Aalborg Universitet

AALBORG UNIVERSITY

DENMARK

Hardware Aspects Of Fixed Relay Station Design For Ofdm(A) Based Wireless Relay Networks

Can, Basak; Portalski, Maciej; Le Moullec, Yannick

HARDWARE ASPECTS OF FIXED RELAY STATION DESIGN FOR OFDM(A) BASED WIRELESS RELAY NETWORKS

Başak Can[1], Maciej Portalski[1], Yannick Le Moullec[1,2]

[1]Department of Electronic Systems [2]Center for Software Defined Radio
Aalborg University, Denmark
Emails: bc@es.aau.dk, mportal1@es.aau.dk, ylm@es.aau.dk

ABSTRACT

Hardware aspects of infrastructured based (i.e., fixed) relay station design with focus on digital baseband processing in OFDM(A) based systems are investigated. A novel architecture for the digital baseband processor for fixed relay stations is proposed to minimize the relay complexity and the implementation cost. Two methods of signal forwarding -Amplify and Forward (AF) and -Decode and Forward (DF) are developed and implemented on FPGA, exploiting parallel processing and a pipelined system architecture. In contrast to common heuristical conclusions existing in the literature, we conclude that the AF based implementation introduces significant hardware complexity resulting from high memory usage. On the other hand, the DF implementation requires higher clocking frequencies, making it more power-consuming as comapred to AF-based relaying.

Index Terms— Fixed relay station, wireless relay network, Implementation, FPGA, Design space exploration

1. INTRODUCTION

Achieving high data rates in future wireless communication systems presents numerous challenges. Issues such as pathloss or multipath fading effects result in the need for enhancing wireless network infrastructure to achieve better performance. The increase in data rate causes a proportional decrease in bit energy. As a result, maintaining ubiquitous system coverage at increased data rates would typically require larger density of Base Stations (BS) as well as increased transmit power level [1].

A promising alternative to such costly solution is to introduce multi-hop transmission either by deploying simple fixed relay nodes or by exploiting subscriber equipment present in the system as intermediate nodes in wireless transmission. The latter solution presents problematic issues such as the need of exploiting complex dynamic routing mechanisms [2], increase in power consumption and hardware complexity of subscriber equipment due to additional signal processing and data transmission needed for the forwarding process [3]. An efficient solution is to integrate multi-hop capability into the infrastructure of wireless communication systems by means of fixed Relay Station (RS) deployment. The benefits of incorporating fixed relays in wireless communications are: coverage enhancement [1], [4], throughput and capacity enhancements [1], [4], power and cost efficiency [1]. Achieving the benefits of infrastructure based wireless multi-hop networking requires efficient and low-cost RS hardware design with consideration of not only performance, but also feasibility, hardware complexity and power consumption metrics. Although numerous theoretical derivations concerning infrastructure based multi-hop wireless networks appear in recent publications [1], [2], [4], very few consider issues related to actual hardware implementations of fixed relay stations [5]. Rare examples of infrastructure based relay network testbeds are either based on low-complexity hardware and simple transmission protocols [5] or exploit IEEE 802.11 protocol implemented on PC based stations [6]. They are, however, not suitable for realizing relay support for Metropolitan Area Networks (MAN) such as IEEE 802.16j. This presents a need for research on fixed RS hardware solutions with focus on high performance and low complexity.

The research described in this paper contributes to this need by presenting feasibility, performance and complexity analysis for efficient RS design. Since future wireless systems are very likely to exploit Orthogonal Frequency Division Multiplexing / Multiple Access (OFDM(A)) to achieve high performance at relatively low cost, this work focuses on baseband processing hardware for OFDM(A) based systems. The design and evaluation presented in this paper can be useful for IEEE 802-16j, 802-16e, and 802-16m based networks.
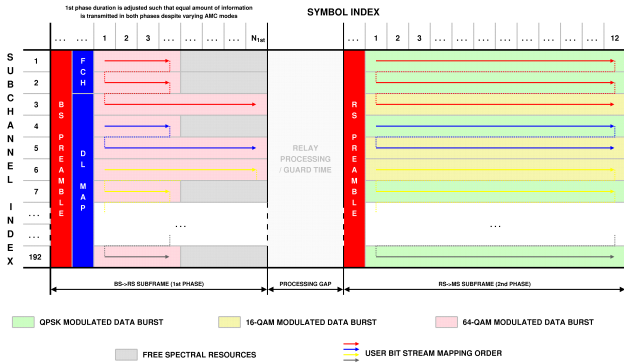
Fig. 1. OFDMA DL frame structure supporting two-hop Time Division Duplexed (TDD) transmission [7]

2. FUNCTIONAL SPECIFICATION AND INITIAL ARCHITECTURE

2.1. Functional specification of the system

Figure 1 presents the OFDMA frame structure considered in this work. The frame consists of blocks such as:

- BS preamble - first symbol in the first subframe used for frame synchronization and initial estimation of BS− >RS channel at the RS

- Frame Control Header (FCH) - fragment containing control information concerning general frame structure such as DL map length and subframe duration

- DL MAP - structure following the FCH containing detailed information concerning subchannel parameters and user scheduling. This includes user< − >subchannel mapping and user burst profiles.

- First phase user data bursts - user bit streams transmitted from BS to RS using a number of subchannels according to the user allocation scheme transmitted in the DL map

- Relay processing gap - minimal time required for signal processing at the RS in order to produce OFDMA symbols for RS− >MS transmission. The gap also includes a guard time needed for radio circuitry to switch from Rx to Tx mode. The influence of utilizing various forwarding methods and digital hardware architectures on the duration of the processing gap is one the main investigations presented in this work.

- Optional RS preamble - first symbol in the second subframe used for frame synchronization and initial estimation of $RS− > MS$ channel at the MS

- Second phase user data bursts - user bit streams transmitted from RS to MS using a number of subchannels according to the user allocation

The following are considered to maintain system efficiency:

- For AF forwarding, the same Modulation and Coding (MCS) modes are used in first and second transmission phase for each subchannel because it is not possible to change the MCS mode between transmission phases without signal decoding and re-encoding. For DF forwarding changing the MCS modes over different phases [7] is possible. This enables to adjust the MCS modes for both transmission phases independently.

- The length of the second subframe is fixed to 12 OFDMA data symbols. For AF forwarding both subframes contain 12 OFDMA symbols due to having same MCS levels in both phases. For DF forwarding the first subframe duration in each subchannel is determined by the MCS chosen for each sub-channel.

- Only one user is allocated to each subchannel. Each user can be allocated to more than one subchannel according to a proper scheduling algorithm. Allocation of spectral resources in the first subframe is based on maintaining the same user< − >subcarrier allocation in both transmission phases. Equal amount of information is transmitted in the first and second phases. For DF forwarding this can result in unused spectral resources in the first phase due to different MCS modes in both transmission phases as shown in Figure 1.

- To draw complexity measures for most demanding cases, we assume that relaying is used over all the frequency band.

The OFDMA based RS nodes require a specialized hardware architecture for baseband signal processing. Such an architecture is proposed in the Subsection 2.2.

2.2. First version of the architecture

The digital part of an OFDM(A) transceiver typically consists of several dedicated signal processing modules. The RS baseband processing module integrates functionalities related to signal reception, processing for forwarding and transmission to the next network node. Figure 2 shows the proposed structure of a digital baseband processing module for OFDM(A) based RS transceivers.

A crucial component of the RS baseband processor is the amplification/regeneration block which is responsible for the processing needed for forwarding. This module can be realized as a multicomponent entity with an internal architecture depending on the exploited forwarding method. For AF-based forwarding, signal processing at the RS is comprised of multiplying consecutive frequency-domain signal values with amplification coefficients calculated from SNR value for each subchannel [7]. Therefore, for AF forwarding a fairly simple amplification architecture as shown in Figure 3 can be used.
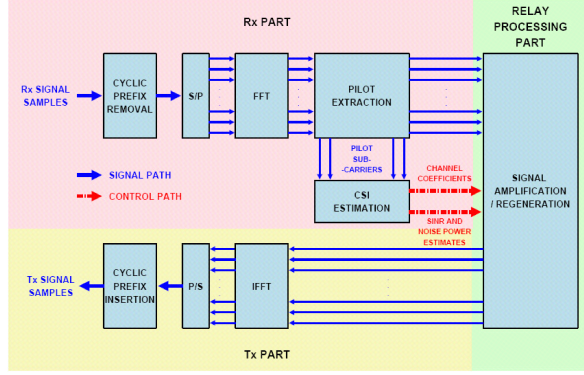
Fig. 2. The proposed digital baseband processing module for OFDM(A) based RS transceivers

For DF forwarding, signal processing is significantly more complex as compared to AF due to the need for complete signal decoding and re-encoding which typically requires implementing components such as ML detector, Viterbi decoder, convolutional codes (CC) encoder and constellation symbol demapper/mapper. The proposed structure is shown in Figure 4.
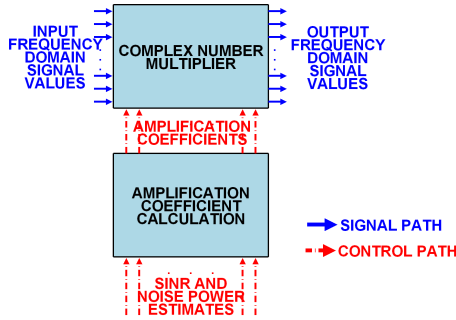

Fig. 3. Proposed architecture of the signal amplification module for AF based relaying

The presented regeneration module architecture enables implementing DF forwarding in OFDM(A) based systems with CC-based Forward Error Correction (FEC). Such a solution can be considered to be costly in terms of hardware resource usage.

3. DESIGN SPACE EXPLORATION

3.1. Hardware Platform Selection

Qualitative metrics and criteria considered in selecting the hardware platform for the design include the following:

- Priority of achieving high system performance for real-time operation

- Since fixed RS is considered, low priority is given to metrics such as chip area and power consumption
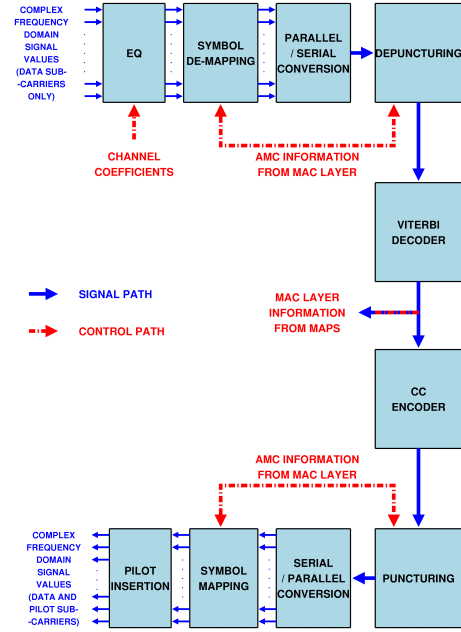

Fig. 4. Proposed architecture of the signal regeneration module for DF based relaying

- Platform support for component-based design and rapid system prototyping

- Availability of OFDM(A) related Intellectual Property (IP) components supported by the platform

Based on these criteria, Field Programmable Gate Array (FPGA) technology has been selected as the hardware platform for further development.

3.2. Pipelined System Architecture

With FPGA as the hardware platform and the RS modular baseband processor structure presented in 2, an efficient pipelined system implementation is possible. Figure 5 presents a pipelined symbol-after-symbol processing scheme with the two-hop OFDM(A) frame proposed in Section 2. Thanks to the simple data flow path and constant symbol rate, no complex hardware resource scheduling algorithms are required for the investigated system. Pipeline synchronization can in that case be based solely on the data flow. This results in relatively low hardware complexity while maintaining high system performance. For static-scheduled pipelined processing with data-flow synchronization and constant input data rate, we suggest two basic requirements for efficient data processing:

- Processing throughput of each pipeline stage defined as number of data tokens processed in a unit of time should not exceed the input data rate

- Pipeline stalls caused by external events and data dependencies should be avoided
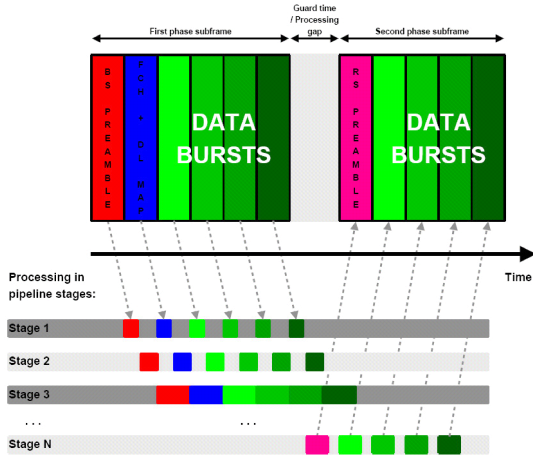
Fig. 5. Pipelined two-hop OFDM(A) frame processing scheme

Fig. 6. Modified architecture of the signal regeneration module used for DF forwarding

Direct mapping of the component-based system architecture presented in Section 2 into a pipelined structure presents two basic challenges for maintaining constant data flow in the pipeline:

- Time Division Duplexed (TDD) system operation requires that processed symbols are generated after the first phase of transmission is finished. This issue concerns both AF and DF schemes and results in the need for data buffering within the processing pipeline. Thus it provides proper subframe synchronization. The data buffer can be placed between various stages of the pipeline depending on design requirements and optimization goals

- Multi-user system operation and MCS utilization require that FEC encoding is applied independently for each user data burst and each subchannel. For DF forwarding this presents a challenge: data needs to be decoded to bit level and re-encoded with a proper FEC scheme for each subchannel separately. Since FEC encoders and decoders are state-based devices, this implies maintaining encoder and decoder states for each subchannel while processing each OFDM(A) symbol. A solution is to constantly store trellis state of the Viterbi decoder and CC register state for each subchannel and to switch the encoder and decoder processing contexts each time a different subchannel is processed.

The TDD operation issue is solved by placing a buffer in the processing pipeline. For AF forwarding it is placed at the end of the pipeline to minimize the processing gap between subframes. This enables the stored samples to be output directly to the Tx part without unnecessary delays. For DF forwarding, buffering can be performed on bit level by storing user data after the puncturing stage. This minimizes the buffer size since data stored in bits or short bit sequences result in smaller buffer size as compared to storing FFT points as in the case of the AF scheme.
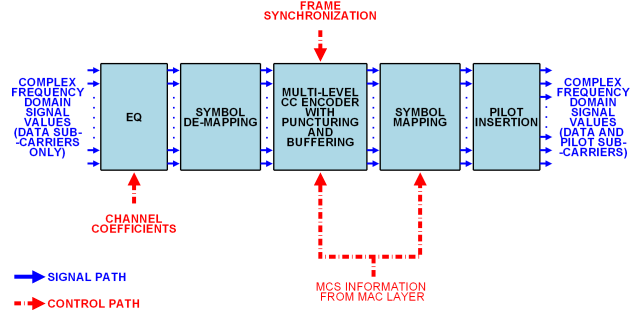
The FEC processing issue is solved by combining the idea of parallel subchannel processing with a simplification to the system structure based on the system model considered in this work. The Line-Of-Sight (LOS) conditions in the $BS- >$ RS channel and the deployment of the relays at strategic positions in the cell enable using high modulation levels (e.g., 64QAM) without the need for FEC [8]. This simplifies the baseband processing pipeline structure by removing the Viterbi decoder at the RS.

Additionally, to maintain system synchronization on bit level, all bit-level components such as parallel CC encoders, puncturer and multi-level bit buffer are integrated in a single hardware component. Such solutions enable constant and efficient data flow in the processing pipeline while maintaining relatively low hardware complexity. Based on the above statments, a modified structure of the signal regeneration module used in case of DF forwarding is proposed and presented in Figure 6.

4. IMPLEMENTATION RESULTS

A reference design realizing the IEEE 802.16e (Mobile WiMAX) PHY layer provided by Altera Corporation [9] is used as a supplementary source of IPs. The overall architecture is shown in Figure 7. The inner details of the developed blocks can be found in [10].

Table 1 presents the results for the component cycle count evaluation. Due to operation in two separate clocking domains (the slow clocking domain is used at the boundary components of the system such as cyclic prefix removal and cyclic prefix insertion with the slow clock frequency matching exactly the input data rate for proper I/O synchronization. The fast clocking domain covers all signal processing blocks in order to increase system performance and efficiency in hardware resource usage), the CP removal and CP insertion components are evaluated only in terms of the fast clocking frequency. Cycle count measures are obtained for the most demanding case, i.e., 64QAM modulation. An evaluation of the minimal system clock frequency can be performed using the obtained cycle count values. The terms f_{suf} and f_{nec} rep-
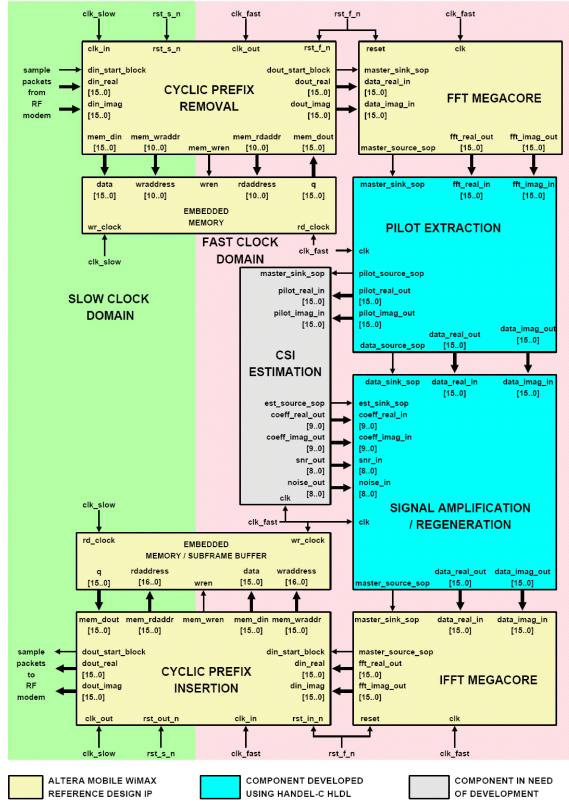
Fig. 7. RTL RS baseband processor architecture

Component name	Latency cycles (C_{LAT})	Output cycles (C_{OUT})	Non-idle cycles ($C_{LAT} + C_{OUT}$)
CP removal	-	2048	2048
FFT/IFFT	3712	2048	5760
Pilot extraction (AF)	1890	1729	3619
Pilot extraction (DF)	1890	1537	3427
EQ	7878	9216	17094
Symbol demapper	24	9216	9240
Symbol mapper (64QAM case)	12	9216	9228
Pilot insertion (64QAM case)	9211	2048	11259
Signal amplification	6918	2048	8966
CP insertion	2048	-	2048

Table 1. Cycle counts obtained from component simulations

resent the minimal and necessary system clock frequencies, respectively. In order to prevent data loss or pipeline stalls, f_{nec} guarantees that each data token doesn't not overlap. Due to back-to-back OFDM(A) symbol arrangement in the frame, data input period equals to the OFDM(A) symbol duration which is $T_{IN} = 102.8\mu s$ [11]. In case of AF forwarding, the calculated clock frequency values are: $f_{nec}^{AF} = 19.9MHz$ and $f_{suf}^{AF} = 87.2MHz$ The bottleneck component responsible for high f_{suf} value in this case is the signal amplification block with 8966 busy cycles within a single symbol processing period.

The architecture is coarse-grained data processing oriented. Therefore the components operate on entire data packets and do not process data from the input before a complete output data packet is produced. The sufficient clock frequency value in that case is also the necessary minimal system clock frequency which is $f_{min}^{AF} = 87.2MHz$. In case of DF forwarding the calculated frequency range is expressed by: $f_{nec}^{DF} = 89.6MHz$ and $f_{suf}^{DF} = 166.3MHz$ In case of DF implementation, the bottleneck component is the EQ block which as in case of the signal amplification module determines the system clock frequency at the f_{suf} value of $f_{min}^{DF} = 166.3MHz$. The differences in $< f_{nec}, f_{suf} >$ ranges illustrate different requirements for clocking frequency in case of both designs. The DF implementation requires approximately two

Component name	ALMs (N_{ALM})	RAM bits (N_{RAM})	9-bit multipliers (N_{MULT})
CP removal	101	65536	0
FFT/IFFT	725	311296	36
Pilot extraction (AF)	134	55296	0
Pilot extraction (DF)	134	55296	0
EQ	1455	57600	0
Symbol demapper	430	512	0
Symbol mapper	61	272	0
Pilot insertion	147	49152	0
Signal amplification	1004	58176	0
CP insertion	111	131072	0
bit-level buffer (DF) (estimated value)	0	110592	0
symbol-level buffer (AF) (estimated value)	0	655360	0

Table 2. Component synthesis results on Altera Stratix II platform

times higher clocking frequency as compared to AF, which can significantly affect the system's power consumption. Moreover, eliminating the bottleneck lying in the channel EQualizer (EQ) block internal architecture (namely a costly division needed to remove the channel coefficient) can lead to clock frequency reduction to a value not smaller than $f_{nec}^{DF} = 89.6MHz$. On the other hand, eliminating bottlenecks in the AF implementation can lead to clock frequency as low as $f_{nec}^{AF} = 19.9MHz$.

The component synthesis results[1] enable to evaluate the designs in terms of utilization of specific hardware resources as well as to calculate overall hardware complexity cost. Table 2 presents hardware utilization measures of the implemented components.

For the Altera Stratix II family, hardware complexity is characterized by the number of Adaptive Logic Module units (N_{ALM}), embedded RAM bits (N_{NRAM}), and dedicated 9-bit multipliers (N_{NMULT}). In order to evaluate and compare hardware complexity of various system components, the following hardware complexity cost function is used:

$$HWCOMP = \alpha N_{ALM} + \beta N_{NRAM} + \gamma N_{NMULT}$$

The terms α, β and γ are weights reflecting the significance of each metric. Weight values are selected according to the number of resources available for each type typically placed on a single FPGA chip. Analysis of the Stratix II family architecture results in applying weights of $\alpha = 100$, $\beta = 1$ and $\gamma = 10000$ based on average proportions concerning on-chip resource availability [10].

The results of hardware complexity evaluation are shown in Figure 8. The evaluation shows two main observations:

- FFT / IFFT blocks require significantly more hardware resources as compared to other system components

- Symbol-level buffering utilized in AF based schemes introduces significant RAM consumption due to the need of storing complex numbers in contrast to storing short bit sequences as in case of DF implementation.

5. CONCLUSION

Two signal forwarding methods - AF and DF - were investigated and evaluated in terms of implementation feasibility, hardware requirements and processing performance. Both forwarding methods prove to be feasible in terms of hardware implementations in OFDM(A) based wireless relay networks. The implementation models developed meet all the design constraints for real-time operation at achievable system clock frequencies. The proposed pipelined processing architecture introduces high processing performance which enables to minimize the processing gap to zero.

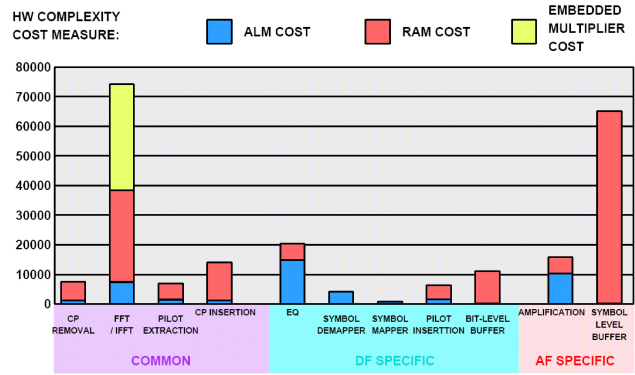[1]Obtained by using the Quartus II environment

Fig. 8. Hardware complexity evaluation of system components synthesized on Stratix II FPGA

In terms of overall hardware complexity reflecting hardware resource requirements relatively to their availability, the AF implementation is more complex due to higher memory usage as compared to DF based systems. Interestingly, this observation contradicts the assumptions typically found in the literature. On the other hand, DF baseband processing hardware requires higher clocking frequencies which results in increased power consumption at the RS. Feasibility of both methods depends on specific application characteristics and limitations such as availability of power supply or hardware resources at the RS.

6. REFERENCES

[1] R. Pabst et al., "Relay-based deployment concepts for wireless and mobile broadband radio," *IEEE Communications Magazine*, , no. 42, pp. 80–88, September 2004.

[2] H. Li and D. Yu, "Comparison of ad hoc and centralized multihop routing," October 2002.

[3] A. Lindgren and O. Schelen, "Infrastructured ad hoc networks," August 2002.

[4] C. Hoymann et al., "Fireworks. Flexible Relay Wireless OFDM-Based Networks. Cellular Deployment Concepts for Relay-Based Systems," *Technical report, Infor- mation Society Technologies*, January 2007.

[5] A. Bletsas and A. Lippman, "Implementing cooperative diversity antenna arrays with commodity hardware," *IEEE Communications Magazine*, , no. 44, pp. 33–40, December 2006.

[6] J. Bicket et al., "Architecture and Evaluation of an Unplanned 802.11b Mesh Net-work," August 2005.

[7] B. Can, H. Yomo, and E. De Carvalho, "Link adaptation and selection method for ofdm based wireless relay networks," *JCN Journal on MIMO OFDM and its Applications*, June 2007.

[8] B. Can, *Link Adaptive Transmission Techniques for OFDMA Based Wireless Relay Networks*, Ph.D. Thesis, Aalborg University, Denmark, 2008.

[9] Altera Corporation, "A Scalable OFDMA Engine for Mobile WiMAX," 2006.

[10] M. Portalski, "Hardware aspects of fixed relay station design for ofdm(a) based wireless relay networks," 2007.

[11] WiMAX Forum, "Mobile WiMAX - Part I: A Technical Overview and Performance Evaluation," Tech. Rep., 2006.