

Research Article

Design and Implementation of a Single-Frequency Mesh Network Using OpenAirInterface

Florian Kaltenberger, Rizwan Ghaffar, Raymond Knopp, Hicham Anouar, and Christian Bonnet

Eurecom, 2229, Route des Cretes, B.P. 193, 06904 Sophia Antipolis, France

Correspondence should be addressed to Florian Kaltenberger, florian.kaltenberger@eurecom.fr

Received 1 June 2009; Revised 20 October 2009; Accepted 11 January 2010

Academic Editor: Faouzi Bader

Copyright © 2010 Florian Kaltenberger et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

OpenAirInterface is an experimental open-source real-time hardware and software platform for experimentation in wireless communications and signal processing. With the help of OpenAirInterface, researchers can demonstrate novel ideas quickly and verify them in a realistic environment. Its current implementation provides a full open-source software modem comprising physical and link layer functionalities for cellular and mesh network topologies. The physical (PHY) layer of the platform targets fourth generation wireless networks and thus uses orthogonal frequency division multiple access (OFDMA) together with multiple-input multiple-output (MIMO) techniques. The current hardware supports 5 MHz bandwidth and two transmit/receive antennas. The media access (MAC) layer of the platform supports an abundant two-way signaling for enabling collaboration, scheduling protocols, as well as traffic and channel measurements. In this paper, we focus on the mesh topology and show how to implement a single-frequency mesh network with OpenAirInterface. The key ingredients to enable such a network are a dual-stream MIMO receiver structure and a distributed network synchronization algorithm. We show how to implement these two algorithms in real-time on the OpenAirInterface platform. Further more, we provide results from field trials and compare them to the simulation results.

1. Introduction

The design and implementation of next generation wireless networks is a very challenging task. To ensure optimal performance it is necessary to carry out performance evaluations and field trials in parallel to standard development. Easily reconfigurable testbeds are a convenient way to investigate new ideas and to tackle many problems at an early development stage.

Novel ideas for wireless networks are usually first studied using computer simulations based on some kind of model of the network, the hardware and the radio channel. These models usually make assumptions in order to simplify or isolate the problem at hand. However, it might turn out that the assumptions are not fulfilled in a real environment. An easily reconfigurable experimental platform allows to study novel algorithms under realistic conditions. Comparing

simulation results with results from lab tests and field trials reveals if initial assumptions were correct or if they need to be refined.

This paper presents the Eurecom testbed OpenAirInterface, which is an experimental real-time, open-source hardware and software platform for future wireless networks. OpenAirInterface can be seen as a mock standard for realistic experimentation purposes which retains the salient features of a real radio system, without all the required mechanisms one would find in a standard used in deployment of commercial networks. Its aim is to study techniques such as multicell cooperative techniques, distributed synchronization, interference coordination and cancellation.

OpenAirInterface features an open-source software modem written in C comprising physical and link layer functionalities for cellular and mesh network topologies. This software modem can be used either for extensive

computer simulations using different channel models or it can be used for real-time operation. In the latter case, it is run under the control of the real-time application interface (RTAI) which is an extension of the Linux operating system.

The use of an open-source software modem has several advantages. Firstly, the same code can be initially debugged and tuned in simulation before using it in the real-time modem (where debugging and performance analysis is much harder). Secondly, the system is very flexible and parameters like frame structure, pilot placement, and so forth, can be changed rather easily. Thirdly, researchers can implement new ideas rather fast, without having to use very sophisticated hardware description languages (HDL). Last but not least since all code is open-source, other researchers can use easily adjust the modem to their needs and collaboration is fostered.

Other highlights of the OpenAirInterface platform are its usage as measurement platform or as an emulation platform which allow to study different aspects of a wireless network in isolation. In the emulation mode, the physical layer is abstracted and emulated over the ethernet. This approach allows to test and investigate MAC and link-level algorithms without using the radio interface [1]. The OpenAirInterface can also be used to perform channel measurements which can be used for channel characterization and capacity analysis [2].

Apart from a general overview of OpenAirInterface, in this paper we present OpenAirMesh—a specification of wireless mesh network and its implementation on the OpenAirInterface platform. OpenAirMesh exemplifies two major challenges in future wireless networks. The first challenge is interference which is caused by a very tight frequency reuse in order to increase the network throughput. Interference is especially strong for users at the cell edge severely limiting user's throughput. We propose a low-complexity dual-stream MIMO receiver that is able to cancel out interference from a neighboring cell and show its implementation and performance on the OpenAirInterface. This example also highlights the insight that OpenAirInterface provides for developing multicell algorithms. More concretely it was found out that frequency offsets and receive correlation have a very strong influence on the receiver performance and can thus not be neglected in the simulations.

Another big challenge in future wireless networks is synchronization between nodes, especially indoors where a reference timing signal such as the one provided by the global positioning system (GPS) cannot be used. Synchronization is needed for example for the dual-stream MIMO receiver and to enable the collaboration between base-stations both on the media access (MAC) and the physical (PHY) layer. The distributed synchronization algorithm proposed in this paper can be interpreted as a form of firefly synchronization [3].

1.1. Related Work. Many people have recognized the advantages of testbeds for wireless networks. However, many of them focus on a particular layer, such as the physical layer, the link layer or the network layer. A good overview of

testbeds that focus on the physical layer and especially MIMO communications is given in [4, 5].

Most of the testbeds that provide both physical and link layer functionalities (like OpenAirInterface) are either based on the Universal Software Radio Peripheral (USRP) from Ettus Research [6] together with the GNU software radio [7] or the wireless open-access research platform (WARP) from Rice University [8]. For example, the Hydra testbed of the University of Texas in Austin [9] is based on the GNU radio platform while the WAPRnet testbed [10] is based on WARP platform. The WiTestLab from the Polytechnic Institute of NYU [11] has been experimenting with both platforms. Another testbed example is the Cognitive Radio Testbed from Berkley [12], which uses the Berkeley Emulation Engine 2 (BEE2) [13] as an implementation platform.

Compared to the OpenAirInterface platform, the GNU radio project does not provide a full reference design, but only building blocks. Further, a MAC layer implementation is missing in the current distribution. Also the USRP hardware has its limitations, mainly due to the connection to the PC over USB or Ethernet, which severely limits the achievable system throughput. Like the OpenAirInterface, WARP is also a full software defined radio (SDR), but physical layer algorithms have to be developed either directly in VHDL or using the Xilinx System Generator toolchain for Matlab. Compared to the use of C language in OpenAirInterface, the use of VHDL is more cumbersome and time consuming. Also, the Xilinx System Generator is not openly available. The BEE2 platform is a very flexible hardware platform, which has been designed for a multitude of applications. However, no software modem exists for this platform.

Last but not least we mention here the two testbeds developed within the EASY-C project (<http://www.easy-c.de/>) that were set up in Berlin [14] and in Dresden [15] (both Germany). The project is a cooperation between German universities, research centers and industry and focuses on LTE-Advanced technologies. However, both testbeds use proprietary hardware and software and are not openly accessible.

1.2. Organization. Section 2 gives an overview of the OpenAirInterface experimental platform. Section 3 presents the network, the link layer and the physical layer architecture of OpenAirMesh—a mesh network built using the OpenAirInterface. Section 4 describes the two building blocks for the implementation of OpenAirMesh: a novel low-complexity dual-stream receiver architecture and the distributed synchronization algorithm. Finally we show results from computer simulations as well as real experiments in Section 5. We conclude the paper in Section 6.

Notation. Let \mathbb{C} denote the set of complex numbers. Scalars are denoted by x . Column vectors and matrices are denoted by \mathbf{a} and \mathbf{A} and their elements are denoted by a_i and $A_{i,j}$ respectively. Transpose and Hermitian transpose are denoted by \cdot^T and \cdot^H . \mathbf{I}_M is the identity matrix of size M and $\mathbf{0}_M$ is an M -dimensional vector of zeros. The Euclidean (ℓ_2) norm of a vector \mathbf{a} is denoted by $\|\mathbf{a}\|$ and the Frobenius norm of a matrix \mathbf{A} is denoted by $\|\mathbf{A}\|_F$. \mathbb{E} denotes expectation, and

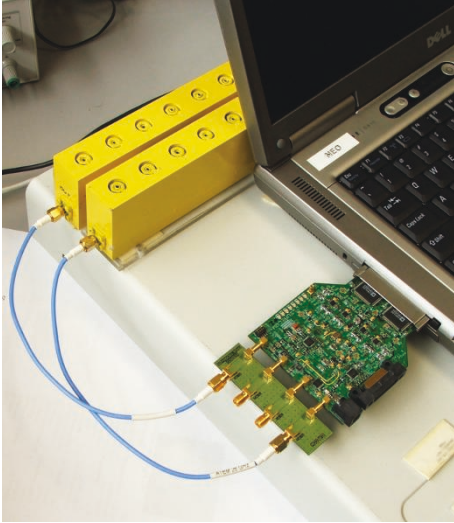


FIGURE 1: The CBMIMO1 board.

$\mathcal{CN}(\mathbf{m}, \mathbf{C})$ denotes a multivariate proper complex normal distribution with mean vector \mathbf{m} and covariance matrix \mathbf{C} .

2. OpenAirInterface Overview

The OpenAirInterface platform consists of both hardware and the software components. Additionally it comprises different simulation tools as well as collaborative web tools. The hardware components are described in Section 2.1. In Section 2.2, we describe the basic organization of the OpenAirInterface software components (which are available under the GNU GPL from the OpenAirInterface website (<http://www.openairinterface.org/>)).

2.1. Hardware Components. In OpenAirInterface there are two different hardware modules available: CardBus MIMO1 (CBMIMO1) and its successor Express MIMO. All current activities (including the experiments described in this paper) are based on CBMIMO1. In the following we will describe the main characteristics of the two boards.

The CBMIMO1 board (cf. Figure 1) comprises two time-division duplex (TDD) radio frequency (RF) chains operating at 1.900–1.920 GHz with 5 MHz channels and 21dBm transmit power per antenna for an orthogonal frequency division modulated (OFDM) waveform (EURECOM has a frequency allocation for experimentation around its premises in Sophia Antipolis). The cards house a medium-scale field programmable gate array (FPGA) (Xilinx X2CV3000) which makes use of the open-source LEON3 embedded processor from Gaisler research [16]. In the current version, the FPGA implements the interfaces with the Peripheral Component Interconnect (PCI) bus, with the RF frontend as well as with the A/D and D/A converters. The card can be connected to a host PC (in our lab we use Dell Precision M2300 laptops) using a CardBus PCI interface. See Table 1 for an overview of the card's components.

TABLE 1: Hardware characteristics of the CardBus MIMO1 card.

FPGA Components	Xilinx Virtex 2 3000
Data Converters	2x AD9832 (dual 14-bit 128 Msps D/A, dual 12-bit 64 Msps A/D)
MIMO Capability	2×2
RF TX Chipset	2x Maxim MAX2395 (1900–1920 MHz) Zero-IF
RF RX Chipset	2x Maxim MAX2393 (1900–1920 MHz) Zero-IF
TX Power	21 dBm per antenna
RX Noise Figure	6-7 dB at highest gain setting
Bus Interface	32-bit PCI (CardBus)
Configuration	Flash EEPROM, Xilinx JTAG port (FPGA and EEPROM)

TABLE 2: Hardware characteristics of the Express MIMO card.

FPGA Components	Virtex 5 LX330, Virtex 5 LX110T
Data Converters	4x AD9832 (dual 14-bit 128 Msps D/A, dual 12-bit 64 Msps A/D)
MIMO Capability	4×4 Quadrature, 8×8 low-IF
Memory	128 Mbytes/133 MHz DDR (LX110T), 1-2 Gbytes DDR2 (LX330)
Bus Interface	PCIExpress 8-way
Configuration	512 Mbytes Compact Flash (SystemACE), JTAG

Express MIMO is a baseband processing board, which provides significantly more processing power and bandwidth than CBMIMO1 and will be used for future applications. It comprises two FPGAs: one Xilinx XC5VLX330 for real-time embedded signal processing applications [17] and one Xilinx XC5VLX110T for control. The card uses an eight-way PCI express interface to communicate with the host PC. The card employs four high-speed A/D and D/A converters from Analog Devices (AD9832) allowing to drive four RF chains using quadrature modulation or eight RF chains in low intermediate frequency (IF) for bandwidths of up to 20 MHz; See Table 2 for an overview of the card's components. A RF board for Express MIMO called Agile RF is also available. It offers significantly more RF functionality in terms of tuning range and channel bandwidth than CBMIMO1. The tuning range per RF chain is 180 MHz–8 GHz with 20 MHz channels.

2.2. Software Components. The software components are organized into four areas (folders), which correspond more or less to the different layers of the Open Systems Interconnection (OSI) reference model. The areas also correspond to the directory structure on the OpenAirInterface Subversion (SVN) server.

(a) *Openair0: Wireless Embedded System Design.* This folder mainly contains descriptions of the CBMIMO1 and Express

MIMO hardware and the firmware for the corresponding FPGAs.

(b) *Openair1: Baseband Signal Processing.* This folder contains the code for the physical layer software modem along with RTAI/Linux device drivers and user-space tools to control the hardware. It also contains simulation environments and channel models to test the code without the hardware or to do performance simulations. Further, openair1 provides also the functionality for the Eurecom MIMO OpenAir Sounder (EMOS) to perform MIMO channel measurements over multiple users [2].

(c) *Openair2: Medium-Access Protocols.* This folder contains the layer 2 protocol stack development for PCs along with Linux networking device drivers for Internet Protocol (IP) and Multiprotocol Label Switching (MPLS) interconnection. This pertains to both cellular and mesh network topologies. The folder also contains an abstraction of the PHY layer, providing an efficient emulation platform for layer 2 and higher algorithms.

(d) *Openair3: Wireless Networking.* This contains the layer 3 protocol stack development for both all-IP cellular and IP/MPLS mesh networks.

3. OpenAirMesh Specification

In this section, we present the specification of OpenAirMesh, a mesh network built using the OpenAirInterface [18]. We start off by describing the network topology in Section 3.1. In Section 3.2 we describe the layer 2 and finally in Section 3.3 the physical layer. A more detailed specification can be found in [19].

3.1. Network Topology. In OpenAirMesh, the network is organized in clusters, where nodes can either take the role of a cluster-heads (CHs) or a mesh router (MR). CHs are typically the best-connected nodes in a particular geographical area and manage radio resources within the cluster. MRs are used to relay information between CHs. An example of a mesh architecture with 5 nodes is shown in Figure 2.

3.1.1. Cluster Head. The primary role of the CH is to manage radio resources in their cluster, much as a base-station would do in a cellular network. The cluster is defined as the set of nodes which are characterized by one-hop connectivity with the CH. The system is designed as a TDD system, where CHs and MRs transmit in alternating transmission time intervals. Thus—due to the half duplex constraint—direct CH \leftrightarrow CH communication is not supported. The downlink (CH \rightarrow MR) signaling channels allow for the CH to schedule transmission of labels (in the form of time and frequency mappings on the radio resource) each of which carry different types of traffic throughout the mesh network according to pre-defined quality-of-service (QoS) descriptors. The Uplink (UL) signaling channels (MR \rightarrow

CH) are used for relaying bandwidth requirement indicators and channel quality measurements from nodes within the cluster. These feed the scheduling algorithms residing in the CH and allow for proper resource allocation satisfying QoS negotiations carried out using Layer 3 (L3) signaling. The latter are beyond the scope of the description in this paper.

3.1.2. Mesh Router. The primary role of an MR is to interpret the scheduling information from the CH on the downlink (DL) signaling channels in order to route the traffic corresponding to the scheduled labels on the allocated physical resources. MRs can also be connected to more than one cluster at the same time. Since all CHs transmit on the same time-frequency resources, MRs must be able to cancel interference; see Section 4.1 for details.

3.2. Layer 2 Protocol Stack. The OpenAirMesh Layer 2 protocol stack is depicted in Figure 3 and comprises:

- (i) A IP/MPLS networking device (nonaccess stratum (NAS) driver) responsible for provision of IP/MPLS layer services to Layer 2 and vice-versa
- (ii) A Radio resource control (RRC) entity responsible for MAC layer signaling for configuration of logical channels and retrieval of measurement information.
- (iii) A Radio Link Control (RLC) entity which is responsible for hybrid automatic repeat request (HARQ) protocols and IP packet segmentation
- (iv) A convergence protocol (PDCP) responsible for IP interface and related functions (header compression, ciphering, etc.)
- (v) A scheduling and multiplexing unit to control the media access (MAC).

The information flow is organized into different traffic queues.

- (i) Radio bearers are the user-plane traffic queues at the PDCP-RLC interface
- (ii) Signaling radio bearers are the control-plane traffic queues at the RRC-RLC interface
- (iii) Logical channels are the traffic queues at the RLC-MAC interface (both control and user-plane data, see Table 3)
- (iv) Transport channels are the traffic queues at the MAC-PHY interface which are mapped to physical channels by PHY (see Table 4).

The MAC layer scheduling and multiplexing entity is responsible for scheduling control plane and user plane traffic (logical channels) on the physical OFDMA resources (transport channels). It is important to note that although dedicated resources are configured at the input of the MAC layer, the physical resources allocated in the scheduling entities (with exception of the CHBCH) are dynamically allocated every CH transmission time interval (TTI) and thus all physical resources are shared. The BCCH is multiplexed

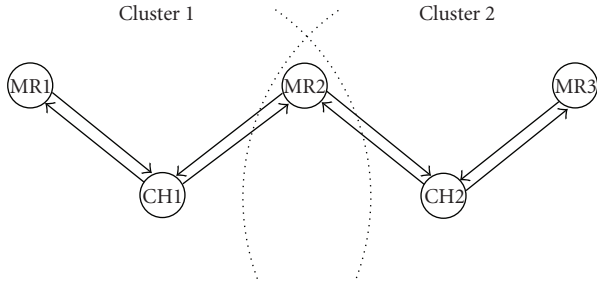


FIGURE 2: The mesh network topology is organized in clusters. Each cluster is controlled by a cluster head (CH). Other nodes in the network are called mesh routers (MR) since they can be used to relay information between CHs.

TABLE 3: Logical channels.

Broadcast Control Channel	BCCH	Low bit-rate control channel used by a CH for broadcasting basic information to nodes in the cluster.
Common Control Channel	CCCH	Low bit-rate control channel used both during the attachment or association phase of a new node.
Dedicated Control Channel	DCCH	Access-layer signaling information (RLC return channels, RF measurement reporting, traffic measurement reporting, power control, etc.) to the corresponding node.
Dedicated Traffic Channel	DTCH	Variable bit-rate traffic channel with negotiated QoS parameters used by the mesh network to transport data traffic corresponding to a particular flow.

in the scheduling entity responsible for generation of the CH-BCH transport channel along with MAC-layer signaling. MAC signaling concerns both allocations of CH-SACH in the current frame and MR-SACH in the next frame (uplink, downlink and direct link map of PHY resources). The CCCH (uplink) is used exclusively during the attachment phase of the MR with a particular cluster and corresponds to the only random-access resources allocated by the CH in the frame.

The DCCHs are multiplexed along with user-plane traffic DTCHs on the available CH-SACH resources. Based on measurement and feedback information, SACH scheduling (see Figure 4) aims to respect the negotiated QoS of each logical channel, while maximizing the aggregate spectral efficiency of the data streams. Different wideband scheduling policies taking into account both queuing measures from RLC and channel quality feedback can be accommodated (see, e.g., [20]). Channel quality information is signaled between corresponding MAC-layers based quantized wideband channel estimates received from PHY.

TABLE 4: Transport and physical channels. Each of the transport channels is mapped to a corresponding physical channel of the same name.

CH Broadcast Channel	CH-BCH	Broadcast control channel which houses MAC-layer signaling for CH and MR physical resource scheduling as well as layer 2 radio-resource control (RRC) signaling for topology and QoS management.
CH Scheduled Access Channel	CH-SACH	Data channel (for both control and user-plane logical channels) used by CH to communicate with a node in its cluster.
MR Broadcast Channel	MR-BCH	Broadcast resource used by MR to extend the coverage of a cluster during topological discovery.
MR Scheduled Access Channel	MR-SACH	Data channel (for both control and user-plane logical channels) used by MR to communicate with a CH.
Random Access Channel	RACH	Signaling channel used by a MR during the association phase with the CH.

TABLE 5: Additional physical channels.

Physical CH Synchronization Channel	CHSCH	Pilot resource reserved to a CH which is responsible for delivering synchronization information to nodes in the cluster.
Physical Synchronization Channel	MRSCH	Pilot resource used by a MR to allow the CH to estimate the channel of an MR.

TABLE 6: OFDMA Parameters in OpenAirMesh.

Symbol (DFT/IDFT) size	256 samples
Prefix length	64 samples
Useful carriers	160
Number of subbands (chunks)	16
Data carriers per subband	8
Pilots per subband	2
OFDM symbols/frame	64

3.3. *Physical Layer.* The physical layer of the platform uses orthogonal frequency division multiple access (OFDMA) together with multiple-input multiple-output (MIMO) techniques and is similar to that of fourth generation wireless networks such as WiMAX or LTE. The parameters for OpenAirMesh are given in Table 6.

The MIMO-OFDMA system provides the means for transmitting several multiple-bitrate streams (multiplexed over sub-carriers and antennas) in parallel. Moreover, PHY

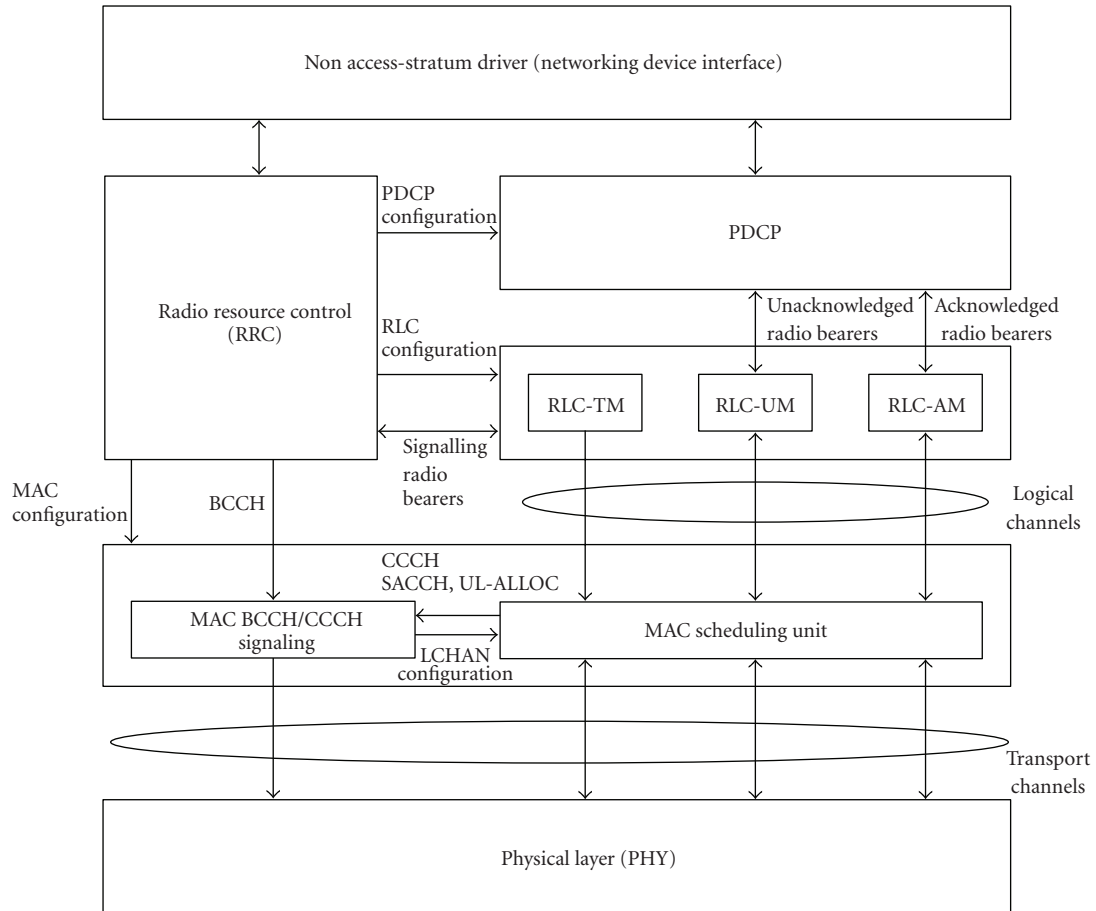


FIGURE 3: OpenAirMesh CH Layer 2 Protocol Stack.

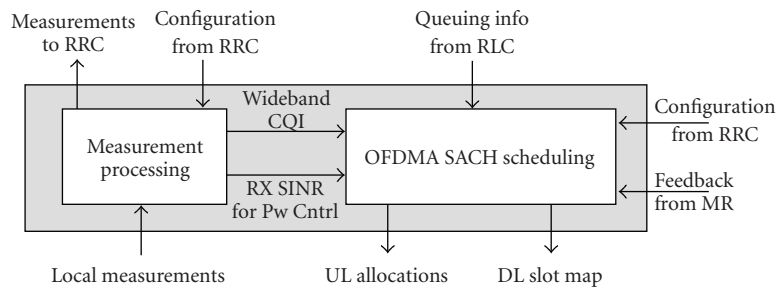


FIGURE 4: Overview of the MAC Scheduler.

signaling strategies are included to provide the means for exploiting channel state feedback at the transmitters in order to allow for advanced PHY allocation of OFDMA resources via the MAC.

In addition to the physical channels of Table 4, there are two synchronization channels (see Table 5) which are used for parameter estimation. In the following we describe the framing and channel multiplexing as well as the coding and modulation scheme.

3.3.1. Framing and Channel Multiplexing. The physical resources are organized in frames of OFDM symbols. One

frame consists of 64 OFDM symbols and is divided equally in a CH transmission time interval (TTI) and a MR TTI (see Figure 5). The first four symbols of the CH TTI are reserved for pilot symbols. Each CH transmits one common pilot symbol (CHSCH₀) at position 0 and one dedicated pilot symbol (CHSCH_{*i*}) at position $i \in \{1, 2, 3\}$. This way, we can ensure orthogonality between the pilots of different CHs received at one MR. The pilot symbols are followed by the broadcast channel (CH-BCH), which contains 128 data subcarriers and 32 pilot subcarriers. The remaining 20 OFDM symbols of the CH TTI frame are divided into 16 resource blocks (RB), which constitute the multiplexed

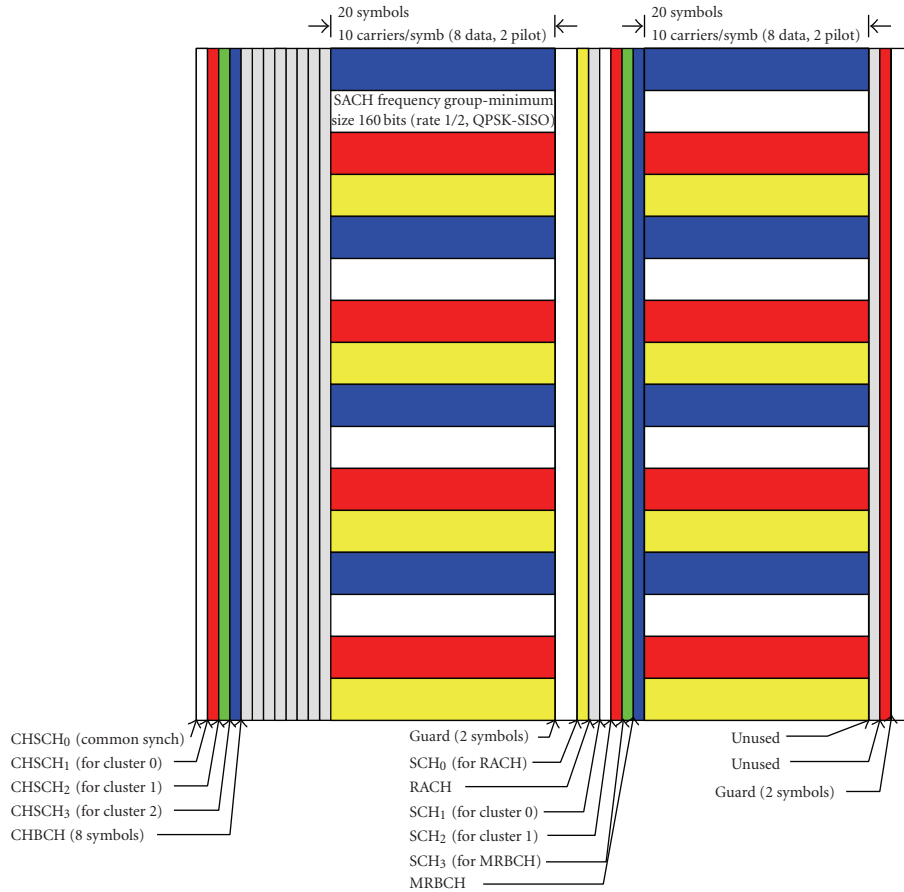


FIGURE 5: OpenAirMesh frame structure.

scheduled access channels (CH-SACH). Each RB contains 8 subcarriers for data and 2 pilot subcarriers (one for each CH), which are used for frequency offset compensation.

The MR TTI contains the random access channel (MR-RACH) with an associated pilot symbol (SCH₀). The next two symbols are reserved for pilots. Each MR transmits a pilot symbol SCH_{*i*}, $i \in \{1,2\}$ corresponding to the cluster it belongs to. This way we can ensure orthogonality between the pilots of different CHs. The pilot symbols are followed by the uplink broadcast channel (MR-BCH) with an associated pilot symbol (MRSCH). The rest of the uplink frame contains the multiplexed scheduled access channels (MR-SACH). The end of the CH and MR TTIs are protected by a guard interval of two symbols. All pilots are designed for MIMO and/or multiuser channel estimation at the corresponding end.

3.3.2. Coding and Modulation. OpenAirMesh makes use of punctured binary codes (64-state rate 1/2 convolutional or 8-state rate 1/3 3GPP/LTE Turbo code). Puncturing can use either 3GPP rate matching or random puncturing in order to fine tune the coding rate to adapt to configurable transport block sizes delivered to PHY by the MAC. The overall coding sub-system is shown in Figure 6. New transport blocks arriving from the MAC layer (based on multiuser scheduling) are coded using a CRC extension and the chosen

binary code. These are then fed to the active transport block buffer along with those that are to be retransmitted. Each transmitted block is punctured and then passed to a bit-interleaver and modulation mapper (BICM). OpenAirMesh supports QPSK, 16-QAM and 64-QAM modulation. The transmitted transport blocks can be split into two spatial streams in the case of point-to-point MIMO transmission.

The modulated symbols are then multiplied by an adjustable amplitude and passed to the space-time-frequency (STF) parser. The STF parser multiplexes the pilot symbols and the data symbols into OFDM symbols, taking into account the sub-band allocation from the scheduler. In the case of one available spatial stream the STF parser also performs fast antenna cycling, that is, every subcarrier is transmitted from a different antenna. This way each stream can access all the degrees of freedom of the channel. In the case of two spatial streams the STF parser guarantees that both streams use different antennas in the same time/frequency dimension. This is a form of superposition coding since the two streams are combined additively in the air through the use of multiple transmit antennas. Last but not least the symbols are transformed to the time domain using an IFFT and a cyclic prefix is appended.

This design allows to use the same transmitter and receiver structure both for point-to-point MIMO as well

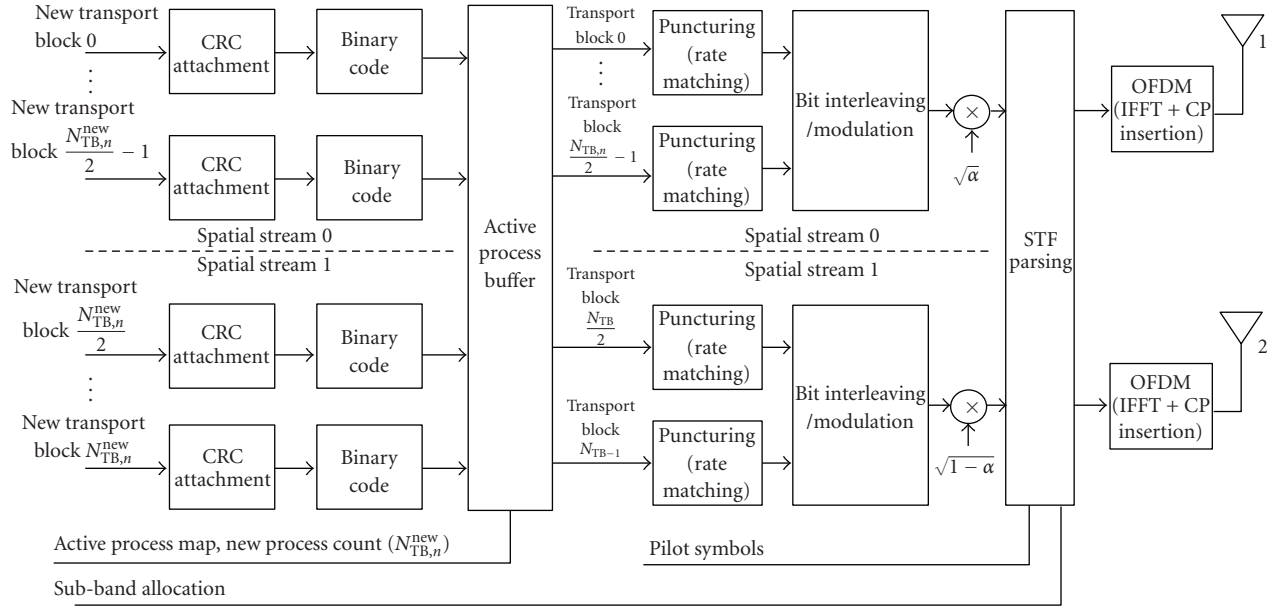


FIGURE 6: OpenAirMesh Coded Modulation and HARQ.

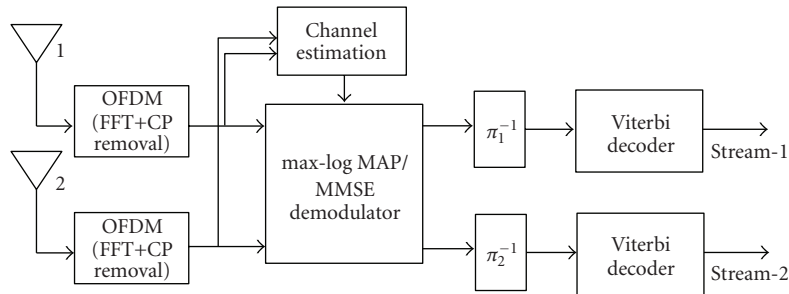


FIGURE 7: Dual-stream receiver structure.

as distributed MIMO transmission. In the latter case one spatial stream is used at each source and the second stream originates in another part of the network, either in the same cluster or an adjacent cluster. A particular user can decode both streams or simply select the one it requires. In Section 4.1 we derive a low-complexity successive interference cancellation (SIC) receiver for this design.

4. Implementation of OpenAirMesh

In this section, we show how OpenAirMesh as specified in Section 3 can be implemented as a single-frequency network. The solution makes use of a distributed network synchronization procedure and a dual-stream MIMO interference cancellation receiver. In this section we describe these novel solutions and their implementation on OpenAirInterface. We present results from simulations and field trials in Section 5.

The implementation is based on the CBMIMO1 hardware (cf. Section 2.1) and is thus restricted to two antennas. Therefore we can process up to two spatial streams coming

from two different CHs. The extension of the receiver structure to more antennas can be found in [21].

4.1. Dual-Stream MIMO Receiver Architecture. In this section, after the general overview of the receiver structure in Section 4.1.1, we describe two different dual-stream multiantenna demodulators, namely a linear minimum mean squared error (MMSE) receiver (see Section 4.1.3) and an approximate maximum likelihood receiver (see Section 4.1.4) [22–24]. The derivations are based on the signal model presented in Section 4.1.2.

4.1.1. Receiver Architecture. The overall receiver structure is shown in Figure 7. After the CP removal and the FFT, a channel estimation based on the least squares algorithm is performed for CH i using the corresponding synchronization symbol (CHSCH $_i$). Further, the frequency offset is estimated using the pilot subcarriers in the CHBCH. The main block of the receiver is either the spatial MMSE filter or the reduced complexity max-log MAP receiver described in the following subsections. Finally we perform inverse bit interleaving and Viterbi decoding.

4.1.2. Signal Model. Consider the scenario depicted in Figure 2 with two clusterheads but only one MR (MR2). We assume that each CH has n_t antennas and MR2 has n_r antennas. Let $\mathbf{x}_{m,q}^{(j)}$ denote $n_t \times 1$ vector of the transmit symbols for subcarrier q of OFDM symbol m of CH j , $j = 1, 2$. We assume that the transmit symbols are taken from a signal set $\chi_j \subseteq \mathbb{C}$ of size $|\chi_j| = M_j$ with a Gray labeling map $\mu_j : \{0, 1\}^{\log_2 M_j} \rightarrow \chi_j$.

Cascading the IFFT and the CP extension at the CHs and the FFT and the CP removal at MR2, the received signal at MR2 at q th frequency tone and the m th OFDM symbol can be expressed as

$$\mathbf{y}_{m,q} = \mathbf{H}_q^{(1)} e^{2\pi j \phi_1 m} \mathbf{x}_{m,q}^{(1)} + \mathbf{H}_q^{(2)} e^{2\pi j \phi_2 m} \mathbf{x}_{m,q}^{(2)} + \mathbf{z}_{m,q}, \quad (1)$$

where $\mathbf{H}_q^{(1)}$ and $\mathbf{H}_q^{(2)}$ denote the $n_r \times n_t$ MIMO channel between CH1 and MR2 and between CH2 and MR2. The channel is assumed to be frequency selective (i.e., it varies with subcarrier index q) and block fading (i.e., constant over the OFDM symbols of a frame). ϕ_1 is the frequency offset between CH1 and MR2 and ϕ_2 is the one between CH2 and MR2 in radians (to convert them to Hertz, multiply by the OFDM symbol rate). $\mathbf{z}_{m,q} \in \mathbb{C}^{n_r}$ is the vector of circularly symmetric complex white Gaussian noise of variance σ^2 .

Since each clusterhead transmits only one spatial stream and antenna cycling is used, only one element of $\mathbf{x}_{m,q}^{(j)}$, $j = 1, 2$ is nonzero for every m and q . We identify this nonzero element with $x_{m,q}^{(j)}$, $j = 1, 2$ and can rewrite (1) equivalently as

$$\mathbf{y}_{m,q} = \mathbf{h}_q^{(1)} e^{2\pi j \phi_1 m} x_{m,q}^{(1)} + \mathbf{h}_q^{(2)} e^{2\pi j \phi_2 m} x_{m,q}^{(2)} + \mathbf{z}_{m,q}, \quad (2)$$

where $\mathbf{h}_q^{(1)}$ and $\mathbf{h}_q^{(2)}$ are the equivalent channel vectors for the nonzero elements. The complex symbols $x_{m,q}^{(1)}$, $x_{m,q}^{(2)}$ of the 2 streams are assumed to be independent and of variances σ_1^2 and σ_2^2 respectively. Assuming that the first stream is the desired stream, the signal to noise ratio (SNR) is given by σ_1^2/σ^2 and the signal to interference ratio (SIR) by σ_1^2/σ_2^2 .

4.1.3. MMSE Receiver. Linear spatial filters such as minimum mean square error (MMSE) and zero forcing (ZF) filters can be used to minimize the level of interference in the former case while nulling out the interference in the latter case. Linear MMSE filters exhibit better performance compared to ZF ones and are thus being considered as favorable candidates for future wireless systems [25, 26]. However, it is well known that MMSE detection for non Gaussian alphabets in low dimensional systems (low number of interferers) is sub-optimal [27] and moreover MMSE detection cannot exploit the interference structure.

The frequency domain MMSE filter \mathbf{M}_q is given as

$$\mathbf{M}_q = \left(\sigma^2 \mathbf{P}^{-1} + \mathbf{H}_q^H \mathbf{H}_q \right)^{-1} \mathbf{H}_q^H, \quad (3)$$

where \mathbf{P} is the diagonal power distribution matrix with the diagonal as $[\sigma_1^2, \sigma_2^2]$ and $\mathbf{H}_q = [\mathbf{h}_q^{(1)} \quad \mathbf{h}_q^{(2)}]$.

The estimates of the transmitted symbols $\hat{\mathbf{x}}_{m,q} = [\hat{x}_{m,q}^{(1)}, \hat{x}_{m,q}^{(2)}]^T$ are computed in three steps. Firstly, the frequency offset needs to be compensated by computing $\mathbf{y}'_{m,q} =$

$e^{-2\pi j \phi_1 m} \mathbf{y}_{m,q}$. Secondly the spatial filter \mathbf{M}_q is applied to $\mathbf{y}'_{m,q}$ by computing $\tilde{\mathbf{x}}_{m,q} = \mathbf{M}_q \mathbf{y}'_{m,q}$. Finally an unbiased operation is performed by computing $\hat{\mathbf{x}}_{m,q} = \Gamma_q^{-1} \tilde{\mathbf{x}}_{m,q}$ where $\Gamma_q = \text{diag}(\mathbf{M}_q \mathbf{H}_q)$.

Post detection interference is assumed to be Gaussian which on one hand reduces the computational complexity but on the other adds to the sub-optimality of MMSE detection. MMSE pre-processing decouples the spatial streams and the bit metric for the i -th bit for bit value b of the symbol $x_{m,q}^{(k)}$ on k th stream is given as

$$\lambda_k^i(\mathbf{y}_{m,q}, b) \approx \max_{x_{m,q}^{(k)} \in \chi_{k,b}^{(k)}} \left[-\frac{\gamma_k^2}{N_0} \left| \hat{x}_{m,q}^{(k)} - x_{m,q}^{(k)} \right|^2 \right] \quad (4)$$

for $k = 1, 2$, where γ_k is the i th diagonal element of $\Gamma_q \cdot \chi_{k,b}^i$ denotes the subset of the signal set $\chi_{m,q}^{(k)} \in \chi_k$ whose labels have the value $b \in \{0, 1\}$ in the position i . Based on these bit metrics, bit log likelihood ratios (LLRs) are calculated which after de-interleaving are passed to the channel decoder.

Implementation. The core of the MMSE receiver is the matrix inversion that is needed to calculate the filter

$$\mathbf{M}_q = \underbrace{\left(\sigma^2 \mathbf{P}^{-1} + \mathbf{H}_q^H \mathbf{H}_q \right)^{-1}}_{=: \mathbf{A}_q} \mathbf{H}_q^H. \quad (5)$$

\mathbf{M} needs to be calculated for every subcarrier and every frame, but we drop the indices for notational convenience.

Since we are limited to a 2×2 MIMO system, the matrix inversion can be calculated directly using Cramer's rule $\mathbf{A}_q^{-1} = (1/\det(\mathbf{A}_q)) \text{adj}(\mathbf{A}_q)$, where $\text{adj}(\mathbf{A}_q)$ denotes the adjoint of the matrix \mathbf{A}_q . Care has to be taken to properly scale the intermediate results in a fixed point implementation. The entries of the channel matrices \mathbf{H}_q are stored in signed 16 bit-wide variables, but their resolution is limited to 14 bit due to the A/D converters. Since the calculation of the determinant $\det(\mathbf{A}_q)$ involves terms up to the fourth power, the dynamic range of the determinant can reach up to 48 bit. In order to handle this high dynamic range we first use a 64 bit-wide variable to calculate $\det(\mathbf{A}_q)$ thus not losing any accuracy. This intermediate result is then shifted such that $\max \det(\mathbf{A}_q)$ uses 16 bits.

In order to calculate the inverse of $\det(\mathbf{A}_q)$, we interpret all numbers as fractional Q15 numbers and use standard fixed-point arithmetic. Intermediate results are stored in double precision. The inverse is scaled back by the mean (over all subcarriers) of $\det(\mathbf{A}_q)$ and saturating to 16 bit. Finally, the MMSE filter matrix is calculated according to $\mathbf{M}_q = (1/\det(\mathbf{A}_q)) \text{adj}(\mathbf{A}_q) \mathbf{H}_q^H$, scaling the intermediate results always to 16 bits.

The high dynamic range of the determinant can cause severe problems. Especially in a frequency selective channel its inverse may saturate on some carriers and can be zero on some other frequencies. This is one of the reasons why the MMSE receiver has a worse performance than the max-log MAP detector described in the next subsection.

4.1.4. Low Complexity max-log MAP Detector. This detector is a low complexity version of max-log MAP detector and is based on the matched filter outputs [21]. Its low complexity is based on the reduction of one complex dimension. Instead of attenuating the interference this detector exploits its structure and mitigates its effect. Without loss of generality, consider the first stream being the desired stream.

Contrary to the MMSE detection we do not compensate the frequency offsets in the received signal, but instead we integrate them in the channel estimates. Therefore define $\mathbf{h}_{m,q}^{(k)} = \mathbf{h}_q^{(k)} e^{2\pi j \phi_k m}$, $k = 1, 2$. For clarity we omit the subindices m, q in the following derivation. The max-log MAP bit metric for bit b of the desired stream x_1 is given as [28]

$$\begin{aligned} \lambda_1^i(\mathbf{y}, b) &\approx \min_{x_1 \in \mathcal{X}_{1,b}^i, x_2 \in \mathcal{X}_2} \|\mathbf{y} - \mathbf{h}_1 x_1 - \mathbf{h}_2 x_2\|^2 \\ &= \min_{x_1 \in \mathcal{X}_{1,b}^i, x_2 \in \mathcal{X}_2} \left\{ \|\mathbf{y}\|^2 + \|\mathbf{h}_1 x_1\|^2 - (2y_1 x_1^*)_R \right. \\ &\quad \left. + (2p_{12} x_1^* x_2)_R - (2y_2 x_2^*)_R + \|\mathbf{h}_2 x_2\|^2 \right\}, \end{aligned} \quad (6)$$

where $y_1 = \mathbf{h}_1^H \mathbf{y}$ be the matched filter output for the first stream and $p_{12} = \mathbf{h}_1^H \mathbf{h}_2$ be the cross correlation between the first and the second channel. Note that subscripts $(\cdot)_R$ indicates the real part. Writing terms in their real and imaginary parts, we have

$$\begin{aligned} \lambda_1^i(\mathbf{y}, b) &\approx \min_{x_1 \in \mathcal{X}_{1,b}^i, x_2 \in \mathcal{X}_2} \left\{ \|\mathbf{h}_1 x_1\|^2 - (2y_1 x_1^*)_R \right. \\ &\quad \left. + (2(p_{12,R} x_{1,R} + p_{12,I} x_{1,I}) - 2y_{2,R}) x_{2,R} \right. \\ &\quad \left. + \|\mathbf{h}_2\|^2 x_{2,R}^2 + (2(p_{12,R} x_{1,I} - p_{12,I} x_{1,R}) \right. \\ &\quad \left. - 2y_{2,I}) x_{2,I} + \|\mathbf{h}_2\|^2 x_{2,I}^2 \right\}, \end{aligned} \quad (7)$$

where $(\cdot)_I$ indicates imaginary part. For x_2 belonging to the equal energy alphabets (such as QPSK), the values of $x_{2,R}$ and $x_{2,I}$ which minimize (7) need to be in the opposite directions of $2(p_{12,R} x_{1,R} + p_{12,I} x_{1,I}) - 2y_{2,R}$ and $2(p_{12,R} x_{1,I} - p_{12,I} x_{1,R}) - 2y_{2,I}$ respectively thereby evading search on alphabets of x_2 and reducing one complex dimension of the system. The bit metric is therefore written as

$$\begin{aligned} \lambda_1^i(\mathbf{y}, b) &\approx \min_{x_1 \in \mathcal{X}_{1,b}^i, x_2 \in \mathcal{X}_2} \left\{ \|\mathbf{h}_1 x_1\|^2 - (2y_1 x_1^*)_R \right. \\ &\quad \left. - |(2(p_{12,R} x_{1,R} + p_{12,I} x_{1,I}) - 2y_{2,R})| |x_{2,R}| \right. \\ &\quad \left. - |(2(p_{12,R} x_{1,I} - p_{12,I} x_{1,R}) - 2y_{2,I})| |x_{2,I}| \right\}. \end{aligned} \quad (8)$$

For non equal energy alphabets (such as 16-QAM), it is the minimization problem of a quadratic function again

trimming one complex dimension of the system. In that case, the real and imaginary parts of x_2 which minimizes (6) are given as

$$\begin{aligned} x_{2,R} &\rightarrow - \frac{(p_{12,R} x_{1,R} + p_{12,I} x_{1,I}) - y_{2,R}}{\|\mathbf{h}_2\|^2}, \\ x_{2,I} &\rightarrow - \frac{(p_{12,R} x_{1,I} - p_{12,I} x_{1,R}) - y_{2,I}}{\|\mathbf{h}_2\|^2}, \end{aligned} \quad (9)$$

where \rightarrow indicates the quantization process in which amongst the finite available points, the point closest to the calculated continuous value is selected.

The reduced complexity max-log MAP detector has a much lower complexity than the MMSE receiver [22]. Furthermore, it can be implemented without any division and therefore it is numerically more stable than the MMSE receiver.

4.2. Network Synchronization. The dual-stream MIMO receiver described above requires timing and frequency synchronization. It has to be assured that the transmit frames are aligned and that the carrier frequency offsets between different nodes are small. The accuracy of the timing, that is, the time difference between signals coming from the two different CHs, has to be smaller than the CP length of the OFDM system. Although carrier frequency offsets are compensated in the receiver, large frequency offsets cause intercarrier interference and thus degrade the performance of the receiver. We will evaluate the maximum allowable frequency offset in Section 5 by simulation.

Timing synchronization can be achieved by using high accuracy reference clocks, such as rubidium oscillators or global positioning system (GPS) receiver. However, the rubidium oscillators are very expensive (in comparison with other components of the receiver) and very large. GPS receivers on the other hand are not able to operate indoors. Therefore we will take a distributed network synchronization approach.

In nature, distributed synchronization scheme can be observed on the flashing of fireflies [29]. Recently, this nature-inspired scheme has been applied to synchronization in wireless networks [30–33]. However, most of these works consider the isolated synchronization problem and neglect the actual data communication. The pulse-coupled oscillator model (the model inspired by firefly synchronization) assumes that nodes have to be listening to all other nodes except during its own transmission of the synchronization pulse and immediately afterwards (refractory period). Therefore data transmission can only take place in the refractory period. However, this period must not be very long because otherwise the system becomes unstable [32].

In OpenAirMesh, we follow a similar approach as in [3] for distributed timing synchronization. It is based on the two physical channels CHSCH and MRSCH (see Table 5) which are transmitted in alternating TTIs from the CHs and MRs respectively. Initially we declare one CH to be the primary CH, which is the reference clock in the system. The primary CH continuously sends out a synchronization

signal (the CHSCH) that allows every MR within the CH's broadcast region to synchronize to the network. As soon as a MR is synchronized (i.e., when it can detect the CHBCH successfully), it sends out a synchronization signal itself (the MRSCH). A secondary CH not within the broadcast domain of the primary CH can use the MRSCH to synchronize to the network. As soon as a secondary CH is synchronized to the network, it also sends out a CHSCH, allowing further MRs to synchronize, and so on. A positive side-effect of this method is that several MRs form a distributed antenna array when sending out the MRSCH. This means that the secondary CH can benefit from this array gain when detecting the MRSCH.

For the carrier phase synchronization, we use off-line calibration prior to the system deployment. However, the granularity of the calibration on the CBMIMO1 cards is in the order of 500 Hz, causing residual frequency offsets.

4.3. Integration. The dual-stream MIMO receiver and the distributed network synchronization procedure described above enables the implementation of a single-frequency mesh network. Since all CHs are synchronized and transmit on the same frequency, MRs that are in the broadcast domain of two such clusters must use the dual-stream MIMO receiver to decode the CHBCHs of both CHs concurrently. But the receiver can also be used for the SACH (both in the uplink and in the downlink) allowing the CHs to schedule their resources independently (and thus significantly reducing the signalling overhead). For the downlink SACH, the receiver is used exactly the same way as for the CHBCH. On the uplink the MR transmit two independent data streams as described in Section 3.3.2 and the CHs decode only the stream dedicated for them, treating the other one as interference.

5. Experiments and Results

In this section, we investigate the performance of the two dual-stream receiver structures described in the previous section. Firstly, in Section 5.1, we perform computer simulations of the two receiver structures using a simple synthetic channel model. Secondly, in Section 5.2 we present performance results from the real-time implementation on the OpenAirInterface platform. Last but not least, in Section 5.3 we present field trial experiments that were conducted within the CHORIST project (<http://www.chorist.eu/>) close to Barcelona, Spain in February 2009.

All performance comparisons (both for simulation and lab tests) were done using the broadcast channel (BCH) of the primary clusterhead (CH1) with interference from the BCH from the secondary clusterhead (CH2). The BCH uses QPSK modulation and rate 1/2 convolutional code. The block length is 1056 bits, which corresponds to 8 OFDM symbols with 132 data subcarriers each. We use 2 antennas on all nodes.

5.1. Computer Simulations. In the computer simulations we isolate and study the effect of the following phenomena on the performance of the receiver: (i) channel state information

at the receiver (CSIR), (ii) frequency selective fading versus frequency flat fading, (iii) Rayleigh fading versus Ricean fading, (iv) receive antenna correlation, and (v) frequency offsets.

5.1.1. Channel Model. For the simulations the 2×2 MIMO channel matrices $\mathbf{H}_q^{(1)}$ and $\mathbf{H}_q^{(2)}$ are modeled as spatially white and independent. The channel is assumed to be constant during a block and varies independently between blocks. We use both a frequency flat fading model as well as a frequency selective model. In the frequency flat case the channel matrices stay constant over all subcarriers q with channel coefficients drawn from a Rayleigh distribution with unit variance. In the frequency selective case we model the channel as a tapped delay line with 8 sample-spaced taps with an exponential decaying power delay profile. Each tap undergoes Rayleigh fading. If line-of-sight (LOS) is present, the first tap undergoes Ricean fading. Receive correlation is modeled by multiplying (from the left) the MIMO channel matrices with the square root of the receive correlation matrix $R_{Rx} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$.

5.1.2. Simulation Results. The two receiver structures described in Section 4.1 were implemented in fixed-point C. The simulation model follows the model (1) with the difference that the C simulator includes the IFFT and CP insertion at the transmitter and the corresponding FFT and CP removal at the receiver. The channel is thus simulated in the time domain rather than in the frequency domain. Further, we can simulate carrier frequency offsets. By not adding any noise on the pilot symbols we can also simulate the case of perfect CSIR. This allows us to study the impact of imperfect channel estimates on the two receiver structures.

We perform Monte Carlo simulations with the MMSE receiver and the max-log MAP receiver. We fix the SNR for the first stream and vary the interference from the second stream on the y -axis. Each figure shows the frame error rates of the first stream for both receivers and several SNR values. We only show a representative subset of the simulation results.

- (i) Figure 8 shows results for the frequency flat Rayleigh fading channel with perfect CSIR.
- (ii) Figures 9, 10, and 11 show results for the frequency flat and the frequency selective Rayleigh fading channel as well as for a frequency selective Ricean fading channel with a Ricean K-factor of $K = 10$.
- (iii) Figure 12 shows results for a frequency flat Rayleigh fading channel with a receive antenna correlation of $|\rho| = 0.75$.
- (iv) Figure 13 shows results for the frequency selective channel, where CH2 has a frequency offset of 1500 Hz w.r.t. the receiver.

We can make the following observations. First of all the max-log MAP receiver always performs better than the MMSE receiver. Together with the fact that the max-log MAP

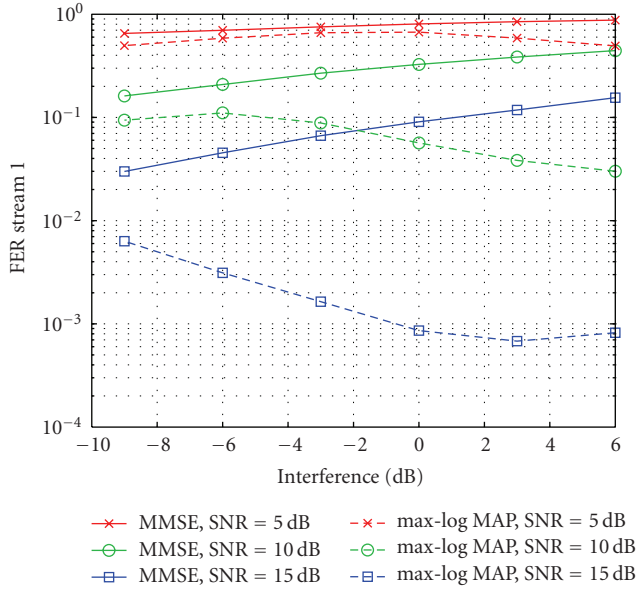


FIGURE 8: Frame error rate of the first stream of the of the MMSE and max-log MAP receiver for a frequency flat Rayleigh fading channel with perfect CSIR. Results are plotted for different SNR levels of the first stream. The x -axis denotes the interference of the second stream w.r.t. the first stream.

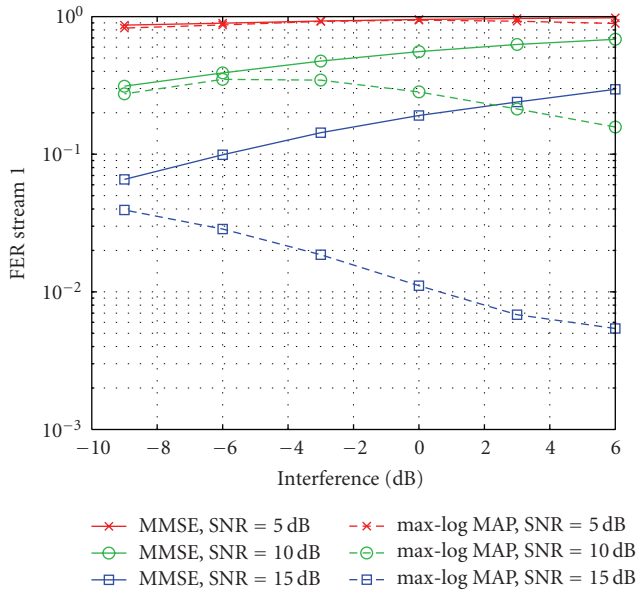


FIGURE 9: Frame error rate of the first stream of the of the MMSE and max-log MAP receiver for a frequency flat Rayleigh fading channel. Results are plotted for different SNR levels of the first stream. The x -axis denotes the interference of the second stream w.r.t. the first stream.

receiver has actually less complexity than the MMSE receiver [22], it is clearly the first choice for such a system. Also note that the performance of the max-log MAP receiver actually gets better when the interference gets stronger. Channel estimation errors have a stronger impact on the max-log MAP receiver than on the MMSE receiver. On the other

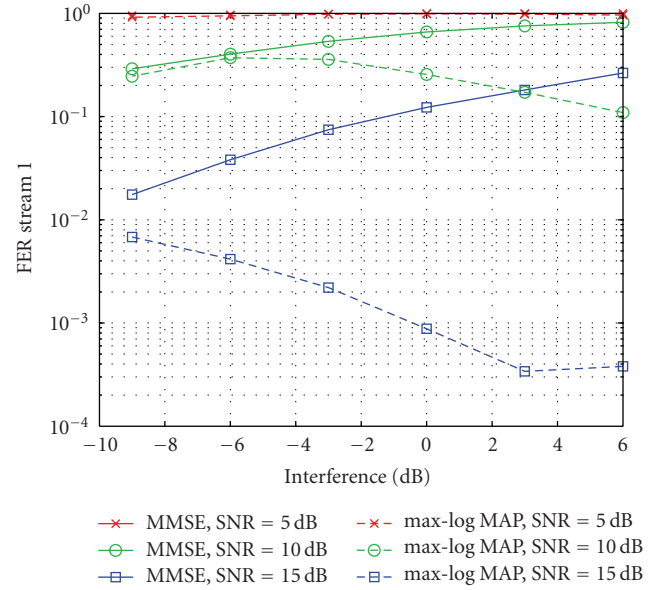


FIGURE 10: Frame error rate of the first stream of the of the MMSE and max-log MAP receiver for a frequency selective Rayleigh fading channel. Results are plotted for different SNR levels of the first stream. The x -axis denotes the interference of the second stream w.r.t. the first stream.

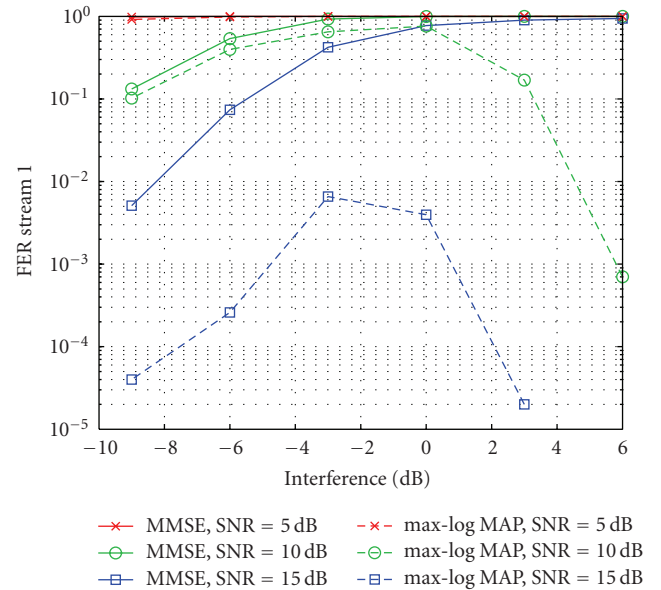


FIGURE 11: Frame error rate of the first stream of the of the MMSE and max-log MAP receiver for a frequency selective Ricean fading channel with a K-factor of $K = 10$. Results are plotted for different SNR levels of the first stream. The x -axis denotes the interference of the second stream w.r.t. the first stream.

hand it can be seen that in a frequency selective channel the max-log MAP receiver profits most from the the additional diversity, while the performance of the MMSE receiver hardly improves. This is due to the fact that the max-log MAP receiver has full diversity gain while MMSE receiver loses one order of diversity [34].

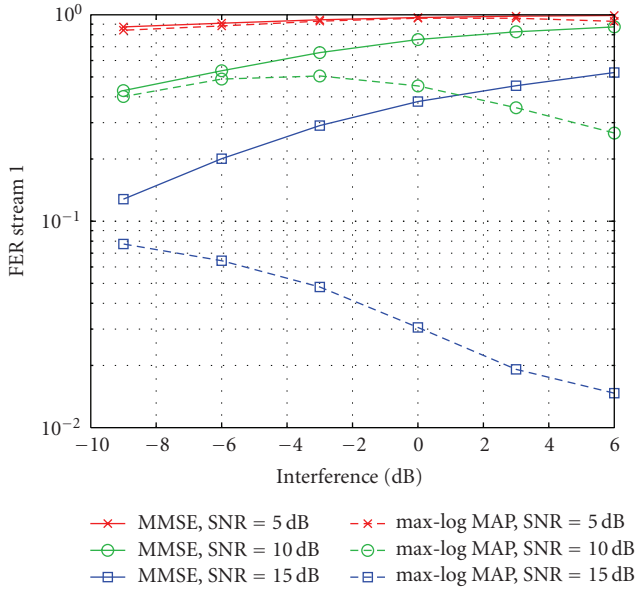


FIGURE 12: Frame error rate of the first stream of the of the MMSE and max-log MAP receiver for a frequency flat Rayleigh fading channel with a receive correlation of $|\rho| = 0.75$. Results are plotted for different SNR levels of the first stream. The x -axis denotes the interference of the second stream w.r.t. the first stream.

In a LOS Ricean fading channel, the performance of the max-log MAP receiver is better than in a Rayleigh fading channel only if the power of the interference is either stronger or weaker than the desired signal. If the powers are similar, a LOS component is not beneficial for the performance. Frequency offsets also have a very strong negative impact on the performance of the system. In fact as the interference gets stronger the max-log MAP receiver is not able to cancel out the interference as good as in the case with no frequency offsets. Last but not least, receive correlation also degrades the performance of the system.

5.2. Lab Tests

5.2.1. Test Setup. The dual stream receiver was tested in the lab using an extended version of the Eurecom MIMO OpenAir Sounder (EMOS) [2]. The EMOS can be seen as a stand-alone version of the physical layer of the OpenAirMesh testbed. Only the synchronization symbols (CHSCH, MRSCH) and the broadcast channels (CHBCH, MRBCH) are transmitted. Instead of the scheduled access channels (SACH), additional pilot symbols are transmitted that can be used for channel sounding purposes, but this functionality is not presented in this paper. Instead we record the frame error rates (based on the CRC check) of the CHBCH. Note that the real-time system uses the same fixed point code for the receiver as the simulator.

For the experiments we set up three nodes (CH1, CH2 and MR2) in our lab. To simulate different SNR levels at the receiver we changed the transmit powers of the two CHs between -20 dBm and 0 dBm.

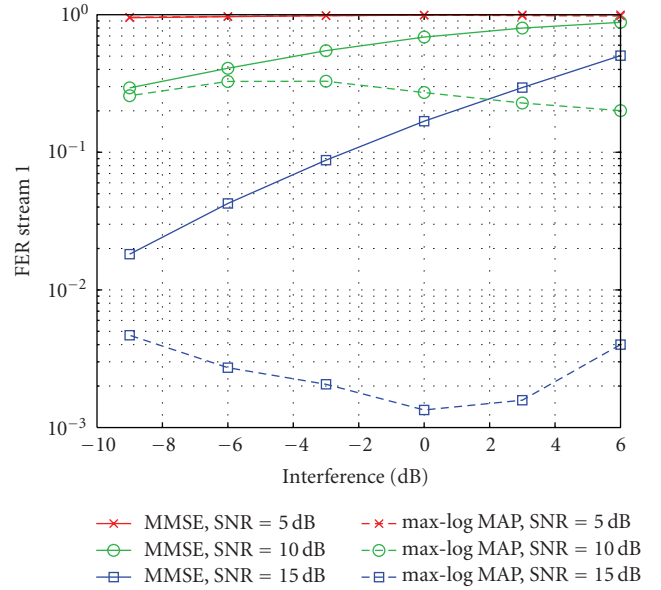


FIGURE 13: Frame error rate of the first stream of the of the MMSE and max-log MAP receiver for a frequency selective Rayleigh fading channel. CH2 has a frequency offset of 1500 Hz w.r.t. the receiver. Results are plotted for different SNR levels of the first stream. The x -axis denotes the interference of the second stream w.r.t. the first stream.

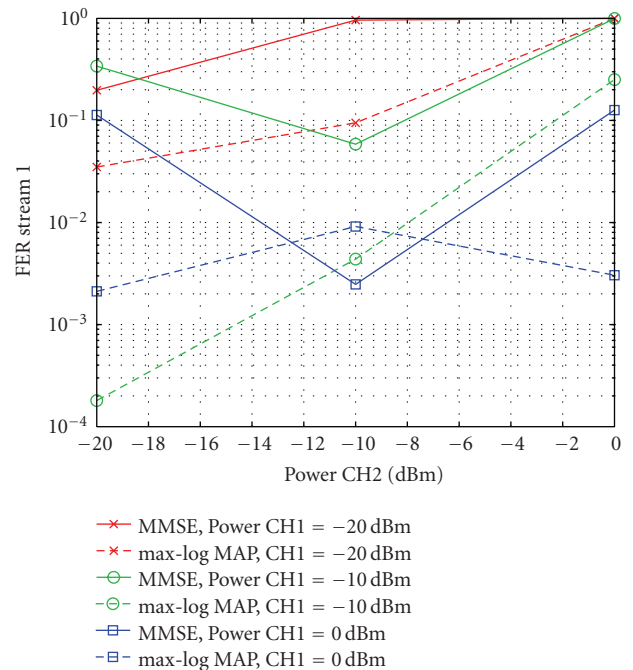


FIGURE 14: Measured frame error rate of the first stream using real-time MMSE and max-log MAP receiver in the Eurecom lab. Results are plotted for different power levels of the two clusterheads.

5.2.2. Results. Figure 14 shows the FER of the first stream for different SNR values for the max-log MAP and the MMSE receiver respectively. It is worth noting that in the test setup we encountered a frequency offset of the second



FIGURE 15: Five node mesh network trial in Bellaterra, Spain. MR1 is located at the roof of the red fire brigade building. CH1 and MR2 are located on opposite sides of the parking lot in front of the building. CH2 and MR3 are on the other side of the building.

stream of around 500 Hz, while there was very little offset on the first stream (less than 10 Hz). This is very likely the reason why the interference cannot be cancelled out that well at a high interference level. Another difference to the simulations is the correlation of the MIMO channel, which was approximately $|\rho| = 0.9$ in the measurements. As we have seen in the simulations this has a very strong impact on the results. Nevertheless, we can still observe similar trends in the measurements as in the simulations, such as that the MMSE receiver has in general a worse performance than the max-log MAP receiver.

5.3. Field Trials. One major application of OpenAirMesh platform is the demonstration of rapidly-deployable broadband ad-hoc communications systems for public safety units in interventions following natural disasters and industrial accidents. Such a demonstration took place in February 2009 in Bellaterra, Spain in the context of the European project CHORIST, which is funded by the 6th framework program of the European Commission.

During the trials we set up a mesh network with five nodes as depicted in Figure 2 on the parking lot of the fire brigade building in Bellaterra (see Figure 15). MR1 was placed on the roof of the building and served as an edge router establishing the connection to the core network and the control room. CH1 and MR2 were placed on the parking lot in front of the building. CH2 and MR3 were placed behind the building, such that there was no connection between CH1 and MR3 as well as between CH2 and MR1. MR2 was in the broadcast domain of both CHs, relaying traffic between them.

Both MR2 and MR3 were used as gateways to other networks. Two different end-to-end applications were tested on the network: a video surveillance application and a push-to-talk VoIP application. During the trials we managed to establish a reliable connection (in the sense that both applications were running smoothly) between MR1 and MR3. See the CHORIST website (<http://www.chorist.eu/>) for more details and a video of the demo.

6. Conclusions

In this paper, we have shown the feasibility of distributed network synchronization and distributed MIMO on the real-time open-source OpenAirInterface platform. We conclude this paper by describing a few lessons we have learned during the implementation and the field trials.

Synchronization is a prerequisite for the dual-stream MIMO receiver described in this paper and other cooperative communication schemes. We have seen that the proposed synchronization is feasible for small scale networks in indoor and medium-range outdoor scenarios. For larger networks, the requirement of a single reference clock is somewhat restrictive, since when it fails the whole network fails. Also it is not proven that the algorithm is stable in larger networks. We are planning to investigate this issue in future works.

As for the implementation of the dual-stream MIMO receiver we have seen that the reduced complexity max-log MAP detector has several advantages over the linear MMSE receiver. First of all its performance is much better (both diversity and coding gain), especially when the interference level is high. Further it can be implemented without any divisions which is very advantageous on a fixed point processor. The implementation of the MMSE receiver on the other hand requires a matrix inversion, which is not trivial using fixed point arithmetic.

During the trials we have also seen that the dual-stream MIMO receiver is very sensitive to channel conditions. The best performance is achieved if the two transmitters have a line of sight to the receiver and if the receive correlation is small. However, positioning the nodes and their antennas in such a way is not trivial. In case of the max-log MAP receiver significant differences in the received powers from the two sources can also improve the performance.

In future work we would like to include distributed space-time coding and collaborative beamforming into OpenAirInterface. This could for example be used in a multiple relay channel, when several relays are placed between two clusterheads. One particular aspect we would like to investigate are the consequences of such scenarios on design aspects related to spatial HARQ and channel coding mechanisms.

Acknowledgments

Eurecom's Research is supported in part by its industrial partners: Swisscom, Thales, SFR, Orange France, ST Ericsson, SAP, Cisco, BMW Group, Monaco Telecom, Symantec. This work is also funded in part by the European Commission through project CHORIST (ICT/FP6), the French National Research Agency (ANR) through projects AIRNET and HNPS, and by the Wiener Wissenschafts-Forschungs-und Technologiefonds (WWTF) through the project PUCCO. Parts of this work were presented at the COST2100 Workshop on MIMO and Cooperative Communications, June 2008, Trondheim, Norway.

References

- [1] H. Anouar, C. Bonnet, D. Câmara, et al., “An overview of OpenAirInterface wireless network emulation methodology,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 2, pp. 90–94, 2008.
- [2] F. Kaltenberger, M. Kountouris, D. Gesbert, and R. Knopp, “On the trade-off between feedback and capacity in measured MU-MIMO channels,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 9, pp. 4866–4875, 2009.
- [3] A. Tyrrell and G. Auer, “Biologically inspired intercellular slot synchronization,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, Article ID 854087, 12 pages, 2009.
- [4] M. Rupp, S. Caban, and C. Mehlführer, “Challenges in building MIMO testbeds,” in *Proceedings of the 15th European Signal Processing Conference (EUSIPCO '07)*, pp. 1833–1839, Poznan, Poland, September 2007.
- [5] J. A. García-Naya, M. González-López, and L. Castedo, “An overview of MIMO testbed technology,” in *Proceedings of the 4th International Symposium on Image/Video Communications over Fixed and Mobile Networks (ISIVC '08)*, Bilbao, Spain, July 2008.
- [6] Ettus Research, “The USRP product family,” 2009, <http://www.ettus.com/>.
- [7] GNU Radio, “The GNU software radio,” 2009, <http://www.gnuradio.org/>.
- [8] Rice University, “WARP project: wireless open-access research platform,” 2009, <http://warp.rice.edu/>.
- [9] K. Mandke, S.-H. Choi, G. Kim, et al., “Early results on hydra: a flexible MAC/PHY multihop testbed,” in *Proceedings of the 65th IEEE Vehicular Technology Conference (VTC '07)*, pp. 1896–1900, Dublin, Ireland, April 2007.
- [10] S. Gupta, C. Hunter, P. Murphy, and A. Sabharwal, “WARP-net: clean slate research on deployed wireless networks,” in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09)*, pp. 331–332, 2009.
- [11] T. Korakis, M. Knox, E. Erkip, and S. Panwar, “Cooperative network implementation using open-source platforms,” *IEEE Communications Magazine*, vol. 47, no. 2, pp. 135–141, 2009.
- [12] S. M. Mishra, D. Cabric, C. Chang, et al., “A real time cognitive radio testbed for physical and link layer experiments,” in *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 562–567, November 2005.
- [13] C. Chang, J. Wawrzyn, and R. W. Brodersen, “BEE2: a high-end reconfigurable computing system,” *IEEE Design & Test of Computers*, vol. 22, no. 2, pp. 114–125, 2005.
- [14] Fraunhofer Institute for Telecommunications (Heinrich-Hertz-Institut), “Berlin LTE-Advanced Testbed,” 2009, <http://www.hhi.fraunhofer.de/en/departments/broadband-mobile-communication-networks/research-topics/berlin-lte-advanced-testbed/>.
- [15] R. Irmer, H.-P. Mayer, A. Weber, et al., “Multisite field trial for LTE and advanced concepts,” *IEEE Communications Magazine*, vol. 47, no. 2, pp. 92–98, 2009.
- [16] Gaisler Research, “Leon3 processor,” 2009, <http://www.gaisler.com/cms/>.
- [17] N.-U.-I. Muhammad, R. Rasheed, R. Pacalet, R. Knopp, and K. Khalfallah, “Flexible baseband architectures for future wireless systems,” in *Proceedings of the 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools (DSD '08)*, pp. 39–46, 2008.
- [18] H. Anouar, C. Bonnet, F. Kaltenberger, and R. Knopp, “OpenAirMesh—an experimental platform for cooperative mesh networks,” in *Proceedings of the 1st COST2100 Workshop on MIMO and Cooperative Communications*, Trondheim, Norway, June 2008.
- [19] R. Knopp, “Detailed report on broadband prototype implementation and updated system specifications,” CHORIST, Project Deliverable SP4.D11, July 2009, <http://www.chorist.eu/>.
- [20] M. Realp, A.I. Pérez-Neira, and R. Knopp, “Delay bounds for resource allocation in wideband wireless systems,” in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, vol. 10, pp. 4385–4390, June 2006.
- [21] R. Ghaffar and R. Knopp, “Interference suppression for next generation wireless systems,” in *Proceedings of the 69th IEEE Vehicular Technology Conference (VTC '09)*, pp. 1–5, Barcelona, Spain, April 2009.
- [22] R. Ghaffar and R. Knopp, “Dual-antenna BICM reception with applications to MIMO broadcast and single frequency cellular system,” in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '08)*, Cannes, France, September 2008.
- [23] R. Ghaffar and R. Knopp, “A MIMO broadcast strategy and interference cancellation in single frequency cellular system,” in *Proceedings of the 1st COST2100 Workshop on MIMO and Cooperative Communications*, Trondheim, Norway, June 2008.
- [24] F. Kaltenberger, R. Ghaffar, and R. Knopp, “Low-complexity distributed MIMO receiver and its implementation on the OpenAirInterface platform,” in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '09)*, Tokyo, Japan, September 2009.
- [25] E. Dahlman, H. Ekström, A. Furuskär, et al., “The 3G long-term evolution—radio interface concepts and performance evaluation,” in *Proceedings of the 63th IEEE Vehicular Technology Conference (VTC '06)*, vol. 1, pp. 137–141, Melbourne, Canada, May 2006.
- [26] I. Medvedev, B. A. Bjerke, R. Walton, J. Ketchum, M. Wallace, and S. Howard, “A comparison of MIMO receiver structures for 802.11N WLAN—performance and complexity,” in *Proceedings of the 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '06)*, pp. 1–5, Helsinki, Finland, September 2006.
- [27] H. Poor and S. Verdú, “Probability of error in MMSE multiuser detection,” *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 858–871, 1997.
- [28] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, 1998.
- [29] J. Buck and E. Buck, “Synchronous fireflies,” *Scientific American*, vol. 234, no. 5, pp. 74–79, 1976.
- [30] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. H. Strogatz, “Distributed synchronization in wireless networks: global synchronization via local connections,” *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 81–97, 2008.
- [31] A. Tyrrell, G. Auer, and C. Bettstetter, “Fireflies as role models for synchronization in ad hoc networks,” in *Proceedings of the 1st Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS '06)*, Cavalese, Italy, 2006.
- [32] Y.-W. Hong and A. Scaglione, “A scalable synchronization protocol for large scale sensor networks and its applications,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 5, pp. 1085–1099, 2005.

- [33] A.-S. Hu and S. D. Servetto, "On the scalability of cooperative time synchronization in pulse-connected networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2725–2748, 2006.
- [34] R. Ghaffar and R. Knopp, "Spatial interference cancellation and pairwise error probability analysis," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–5, Dresden, Germany, June 2009.