinks and signal processing blocks that are connected together to form transmitter and receiver chains. Figure 1 shows how GNU Radio works from the PHY layer up to the application layer. At the lower level are the USRP Hardware Drivers (UHDs) that provide an application programming interface (API) for USRPs and guarantee correct information exchange between USRPs and PCs (Personal Computers) through an Ethernet cable. Then, GNU Radio provides a library of signal processing blocks in C++, which implement various functionalities of the transmit and receive architectures. These C++ blocks can be used and combined by the final user through Python; in fact, GNU Radio provides a Python interface for each C++ block, as well as tools to connect the various blocks. This way, the simplified wrapper and interface generator (swig) allows C++ code to be used from Python.

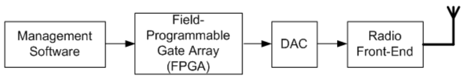
### B. Testbed Hardware

The hardware used in order to implement the testbed is comprised mainly of PCs and USRPs as shown in Fig. 2(a). More specifically, each node of the testbed houses a PC, two USRPs and a MIMO cable.

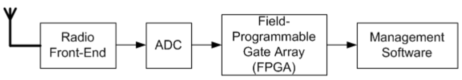
USRPs are a family of computer-hosted hardware produced by Ettus Research$^{TM}$ [16], and designed *with* GNU Radio. A single USRP unit consists of a motherboard and a daughterboard, where the latter is the radio front-end. The USRP N210 was chosen as the motherboard and the XCVR2450




(a) Testbed Node




(b) Transmitter chain




(c) Receiver chain

Figure 2. Testbed hardware chains

as the daughterboard due to their computational capabilities, as well as their ability to operate in the frequency ranges of $[2.4; 2.5]$ GHz and $[4.9; 5.85]$ GHz, which are the frequencies used by the IEEE 802.11n standard for multiple antenna wireless networks.

**PAOLO: Describe shortly also the features of the PCs.**

Figure 2(b) and Fig. 2(c) depict the transmitter and the receiver hardware chains, respectively. When a USRP is used as a transmitter, the management software produces a signal that is converted from digital to analog (DAC) so that it can be suitable for the radio front-end for subsequent transmission over the air. On the receiver side, the radio front-end receives the signal which is converted from analog to digital (ADC) and sent to the management software.

The connection between PCs and USRPs is through an Ethernet cable, while the connection between the two USRPs is provided by a so called MIMO cable. The MIMO cable is designed to guarantee synchronization between the two units, i.e., coherence in sampling clocks and local oscillators, which is essential for implementation of any MIMO technique. A MIMO cable can be used in a variety of ways. The so called *shared Ethernet mode* was adopted in this instance, where one USRP is attached to the Ethernet cable and clock reference; time reference and *data* are communicated over the MIMO cable. This method differs from the dual Ethernet mode, where both USRPs are connected to the PC through an Ethernet cable, and no data are communicated over the MIMO cable.

**PAOLO: comment on the rationale behind the above choice; I guess: it is easier to program?**

### C. Testbed Implementation

Considering that the entire USRP design is open source, as well as the GNU Radio software, a fully scalable open source SDR system is implemented, enabling host based signal processing on low cost hardware. The testbed implements Orthogonal Frequency-Division Multiplexing (OFDM), where the main signal is split in a sub-set of independently modulated signals on orthogonal sub-carriers. Three modulation schemes have been coded: binary phase-shift keying (BPSK), quadrature PSK (QPSK) and 8PSK, which differ in the number of bits mapped into a constellation point, i.e., a symbol.

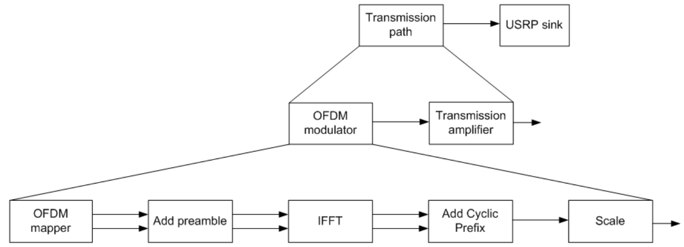



Figure 3. GNU Radio transmit chain

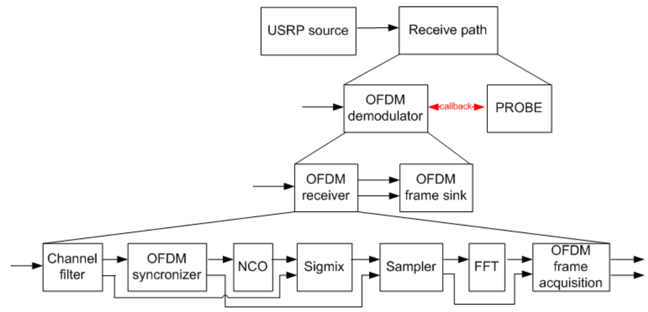



Figure 4. GNU Radio receiver chain

The transmitter software implemented in GNU Radio is shown in Fig. 3 and its main functionalities are: (i) an application that generates packets, (ii) an OFDM mapper that produces as many OFDM symbols are needed in order to hold a full packet according to the modulation scheme used, (iii) a preamble adder that inserts pre-modulated known symbols before each packet in order to perform synchronization and to estimate the communication channel at the receiver, (iv) an Inverse Fast Fourier Transform (IFFT) that produces OFDM symbols, (v) a cyclic prefix adder that copies a portion of symbols located at the end of the OFDM symbol, at the begin of the same symbol, and finally, (vi) a scale factor that normalizes the signal power of the symbols. **The application block – item (i) – is not reported in Fig 3. All others blocks instead are reported in the Fig.**

Figure 4 shows the receiver software implemented in GNU Radio. The main component is the OFDM demodulator which performs synchronization on the received data, demodulates symbols into bits and packs bits into packets. The synchronization stage is the most complex because it performs several operations: (i) frame detection, (ii) Schmidl and Cox preamble correlation [17] in order to perform timing synchronization and fine frequency offset correction, (iii) cyclic prefix removal, (iv) Fast Fourier Transformation (FFT), so that symbols are transformed to the frequency domain, and, (v) sub-carriers equalization using the least square estimation technique based on the received preamble and the known symbols in order to remove the distortion introduced by communication channel effects.

*D. Parameters and Scenario*

Several parameters need to be appropriately chosen in order to set up a SDR testbed. These parameters are divided in three sets:

- USRP parameters
- OFDM parameters
- Application parameters

**USRP parameters** can be summarized as follows: (i) central frequency, (ii) sample rate, and, (iii) transmitter/receiver power. The central frequency used for transmission/reception depends on the daughterboard of the USRP, in our implementation it can vary within the two ranges $[2.4; 2.5]$ GHz and $[4.9; 5.85]$ GHz. The sample rate is the speed at which samples are clocked in and out of the transmitter and receiver. The sample rate is limited by the following factors: (i) PC's speed, which is 2 MS/s (MegaSamples/s) in the described setup; (ii) Ethernet bound, i.e., $\frac{1\text{Gb/s}}{32} = 31.25\text{MS/s}$, where 32 is the number of bits per complex sample; and, (iii) USRP limit, which is 25 MS/s due to hardware constraints. Hence, the chosen sample rate equals 2 MS/s, corresponding to the tighter among the above limits. The transmitter power depends on transmitter gain and transmitter amplitude, while the receiver power depends only on the receiver gain. More specifically, transmitter and receiver gains compensate losses in USRPs, and they need to be properly set depending on the network topology. The transmitter amplitude, instead, is a scale factor related to the format of the signals expected by the UHD. Each OFDM symbol is multiplied by this constant to fall in the range expected by the hardware. It depends on the network topology, the used central frequency, and the OFDM modulator. USRP parameter ranges are summarized in Table I.

**OFDM parameters** are: (i) OFDM modulation scheme, i.e., one of BPSK, QPSK or 8PSK in our set up; (ii) FFT length: total number of sub-carriers in which the available bandwidth is divided; (iii) occupied tones: sub-carriers used for actual data transmission; (iv) cyclic prefix (CP) length: sub-carriers in the cyclic prefix; (v) OFDM symbol length: sum of FFT length and CP length. Note that for each OFDM symbol, the 2 central sub-carriers are not used for data transmission, hence the number of occupied tones needs to be considered excluding the 2 central sub-carriers. In Table II, the used OFDM parameters are shown. The first two settings are as defined in the IEEE 802.11n standard [18], while the latter is as suggested by the Ettus Research$^{TM}$ and described in the GNU Radio tutorials. Given the sample rate and the OFDM parameters we can obtain the occupied bandwidth as is Eq. (1)

$$bandwidth = \frac{sample\ rate \cdot occupied\ tones}{FFT\ length} \qquad (1)$$

**Application parameters** are the number of transmitted packets and their size. Each packet contains header, payload and pad, due to compatibility with USRP old drivers, as depicted in Fig. 5. The header is 4 bytes long and it consists of an access code of 4 bits and the packet length of 12 bits, hence the maximum length of a packet is 4096 bytes. This information is repeated twice because the receiver node examines the correctness of the header checking if the two halves are equal. The payload contains data and a cyclic redundancy check of 32 bits (CRC32) that is used by the receiver node to check if the payload has been correctly received.

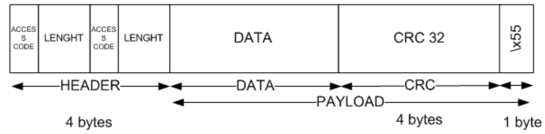



Figure 5. Packet structure

| | |
|---|---|
| Central frequency | 2.4 GHz and 5.2 GHz |
| Sample rate | 2 MS/s (Mega samples per second) |
| Tx gain | (0 : 35] |
| Rx gain | (0 : 99] |
| Tx amplitude | (0 : 1) |

Table I
USRP PARAMETERS

In order to understand the complexity of the system that is implemented, considering the combination of all the transmitter and receiver parameters, one obtains 967680 different tests; considering 10 runs for each test, more than 10 millions runs should be performed to precisely identify the best configuration. This is clearly infeasible, and the alternative approach of investigating a smaller parameter space by fixing some parameters has been pursued instead. For example, the transmitter power is set to the maximum value, and only some values for the receiver power are examined. Using these assumptions, the number of tests is reduced to about 10000 runs on which the outcomes presented in Section IV are based.

Finally, Fig. 6 shows the transmitter and receiver location during the experiments. The distance between transmitter and receiver is 5 meters and they are separated by a wall of wood and steel.

## IV. Performance Evaluation

In this section, the parameters used during this experimental campaign are analyzed, and subsequently the outcomes of a set of measurements on a single communication link are shown.

*A. Scenario setup*

Experiments were conducted during the night time in order to avoid interference from external wireless devices. Results are averaged on 10 experiments, and confidence intervals are

| FFT length | Occupied tones | CP length |
|---|---|---|
| 64 | 56 | 16 |
| 128 | 112 | 32 |
| 512 | 200 | 128 |

Table II
OFDM PARAMETERS

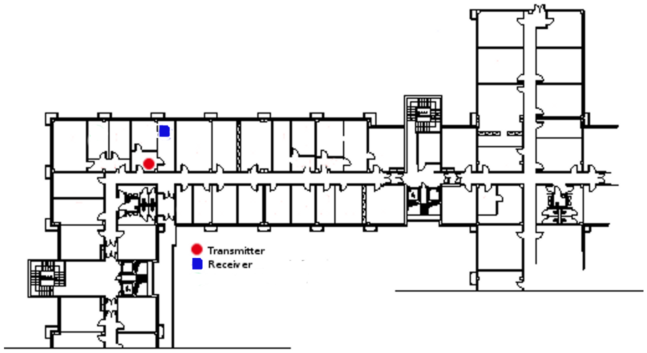



Figure 6. Testbed Scenario

| Packet size (bytes) | Theoretical Bit rate (Kb/s) | Measured Bit rate (Kb/s) |
|---|---|---|
| 500 | 618.75 | 571.54 ± 0.013 |
| 1500 | 618.75 | 606.65 ± 0.008 |
| 4000 | 618.75 | 614.67 ± 0.019 |

Table III
THEORETICAL AND MEASURED BIT RATES (AT THE TRANSMITTER SIDE) FOR DIFFERENT PACKET SIZES, MODULATION IS BPSK, FFT=512, CP=128, OCCUPIED TONES = 200.

reported in the plots. Metrics used in order to evaluate our testbed are the following: the number of bits received and correctly received in the unit of time (Kb/s), and the percentage of received, correctly received and lost packets.

### *B. Results*

To being with, a comparison is presented between the theoretical and measured bit rates. The theoretical bit rate is given by:

$$\begin{aligned} bit\ rate = symbol\ rate \cdot (occupied\ tones - 2) \cdot \\ \cdot\ bit\ per\ symbol \cdot encoding\ rate, \end{aligned} \tag{2}$$

where the encoding rate depends on the Forward Error Correction (FEC)[1]. The symbol rate is given by:

$$symbol\ rate = \Delta f \cdot \frac{T_S}{T_T} = 3.125 \text{ KHz}, \tag{3}$$

where $T_T$ is the time to transmit an OFDM symbol and is equal to 320$\mu$s, while $T_S$ is the time to transmit a symbol. $\Delta f$ is the bandwidth of a sub-carrier and is given by:

$$\Delta f = \frac{sample\ rate}{FFT\ length} = \frac{2 \text{ MHz}}{512} = 3.90625 \text{ KHz}. \tag{4}$$

In Table IV, the theoretical and the measured bit rates at the receiver are depicted varying the modulation scheme. OFDM parameters are as defined in the third row of Table II. In these tests, the packet size was set to 1500 bytes. For smaller packet sizes, lower bit rates were obtained because of the higher overhead, as shown in Table III.

In order to understand how our testbed is affect by OFDM parameters, in Fig. 7 the three configurations in Table II

[1]FEC has not yet been implemented, and therefore in this case the encoding rate is equal to 1.

| Modulation | Theoretical Bit rate (Kb/s) | Measured Bit rate (Kb/s) |
|---|---|---|
| BPSK | 618.75 | 606.65 ± 0.008 |
| QPSK | 1237.5 | 1177.24 ± 0.043 |
| 8PSK | 1856.25 | 1715.00 ± 0.156 |

Table IV
THEORETICAL AND MEASURED BIT RATES.

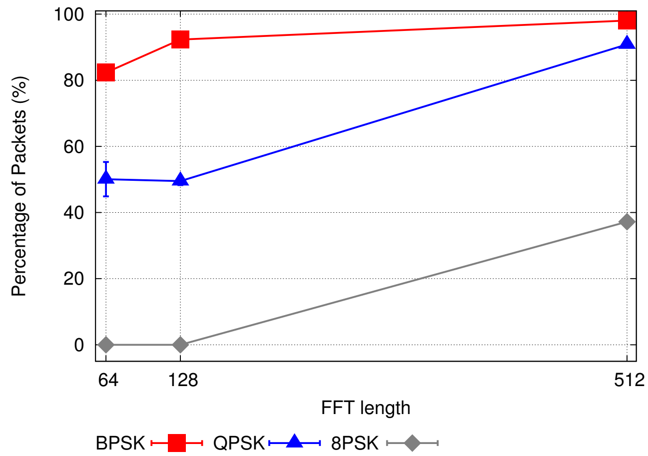



Figure 7. OFDM parameters comparison

are evaluated. Note that the main difference between these configurations is the ratio between occupied and unoccupied carriers, and hence the bandwidth of the signal as shown in Eq. (1). The comparison is by mean of percentage of correctly received packets. We can notice how having more space between occupied tones helps in synchronization and hence the percentage of correctly received packet is the highest with the third set of parameters. Hereafter, we set the FFT length to 512, occupied tones to 200 and CP length to 128.

Figure 8 shows the comparison between the testbed working with central frequency equal to 2.4 GHz or to 5.2 GHz. A proper set of USRP parameters improves the performance of the system in terms of bit rate. Knowing that at higher frequencies the system needs more transmit power, the transmitter amplitude is increased from 0.2 to 0.4 when the system works at 5.2 GHz. Subsequently, a large set of tests was carried out to assess the optimal USRP parameters, in accordance with the suggestion in [16]. Due to limited space, it is not possible to report all results, and therefore only the main findings are listed. It was realized that the transmission power should be kept as high as possible, except when the transmitter and receiver nodes are very close, lets say one or two meters apart and in line of sight. Regarding the receiver gain, instead, values ranging in $[40 - 80]$ produce the best results in terms of percentage of received and correctly receive packets. In any case, all aforementioned parameters depend on the network topology and the central frequency used as shown in Fig. 8. In order to perform the measurements described in the Section IV-C, the optimal configuration for transmitter amplitude, transmitter gain and receiver gain is sought.

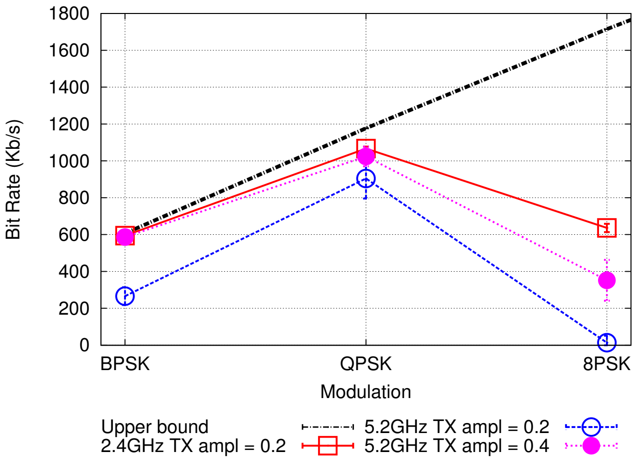



Figure 8. Central Frequency Comparison

| | SISO | MRC |
|---|---|---|
| Transmitter amplitude | 0.2 | 0.2 |
| Transmitter gain | 25 | 25 |
| Receiver gain | 60 | 80 |

Table V
SISO AND MRC OPTIMAL SETTINGS

**Algorithm 1:** SNR Computation

**for** $i = 0$; $i <$ *occupied tones*; $i++$; **do**
  $dist_i = |received\ symbol - closest\ constellation\ symbol|$

$$sum = \sum_{i=1}^{occupied\ tones} \frac{1}{|dist_i|^2};$$

$$\text{SNR} = 20 \log_{10} \sqrt{\frac{sum}{occupied\ tones}}$$

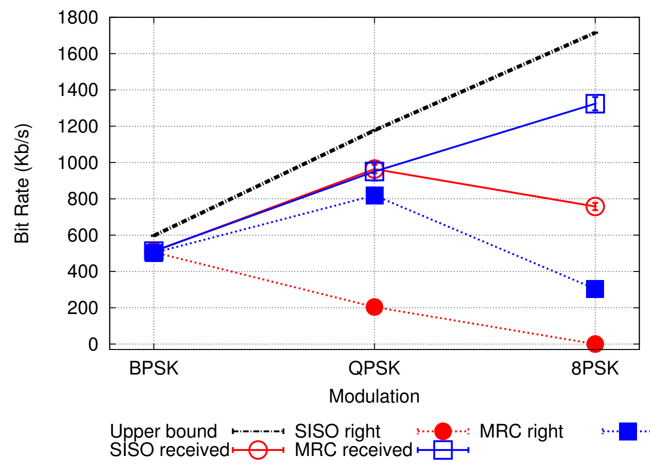



Figure 10. SISO vs. MRC, packet level analysis.

### C. SISO vs MRC performance evaluation

A set of tests was performed in order to compare a SISO and a SIMO system. In a SIMO system, the channel experienced by each receiver antenna is randomly varying in time and independently from the channel experienced by the other receiver antenna (if the two antennas are sufficiently separated). The technique used in order to implement the SIMO system lies on the principles of diversity combing, i.e., the MRC technique. According to MRC, signals coming from different antennas are multiplied by a weight proportional to the signal amplitude, summed up together, and then normalized. Weights are chosen so that strong signals are further amplified and weak signals are attenuated.

A set of tests was performed to measure the performance gain that one can obtain in practice by using MRC, with respect to the simpler SISO alternative. Both systems were tuned to their optimal configuration parameters, which are reported in Table V, and the central frequency is set to 2.4 GHz.

A Signal to Noise Ratio (SNR) analysis was performed in order to confirm the theoretical 3 dB array gain improvement [2]. The array gain refers to the average increase in SNR at the receiver that arises from the coherent combining of signals at the transmitter or receiver. Alternatively, this is the gain achieved by weighting the signals from each antenna based on the knowledge of the channel state information. The SNR was computed for each OFDM symbol in each received packet as in Algorithm1.

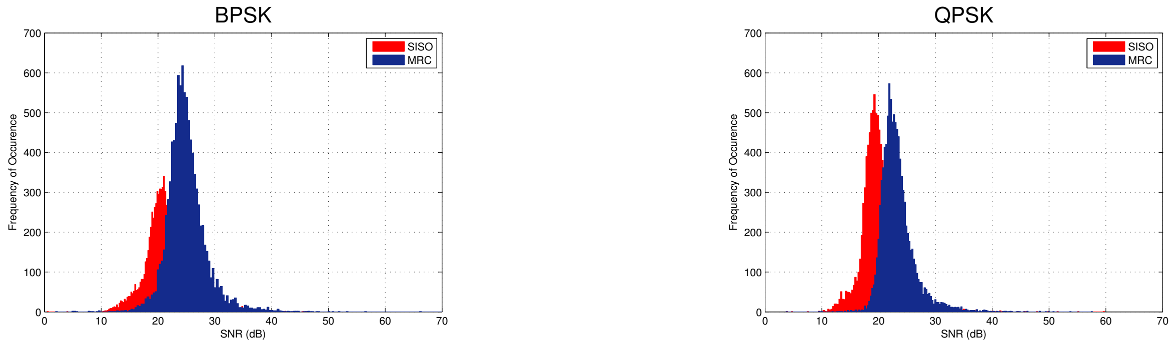
Figure 9 shows the frequency of observing different SNR values during the experiment time, and the difference in dB of the average SNR (**PAOLO: is this the average SNR?**) for different modulation schemes. Comparing the SNR values of the SISO and MRC cases, it is evident from the figures and the table that the MRC achieves the theoretical 3 dB array gain improvement irrespective of the modulation scheme.

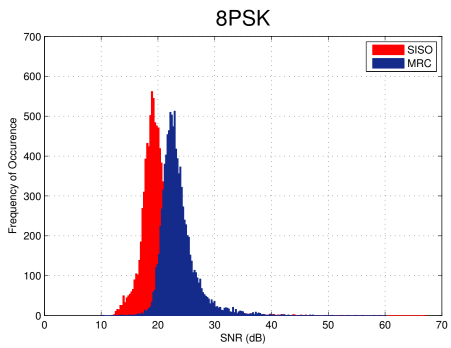
In order to enhance the above mentioned findings, in Figure 10 the advantage in terms of the bit rate of a SIMO link for different modulation schemes is shown. The *Upper bound* line is the theoretical bit rate. Note that the bit rate improvement depends on the modulation scheme and it is up to 400% with QPSK modulation.

## V. CONCLUSIONS

In this paper, hardware and software used in order to implement a SDR testbed were presented and briefly analyzed. The software used was GNU Radio and the hardware Universal Software Radio Peripherals produced by Ettus Research$^{TM}$. Experiments were conducted and the results showed in order to analyze OFDM, USRP and application parameters. The MRC receiver diversity technique was implemented, confirming the theoretical 3 dB array gain improvement over the SISO system in this practical set-up, and showing up to 400% improvement in bit rate with QPSK modulation.

The next steps in our MIMONet testbed development are implementation of MIMO techniques encompassing spatial multiplexing, interference suppression, and other diversity techniques.




| Modulation | Difference (dB) |
|---|---|
| BPSK | 3.4608 |
| QPSK | 3.3134 |
| 8PSK | 3.4042 |

Figure 9. SNR gain of MRC for different modulation schemes.

## References


[1] M. Dillinger, K. Madani, and N. Alonistioti. *Software Defined Radio: Architectures, Systems and Functions*. John Wiley & Sons, 2003.

[2] M. Jankiraman. *Space-Time Codes and MIMO Systems*. Artech House, August 2004.

[3] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela. V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel. In *IEEE International Symposium on Signals, Systems, and Electronics*, pages 295–300, 1998.

[4] A. Adjoudani, E. C. Beck, A. P. Burg, G. M. Djuknic, T.G. Gvoth, D. Haessig, S. Manji, M. A. Milbrodt, M. Rupp, D. Samardzija, A.B. Siegel, T. Sizer, C. Tran, S. Walker, S. A. Wilkus, and P. W. Wolniansky. Prototype experience for MIMO BLAST over third generation wireless system. *IEEE Journal on Selected Areas in Communications*, 21(3):440–451, Apr. 2003.

[5] D. Garrett, G. Woodward, L. Davis, G. Knagge, and C. Nicol. A 28.8 Mb/s 4×4 MIMO 3G high-speed downlink packet access receiver with normalized least mean square equalization. In *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, pages 420–536. IEEE, 2004.

[6] S. H. Won, D.-S. Lyu, and H. J. Park. Physical layer implementation and evaluation of multiple input multiple output-orthogonal frequency division multiplexing (MIMO-OFDM) system. In *IEEE International Conference on Communication Technology Proceedings (ICCT)*, volume 2, pages 1348–1352, 9–11 Apr., Beijing, China, 2003.

[7] T. Kaiser, A. Wilzeck, M. Berentsen, and M. Rupp. Prototyping for Mimo Systems - An Overview. In *European Signal Processing Conference (EUSIPCO)*, 2004.

[8] G. Holland, N. Vaidya, and Paramvir Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 236–251, 16–21 Jul., Rome, Italy, 2001.

[9] P. De, A. Raniwala, S. Sharma, and T.C. Chiueh. Design considerations for a multihop wireless network testbed. *IEEE Communications Magazine*, 43(10):102–109, October 2005.

[10] E. Nordstrom, P. Gunningberg, and H. Lundgren. A testbed and methodology for experimental evaluation of wireless mobile ad hoc networks. In *IEEE International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, pages 100–109, 23–25 Feb., Trento, Italy, 2005.

[11] R.M. Rao, S. Lang, C. Oberli, D. Browne, J. Bhatia, J.-F. Fr, P. Gupta, D.N. Liu, S.G. Wong, M. Fitz, B. Daneshrad, and O. Takeshita. Multi-antenna testbeds for research and education in wireless communications. *IEEE Communications Magazine*, 42(12):72–81, Dec. 2004.

[12] S. Caban, C. Mehlführer, A. L. Langwieser, R.and Scholtz, and M. Rupp. Vienna MIMO Testbed. *EURASIP Journal on Advances in Signal Processing*, pages 1–14, 2006.

[13] M. Takai, R. Bagrodia, M. Gerla, B. Daneshrad, M. Fitz, M. Srivastava, E. Belding-Royer, S. Krishnamurthy, M. Molle, P. Mohapatra, U. Mitra, and J. Evans. Scalable Testbed for Next-Generation Wireless Networking Technologies. In *IEEE International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, pages 162–171, 25 Feb., Trento, Italy, 2005.

[14] W. Kim, O. Khan, K.T. Truong, S.H. Choi, R. Grant, H.K. Wright, K. Mandke, R.C. Daniels, R.W. Heath, and S.M. Nettles. An experimental evaluation of rate adaptation for multi-antenna systems. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 2313–2321, 19–25 Apr., Rio De Janeiro, Brazil, 2009.

[15] http://gnuradio.org/.

[16] http://www.ettus.com/.

[17] T.M. Schmidl and D. C. Cox. Robust frequency and timing synchronization for OFDM. *IEEE Transactions on Communications*, 45(12):1613–1621, 1997.

[18] 802.11n-2009 - ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac)and physical layer (phy) specifications amendment 5: Enhancements for higher throughput. http://standards.ieee.org/findstds/standard/802.11n-2009.html, 2009.