JYVÄSKYLÄ STUDIES IN COMPUTING
154

Henrik Martikainen

PHY and MAC Layer Performance Optimization of the IEEE 802.16 System

JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 154

Henrik Martikainen

PHY and MAC Layer Performance Optimization of the IEEE 802.16 System

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen auditoriossa 1
syyskuun 7. päivänä 2012 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, auditorium 1, on September 7, 2012 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2012

PHY and MAC Layer Performance Optimization of the IEEE 802.16 System

JYVÄSKYLÄ STUDIES IN COMPUTING 154

Henrik Martikainen

PHY and MAC Layer Performance Optimization of the IEEE 802.16 System

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2012

Editors
Timo Männikkö
Department of Mathematical Information Technology, University of Jyväskylä
Pekka Olsbo, Ville Korkiakangas
Publishing Unit, University Library of Jyväskylä

URN:ISBN:978-951-39-4830-6
ISBN 978-951-39-4830-6 (PDF)

ISBN 978-951-39-4829-0 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2012

ABSTRACT

Martikainen, Henrik
PHY and MAC Layer Performance Optimization of the IEEE 802.16 System
Jyväskylä: University of Jyväskylä, 2012, 80 p.(+included articles)
(Jyväskylä Studies in Computing
ISSN 1456-5390; 154)
ISBN 978-951-39-4829-0 (nid.)
ISBN 978-951-39-4830-6 (PDF)
Finnish summary
Diss.

This thesis concentrates on how to improve the performance of the IEEE 802.16 system in three main areas. The first problem area is system throughput optimization by selecting the optimal MAC PDU size when ARQ is used. A method for selecting the optimal size is proposed and evaluated. In the second part, various ARQ features are evaluated and also the performance of the ARQ and HARQ error correction methods are evaluated. Finally, the two different frame optimization problems are investigated. First the three duplexing modes of IEEE 802.16 are compared, namely Time Division Duplexing, Full Frequency Division Duplexing and Half Frequency Division Duplexing. A novel group division algorithm is proposed for H-FDD. Second, the relay mechanism of IEEE 802.16 is presented and its performance evaluated. The performance evaluation of all presented problems is done by simulations with the WINSE extension which is build on top of the NS-2 network simulator.

Keywords: IEEE 802.16, WiMAX, NS-2, Performance, WINSE, ARQ, HARQ, Duplexing, FDD, TDD, H-FDD

Author

Henrik Martikainen
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Supervisors

Prof. Timo Hämäläinen
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Dr. Alexander Sayenko
Industrial Environment 3GPP standardization
Nokia Siemens Networks
Finland

Reviewers

Dr. Dmytro Andrushko
Telecommunication Systems Department
Kharkov National University of Radio Electronics
Ukraine

Dr. Andrey Garnaev
Department of Computer Modelling and Multiprocessor Systems
St Petersburg State University
St Petersburg, Russia

Opponent

Prof. Jyri Hämäläinen
Department of Communications and Networking
Aalto University
Espoo, Finland

ACKNOWLEDGEMENTS

This dissertation has taken a lot of work from me but it would not have been possible without the help of many people. First of all, I would like to thank my supervisor Professor Timo Hämäläinen for giving me the possibilities to work at the university and relentlessly finding new funding opportunities. Also, I'm in great gratitude for Alexander Sayenko for leading the project, helping me in all the possible ways and being a great source of inspiration. The research was done in co-operation with Nokia and Nokia Siemens Networks and would not have been possible without them. In addition, I would like to thank all the other people who I worked with and who helped me in my publications: Olli Alanen, Vitaliy Tykhomyrov, Oleksandr Puchko and Vesa Hytönen. All work and no play would make Henrik a dull boy and therefore I want to thank all the great people who ever worked or had a cup of coffee or tea with me.

I am very grateful for my reviewers Dr. Dmytro Andrushko and Dr. Andrey Garnaev for their effort in reviewing the dissertation and Professor Jyri Hämäläinen for being my opponent. I also would like to thank Steve Legrand for the language review of this dissertation. In addition, I would like to thank Magister Solutions and Janne Kurjenniemi for the possibilities to work with the thesis. And all the pool games and interesting lunch discussions with colleagues at Magister.

The following foundations have helped me completing this thesis with their funding and I would like to thank them:

- Emil Aaltosen säätiö
- HPY:n tutkimussäätiö

Finally, I would like to thank my family for their full support during my whole life. And all my friends who have believed in me the whole time even when I did not. Last and most importantly, I thank Reetta for her love and support.

I am sure that there are still persons who I did not remember to mention. I'm sure you know who you are and I thank you all. Kiitos.

GLOSSARY

3G	3rd Generation
3GPP	3rd Generation Partnership Project
ACK	Acknowledge
AMC	Adaptive Modulation and Coding
ARQ	Automatic Repeat Request
BE	Best Effort
BLER	BLock Error Rate
BS	Base Station
CDF	Cumulative Distribution Function
CID	Connection IDentifier
CINR	Carrier to Interference and Noise Ratio
CRC	Cyclic Redundancy Check
DCD	Downlink Channel Descriptor
DL	Downlink
DL-MAP	Downlink Map
DSL	Digital Subscriber Line
ertPS	Extended Real-time Polling Service
E2E	End-to-End
FCH	Frame Control Header
FPS	Frames Per Second
FSH	Fragmentation Subheader
FTP	File Transfer Protocol
FUSC	Fully Used Subchannel
F-FDD	Full Frequency Division Duplexing
GMH	General Management Header
GMSH	Grant Management Subheader
HARQ	Hybrid Automatic Repeat Request
H-FDD	Half Frequency Division Duplexing
IE	Information Element
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LTE	Long Term Evolution
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
NACK	Not ACKnowledge
nrtPS	Non-Real-Time Polling Service
NS-2	Network Simulator v2
OFDM	Orthogonal Frequency Division Multiplex
OFDMa	Orthogonal Frequency Division Multiple Access
PDU	Protocol Data Unit
PHY	Physical Layer
PSH	Packing Subheader

PUSC	Partially Used Subchannel
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RTG	Receive/transmit Transition Gap
rtPS	Real-Time Polling Service
SDU	Service Data Unit
SS	Subscriber Station
TCP	Transmission Control Protocol
TDD	Time Division Duplexing
TTG	Transmit/receive Transition Gap
UDP	User Datagram Protocol
UGS	Unsolicited Grant Service
UMTS	Universal Mobile Telecommunications System
UL	Uplink
UL-MAP	Uplink Map
VoIP	Voice over IP
WiMAX	Worldwide Interoperability for Microwave Access

LIST OF FIGURES

LIST OF TABLES

CONTENTS

LIST OF INCLUDED ARTICLES

PI H. Martikainen,A. Sayenko, O. Alanen and V. Tykhomyrov. Optimal MAC PDU Size in IEEE 802.16. *4th International Telecommunication NEtworking WorkShop on QoS in Multiservice IP Networks*, 2008.

PII H. Martikainen, O. Alanen and V. Tykhomyrov. Impact of portable device restrictions on IEEE 802.16 performance. *The 2nd IEEE International Interdisciplinary Conference on Portable Information Devices*, 2008.

PIII H. Martikainen, O. Alanen and A. Sayenko. ARQ parameters for VoIP in IEEE 802.16 networks. *Wireless Telecommunications Symposium*, 2009.

PIV H. Martikainen. Analysis of Duplexing Modes in the IEEE 802.16 Wireless System. *European Wireless*, 2010.

PV A. Sayenko, H. Martikainen and O. Puchko. Performance Comparison of HARQ and ARQ Mechanisms in IEEE 802.16 Networks. *The 11-th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems*, 2008.

PVI V. Tykhomyrov, A. Sayenko, H. Martikainen and O. Alanen. Analysis and Performance Evaluation of the IEEE 802.16 ARQ Mechanism. *Journal of Communications Software and Systems, vol 4, issue 1*, 2008.

PVII A. Sayenko, O. Alanen, H. Martikainen, V. Tykhomyrov, O. Puchko, V. Hytönen, T. Hämäläinen. WINSE: WiMAX NS-2 Extension. *Simulation: Society for Computer Simulation International*, 2011.

PVIII A. Sayenko, O. Alanen and H. Martikainen. Analysis of the Non-transparent In-band Relays in the IEEE 802.16 Multi-hop System. *IEEE Wireless Communications & Networking Conference*, 2010.

1 INTRODUCTION

Broadband connections have rapidly become more common and most Finnish households have a fixed, wired broadband connection. A broadband connection has an instantaneous bandwidth of more than 1 MHz and it supports data rates greater than about 1.5 Mbps [2]. Finland, however, is a sparsely populated country and it is not economical to run a copper or optical wire to everywhere. The same applies to many places around the world. IEEE has defined a wireless 802.16 standard called WiMAX which might solve the problem. It can provide a high-speed wireless access to the Internet for home and business subscribers. WiMAX has a range of up to tens of kilometres and speed of tens of Mbps, and it can also provide a fast citywide access or fast access for example in trains.

IEEE 802.16 started as a line-of-sight (LOS) solution on the 10GHz to 66GHz band to above last-mile problem where fibre connection was not an option. This 802.16 standard called Wireless MAN-SC was introduced in 2001. The 802.16a amendment was completed in 2003, bringing support for NLOS deployments in the 2GHz to 11GHz. Additional revisions were made in 2004, and the revised complete standard 802.16-2004 was introduced. In 2005, the 802.16e-2005 amendment, which brought support for high-speed mobility, was then published. Since the IEEE 802.16 standards have a very broad scope, the WiMAX Forum was formed. Its task was to ensure interoperability between vendors by creating WiMAX profiles and certifying products which comply with these profiles. The existence of WiMAX-certified products was a major milestone in the history of IEEE 802.16/WiMAX [3].

1.1 Performance in Broadband Wireless Networks

There are many ways to make data transmission more efficient and reliable with an error-prone wireless channel. Due to varying behaviour at the lower levels, the higher level protocols could treat wireless connections differently. Compared to wired connections, wireless connections have a greater number of packet drops caused by errors, these drops being usually caused by congestion. Also the drops in wireless connections occur in bursts. In cellular wireless networks also handoffs might cause additional packet drops.

1.1.1 Recovering from Wireless Errors

TCP is the most used protocol in the Internet. It is well known that its performance is not optimal with high error-rate wireless channels where packet drops are not caused by congestion but from errors at the wireless link. Wired connections do not have many errors, and therefore TCP is assuming that those drops are caused by congestions. TCP reacts to this by reducing its transmission window size, initiates congestion-control mechanisms and resets its retransmissions. These actions decrease the load on the links, which is useful in congestion situations. However, if the drops are caused by errors at the wireless link, the end-to end throughput is unnecessarily reduced. This is especially valid when the errors are sporadic, which is usually the case with wireless links [4].

One solution to the problem is to improve the TCP behaviour in wireless environment. The sender should be aware of the wireless links and adjust its behaviour accordingly. If TCP could have the information that drop was caused by errors rather than congestions the slow start halving could be avoided. Other options are to shrink the advertised TCP window on error situation or split the TCP connection [5, 6, 7]. The Eifel scheme suggests adding time stamps to acknowledgements, thus differentiating among acknowledgements generated by the initial transmission and retransmission [8]. Also, selective acknowledgements help improving the TCP performance in wireless networks [4]. In addition there are many end-to-end schemes, which proactively try to adjust the congestion window to an optimal level. These include methods such as TCP Vegas, TCP Reno, TCP Westwood and TCP-Jersey, and they work better with random errors than traditional end-to-end schemes do. In general, TCP solutions tackle a specific problem and it is difficult to create a solution which would work in a heterogeneous network [9].

1.1.2 Preventing Wireless Errors

Instead of trying to make TCP adapt to behaviour at the wireless links, wireless links can adapt to TCP behaviour. The purpose of reliable link-layers protocols (RLP) is to correct wireless errors locally so that TCP will not notice them. RLP protocols may cause duplicate TCP transmissions if retransmissions on the wireless take too long time. To avoid this, local retransmissions should be interrupted after some time. In addition to TCP, reliable link layer is also important with other protocols such as UDP. Error correction and prevention is at its most efficient when done on link level. This is because then link-specific methods can be used, and also the delays for the retransmissions are smaller. The practical advantage of implementing error correction at the link layer is that then there is no need to modify existing TCP implementations. Also link layer error correction is much faster and efficient than relying on TCP retransmissions [10].

The error probability for a transmission on a wireless channel is highly dependent on the radio channel conditions and on what kind of modulation, coding and other radio signal parameters are chosen. When the radio conditions are good, a more efficient modulation and coding scheme (MCS) can be used. If the radio conditions become worse, then the MCS should be changed to a more robust one or the error probability will become too high or even make the correct reception of data impossible. That is why all the modern radio technologies use link adaptation (LA) methods where MCS is adapted based on the current radio link conditions to minimize the loss of performance caused by the errors [11, 12, 13].

One way to make errors less likely at the wireless channel is by using FEC (forward error correction) method. FEC relies on sending redundant data along with the actual data. This redundant data can then be used at the receiver to detect and correct errors to some degree. While the FEC coding increases the probability of a successful receipt it also significantly increases the amount of transmitted data.

Another principal error correction method is ARQ (automatic repeat request), which is sometimes called BEC (backward error correction). In this method, the errors can be detected but not corrected. Once the errors have been detected, the link layer can use a feedback mechanism to tell the sender to resend the erroneous block of code. Hybrid automatic repeat request (HARQ) is a combination of FEC and ARQ. If the packet is received without errors, then the behaviour is similar to ARQ, i.e. the sender is signalled that the packet was successfully received. However, if the packet contains errors, instead of discarding the packet as with ARQ, it is kept in the receiver's buffer. Then the sender will send the packet again, possibly with additional FEC information, and the receiver can combine the information of all transmissions. Thus the probability of successful decoding increases with every retransmission. The benefit of ARQ over HARQ is that it uses a lot less redundant data to detect the errors. In principle, ARQ based methods are more efficient when the error probability is lower, and HARQ methods are more efficient when the error probability is higher [14, 15].

In addition to errors caused by noise and interference, errors can be caused by packet collisions. Wired MAC (medium access control) protocols like Ethernet use CSMA/CD (Carrier Sense Multiple Access/Collision Detection)mechanism, which allows the sending station to detect when a collision occurs and abort the transmission. On wireless media, the sender usually can not listen while sending, and even if it can, the signal of the sender's own transmission is much stronger than the signal of others, and therefore collisions can not be detected. Instead, for example IEEE 802.11[11] uses the CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) mechanism. However, CSMA/CA does not work very well in busy networks and leads to inefficient usage of radio resources [14].

A more advanced way to solve collisions is to have a special station which does the scheduling of all frames. This station is often referred to as base station. For example the IEEE 802.11[11] standard defines a simple centralized scheduling solution called point coordination function. It is not, however, widely supported or required by the WiFi Alliance[16]. However, all modern mobile network technologies like UMTS, LTE and IEEE 802.16 use centralized scheduling [11, 12, 13].

1.1.3 Overhead of Wireless Connections

Errors and collisions are not the only things which impact on the performance of wireless broadband connections. In addition, various wireless properties bring additional overhead which does not exist with wired connections. On the PHY level, the radio receivers need synchronization, there are transmission gaps for various reasons, and preambles and pilot signals are needed. Also there needs to be transmission opportunities for CQI messages and HARQ feedbacks. Different duplexing modes also need different separation; frequency separation for FDD (frequency-division duplexing) systems and time separation for TDD (time-division duplexing) systems.

Wireless MAC protocols also create additional overhead. Each protocol introduces additional headers to the packet. If centralized scheduling is used, then contention opportunities or polling use some resources. In addition, the base station needs to inform the scheduling decisions. Also, other kinds of management messages are used and this is a major cause of downlink overhead[17]. And if centralized scheduling is not used then collisions can happen, which also decreases the transmission efficiency similarly to overhead.

The third source of overhead are tradeoffs which are made to improve reliability. The size of the MAC layer PDU has a significant impact on the probability that it will be dropped. The bigger the PDU the bigger the probability that it can be dropped. Decreasing the PDU size will therefore improve reliability at the cost of increased overhead. Error correction and detection methods form another tradeoff. If they are used, some bandwidth is used for redundant data and acknowledgements. The same applies to the selection of a modulation and coding scheme (MCS). A more efficient MCS will allow more bits to be sent but at the same time error probabilities will increase.

1.2 Problem Statement

The previous section describes general performance problems of a wireless system. This thesis is concentrating on how to improve the performance of the IEEE 802.16 system. The WiMAX specification[13] leaves a lot of open questions on how the actual implementations should be done. WiMAX Forum has defined WiMAX Forum Mobile System Profiles [18] which define the mandatory features from the specification. Still, it does not tell how to use these features the most efficient way. This thesis is looking at how to improve the performance on three main areas.

The first problem area is optimal MAC PDU usage. The size of the MAC

PDU has impact on how much data must be retransmitted when errors occur and also how much overhead arise from the PDU headers. By estimating the error ratio of the channel the optimal size of PDU can be decided which leads to the best performance. This problem is investigated in Chapter 2.

There are two retransmissions mechanisms in the IEEE 802.16 system called Automatic Repeat Request (ARQ) and Hybrid Automatic Repeat Request (HARQ). The second problem is how to use ARQ efficiently, how ARQ and HARQ performance differs and what is the optimal number of bursts per frame for ARQ and HARQ. These issues are looked in Chapter 3.

The third and final problem area is frame optimization in the IEEE 802.16 system. There are three main duplexing modes in IEEE 802.16: Time Division Duplexing (TDD), Full Frequency Division Duplexing (F-FDD) and Half Frequency Division Duplexing (H-FDD). Although the choice between frequency and time division is often governed by the available spectrum there are still open questions on how the performance can be optimized with different duplexing modes. Especially with H-FDD the specification does not tell how the frame should be divided and how the users should be divided between groups. Also IEEE 802.16 system has a support for wireless relay nodes. The relays nodes use part of the frame to transmit data between the basestation and the subscriber stations. These problems are introduced and studied in Chapter 4.

1.3 Outline of the Dissertation

The rest of the dissertation is organized as follows.

Chapter 2 describes the IEEE 802.16 MAC format and presents the MAC PDU size optimization problem. The related research work is presented and then the MAC PDU size optimization method is presented. The presented method is then evaluated by means of simulations. The target of the optimization is to improve system throughput.

Chapter 3 presents the retransmission methods of the IEEE 802.16 system. First ARQ and its various features are presented. Then HARQ is presented. The ARQ and HARQ performance is studied with network simulations. In most of the cases the performance criteria is throughput, except for VoIP and ARQ subcase where packet delays and drops are evaluated.

Chapter 4 presents the three duplexing modes of IEEE 802.16: TDD, F-FDD and H-FDD. Also a way to select the H-FDD group size and selecting the group for subscribers are presented. Duplexing mode performance is then analysed. In the second part of the chapter relay mechanism is presented and analysed. The performance evaluation criteria is system throughput and user fairness.

Finally the Chapter 5 concludes the thesis. Also the applicability of presented optimization methods to other wireless technologies is discussed.

1.4 Main Contribution

During the work on the subject of this dissertation, the author has produced and co-authored several publications.

Publication **PI** proposes a method estimate the optimal size of IEEE 802.16 MAC PDU when the error probability of the channel is known. The presented method is then evaluated by comparing the analytical result with simulations. Also the ARQ block rearrangement feature is presented and it is evaluated what is the optimal PDU size if block rearrangement is not used. The author was responsible for the main idea of the PDU size estimation method, implemented the ARQ block rearrangement feature, various ARQ features and simulation scenario. The author was also responsible for performing the simulations and main person doing the analysis of the work.

In publication **PII** restrictions of mobile devices which impact on how IEEE 802.16 ARQ features can be used is analyzed. The publication investigates how the lack of ARQ block rearrangement, small ARQ window, large ARQ blocks and lack of some ARQ feedback types impact on application throughput. The author is responsible for implementing ARQ block rearrangement, some ARQ features to the WINSE simulator, main idea and simulation analysis.

The way ARQ parameters should be configured for VoIP traffic is analysed in publication **PIII**. The author is the main contributor in this publication and is responsible for running the simulation. The author is also the main person responsible for the idea and simulation analysis. Publication **PIV** compares TDD, F-FDD and H-FDD duplexing modes in IEEE 802.16 system. The performance and the fairness of the three duplexing modes is compared. Also the difference of introduced two H-FDD group division algorithms is analysed. The author is responsible for H-FDD implementation and otherwise the contribution is similar to previous publication.

In publication **PV** the performance of ARQ and HARQ retransmission methods is evaluated. The author is responsible for is for implementing some ARQ features and in addition contributed with running and analysing the simulation results.

Publication **PVI** analyses ARQ features like ARQ feedback type selection, piggy-packed feedbacks, ARQ transmission window and ARQ timers. The author was co-author and responsible for ARQ rearrangement, ARQ feedback piggy-packing and ARQ timer implementations. The simulations on those topics was also run by author. In addition the author contributed with analysis of the results.

In publication **PVII** the WINSE NS-2 extension is presented. The extension adds IEEE 802.16 support to NS-2 network simulator. The work in this publication is co-authored. The contribution of the author include helping in designing the extension, implementing ARQ and MAC related features, and analysing of the simulation results.

Publication **PVIII** presents IEEE 802.16 relay options and analyses non-transparent in-band relay option in detail. In this research the author contributed in running and analysing the simulation results.

2 IEEE 802.16 MAC PDU

This chapter describes the optimal MAC PDU size problem and proposes a way to estimate the optimal size based on the link quality. First, related research is presented in Section 2.1. Section 2.2 introduces the IEEE 802.16 MAC PDU format, and Section 2.3 presents a method to calculate the optimal size of the PDU. Section 2.4 describes the WINSE simulator, which is used in the simulations. The method presented is then evaluated by means of simulations in Section 2.5, and the conclusions are in Section 2.6.

2.1 Related Research

The idea of adapting the PDU or frame size, based on channel quality, is not new. In [19], [20] and [21], a frame size optimization is done once during the construction of the network, and the frame size is the same for the whole network. This method might be valid for a wired network where the conditions are not likely to change over time unless the network structure is changed.

In [22], the authors propose to adapt frame size that is based on link quality in the WaveLAN wireless environment. They highlight many important aspects like the fact that wireless layer should be transparent to the higher layers and the requirement for an error correction method like ARQ which is configured so that the higher layer protocols like TCP are not disturbed. Also the authors propose a channel quality feedback mechanism which would be used to select an optimal frame size.

The MAC performance of IEEE 802.16 in general, including PDU size, is studied in [23], but the authors assume errorless channel conditions. In [24], the authors study the MAC PDU optimization problem in IEEE 802.16 relay environment. The idea is to reduce the PDU size for relay to SS link if the SINR for the MCS selected by Adaptive Modulation and Coding (AMC) is too low. The method is found to be useful. However, the study does not investigate what the optimal PDU size should be for the initial BS to relay link. Also, the error modelling is done only for the whole PDU, not for an individual FEC block, and the frame structure and related scheduling issues are not considered.

The optimal MAC PDU size in IEEE 802.16 has also been studied in [25], where authors introduce a variable-sized CRC and PDU size mechanism. The sizes are changed according to the six-level feedback of reception status. The *Other subheader* bit is used to distinguish important PDUs upon transmission. The adaptive CRC field had a minor impact while the adaptive payload size had a significant impact. However, the introduced method requires changes to the standard and additional feedback of reception status.

In [26], the basic idea of optimal PDU size is introduced, but only simple UDP traffic is used and no error correction is enabled. In [27], an adaptive packet size estimation method based on ARQ feedbacks is presented. The idea is interesting, and the results show improvement, but the PHY is OFDM and no scheduling or frame structure issues are considered.

2.2 MAC PDU

This section describes the 802.16 MAC PDU format. Figure 1 show the IEEE 802.16 frame structure when TDD is used. In the beginning of the frame there is the DL-MAP message, which is describing the content of the downlink subframe. The actual user data from the base station to the subscriber stations are sent in downlink bursts. Each burst requires an entry in the DL-MAP message. The more downlink bursts there are per frame the more space DL-MAP occupies. Similarly, there is an entry in UL-MAP for each burst in the uplink subframe. In this chapter, the burst optimization problem is not addressed, but that is investigated in Chapter 3. Inside one downlink or uplink burst there are one or more MAC PDUs, which is illustrated in Figure 2. Here the problem of selecting the optimal PDU size is investigated. The purpose is to find a PDU size which yields the best throughput with different error probabilities of the channel while other system parameters remain constant.

Table 1 shows a list of IEEE 802.16 MAC headers and their sizes when ARQ is enabled. When ARQ is disabled, PSH and FSH are one byte less. GMH is always present, and it contains information like the length of the PDU, CID (Connection IDentifier), CRC presence and subheader presence. GMSH subheader is used by SS to request for more bandwidth. PSH is needed when several application level SDUs are packed into one PDU and FSH when one SDU is fragmented into several PDUs. The size of extended subheaders is at least 3 bytes, the size depending on what extended subheaders are present. The optional MAC PDU payload is located after the headers. The optional CRC field is located after the payload and is used to detect errors in the headers and in the payload.

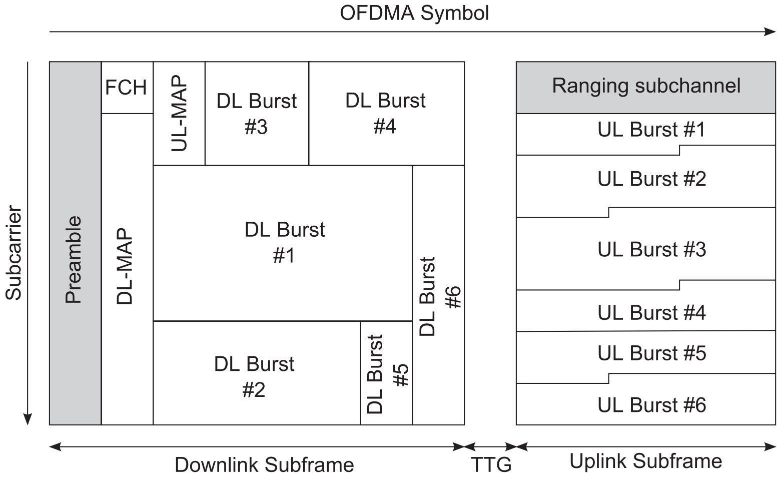


FIGURE 1 IEEE 802.16 TDD frame structure.

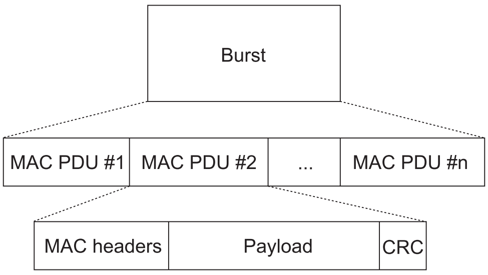


FIGURE 2 IEEE 802.16 burst and PDU structure.

2.3 MAC PDU Size Optimization Methodology

The optimal MAC level PDU size depends on a number of factors. A larger MAC PDU size results in less MAC overhead because there is the mandatory GMH per a PDU. If a connection utilizes the ARQ mechanism, then the PDU must contain CRC as well as FSH or PSH to hold the ARQ BSN. If we assume that the application level SDU is bigger than the preferred MAC PDU, there is no need to pack several SDUs into one PDU. Therefore, only FSH is present. We can neglect GMSH because it appears only in one PDU in a data burst and it is used only in the uplink direction. It is also assumed that other headers are not present. In this case, the MAC level overhead can be approximated as follows [28]:

$$O = S_{\mathrm{GMH}} + S_{\mathrm{FSH}} + S_{\mathrm{CRC}}. \tag{1}$$

Assuming the PDU size is L bytes, the amount of user data S in a PDU is:

$$S(L) = L - O. \tag{2}$$

In this study it is assumed that the size of the headers cannot be decreased

TABLE 1 IEEE 802.16 MAC headers.

Header	Explanation	Size
GMH	general MAC header	6 bytes
GMSH	grant management subheader	2 bytes
PSH	packing subheader	3 bytes
FSH	fragmentation subheader	2 bytes
FFSN	fast feedback allocation subheader	1 byte
-	extended subheaders	3- bytes
CRC	cyclic redundancy check	4 bytes

but is always fixed. Also the payload size could be decreased by using an IEEE 802.16 feature called Payload Header Suppression (PHS). It is relying on the fact that the payload often contains higher level headers, which is often unchanged from packet to packet. It is thus possible to avoid sending the full payload header information in every packet [29]. In this study, the PHS is not studied, but it is assumed that the amount of used data S is true after possible PHS.

$$E_P = 1 - (1 - E_B)^{N_B}. \tag{3}$$

Equation 3 expresses the general error probability for a packet (E_P) when bit error probability (E_B) is known and the packet length is N_B bits. This assumes that the bit errors are uniformly distributed. Similarly to other wireless technologies, 802.16 uses the forward error correction (FEC) mechanism to improve data transmission reliability. In this dissertation, we do not concentrate on the FEC behaviour but rather assume a certain FEC block error rate (BLER) as a function of the chosen modulation and coding scheme (MCS) and the effective signal-to-noise rate (SNR). If we assume some FEC BLER (E), the PDU error rate (E_P) can be calculated using the following expression, where B stands for the FEC block size measured in bytes [30]:

$$E_P(L) = 1 - (1 - E)^{\frac{L}{B}}. \tag{4}$$

This equation is exactly valid when the errors for FEC block are uniformly distributed, PDU size (L) is multiple of the FEC block size (B) and each PDU starts from the beginning of the FEC block. However, in reality the start of the PDU is not aligned with the start of the FEC block, and also the length of the PDU can be virtually anything with byte granularity. The equation therefore presents an estimation of packet error probability. The larger the packet size and the smaller the FEC block size, the more accurate the estimation becomes.

It is important to note that the expression can be applied if Hybrid ARQ (HARQ) is *not* enabled. Otherwise, HARQ retransmission gain influences significantly the BLER. Using previous formulas and taking the overhead and error probabilities into account, the optimal PDU size can be estimated. The efficiency (F) is presented as follows:

$$F(L) = \frac{S(L)}{L}(1 - E_P(L)) = \frac{S(L)}{L}(1 - E)^{\frac{L}{B}}. \tag{5}$$

Equation 5 tells how much user data the PDU contains compared to the total PDU size on average taking the error probability into account. The optimal PDU size is the one which gives the best efficiency value.

Figure 3 shows the analytical results for three different BLER values when the PDU size varies. The FEC block size of 60 bytes comes from the 802.16 OFDMa PHY that we choose for this particular case. As can be seen from Figure 3, BLER has a major effect on the efficiency values with different PDU sizes. As anticipated, the optimal PDU size tends to be smaller when there are more errors in the channel. Using a large PDU size of 500 bytes in the most erroneous case where BLER is 10^{-1} causes a very bad performance compared to the optimal PDU size of about 60 bytes.

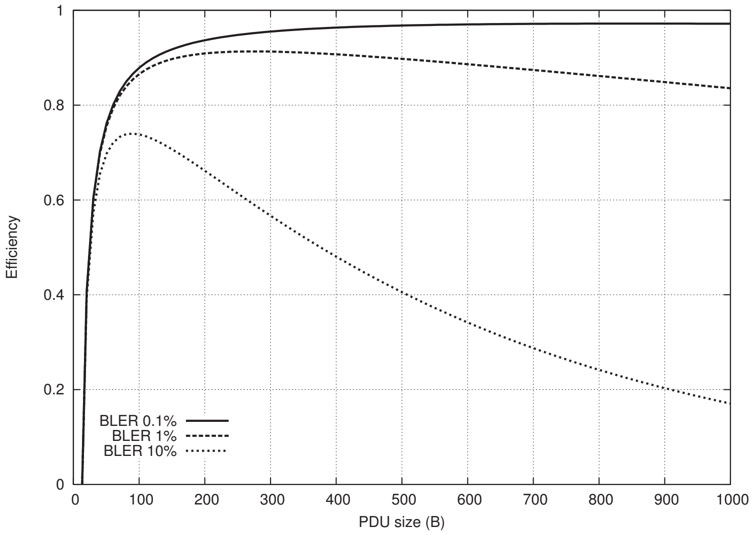


FIGURE 3 Theoretical MAC level efficiency from Equation 5 for different BLER ($O = 12$ bytes, $B = 60$ bytes).

We can use Equation 5 to find the optimal PDU size for the given channel BLER. Figure 4 illustrates the solved Equation 14 with a 60 B FEC block size and overhead of 12 bytes.

$$F = \frac{L-O}{L}(1-E)^{\frac{L}{B}}, \tag{6}$$

$$\ln F = \ln(L-O) - \ln L + \frac{L}{B}\ln(1-E), \tag{7}$$

$$\frac{F\prime}{F} = \frac{1}{L-O} - \frac{1}{L} + \frac{\ln(1-E)}{B}, \tag{8}$$

$$\frac{F\prime}{F} = \frac{O}{L(L-O)} + \frac{\ln(1-E)}{B}, \tag{9}$$

$$F\prime = \frac{L-O}{L}(1-E)^{\frac{L}{B}}\left(\frac{O}{L(L-O)} + \frac{\ln(1-E)}{B}\right), \tag{10}$$

$$(1-E)^{\frac{L}{B}}\left(\frac{O}{L^2} + \frac{L-O}{LB}\ln(1-E)\right) = 0, \tag{11}$$

$$(1-E)^{\frac{L}{B}} > 0, \frac{O}{L^2} + \frac{L-O}{LB}\ln(1-E) = 0, \tag{12}$$

$$\frac{\ln(1-E)}{B}L^2 - \frac{O}{B}\ln(1-E)L + O = 0, \tag{13}$$

$$L = \frac{O}{2} - \frac{\sqrt{(O\ln(1-E))^2 - 4BO\ln(1-E)}}{2\ln(1-E)}. \tag{14}$$

TABLE 2 Modulation and coding schemes (CTC).

MCS	Max. FEC block size (slots)	Slot size (bytes)	Max. FEC block size (bytes)
QPSK 1/2 rep. 6	10	6	60
QPSK 1/2 rep. 4	10	6	60
QPSK 1/2 rep. 2	10	6	60
QPSK 1/2	10	6	60
QPSK 3/4	6	9	54
16-QAM 1/2	5	12	60
16-QAM 3/4	3	18	54
64-QAM 2/3	2	24	48
64-QAM 3/4	2	27	54
64-QAM 5/6	2	30	60

According to the considerations above, if a connection resorts to using a constant PDU size, then this value should be sufficiently small. On good channel conditions, small PDUs do not decrease the performance significantly. However, with a bad channel, small PDUs increase the performance greatly. In a real environment, BLER may vary greatly as a result of changing SNR. So unless reliable information about errors can be gathered in real-time, the PDU size should be small, less than 200 bytes at least.

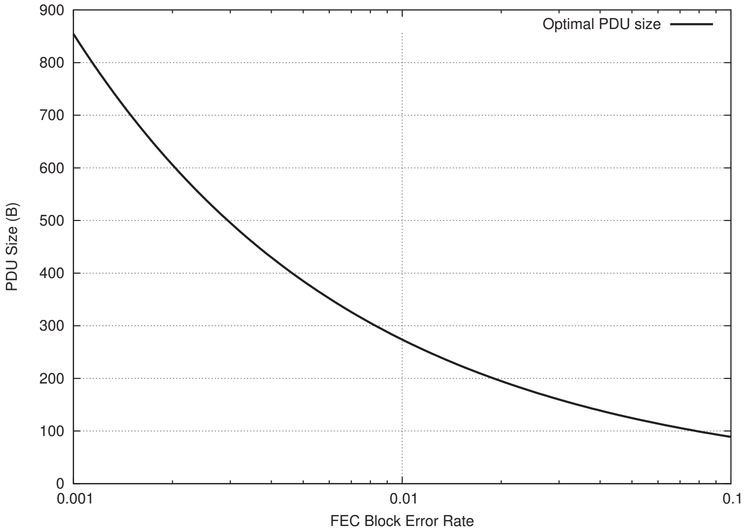


FIGURE 4 Theoretical optimal PDU size from Equation 13 ($O = 12$ bytes, $B = 60$ bytes).

It is worth noting that the BS and the SS measure constantly their Carrier to Noise and Interference ratio (CINR). There is a relationship between the CINR-level, MCS, and BLER. The BS changes MCS to achieve the target BLER. It is then possible to decide one PDU size limit for each MCS by using Equation 13. There could even be several PDU size limits per MCS in order to get the PDU size as close to the optimum as possible. The adaptive PDU size approach is feasible both in the downlink and in the uplink direction. In the downlink direction, the BS knows the target BLER and the control of PDU construction and thus has the full control of both. However, the BS does not have control on how the uplink MAC PDUs are constructed. Still, the SS can do similar PDU size optimization, but the problem is that it does not know what was the BLER target that was used for a particular MCS. In the worst-case, the SS should use a conservative approach and small PDU sizes.

$$\mathrm{BW} = \mathrm{FPS} \cdot N_{\mathrm{slot}} \cdot S_{\mathrm{slot}} \tag{15}$$

$$F(L) \cdot \mathrm{BW} \tag{16}$$

Equation 15 shows a definition for bandwidth. In this, FPS means frames per second, N_{slot} is the number of uplink or downlink slots, and S_{slot} is a single slot size. Equation 16 shows a way to estimate uplink or downlink data for a single SS. This formula assumes that all data is user data; management messages are not taken into account. Figure 6 presents an example of how to use Equation 16 to estimate throughput with different PDU sizes. The figure shows results for seven different MCS and for a single BLER. The FEC block sizes used are mentioned in Table 2.

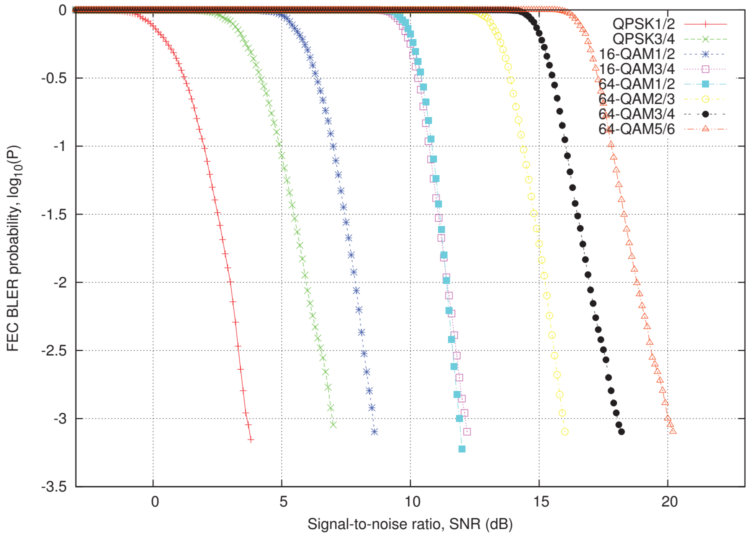


FIGURE 5 802.16 OFDMa CTC FEC BLER curves [1].

2.4 Description of WINSE Simulator

WINSE is a WiMAX extension for the NS-2 simulator. It was started as a small student project and then evolved into a powerful simulation tool. Now several companies use it to study the MAC and QoS in the 802.16 system. Table 3 gives a short overview of features supported in WINSE. WINSE is a dynamic packet level simulator. It supports different types of applications and models application level packets. The focus has been on MAC level implementation. For example the ARQ protocol has been implemented in great detail. See [PVII] for a full description of WINSE's features and modelling details.

2.5 Simulations on PDU Size Optimization

Figure 7 shows the network structure in the simulations. There are one or five subscribers which are sending data in the uplink direction using an FTP application over a TCP connection, which tries to use all the available bandwidth. The scheduler is throughput fair scheduler, i.e. it tries to provide the same throughput for all the subscribers. The subscribers are located 500 m from the basestation and not moving, but still the fast fading is creating fluctuations to the signal. The uplink direction was selected because there is much less overhead generated by MAP etc. management messages. The received data at the FTP server can thus be used directly to evaluate the wireless performance. It must be noted that although there is no user data in the downlink direction still the TCP generates ACK messages in the downlink direction. This means that the ARQ mechanism

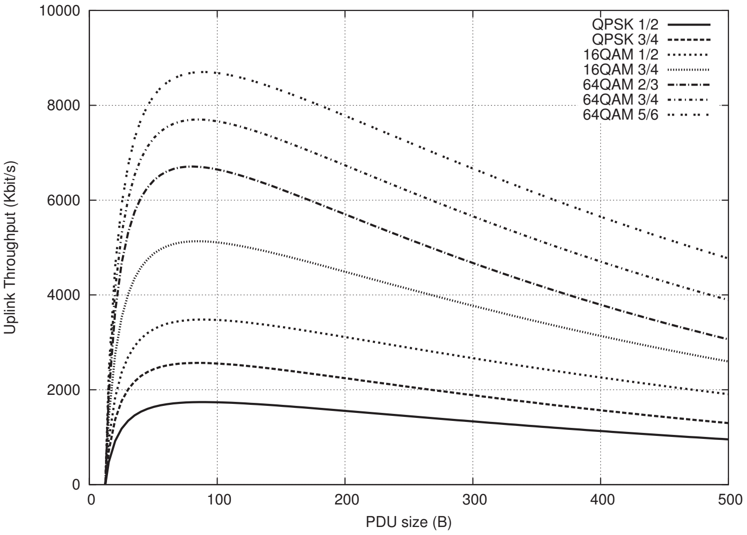


FIGURE 6 Theoretical uplink throughput based on Equation 16 with different MCS. BLER = 10^{-1}, FPS = 200, slots = 245.

TABLE 3 Features supported by WINSE.

PHY
OFDM and OFDMa PHY FEC blocks HARQ: Type I, UL ACK channel Channel reports: REP-RSP and CQICH Link adaptation

MAC
802.16 d/e/j Duplexing modes: TDD, FDD, H-FDD DL broadcast messages: DL-MAP, UL-MAP, DCD, UCD Compressed MAP, sub-MAPs Connections: DL broadcast, basic management, transport PDU construction, fragmentation, packing Bandwidth requests: standalone & piggy-backed ARQ: blocks, feedbacks, timers, transmission window Uplink contention: OFDM and CDMA-based for OFDMa Network entry Handover: SS-initiated, automatic & manual Sleep mode: class I, II, and III

QoS & scheduling
UGS, ertPS, rtPS, nrtPS, BE BS scheduler SS uplink scheduler

Access service network
ASN-GW R4, R6, and R8 interfaces ASN-anchored mobility

needs to send ARQ feedback messages for these TCP ACKs in seed.

A total of 52 combinations of modulation and coding schemes are defined in [13] as burst profiles. In these simulations we use the CTC MCSs because it is the only FEC block coding that is mandated by [18] for both ARQ and HARQ enabled data transmission. Table 2 shows the FEC block and slot sizes of this

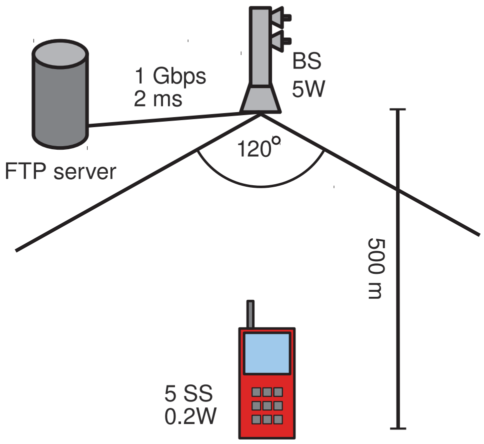


FIGURE 7 Network structure.

TABLE 4 System level parameters.

Parameter	Value
Reuse factor	1/3
Path loss model	802.16m urban macro cell
Fast fading	Jakes model, K=0
Interference level DL/UL	-170 / -167 dBm/Hz
Antenna technique	SISO (1x1)
Antenna pattern BS/SS	3GPP / Omnidirectional
Antenna gain BS/SS	17 / 0 dBi
Antenna height BS/SS	32 / 1.5 m
Tx power BS/SS	5W / 0.2 W
Number of SS	1/5

coding scheme. [1]

2.5.1 Fixed MCS and Constant Error Rate

In the first simulations, there was only 1 SS present. The errors are generated for each FEC block at the given uniform error probability (0.1%, 1% or 10%). The link adaptation is not enabled, and the MCS is fixed to 16QAM 3/4. The purpose of these simulations is to show how the MAC PDU size impacts on the throughput when there is no variable MCS or error probability.

Figure 8 shows the average uplink throughput for the simulation time with a different maximum PDU size limit. Naturally the throughput is higher when there are fewer errors. According to Equation 13, the theoretical optimum PDU sizes for FEC block error rates of $10^{-1}/10^{-2}/10^{-3}$ are 89/273/854 bytes with 1 byte precision. The simulation results show that with 10^{-3} error rate the biggest throughput is achieved when the PDU size limit is roughly 640 bytes or more. With 10^{-2} error rate there are more fluctuations in the results and the largest throughput is achieved with a PDU size limit of 180. It must be noted, though, that the throughput is very similar when the limit is between 120 and 280 bytes.

[1] The WiMAX Forum mobile system profile [18] also mandates the CTC FEC block coding.

TABLE 5 PHY parameters.

Parameter	Value
Frequency band	2.5 GHz
PHY	OFDMa
Cyclic prefix length	1/8
Frames per second	200 (5 ms/frame)
Long preamble	1 symbol
Bandwidth	10 MHz
FFT	1024
TTG+RTG	296+168 PS
DL/UL subchannels	30/35
DL/UL subcarrier alloc.	DL PUSC / UL PUSC
OFDM symbols	47
DL/UL symbols	28/18
DL/UL slots	420/210
Ranging backoff start/end	1/15
Ranging transm. opport.	2
Request backoff start/end	0/15
Request transm. opport.	1
DL/UL channel measurements	preamble / data burst
Channel report type / interval	CQICH / 20ms
Channel measurements filter	EWMA, α = 0.25
UL Power Control	Closed loop
Link adaptation target FEC BLER	$10^{-1}/10^{-2}/10^{-3}$

So the theoretical optimum of 273 bytes is a good choice. When the BLER is 10^{-1}, the 60 byte PDU size limit is clearly the best choice. Overall, the results agree well with the Equation 13.

2.5.2 Link Adaptation and Variable Error Rate

In the previous section it was shown that the presented optimal PDU size estimation method holds true in a simple environment with a fixed error rate and fixed MCS. For simulations in this section, the error rate is not fixed but it is varying due to fast fading. There is no path loss variation because the subscribers are not moving. The details of the PHY model are presented in Section 2.4. The base station is constantly monitoring the quality of UL transmission of each subscriber, and the link adaptation module is selecting the UL MCS of subscribers accordingly. With the FEC BLER curves in Figure 5, the basestation tries to ensure a certain BLER. Also in these simulations there are 5 subscriber stations.

Figure 9 shows the total UL throughput. The different curves correspond to different target FEC block error rates and not to the actual block error rates as in the previous simulation setup. The best throughput is achieved with a BLER of 10^{-2} when the PDU size limit is 100 bytes or more. If the PDU size is less than 10^{-1}, the target gives approximately the same performance. Overall, the best throughput is achieved with a 10^{-2} target BLER and a PDU size limit of 180. This was also the outcome of earlier simulation results [31]. The target BLER of 10^{-3} is never the best option because then LA is very conservative and the more spectral efficient MCSs are not selected.

According to Equation 13, the theoretical optimum PDU sizes for FEC block error rates of $10^{-1}/10^{-2}/10^{-3}$ are 89/273/854 bytes with 1 byte precision. The simulations show that the best throughput is achieved with the PDU size limits

TABLE 6 MAC parameters.

Parameter	Value
MAP MCS	Varied
Compressed MAP	ON
sub-MAPs	OFF
CDMA codes	256
ranging+periodic ranging	64
bandwidth request	192
handover	–
Fragmentation/packing	ON
PDU size	40-1000 B
CRC	ON
ARQ feedback	standalone
ARQ feedback types	all
ARQ feedback interval	20 ms
ARQ block size	64 B
ARQ window	1024
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ timers	
retry	100ms
block lifetime	500ms
Rx purge	500ms

of 120/180/180-360 respectively. One explanation to why theoretical estimation does not match the simulation results is that LA is not able to ensure the BLER to desired level. Figure 10 shows the measured BLER for different target BLER and PDU size limit values. For example for target BLER of 10^{-1} the actual measured BLER is mostly 5-7 %. This means that the PDU size should be larger than the size calculated with the formula. This explains the difference of the theoretical optimal PDU size 70 and the measured 120 bytes. For the target BLER 10^{-2}, the behaviour is opposite. The measured BLER is more than target and this is why the measured optimal PDU size is 180 and not 273 bytes. The same applies to the target BLER 10^{-3}. The reason for the imperfect PDU size selection is that the link adaptation in the basestation is not able to follow the channel perfectly due to delays in CQI mechanisms, filtering applied to CQI measurements and inaccuracies in CQI measurements. Also ARQ mechanism brings uncertainty and distortions to the results.

2.6 Conclusions of PDU Size Optimization

We have proposed Equation 13, which can be used to estimate the optimal size of MAC PDU when the actual BLER of the channel is known. The simulation results show that the estimation method is valid when the actual BLER of the channel is fixed and known by the subscriber station. The results also show that the optimal PDU size can be estimated also when the channel varies and the target FEC BLER for the link adaptation mechanism is known although the estimation is less accurate. This is due to channel variations, imperfections, filtering and delays of the CQI mechanism as well as to added uncertainty caused by ARQ. Because the channel conditions vary, the estimation is less accurate. Also the simulation re-

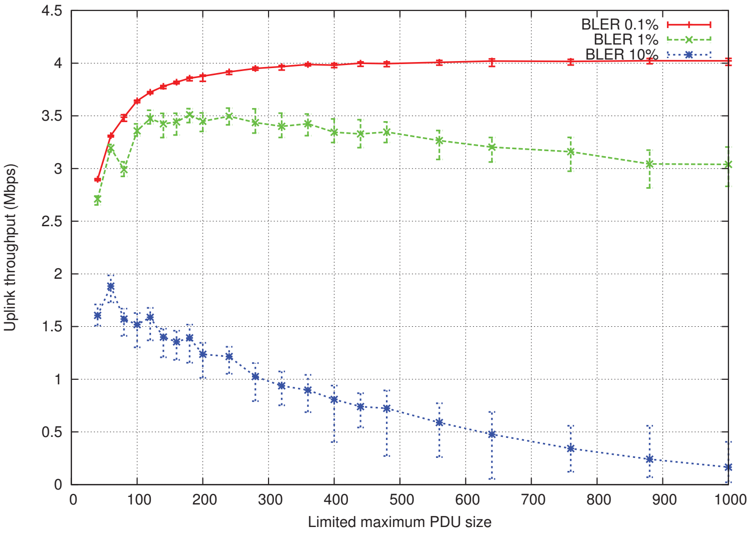


FIGURE 8 1 SS, no LA: Throughput with different FEC target BLER and PDU sizes.

sults show that if the BLER cannot be estimated the PDU size should be between 100 to 200 bytes.

The presented simulations used uplink traffic, on which also the analysis was based. The same ideas can directly be applied to optimal downlink PDU size estimation. In fact, currently the 802.16e specification [13] does not provide any way to signal the preferred PDU size or the link adaptation targets from the basestation to the subscriber station. Thus the subscriber station has to use a conservative small maximum PDU size limit in the uplink direction. This is a clear drawback in the current specification. Adding such a feature would not increase the signalling overhead significantly since the configuration needs be done only once after the connection has been setup.

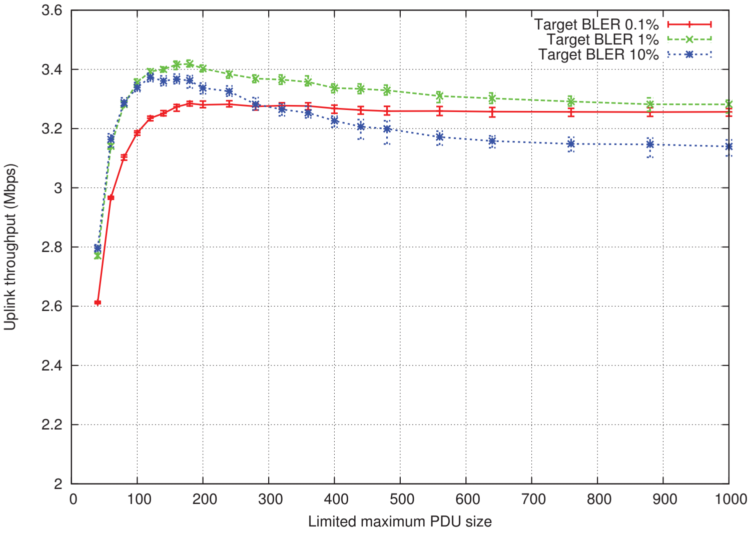


FIGURE 9 5 SS: Throughput with different FEC target BLER and PDU sizes.

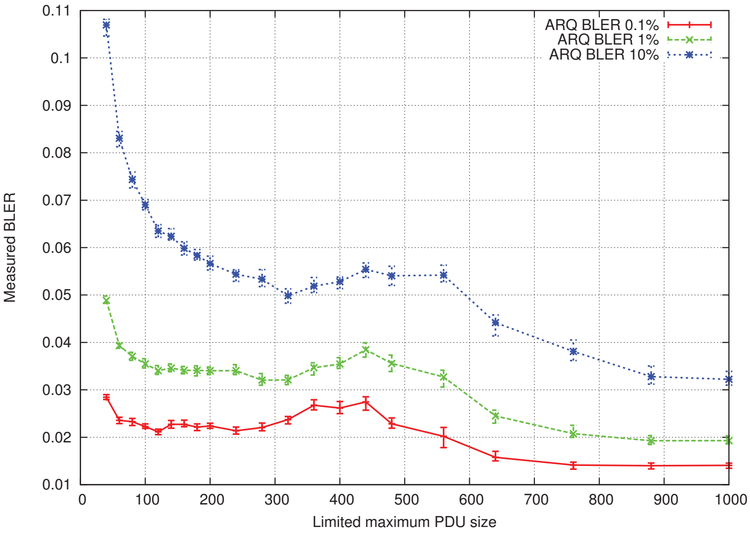


FIGURE 10 5 SS: Measured BLER.

3 RETRANSMISSION MECHANISMS IN IEEE 802.16

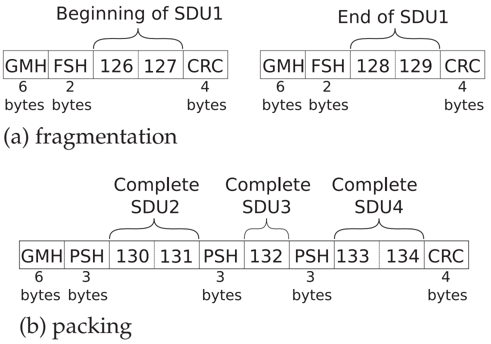
Modern broadband wireless systems provide a number of mechanisms to minimize the errors on the data transmission. In particular, IEEE 802.16 provides two mechanisms: automatic repeat query (ARQ) and hybrid automatic repeat query (HARQ). Both mechanisms are available in the OFDMa PHY, which serves as a basis for mobile 802.16 networks [13]. Both ARQ and HARQ rely on an integrity check to detect channel errors, and use a retransmission process to retransmit lost (i.e., missing or corrupted) data. However, unlike ARQ that works as a part of the upper MAC layer, HARQ requires a more complicated signalling to report ACKs and request retransmissions [32].

This chapter presents ARQ and HARQ mechanisms in the IEEE 802.16 system and evaluates their performance, especially the system throughput. Section 3.1 introduces the ARQ mechanism and shows how its various features impact on the performance. Section 3.4 presents the basics of the HARQ mechanism and compares its performance to ARQ.

3.1 ARQ

If ARQ is enabled for a connection, the extended fragmentation subheader (FSH) or the extended packing subheader (PSH) is used, which is indicated by the extended bit in the general MAC header (GMH). Regardless of the subheader type, there is a block sequence number (BSN) in the subheader that indicates the *first* ARQ block number in the PDU. A PDU is considered to comprise a number of ARQ blocks, each of which is of the same constant size except the final block which may be smaller. The ARQ block size is an ARQ connection parameter negotiated between the sender and the receiver upon a connection setup. It is worth mentioning that the ARQ block is a logical entity, and the block boundaries are not marked explicitly. The remaining block numbers in a PDU can be derived easily on the basis of the ARQ block size, the overall PDU size, and the first block number. Precisely for these reasons the ARQ block size is a constant parameter.

Figure 11 presents ARQ blocks with the fragmentation and packing mechanisms. Block numbers are given with respect to the BSN stored either in the FSH (see Figure 11a) or PSH (see Figure 11b).



(a) fragmentation

(b) packing

FIGURE 11 ARQ blocks with packing and fragmentation mechanisms.

It is important to note that while the 802.16d specification [33] defines an ARQ block size as *any* value ranging from 1 to 2040 bytes, the 802.16e specification [13] has limited it to power of two values ranging from 16 to 1024 bytes, e.g., 16, 32, 64 and so on.

3.1.1 ARQ Timers

The IEEE 802.16 specification defines several ARQ timers. Figure 12 shows how they relate to the ARQ block states. The ARQ block may be in one of the following five states: *done*, *not-sent*, *outstanding*, *discarded*, and *waiting-for-retransmission*. Firstly, as can be seen from Figure 12, any ARQ block begins as *not-sent*. After it has been sent, it becomes outstanding for a period of time termed ACK_RETRY_TIMEOUT, which determines the minimum time interval a transmitter can wait before retransmission of an unacknowledged block for retransmission. The interval begins from the last transmission of the ARQ block. While a block is in *outstanding* state, it is either acknowledged and changed to *done*, or transitions to *waiting-for-retransmission* after ACK_RETRY_TIMEOUT or NACK.

An ARQ block can become *waiting-for-retransmission* before the ACK_RETRY_TIMEOUT period expires if it is negatively acknowledged. An ARQ block may also change from *waiting-for-retransmission* to *done* when an ACK message for it is received or to *discarded* after a timeout ARQ_BLOCK_LIFETIME, which determines the maximum time interval an ARQ block can be managed by the transmitter ARQ state machine, once the initial transmission of the block has occurred. If transmission (or subsequent retransmission) of the block is not acknowledged by the receiver before the time limit is reached, the block is *discarded* [34].

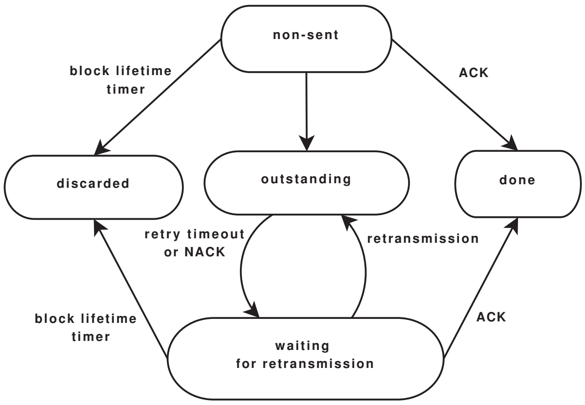


FIGURE 12 ARQ transmit block states.

3.1.2 ARQ Feedback Types

To request a retransmission of blocks (NACK) or to indicate a successful reception of blocks (ACK), a connection uses ARQ block sequence numbers. In turn, the sequence numbers are exchanged by means of ARQ feedback messages. The specification defines the following feedback types: a) selective, b) cumulative, c) cumulative+selective, and d) cumulative+sequence.

The selective feedback type acknowledges ARQ blocks received from a transmitter with a BSN and up to four 16-bit selective ACK maps. The BSN value refers to the first block in the first map. The receiver sets the corresponding bit of the selective ACK map to zero or one, according to the reception of blocks with or without errors, respectively. The cumulative type can acknowledge any number of the ARQ blocks. The BSN number in the ARQ feedback means that all ARQ blocks whose sequence number is equal to or less than BSN have been received successfully. The cumulative+selective type just combines the functionality of the cumulative and selective types explained above. The last type, cumulative+sequence, combines the functionality of the cumulative type with the ability to acknowledge reception of ARQ blocks in the form of block sequences. A block sequence, whose members are associated with the same reception status indication, is defined as a set of ARQ blocks with consecutive BSN values. A bit set to one in the sequence ACK map entity indicates that a corresponding block sequence has been received without errors, and the sequence length indicates the number of blocks that are members of the associated sequence.

When the ARQ feature is declared to be supported, a transmitting side, i.e., a receiver of the ARQ feedbacks, must support all the feedback types described by the 802.16 specification. The sender of the ARQ feedbacks has the ability to choose whatever format it will use. The WiMAX Forum recommendations [18] mandate the support of all the types except the selective ACK.

Figure 13 presents an example in which every feedback type is applied to the same set of ARQ blocks. Selective ACK can acknowledge these 32 blocks in two maps. Cumulative ACK cannot acknowledge all the blocks because there

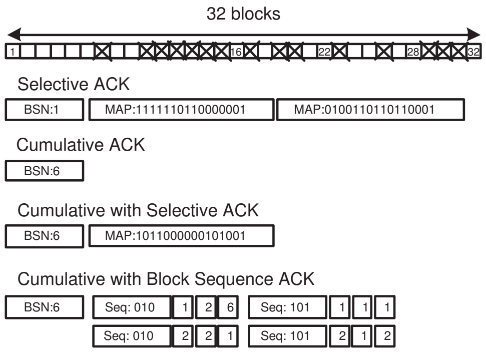


FIGURE 13 Example of ARQ feedback types.

are negative acknowledgements. Thus, only six blocks are encoded. Cumulative+selective ACK can send both positive and negative acknowledgements. However, since there should be 16 blocks per one selective map, some blocks remain unacknowledged. For this particular example, cumulative+sequence ACK can acknowledge only 28 blocks; one message can hold four sequence maps at most, whereas each map can have either two or three sequences. This type does not work effectively in this case because the block sequences are very short.

3.1.3 ARQ Block Rearrangement

While retransmitting a PDU, a connection may face a problem where an allocated data burst is smaller than the PDU size to be retransmitted. This may happen if the BS scheduler allocates data bursts of different sizes, which is usually the case for real-time Polling Service (rtPS), non-real-time Polling Service (nrtPS), and Best Effort (BE) connections. Suppose that the BS allocates a data burst of three slots for the BE connection, and the latter sends a PDU that spans the whole data burst. If this PDU encounters an error, the connection will retransmit it. However, if the BS scheduler allocates later a data bursts of two slots, there is no way to retransmit the original PDU. Fortunately, the connection may rely upon *the retransmission with rearrangement* that allows for fragmenting the retransmitted PDU on the ARQ block size boundaries. If there is a sufficiently small ARQ block size, then the connection may construct a smaller PDU. As an example, Figure 14 shows the rearranged PDU presented in Figure 11a. There are two PDUs with two blocks per each PDU.

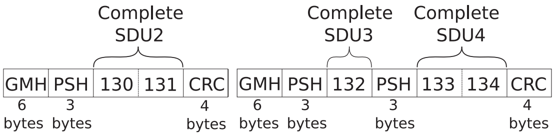


FIGURE 14 Rearranged PDU.

In this subsection we do not focus on the optimal ARQ block size, but rather consider a solution for a case where a sender retransmission policy is not to use the ARQ block rearrangement. The reason this functionality can be absent is the fact that rearrangements involve much more complicated actions with PDUs in the retransmission buffer when compared to the PDU construction. A sender must keep a set of the ARQ timers for each ARQ block. If the retransmission with rearrangement is not implemented, then eventually the sender can associate all those timers with a PDU, which requires much less resources.[1] Furthermore, the rearrangement requires the sender to analyze the PDU and to search for block boundaries on which that PDU can be fragmented.

It is important to note that this problem concerns merely the uplink connections, because having the bandwidth request size, the BS does not know whether it is one big PDU or several smaller ones. In the case of the downlink transmission, the BS can always look inside the queue. Besides, this problem would not be so critical if the BS knew that the connection does not support rearrangements. However, there is no such QoS parameter that would indicate it. On the one hand, the BS can guess that a connection does not rearrange PDUs by monitoring bandwidth request sizes and the number of received bytes. On the other hand, a connection should not rely much upon this functionality because it is not mandated by the specification. Thus, the only safe way is to control the maximum size of transmitted PDUs. It is not a complicated task for the rtPS and nrtPS connections, which should always be allocated such a number of slots that their minimum bandwidth requirements are ensured [28]. Thus, the maximum PDU size can be limited by the minimum data burst size allocated by the BS scheduler. The BE scheduling class is more challenging since the BS scheduler can allocate a data burst of any size. A connection may monitor allocated data burst sizes to control the maximum PDU size. Another possible solution is to send a PDU of the size of one slot. However, such an approach may be unacceptable due to the increased MAC overhead and very small slot size of robust MCSs. As a result, regardless of an approach taken, the BE connection, which does not support retransmissions with rearrangements, should avoid sending large PDUs. In all the cases, Figure 17 should be obeyed. L is the PDU size limit and S_{burst} is the average size of the burst for BE connections or the minimum requirement for rtPS, nrtPS and UGS connections.

$$L < S_{burst} \tag{17}$$

3.1.4 ARQ Transmission Window and ARQ Block Size

At any time a sender may have a number of outstanding and awaiting acknowledgements ARQ blocks. This number is limited by the ARQ transmission window that is negotiated between an SS and the BS during a connection set-up. A

[1] Practically, a sender can associate a timer with a whole PDU even if the ARQ block rearrangement is turned on. However, then it has to perform quite complicated actions with ARQ timers when the retransmitted PDU is partitioned into several PDUs because certain ARQ blocks are retransmitted while the other ones remain in the output buffer.

sufficiently large ARQ window allows for a continuous transmission of data. A connection can continue to send ARQ blocks without waiting for each block to be acknowledged. Conversely, a smaller ARQ window causes a sender to pause a transmission of new ARQ blocks until a timeout or the ARQ feedback is received. Though it may seem that a large transmission window is always the best choice, it is worth noting that a large transmission window leads to increased memory consumption and processing load. Every ARQ block must be stored in the retransmission buffer until a positive feedback is received. Taking into account the largest ARQ block size of 1024 bytes and the maximum ARQ transmission window of 1024 blocks, it is possible to arrive at the conclusion that some mobile and portable devices will not have enough resources to handle this amount of data for each frame.

If we assume a continuous errorless data transmission, then the maximum throughput a connection can achieve is limited by the following expression:

$$\frac{S^{\mathrm{ARQ}}\, W\, \mathrm{FPS}}{\mathrm{DF}}, \tag{18}$$

where S^{ARQ} is the ARQ block size, W is the ARQ transmission window size, FPS is the number of frames per second and DF is the delay factor. In the case of the downlink transmission, the delay factor is always 1 because the BS can allocate a downlink data burst whenever it wants. In the case of the uplink transmission, the delay factor depends on PHY and whether polling is in effect. If the BS polls a connection in *every* frame, then the delay factor is also 1. Otherwise, like in the case of the BE connections, the delay factor is modulation and coding scheme 2 for OFDM and 3 for OFDMa PHY. The reason is that in OFDM PHY, the uplink bandwidth request carries the request size, while in the OFDMa PHY, special CDMA codes are used that do not carry any request size. As a result, once the BS receives the CDMA code, it puts a special uplink CDMA allocation where an SS can transmit the request size.

The ARQ transmission window and the ARQ block size parameters depend on each other. On the one hand, a connection may prefer to work with a small ARQ transmission window. This will result in a necessity of choosing a larger ARQ block size because the throughput may be limited by the transmission window size. A large block size requires fewer resources because a set of the ARQ timers must be associated with a single ARQ block at the sender and at the receiver. At the same time, a connection supporting the retransmission with rearrangement may wish to work with a smaller ARQ block size because that will provide greater flexibility in splitting large PDUs into several smaller ones. Furthermore, the choice for the ARQ block size can be dictated by the device peculiarities, such as the memory page size. These various requirements introduce a cyclic dependency between these two parameters.

We anticipate that the ARQ block size should be the governing parameter, while the ARQ transmission window size should be adapted. The reason is that the ARQ block size has a set of discrete values, while the ARQ transmission window can accept any value within the specified rearrangement range.

3.2 ARQ Simulations

NS2 with WINSE extension was used to run following results. The Figure 15 shows the general network structure. Subscriber stations (SS) are connected to an IEEE 802.16 base station in this scenario. A file server is connected to the base station using a fast low-latency wired connection. The number of SSs and direction of FTP traffic depends on the particular simulation subcase.

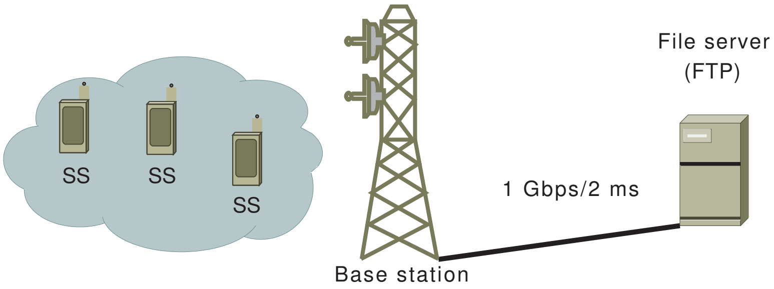


FIGURE 15 Network structure for ARQ simulations.

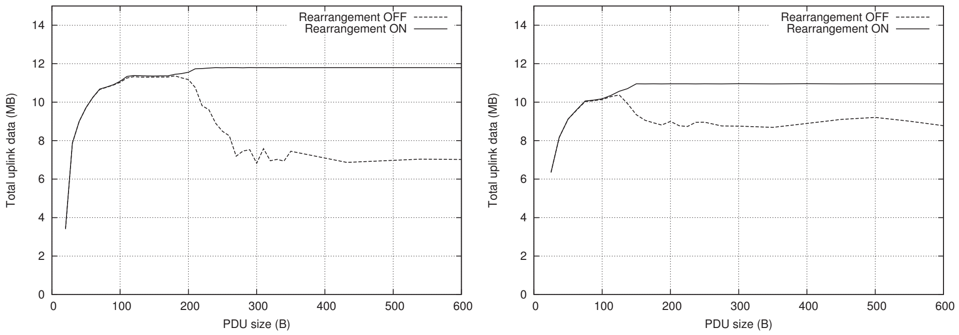
3.2.1 ARQ Block Rearrangement

In this subsection we study the impact of ARQ block rearrangement on the uplink throughput. Figure 16a shows the total transmitted data with and without an ARQ block rearrangement using different MAC level PDU size limits. It can be seen that the best results are obtained by using a rearrangement and PDU size of 200 bytes or more. Without rearrangement performance drops significantly if large PDUs are used. This can be prevented by limiting the PDU size to less than average burst size. In this case the average burst size is about 200 bytes. This also the reason why the performance is the same for PDU sizes larger than 200 bytes: all the subscriber stations are scheduled in each frame and they never have more than 200 bytes of allocated data.

Note that the average burst size depends heavily on the load of the network, type of traffic and scheduler. Figure 16b shows how the PDU size limit is even lower with 30 SS, which will cause more header overhead and decrease the performance. In both cases, it is clear that the ARQ block rearrangement improves the performance. However, if the rearrangement implementation is too complicated for a SS it can still achieve reasonable performance without it by limiting the MAC level PDU size.

3.2.2 ARQ Feedback Types

In this simulation subcase, we study ARQ feedback types. In all the cases, the simulator chooses an appropriate feedback from the allowed ones using an algo-



(a) 20 SS

(b) 30 SS

FIGURE 16 Impact of ARQ block rearrangement and PDU size on total uplink data.

rithm presented in [35]. The traffic is downlink FTP traffic, making it SS's responsibility to create ARQ feedbacks, which are studied here.

Figure 17 show four different simulation cases. In this case, downlink FTP-traffic was used, and SSs are responsible for creating the ARQ feedback messages. In the first case, only selective ARQ feedback type is used. It is clear that this is not efficient since all the blocks have to be acknowledged explicitly. In the second case, also standalone cumulative feedbacks are used. This increases the performance significantly since all the correctly received blocks can be acknowledged with a single cumulative message. In the third case, a combined cumulative+selective type is also used, which again boosts the performance although not as much as in the previous case. The boost is achieved because the combined type can store the information, which is the same as in the separated cumulative and selective types, in one message and therefore reduce the overhead. Finally, in the fourth case also the cumulative+sequence type is allowed. Also this type increases the performance because it can acknowledge more blocks in one message than the cumulative+selective type can.

The only case when the ARQ block size has any significant impact is when only the selective feedback type is used. In that case, the ARQ window of 1024 blocks is not always enough if ARQ block size is 16 bytes. The selective feedback is not able to acknowledge the transmissions as efficiently as the other feedback types, and the ARQ window becomes full in some cases. The impact of ARQ block size would be more significant if there were fewer subscribers and thus more data would be sent to them.

In conclusion, it can be seen that at least cumulative type and the selective type should be used. The extra benefit from cumulative+selective and cumulative+sequence is much smaller. So if creating cumulative+sequence takes too many resources on an SS, the SS can decide not to use that type and still it will not have a major impact on the performance. WiMAX Forum Mobile System Profile [18] mandates that the support for all the types, but the selective one is mandatory. This means that an SS has to able to receive those types but does not mandate the SS to use them. Indeed if the cumulative+sequence type is supported, our feedback selection algorithm does not select the selective type at all; hence the result is exactly the same as in case four.

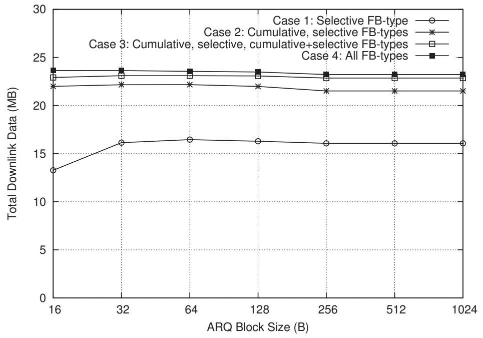


FIGURE 17 Impact of different ARQ feedback types and ARQ block sizes on total downlink data (20 SS).

3.2.3 ARQ Window and ARQ Block Size

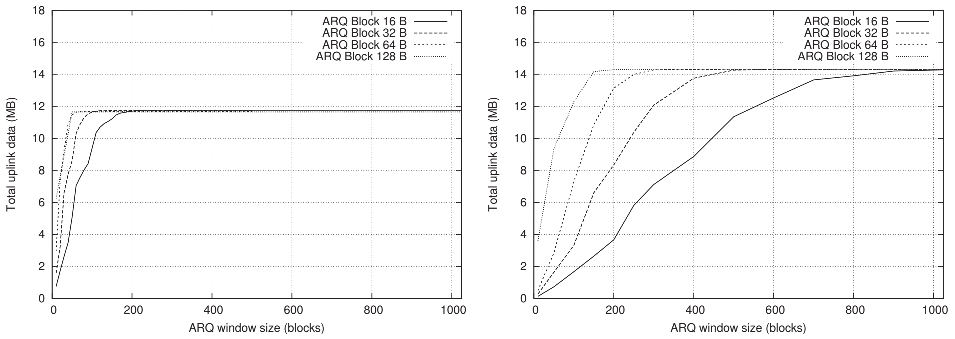
In this simulation scenario, we present the simulation results for different ARQ window and ARQ block size values. Figure 18a shows the results when there are 20 SSs receiving FTP traffic. It can be seen that if the ARQ window is more than 200 blocks it does not restrict the performance at all. Also the results for different block sizes of 16-128 are almost the same if the window is more than 200 blocks.

Figure 18b shows the same scenario when there are only 5 SSs present. This case has similar characteristics compared to the previous one. If the ARQ window is big enough, then the ARQ block size does not matter. However, if the ARQ window is for example 300 blocks, there is a big difference in total uplink data between the ARQ block sizes of 16-128 bytes.

Also if we analyze also the results from Figure 17 with different ARQ block sizes it is clear that the smallest block size is not an optimal selection. In practice, there is no difference between block sizes of 16 and 128 in performance, but the first one requires 8 times more resources for ARQ timers.

3.2.4 ARQ and VOIP Simulations

Although 802.16 has sophisticated QoS classes, which are better suitable for delay critical applications such as VoIP [36], it might be that real networks have only BE subscriptions, as it is the case with current broadband connections. Even if an operator would have its own VoIP services, the users might want to use some other VoIP service, e.g. Skype. Also, even though HARQ is more suitable for VoIP applications due to it's faster feedback mechanism the ARQ mechanism may be more suitable for BE connections. This is why it is important to study the performance of the combination of the BE connection, ARQ and VoIP.

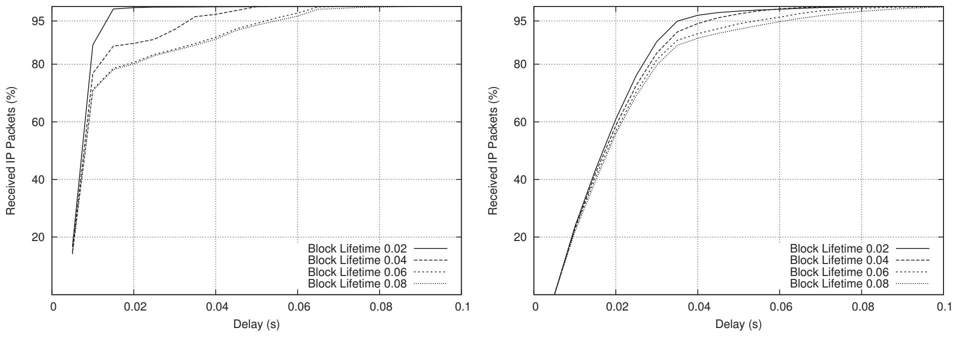


(a) 20 SS

(b) 5 SS

FIGURE 18 Impact of ARQ window size and ARQ block size on total downlink data.

First, we analyse whether the ARQ block lifetime can ensure an upper limit for VoIP delay. For these purposes, we run the same simulation scenario with different ARQ block lifetime values. The downlink VoIP E2E delay CDFs with different block lifetime values are presented in Figure 19a. It shows that the block lifetime does actually limit the maximum delay. The same results for uplink are shown in Figure 19b. However, it can be seen that the block lifetime does not restrict the maximum delay as expected. The reason for this is that the ARQ block lifetime is started only when the ARQ block is transmitted for the first time. Before that, a packet can spend some time in the output buffer waiting for being transmitted. This is especially the case for the uplink BE connection, where an SS has to take part in the uplink contention. Nevertheless, the maximum delay is less than 100 ms and the 95 percentile is less than 60 ms (for uplink) and 50 ms (for downlink).



(a) Downlink

(b) Uplink

FIGURE 19 VoIP IP end-to-end delay CDF.

In addition to the VoIP delay distribution, Figure 20 presents information on the number of dropped VoIP packets. As follows from the results, a small VoIP packet delay is achieved by a significant number of dropped packets due to the expired ARQ block lifetime. At the same time, ARQ block lifetime value of 80 ms provides more than a satisfactory number of transmitted packets, where the number of dropped packets is less than 1%.

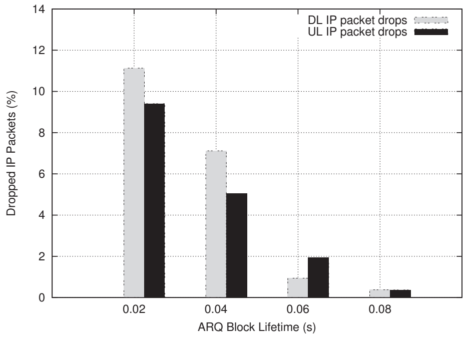


FIGURE 20 Dropped VoIP packets with different ARQ Block Lifetime values.

Based on the these results, it is possible to arrive at the conclusion that ARQ block lifetime of 80 ms is a good tradeoff between the delay requirements and the number of dropped packets.

3.3 HARQ

Hybrid automatic repeat request (HARQ) is a combination of the FEC and ARQ methods. If the packet is received without errors, then the behaviour similar to ARQ, i.e. the sender is signalled that the packet was successfully received. However, if the packet contains errors, instead of discarding the packet as with ARQ it is kept in the receiver's buffer. Then the sender will send the packet again, possibly with additional FEC information and the receiver can combine the information of all transmissions. Thus the probability of successful decoding increases with every retransmission. The benefit of ARQ over HARQ is that it uses a lot less of redundant data to detect the errors. In principle, ARQ based methods are more efficient when the error probability is lower and HARQ methods when the error probability is higher [14, 15].

IEEE 802.16 supports both Type I and Type II HARQ methods. Type I is often referred to chase combining, and with it the redundancy information is kept the same for retransmissions. Type II, incremental redundancy, changes the puncturing pattern with each retransmission, which, more than with Type I, increases the probability of correct decoding of the transmission. Type II can lead to better performance at the cost of increased complexity. Type I was the only mandatory HARQ Type in Release 1.0 of Mobile WiMAX, but Type II has been since made mandatory for WiMAX Mobile System Profile Release 2.0 [37, 18, 38, 39].

The MAC level PDU overhead of HARQ is somewhat less or comparable to the ARQ mechanism. Firstly, the sender must reserve 2 bytes at the end of a

HARQ sub-burst to include the HARQ CRC-16 field. Though there is no need carry the per-PDU CRC-32 field, there is a PDU sequence number (SN) extended subheader that occupies 4 bytes. It should be noted that the PDU SN is optional. However, it is anticipated that it is turned on for most services. Otherwise, PDUs can arrive in the wrong order to the upper MAC causing SDU reassembly problems. Furthermore, if SDUs are delivered in the wrong order to a receiver, it may result in a decreased performance at the application level. Figure 21 illustrates a HARQ sub-burst with one MAC PDU. It is anticipated that there will be one MAC PDU per a HARQ sub-burst because the whole HARQ sub-burst is retransmitted if an error is detected. Though it is possible to have a number of PDUs in a single HARQ sub-burst, it results in a larger MAC overhead.

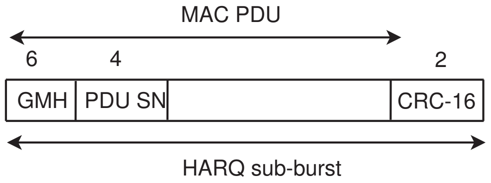


FIGURE 21 HARQ PDU.

The MAP signalling overhead introduced by the HARQ mechanism depends a lot on how the scheduler allocates resources. It makes sense to mention that in 802.16 networks, HARQ data is allotted in a form of HARQ sub-bursts, where a sub-burst is a *one-dimensional* entity that occupies slots in the frequency-first order. Multiple HARQ sub-bursts can be placed into one burst in a two-dimensional allocation. If we assume that all the HARQ sub-bursts are located in a single burst, then the MAP overhead is not large. Figure 22 illustrates a data burst with several HARQ sub-bursts. Conversely, placing a HARQ sub-burst in an independent data bursts creates a large overhead because the MAP message has to encode the data burst and clarify the HARQ sub-burst configuration, e.g., the number of slots, MCS, HARQ mode etc.

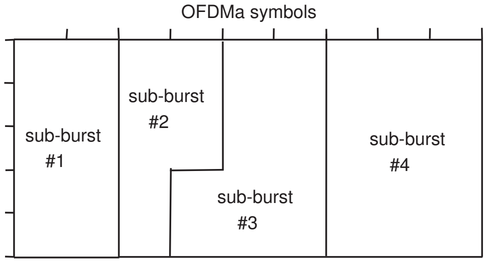


FIGURE 22 HARQ sub-bursts within one data burst.

In addition to HARQ enabled data bursts, the MAP signalling overhead comprises information on downlink slots to transmit ACKs for the uplink bursts, and uplink slots for ACKs sent by SSs for the downlink bursts. Downlink ACKs do not create a significant overhead: there is just a bitmap, where each bit corresponds to a particular burst. The uplink HARQ ACK channel is more demanding. Each HARQ downlink sub-burst requires an uplink transmission opportunity to transmit the HARQ ACK message. Though a single UL ACK message requires only half of an uplink data slot, the resulting overhead may reduce noticeably the amount of available uplink resources. Besides, the HARQ UL ACK channel is a two-dimensional uplink allocation that poses additional constraints for the scheduler.

The HARQ mechanism introduces significant constraints to the BS scheduler. While the initial HARQ transmission can be of any size, the subsequent HARQ retransmissions must be of exactly the same size. Taking into account two-dimensional data allocations in the OFDMa frame, it is easy to imagine the complexity of this problem. Furthermore, as mentioned above, there is also the HARQ UL ACK channel that must be placed in the uplink part.

Taking all the presented assumptions into account, we anticipate that one data burst will always contain one HARQ sub-burst and one HARQ sub-burst will contain one MAC PDU. So unlike with ARQ the MAC PDU size is always unlimited. And as mentioned, HARQ retransmits the whole data burst when an error is detected. However, unlike ARQ, the HARQ can use the information of the first transmission and the success probability increases with each retransmission. Still, as the large data bursts have a higher probability of being dropped, the BS scheduler should consider making smaller allocations. Such a requirement may conflict with certain scheduling policies, such as proportional fair, where the scheduler tends to allot slots when an SS has a good channel performance or where large allocations are made at large time intervals.

3.4 HARQ Simulations

In this section, we will have a look at the performance comparison of the ARQ and HARQ mechanisms. The downlink performance depends a lot on how many data bursts the BS scheduler allocates per single frame, and we conduct simulations where we vary that parameter. The simulation parameters are presented in Table 7. The general network layout is the same as in ARQ simulations and it is visible in Figure 15 except that there are 32 subscriber stations connected to the base station.

Figure 23 presents the total amount of data transferred in the downlink direction during the simulation run with different retransmission mechanisms and number of bursts.

In general, HARQ outperforms the ARQ retransmission mechanism, which is especially the case for medium data bursts i.e. 8 to 16 bursts per frame. ARQ achieves the best performance with large data bursts. That is because with the ARQ the PDU size within the data burst is limited to the optimal value of 140 bytes which is based on Chapter 2 results. Having more burst means that more

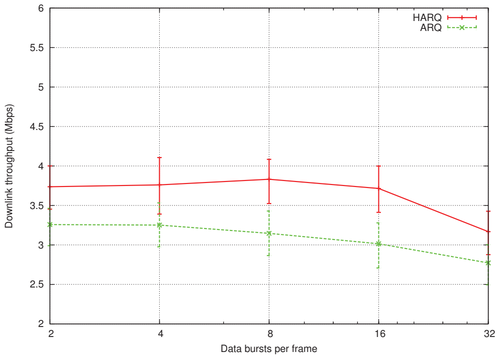


FIGURE 23 The downlink throughput with 32 SS.

signalling overhead is present but the probability of an error per PDU remains the same.

With HARQ, on the other hand, there is only one PDU per HARQ burst. The reason is, as explained in the previous section that the whole HARQ burst will be retransmitted in the case of an error. This means that it makes sense having more bursts per frame. Thus the optimal HARQ performance is achieved with 8 bursts per frame. With more subbursts per frame the signalling overhead of HARQ becomes higher and with 32 bursts the difference to ARQ is smallest. The signalling overhead problem of MAP messages can be eased with sub-maps which utilize different MCS depending on the quality of the channel [40]. Also, the reason why HARQ performance does not drop significantly with large data bursts is that even though a large data burst has a higher probability of being dropped, the HARQ retransmission gain helps when the same data is retransmitted.

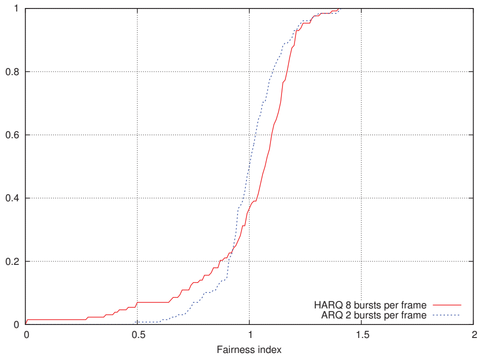


FIGURE 24 The downlink fairness with 32 SS.

Figure 24 shows the throughput fairness. The number of bursts per frame is 8 for HARQ and 2 for ARQ. These burst counts were selected because they provide the best throughput with a given retransmission mechanism. It can be seen that even though there is only 2 bursts per frame for ARQ it still can provide better fairness across subscribers. One explanation for the HARQ fairness tail is that in some cases all the HARQ retransmissions failed. This means that the incomplete SDU is not delivered to upper layers but dropped. Eventually TCP timers will expire and SDU is retransmitted but this will momentarily decrease the performance of that particular subscriber.

3.5 Conclusions of Retransmissions Mechanisms

In this chapter, we have presented the IEEE 802.16 ARQ and HARQ features. The impact of different ARQ features was analyzed. It was shown that it is important to have ARQ block rearrangement feature implemented and that using selective ARQ feedback type alone is not efficient. Also, it was shown that the combination of small ARQ block size and small transmission window might limit maximum throughput especially when there are only few subscribers present. The VoIP topic was also investigated, and it was shown that ARQ block lifetime can be used to limit VoIP end-to-end delay at the expense of increased packet drops.

Finally, we compared the performance of HARQ and ARQ while changing the number of scheduled bursts per frame parameter. Overall, the HARQ had better throughput and its performance was best with 8 bursts per frame. Less bursts meant increased error probability, and with more bursts the increased overhead decreased the performance. ARQ PDU error probability however is controlled by the PDU size, and having more bursts than two per frame meant increased overhead. While HARQ provided better performance in terms of throughput, it also brought more scheduling restraints. Also the throughput fairness of HARQ is worse. For ARQ, on the other hand, the optimal ARQ parameters play important role.

TABLE 7 802.16 network parameters.

Parameter	**Value**
PHY	OFDMa
Bandwidth	10 MHz
FFT	1024
Cyclic prefix length	1/8
TTG+RTG	464 PS
Duplexing mode	TDD
Frames per second	200 (5 ms per frame)
OFDM symbols	47
DL/UL symbols	32/15
DL/UL subcarrier alloc.	DL PUSC/UL PUSC
DL/UL slots	480/175
DL/UL channel measurements	preamble / data burst
Channel report type / interval	CQICH / 20ms
Channel measurements filter	EWMA, $\alpha = 0.25$
MAP MCS	QPSK1/2
Compressed MAP	ON
MAP errors	ON
Ranging transm. opport.	2
Ranging backoff start/end	1/15
Request transm. opport.	8
Request backoff start/end	3/15
CDMA codes	256
ranging+periodic ranging	64
bandwidth request	192
handover	–
HARQ	Type I (CC)
HARQ channels	16
HARQ buffer size	2048 B (per channel)
HARQ shared buffer	ON
HARQ max. retransmissions	4
HARQ ACK delay	1 frame
PDU SN	ON
PDU SN type	long (2 bytes)
Fragmentation/packing	ON
PDU size	140 B
CRC/ARQ	ON
ARQ feedback	standalone
ARQ feedback types	all
ARQ feedback interval	25 ms
ARQ block size	64 B
ARQ window	1024
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ timers	
retry	100ms
block lifetime	500ms
Rx purge	500ms

4 IEEE 802.16 FRAME OPTIMIZATION

In this chapter, the IEEE 802.16 frame optimization methods are introduced. First, the duplexing modes are presented and analyzed. WiMAX system profile allows three different duplexing modes, and each of them have their benefits and drawbacks. Also we propose a solution for H-FDD group ratio and group assignment problems and analyze them. Secondly, the IEEE 802.16 relays are introduced, and the frame partitioning for access and relay is optimized and analyzed.

4.1 Duplexing Modes

Similar to 3GPP LTE, the IEEE 802.16 specification defines two major duplexing modes: TDD and FDD. However, unlike the 3GPP LTE basic deployment scenario, WiMAX Forum chose the TDD frequency bands, FDD having been excluded from the early versions of the system profile. The reason is that the TDD system is simpler in design thus requiring less expensive chipsets at the terminal side. However, lack of a proper FDD support meant that WiMAX could not be deployed to the FDD bands where downlink and uplink reside on different non-adjacent frequency bands. As a result, recent advances in both the IEEE 802.16 standard [41] as well as the system profile [42] added the FDD duplexing mode. To keep the terminal cost as low as for TDD, a special form of the FDD mode, referred to as H-FDD, was proposed. Thus, the 802.16 system can be considered in three duplexing modes: TDD, F-FDD, and H-FDD.

In this chapter, we analyze the duplexing modes available in the 802.16 system and compare their performance. In addition, we study the H-FDD group assignment algorithm and group ratio problems from the IEEE 802.16 MAC and PHY perspective. Although network planning, frequency planning and system design problems for each duplexing modes in the simulations are mentioned, these topics are not considered.

4.1.1 TDD

Time Division Duplexing (TDD) frame structure is presented in Figure 25. One of the main benefits of TDD is that the ratio between the downlink and uplink subframes can be adjusted flexibly, thus adapting the system throughput to operator demands. Still, the whole operator network should have the same ratio. Otherwise, the downlink data from neighbouring cells will cause severe interference to the uplink data transmission in the other cell. This is a drawback since the operator might want to optimize the ratio on a cell-by-cell basis if the traffic profile of different cells varies. The radio implementation in TDD is simpler since there is no need to send and receive simultaneously. It is enough to have a single radio interface with a single encoding/decoding chain. This is particularly important for subscriber stations because it simplifies the hardware design and therefore makes the device cheaper [43]. The uplink MAP message can point to the uplink subframe in the same or next frame. With F-FDD and H-FDD the uplink allocation is always pointing to the next uplink subframe.

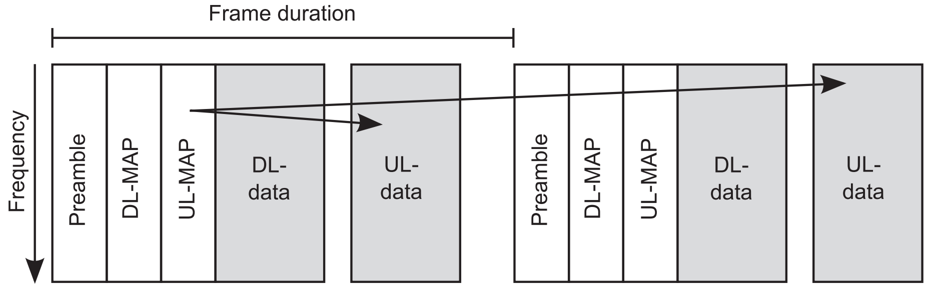


FIGURE 25 TDD frame structure.

One of the drawbacks of TDD is a relatively small uplink subchannelization gain, especially if the downlink sub-frame is considerably larger than the uplink one. It may become a limiting factor for many services and makes operation at the cell edge much more challenging.

4.1.2 F-FDD

Figure 26 presents the frame structure for Full Duplex Frequency Division Duplexing (F-FDD). The uplink and downlink subframes reside on different frequencies, where the frequency bands do not need to be adjacent. This is also a benefit of FDD modes and it means that an operator can use two narrow channel bandwidths where TDD could use only one. The downside of F-FDD is that subscriber stations must be able to send and receive at the same time. This makes the radio implementation more expensive. Also the antenna design might become more complicated if the downlink and uplink reside at frequency bands that differ considerably. Because the frequency bands and channel bandwidths given to the operator are fixed, it is impossible to change the ratio between the downlink and uplink subframes. This is not a desired feature if the network is used as a last-mile data connection, where the traffic nature is very asymmetric. However, even though a typical FDD frequency allocation assumes two bands of the same size, the WiMAX Forum system profile allows bands of different size, e.g., 10 MHz bandwidth for downlink and 5 MHz for uplink. Nevertheless, this is less flexible than in the TDD system.

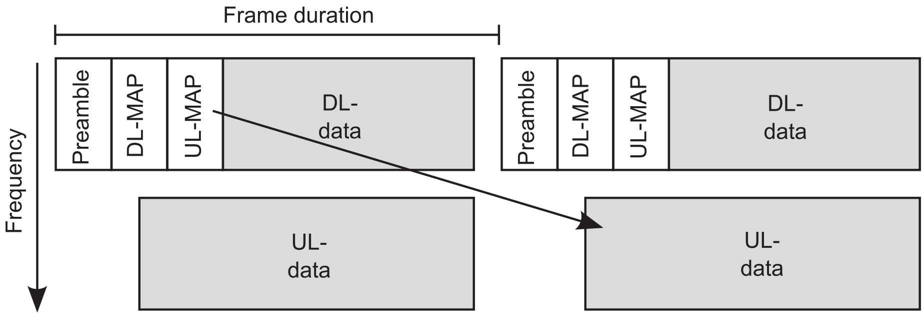


FIGURE 26 F-FDD frame structure.

One of the tempting features of FDD in OFDMA is subchannelization gain. If a subscriber uses fewer subchannels, it can use more power per subchannel and thus increase the received signal-to-noise ratio (SNR). A typical F-FDD system with 5+5 MHz channel bandwidth has twice the number of slots per subchannel when compared to a typical TDD system with a 10 MHz channel bandwidth. This means that subscribers in the F-FDD system use less uplink subchannels for the same amount of data as used in TDD and thus might get better uplink throughput.

4.1.3 H-FDD

The introduction of the H-FDD duplexing mode aimed at solving a few problems of F-FDD. To transmit and receive simultaneously, a terminal must have two antennas with two related radio processing chains. Theoretically, a terminal with one radio chain might have announced itself as being able either to receive or transmit, but then the scheduling becomes much more complicated [44]. The BS must ensure that downlink and uplink bursts are not scheduled for the same terminal at the same moment of time. The H-FDD divides each sub-frame into two groups in such a way that downlink and uplink transmissions occur at different moment of times. The group 1 downlink subframe and group 2 uplink subframe start the frame. A subscriber station belongs logically to one of the aforementioned groups.

There are a number of drawbacks in the H-FDD duplexing mode. Firstly, due to the MAP messages transmitted in both groups, the downlink overhead increases. The total number of burst information included in the MAP messages remains the same, but there are some fixed signalling parts in the MAP messages and also in the DCD/UCD messages which must be sent twice instead of just once. Secondly, there are more gaps on the uplink to allow a terminal to switch

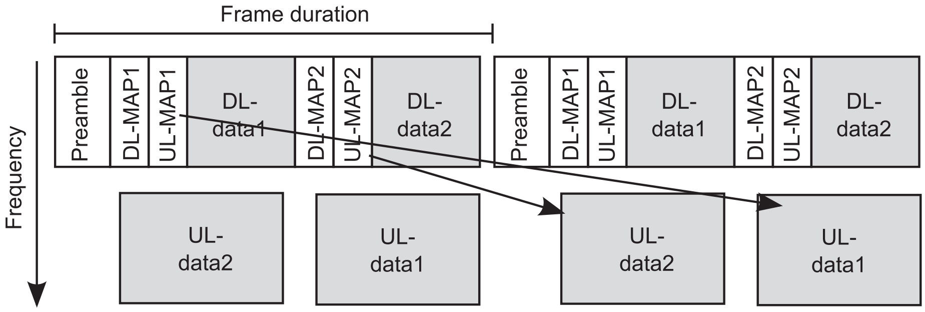


FIGURE 27 H-FDD frame structure.

from a transmitting to a receiving mode. From the network point of view, H-FDD creates a need to implement and run a few additional resource management algorithms that are not present in other duplexing modes.

4.1.4 H-FDD Group Ratio

One of the key questions in the H-FDD system is how to partition the H-FDD frame into two groups. The group ratio plays a crucial role in the system performance, as it allows a subchannelization gain for stations that reside at the cell edge.

A group ratio of 1:1 is the simplest solution for partitioning the H-FDD frame. However, when compared to the TDD mode, this does not provide a significant gain since each group has approximately 21 symbols. It is very close to the typical TDD system configuration where symmetrical services are supported. Of course, if the TDD system has a much larger DL frame, then H-FDD starts to provide an additional gain. Nevertheless, the gain is not as high as for the F-FDD system, where the UL subframe spans all the symbols. It is also worth mentioning that the group ratio of 1:1 simplifies significantly the scheduling process because all two groups are similar from the resource allocation point of view.

An interesting approach is to have unequal group sizes in the H-FDD system, where the first group downlink sub-frame is always of the smaller size. The first group is the default group that new subscribers use. Having the uplink subframe bigger in the first group means that stations joining the network at the cell edge can benefit from subchannelization gain as much as possible. At the same time, a group with a small UL subframe might be allocated stations with a very good uplink performance.

The biggest problem with an unequal group ratio is that it imposes significant constraints on the BS scheduler. For example, if too many stations with a poor uplink performance are assigned to group 1 that has a large UL subframe, then the DL subframe may run out of resources. Similarly, if a station with high uplink and downlink bandwidth requirements is assigned to group 2, then the BS scheduler may fail to ensure its uplink bandwidth needs. Due to a small UL subframe, an UL allocation will span too many subchannels thus causing problems at the cell edge.

The group ratio could be adjusted dynamically and on a cell-by-cell basis, but changing the group ratio would mean an interruption to the service and signalling a new frame configuration to all the subscribers. In the following subsection, we are concentrating only one cell and assume that the group ratio does not change in the course of time.

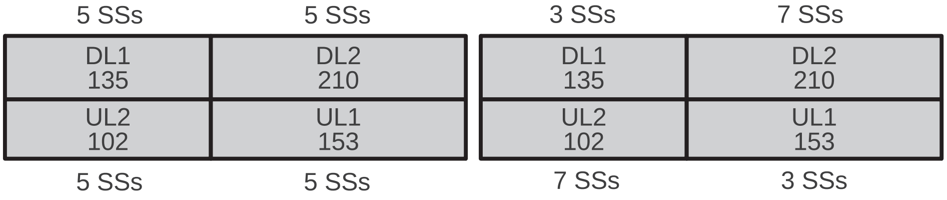
4.1.5 H-FDD Group Balancing

The 802.16 specification[33, 13] or system profile [42] does not say anything about how or how often the H-FDD group balancing should be done. We have implemented two H-FDD group balancing algorithms which, hence, will be referred to as basic and adaptive fair.

In the basic algorithm, the subscriber ratio between groups is kept as close to 1:1 as possible. The subscribers' SNR or type of traffic etc. do not have any impact on the decision, neither is the H-FDD group ratio taken into account. An uneven group ratio means that the users in the group with a bigger UL subframe will get more UL resources and vice versa. Since the subscribers are selected randomly, it probably means that these users also get more bandwidth. An example of group balancing for 10 subscribers with the basic algorithm can be seen in Figure 28a. It is already evident that the basic algorithm cannot provide a good fairness across *all* the stations. The only achievable thing is a good fairness *within* a particular group.

The objective of the adaptive fair group balancing algorithm is to provide a good fairness between all the subscribers in two groups. Adaptive fair consists of two main stages. In the first stage, the traffic direction in the cell is detected. If most of the users have mostly downlink traffic, then downlink fair balancing will be used, and if the traffic is mostly in the uplink direction, then uplink fair balancing is utilized. In the second stage, users are sorted based on the SNR of the detected direction. Then, the users with the worst SNR are placed to H-FDD group 1. The premise of the idea in this solution is that users with bad SNR can benefit from the uplink subchannelization gain since group 1 has a larger uplink subframe size. The balancing between groups is based either on the uplink or downlink fairness criterion so that the bandwidth for every subscriber in the cell should be roughly the same. An example of group balancing done with the adaptive fair algorithm in the downlink case is shown in Figure 28b. In the example, it is assumed that all users use the same MCS.

In both cases the groups are balanced periodically. The H-FDD group switch information is passed to the subscriber in the DL-MAP. When a subscriber changes the group it loses one uplink subframe because the information on the UL-MAP message always refers to the next frame. Thus, rapid switching impacts negatively uplink performance and, in particular, the HARQ functioning.



(a) Basic

(b) Adaptive Fair - DL optimized

FIGURE 28 Example of H-FDD group balancing.

4.2 Performance Comparison of Duplexing Modes

In this chapter, we present the simulation environment and compare the performance of the IEEE 802.16 duplexing modes.

4.2.1 Simulation Environment & Setup

Figure 29 shows the network that we use in a simulation scenario. It comprises a single BS controlling its sector, an FTP server with which data is exchanged, and 32 subscriber stations. In the uplink scenarios, only uplink traffic is present; similarly, there is only downlink traffic in scenarios where the downlink performance is analyzed. The traffic model is the TCP full buffer FTP transmission over the 802.16 BE connection. Thus, regardless of the scenario, there is also traffic in the opposite direction caused by the TCP acknowledgements. Each simulation is run 12 times, and the application level data is measured at a wired link between the FTP server and the base station.

TABLE 8 System level parameters.

Parameter	**Value**
Reuse factor	1/3
Path loss model	802.16m urban macro cell
Fast fading	Jakes model, K=0
Interference level DL/UL	-157 / -159 dBm/Hz
Antenna technique	SISO (1x1)
Antenna pattern BS/SS	3GPP / Omnidirectional
Antenna gain BS/SS	17 / 0 dBi
Antenna height BS/SS	32 / 1.5 m
Tx power BS/SS	5W(TDD), 2.5W(FDD) / 0.2 W

Even though we concentrate on simulations of a single sector, we assume the presence of other cells that generate interference. Table 8 provides information on the relevant parameters. We use the 802.16 UMa propagation [1] with the Jakes fast fading model. For the sake of simulation simplicity, we use a constant interference level. The BS and SS use different antenna patterns and antenna heights. The BS preserves the same transmission power density regardless of the duplexing mode, i.e., 5W for the 10 MHz TDD frame and 2.5W for the 5 MHz FDD mode DL sub-frame. The SS *maximum* transmission power is 0.2W, whereas

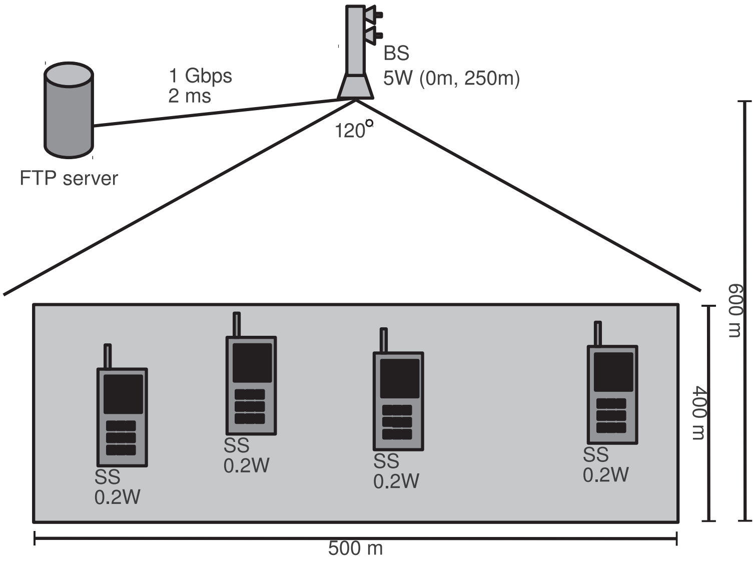


FIGURE 29 General Network Structure.

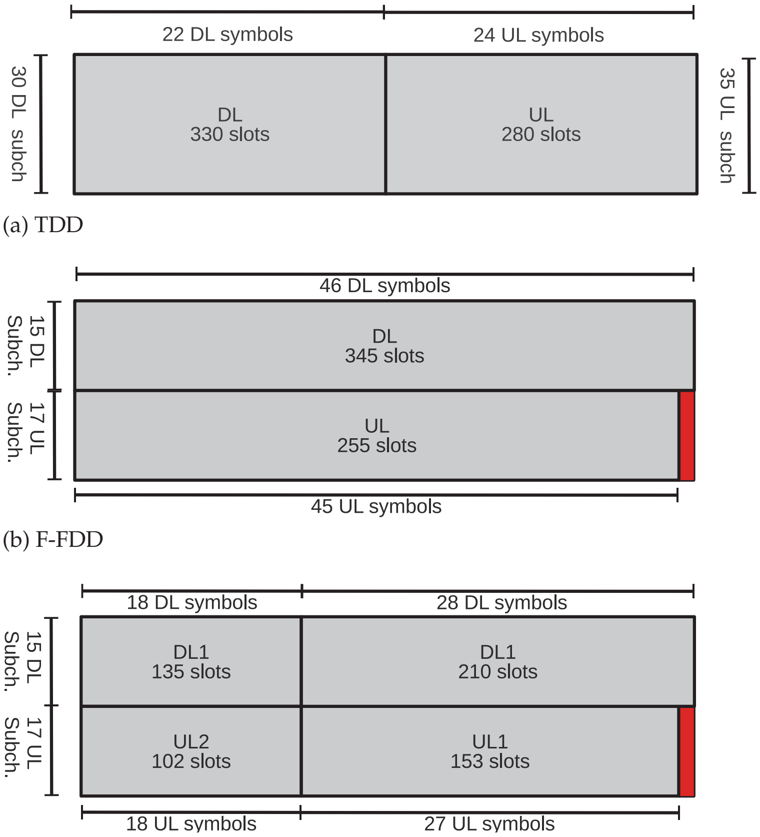
the instantaneous transmission power is governed by the BS power control algorithm.

TABLE 9 TDD/F-FDD/H-FDD PHY parameters.

Parameter	**TDD**	**F-FDD**	**H-FDD**
Frequency band	2.5 GHz		
PHY	OFDMa		
Cyclic prefix length	1/8		
Frames per second	200 (5 ms/frame)		
Long preamble	1 symbol		
Bandwidth	10 MHz	5+5 MHz	
FFT	1024	512	
TTG+RTG	296+168 PS	0+168 PS	
DL/UL subchannels	30/35	15/17	
DL/UL subcarrier alloc.	DL PUSC / UL PUSC		
OFDM symbols	47		
DL/UL symbols	22 / 24	46 / 45	18+28 / 27+18
DL slots	330	345	135+210
UL slots	280	255	102+153
Ranging backoff start/end	1/15		0/15
Ranging transm. opport.	2		1
Request backoff start/end	3/15		2/15
Request transm. opport.	8		4

The 802.16 network parameters are given in Table 9 (duplexing mode specific PHY parameters) and Table 10 (common MAC level parameters). In FDD, there are two 5 MHz bands that impact a choice for the number of FFT points, which ensures the same subcarrier spacing also for TDD that spans one 10 MHz band. It explains the number of subchannels we have in DL and UL directions in different duplexing modes. The TDD DL/UL ratio is chosen in such a way that the number of slots is comparably the same as in the FDD modes. Figure 30a shows number of slots, symbols and subchannels for TDD. For H-FDD, an un-

equal group ratio is utilized to benefit from the UL subchannelization gain. Figure 30b clarifies the number of symbols, subchannels, and slots used in F-FDD mode and Figure 30c in H-FDD mode. It must be noticed that there are unused symbols in the uplink sub-frame in FDD modes due to the UL PUSC structure that mandates the use of 3 symbols per one uplink slot.[1] It is also worth mentioning the initial ranging and bandwidth request parameters. Since there are two independent groups in H-FDD, the number of transmission opportunities per each H-FDD group is reduced two times and the backoff parameters are adjusted accordingly.



(a) TDD

(b) F-FDD

(c) H-FDD

FIGURE 30 Number of slots, symbols and subchannels used in the simulations.

The BS runs the throughput-fair scheduling algorithm, a simple, yet efficient, solution that is capable of allocating slots based on the connection QoS requirements and bandwidth request sizes. It is based conceptually on the deficit round-robin; details of the algorithm are presented in [45]. The reason we did

[1] The real 802.16 system may benefit from those leftovers and use them for other purposes, such as UL sounding. Another option is to allocate there an UL ranging/request channel that should not be aligned on the 3 OFDM symbol boundary.

TABLE 10 Common MAC parameters.

Parameter	Value
DL/UL channel measurements	preamble / data burst
Channel report type / interval	CQICH / 20ms
Channel measurements filter	EWMA, $\alpha = 0.25$
UL Power Control	Closed loop
Link adaptation model	target FEC BLER, 10^{-1}
H-FDD group balancing algorithm	Basic / Adaptive fair
H-FDD group balancing interval	500 ms
MAP MCS	QPSK1/2
Compressed MAP	ON
sub-MAPs	OFF
CDMA codes	256
ranging+periodic ranging	64
bandwidth request	192
handover	–
Fragmentation/packing	ON
PDU size	140 B
CRC	ON
ARQ feedback	standalone
ARQ feedback types	all
ARQ feedback interval	20 ms
ARQ block size / window	16 B
ARQ window	1024
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ timers	
retry	100ms
block lifetime	500ms
Rx purge	500ms

not choose the proportional-fair scheduler [46], which may improve the overall spectral efficiency, is the fact that it tends to decrease fairness, thus making an overall analysis more challenging. In the H-FDD mode, the BS scheduler runs two independent scheduling entities that are responsible for resource allocation in both H-FDD groups. Furthermore, the BS scheduler takes as two independent parameters a preferred number of bursts that should be allocated in the DL and UL directions. It allows for studying the performance of duplexing modes under different scheduling configurations.

In addition to the scheduling algorithm, it is crucial to mention the basics of the UL power control module because the specification does not define an exact algorithm. We implemented a simple closed-loop power control that works in coordination with the BS scheduler. Firstly, every time the BS scheduler makes an UL allocation, it ensures that an SS power budget is not exceeded. If the BS scheduler allocates small UL data grants, then the UL power control increases gradually SS transmission power to benefit from the subchannelization gain. As opposed to that, if the BS scheduler tends to allocate larger UL data grants, the UL power control instructs an SS to decrease the transmission power per subcarrier so that an SS can transmit in more subchannels. All the UL power control commands are carried in the DL broadcast FPC management message. All the SSs also report periodically their UL transmission power values via signalling headers.

The MAC level retransmission mechanism is ARQ, parameters of which are tuned based on our previous research on the ARQ mechanism in the 802.16 networks [47]. The ARQ mechanism also governs the target FEC block error rate of 10^{-1} that is used in the link adaptation module [31]. Even though our simulator supports so-called sub-MAPs that can improve dramatically the performance [40], we decided to turn them off as they also may impact fairness.

While running simulations, we consider a few cases with a different number of preferred bursts per a sub-frame. As is presented later, it helps to explain a difference between TDD and FDD performance and uplink subchannelization gain.

4.2.2 TDD & F-FDD

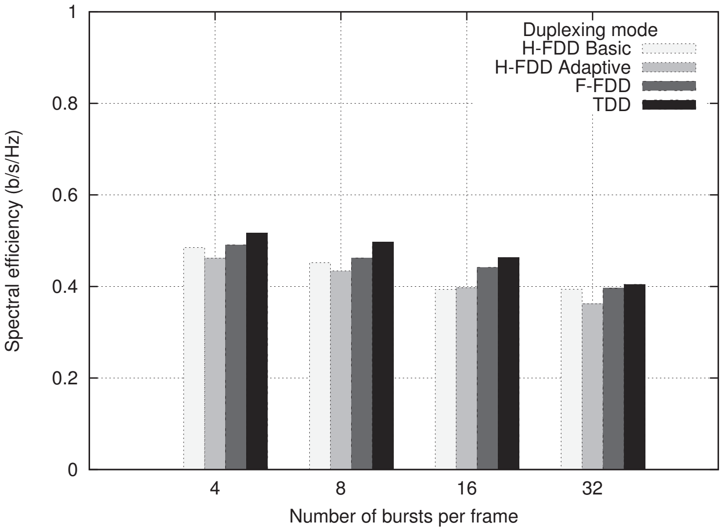


FIGURE 31 Downlink spectral efficiency for different duplexing modes. TDD bars have been scaled up to match the slot count of F-FDD and H-FDD.

First we analyze the difference between TDD and F-FDD. Figure 31 shows the downlink spectral efficiency for F-FDD and TDD. For both duplexing modes the spectral efficiency gets worse when there are more bursts per a subframe. This is because more bursts means that there are more entries in DL-MAP which creates additional overhead. Also with TCP traffic it means that more TCP acknowledgements are sent in the uplink direction, which also increases the UL-MAP size residing in the DL sub-frame. So, regardless of the duplexing mode, the BS scheduler can benefit from the full-buffer traffic by allocating only a few downlink bursts.

In Figure 33a we can see that F-FDD has better downlink throughput for all subscribers. This is due to larger downlink slot count for F-FDD (345 vs. 330), which is visible in the Figure 30b and Figure 30a. The uplink results for 32 SS. In Figure 33b we see that TDD provides better throughput for most of the subscribers, which is explained by higher uplink slot count (280 vs. 255). However,

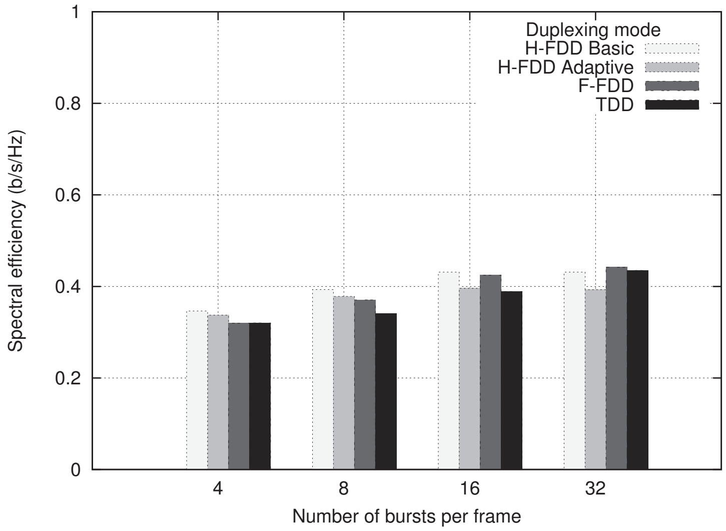
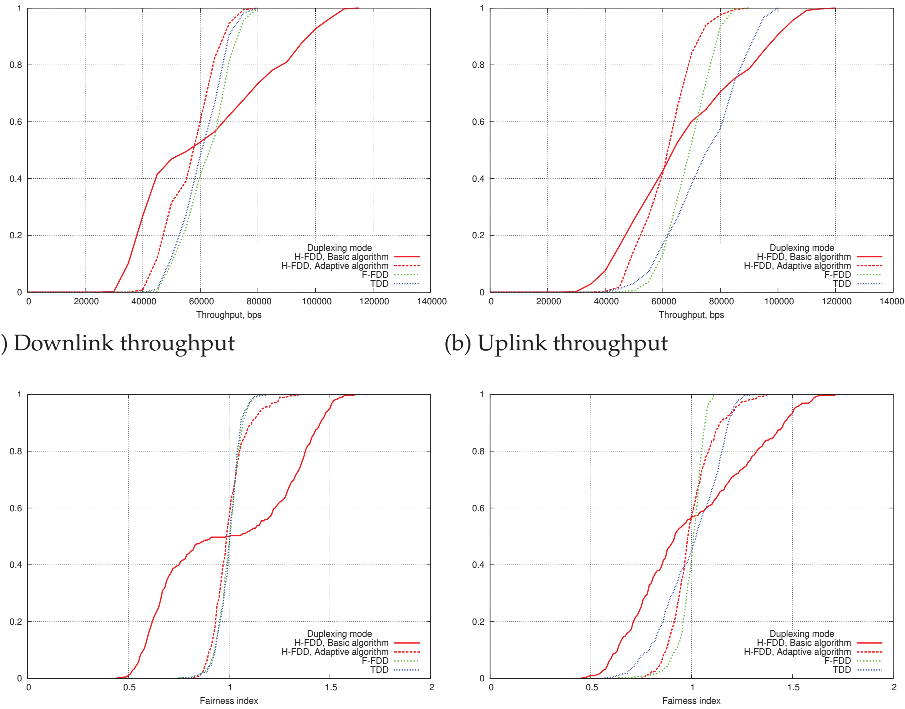


FIGURE 32 Uplink spectral efficiency for different duplexing modes. TDD bars have been scaled down to match the slot count of F-FDD and H-FDD.

F-FDD provides better throughput for the cell edge subscribers which is due to the uplink subchannelization gain; with F-FDD, subscribers send with less subchannels so they can use more power per subchannel.

Downlink fairness for 32 bursts can be seen in Figure 33c and for 4 bursts in Figure 34c. It can be seen that there is no big fairness difference between TDD and F-FDD. Furthermore, the BS scheduler can achieve a good fairness in the downlink direction even for a few data bursts.

Figure 32 shows the uplink spectral efficiency for F-FDD and TDD. It can be seen that with both duplexing modes the performance improves when the number of the bursts per frame increases. This is due to increased subchannelization gain. F-FDD gets the biggest gain already with 16 bursts per frame because there are 17 subchannels in the uplink sub-frame in F-FDD. It means that there is roughly one subchannel per subscriber where it can concentrate all transmission power. With TDD, there are 35 subchannels in the uplink direction, which means that TDD still benefits significantly when the number of bursts is increased from 16 to 32. With 32 bursts per frame, the performance difference between TDD and F-FDD is not big. Also it has to be remembered that in the uplink direction more bursts add to the MAP message size, but the MAP message is carried in the downlink subframe. The MAP message sizes are not measured here, but it is clear that MAP messages are smaller with 16 uplink bursts than with 32 uplink bursts. F-FDD can therefore benefit from full subchannelization gain with less MAP message overhead than TDD. Uplink fairness for 32 bursts can be seen in Figure 33d and for 4 bursts in Figure 34d. With 4 bursts per subframe there is no difference between TDD and F-FDD. With 32 bursts per subframe, F-FDD is providing better fairness than TDD.



(a) Downlink throughput

(b) Uplink throughput

(c) Downlink fairness

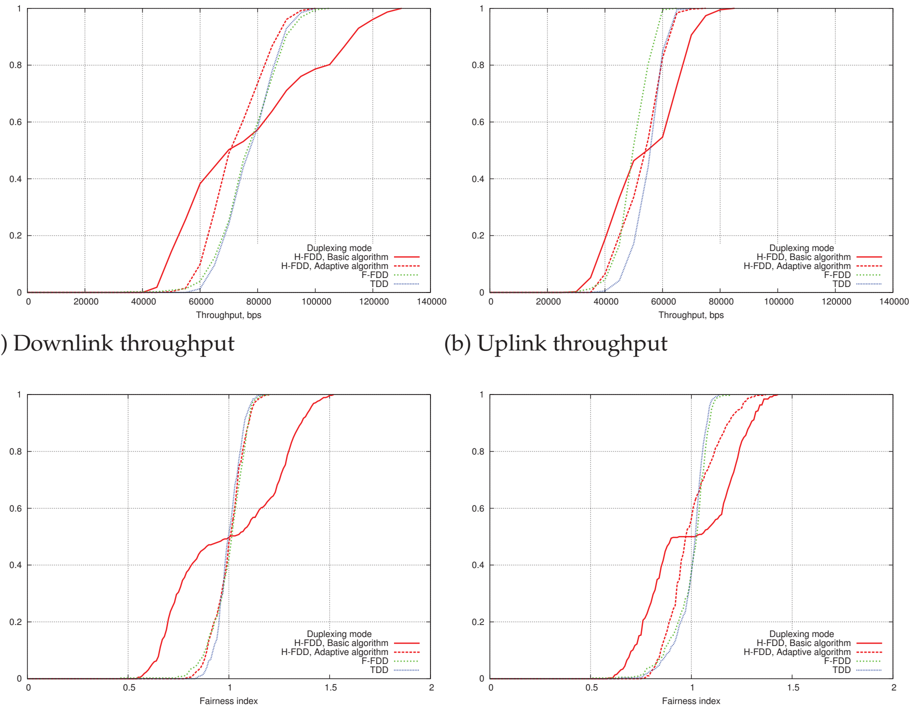
(d) Uplink fairness

FIGURE 33 Duplexing mode comparison throughput and fairness CDFs - 32 bursts per frame.

4.2.3 H-FDD Group Balancing Algorithms

In this section the difference between the two proposed H-FDD group balancing algorithms is analyzed. Figure 31 shows the downlink spectral efficiency for the basic and adaptive fair H-FDD group balancing algorithms. In all the cases, the basic algorithm provides better spectral efficiency than adaptive fair. This can be explained by the bad fairness, which is seen in Figure 33c and Figure 34c. The basic balancing assigns randomly 16 subscribers to the first group and 16 to the second one. The downlink subframe of the first group is smaller than the downlink subframe of the second group, which means that there are fewer slots per subscriber in the first group. This phenomenon can also be seen from throughput CDFs in Figure 33a and Figure 34a.

In the uplink direction, the difference between basic and adaptive fair is similar to that in the downlink. Spectral efficiency for basic is better (Figure 32) but fairness (Figure 33d and Figure 34d) and throughput distribution (Figure 33b and Figure 34b) are worse. The uplink contention mechanism is equalizing the difference between the two groups with basic balancing, so it is not as clear as in the downlink case.



(a) Downlink throughput

(b) Uplink throughput

(c) Downlink fairness

(d) Uplink fairness

FIGURE 34 Duplexing mode comparison throughput and fairness CDFs - 4 bursts per frame.

4.2.4 H-FDD Adaptive Fair, TDD and F-FDD

Because H-FDD with adaptive fair group balancing can provide good fairness, we compare it to the other duplexing modes. In the downlink direction, H-FDD with adaptive fair balancing does not reach the spectral efficiency of F-FDD or TDD, which is seen in Figure 31. This is because H-FDD has more overhead from MAP message headers. Still, Figure 33c shows how fairness with H-FDD adaptive fair is almost as good as with F-FDD and TDD.

In Figure 32, it can be seen that in the uplink direction with 4 and 8 bursts per frame the H-FDD has a better spectral efficiency than F-FDD or TDD, which is explained by worse fairness. With 16 bursts, F-FDD gets the full benefit from subchannelization gain and has the best efficiency while H-FDD and TDD perform equally. With 32 bursts, TDD benefits from subchannelization gain and outperforms H-FDD.

4.3 Conclusions of Duplexing Modes

The simulation results show that, as expected, there is no big difference in performance between the TDD and FDD modes in the downlink direction. In the uplink direction, F-FDD gets the full benefit from subchannelization gain with a fewer number of bursts per a frame. In general, longer uplink sub-frame makes cell edge performance better under F-FDD.

Regardless of the duplexing mode, the number of bursts in the uplink direction should be roughly the same as the number of uplink subchannels. In this case the optimal uplink performance is achieved with 16 bursts for FDD modes and with 32 bursts for TDD. On the other hand, having fewer uplink bursts means that less entries are needed in the MAP messages in the downlink direction, which benefits the FDD modes. In the downlink direction, the optimal number of bursts is 4-8, regardless of the duplexing mode used.

H-FDD has the worst performance, mostly due to two groups that introduce a number of small transmission gaps and increase the DL signalling overhead. While comparing the proposed H-FDD group balancing methods, one can notice that the basic H-FDD balancing algorithm always achieves better spectral efficiency than the adaptive fair balancing, which is explained by bad fairness. The adaptive fair balancing can achieve as good fairness as TDD and F-FDD, except the uplink direction with 4-8 bursts per frame. At the same time, the spectral efficiency is 0-10% behind that of TDD and F-FDD when fairness is similar.

There is no simple choice between TDD and F-FDD since they both have their benefits. TDD has an adjustable DL/UL ratio and F-FDD can utilize better the uplink subchannelization gain. However, it is usually the case that an operator does not select the duplexing mode freely but rather adapts to available frequency bands. Then, if the F-FDD subscriber stations are not available or they are too expensive, H-FDD must be used. It is worth mentioning that the scheduling for H-FDD is more complicated, which creates a burden for the network side. Furthermore, bidirectional traffic mixture will create additional fairness problems. Still, we have shown that in simple cases fairness can be guaranteed with the proposed adaptive fair balancing, where H-FDD has only slightly worse spectral efficiency than TDD or F-FDD.

4.4 Relays

The very high data rate demands for wireless communication systems create a need for more fundamental enhancements, other than just increasing the transmission bandwidth or introducing higher order modulation and coding schemes. Along with technologies, such as MIMO and cooperative multi-point transmission, relaying is seen as a quite promising solution [48]. Relays can be deployed without wired connection installation or microwave links and is thus a cheaper option to fully featured base station. Also the serving base stations do not need any additional hardware to support relay nodes.

For the sake of brevity, we will not divulge into extensive description of the relaying functionality defined in 802.16j. The technical specification is given in [49] and a good technical overview is presented in [50]. However, it is worth to mention the available relaying options. From the viewpoint of the spectrum usage, relays can be either in-band (TTR) or out-band (STR). From the viewpoint of the downlink (DL) management signalling, they can be either transparent or non-transparent. These combinations also define the possible scheduling modes, either centralized or distributed. These are summarized in Table 11.

TABLE 11 Available relay modes in 802.16j.

	transparent	non-transparent
in-band (TTR)	centralized	centralized/distributed
out-band (STR)	–	distributed

Hence, following the 802.16j terminology, BS will refer to the base station, while MR-BS will denote multi-hop relay BS, and RS stands for the relay station.

There is a strong motivation to consider the non-transparent in-band relays working in the distributed scheduling mode. Firstly, an in-band relay reuses the *existent* spectrum instead of requiring a new frequency band. Thus, it is an appealing option for operators that do not have or cannot acquire additional radio resources. Secondly, a non-transparent relay can enhance both *coverage and throughput*, while a transparent relay can improve only throughput within the *existent* cell boundaries. Finally, the distributed scheduling mode makes scheduler implementation simpler and allows for reusing the existent BS software at RS without implementing a complicated centralized scheduling. It also makes the overall scheduling process faster because both MR-BS and RS schedulers work as two independent entities.

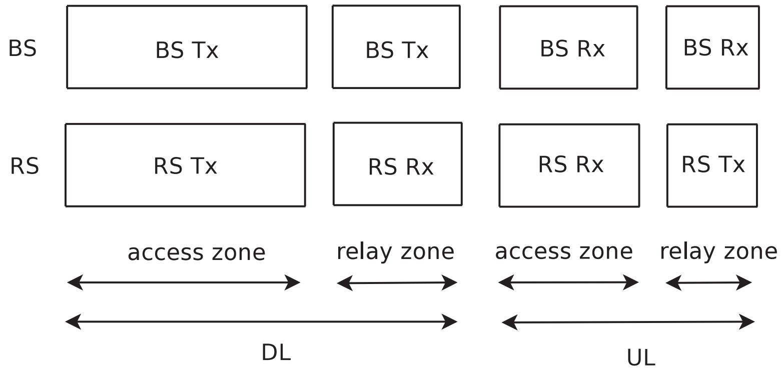


FIGURE 35 Non-transparent in-band relay frame structure.

For the sake of further clarity, Figure 35 presents the frame structure of the non-transparent in-band relay. The relay zone is a one where the MR-BS and RS exchange data. It must be noted that both BS and RS transmit simultaneously in the DL access zone to the associated subscriber stations (SS), thus mutually interfering. A similar situation occurs in the uplink (UL) access link when SSs associated with RS and BS start to interfere with each other. Thus, the non-transparent relays reuse the existent spectrum at the cost of increasing the interference level.

In the following section, we study the problem of how to partition the downlink and uplink subframes into the access and relay zones the most efficient way.

4.5 Simulation Results on Relay Performance

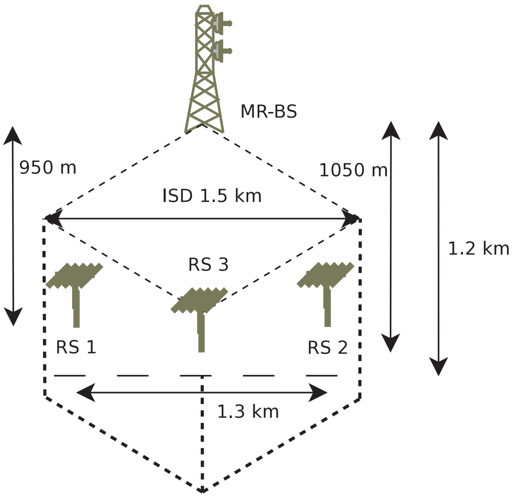


FIGURE 36 Relay Simulation scenario.

For simulations we use the WINSE simulator, which is described in Section 2, with a support for two-hop non-transparent in-band relays as a main solution for the coverage extension problem [51].

The non-transparent RS node runs the same radio resources management mechanisms as a normal BS does, e.g., scheduling [45], channel estimation, link adaptation [31] etc. The scheduler is a *throughput fair* one that is based conceptually on deficit round robin. Even though the proportional fair scheduler might provide a better spectral efficiency [46], we choose the throughput fair scheduler to show, on the throughput fairness, the impact of relaying.

Figure 36 shows a simulation scenario. It is assumed that there is a single BS controlling its sector. To serve an area limited by a dashed line, an operator may deploy additional BSs to cover two more sectors denoted by dotted lines. However, a more cost-efficient solution might be to deploy a few relay nodes, as shown in the figure. It is understandable that deploying additional BSs will bring a better performance at the expense of increased deployment cost: installation and support of two macro BS with microwave links or wired backhaul

TABLE 12 802.16 network parameters.

Parameter	Value
Frequency band	2.5 GHz
Bandwidth	10 MHz
PHY	OFDMa
Reuse factor	1/3
Duplexing mode	TDD
Frame duration	5 ms
CP length	1/8 symbol
TTG+RTG	296+168 PS
OFDM symbols	47
DL/UL symbols	30/15
DL/UL relay zone size	2, 4, 6 / 3 symbols
DL/UL subcarrier alloc.	DL PUSC / UL PUSC
Channel report type / interval	CQICH / 20ms
Channel measurements DL/UL	preamble / data burst
Channel measurements filter	EWMA, $\alpha = 0.25$
Link adaptation model	target FEC BLER, 10^{-1}
Antenna technique	SISO
BS / RS / SS Tx power	10 / 5 / 0.25 W
BS / RS / SS antenna pattern	3GPP / omni / omni
BS / RS / SS antenna gain	17 / 5 / 0 dBi
BS / RS / SS antenna height	32 / 7 / 1.5 m
access / relay link path loss	.16m SMa / .16j TypeA
access / relay link fast fading K factor	0 / 0 dB
DL MAP MCS	QPSK1/2 Rep6. . . QPSK1/2
Compressed MAPs	ON
sub-MAPs	ON, max. 3
Ranging transm. opport.	1
Ranging backoff start/end	0/15
Request transm. opport.	2
Request backoff start/end	1/15
CDMA codes	256
ranging+periodic ranging	64
bandwidth request	192
handover	–
PDU size	140 B
Fragmentation	ON
ARQ feedback	standalone
ARQ feedback types	all
ARQ feedback intensity	20 ms
ARQ block size	64 B
ARQ window	1024
ARQ discard	ON
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ timers	
retry	60 ms
block lifetime/Rx purge	500 ms

connections will cost more than three RS nodes [52]. Thus, for the sake of brevity, we compare a case with a single BS and a case with the MR-BS and RS nodes.

The choice for the number of RSs was motivated based on the cost analysis in [52] that stated that it is better to have a few high-power RS nodes than a number of very low-power ones. While placing three RS nodes, we account for the MR-BS directional antenna gain and its coverage area. While RS3 is placed at the MR-BS main antenna lobe direction, RS1 and RS2 are placed closer to the MR-BS and to the cell edge. Furthermore, since the MR-BS and the non-transparent RSs interfere with each other, we do not put the RS nodes too close to the MR-BS to avoid mutual interference. The final RSs coordinates were tuned after a few simulation runs. However, we do not claim in this paper that they are the optimal ones.

Table 12 presents the key 802.16 parameters used in the simulation, which conforms to [53]. We consider the DL FTP-like continuous TCP transmission over 802.16 BE connections, where the IP-level service data unit (SDU) size is 1000 bytes. It is a good way to analyze the resulting application level throughput and the spectral efficiency. Of course, there is also UL traffic caused by the TCP protocol functioning. It is worth mentioning that to study the relay performance, we consider a different *fixed* DL relay zones size (see Figure 35) of 2, 4, and 6 OFDM symbols. The UL zone size is also fixed and has the constant size of 3 symbols.[2] Unlike the MR-BS, RS uses an omni-directional antenna, has a lower Tx power of 5W and a smaller antenna height. The motivation is that a lower Tx power requires a simpler and a less expensive amplifier chain. The omni-directional antenna simplifies the design and the installation efforts. Furthermore, an omnidirectional antenna at the RS node allows for communicating efficiently to any SS around the RS node.[3]

We assume the sub-urban macro-cell scenario and thus choose the 802.16m SMa propagation model for an access link and the 802.16j TypeA model for the relay link [54]. The latter one is for NLOS communication between the MR-BS and RS nodes because otherwise an operator might deploy a microwave link between two BS sites. The interference modelling accounts for the fact that the MR-BS and the non-transparent RSs transmit simultaneously thus impacting each other. The interference from the neighbouring cells is also taken into account, assuming the reuse 1/3 factor and full load traffic. The fast fading is generated based on the Jakes model with the K factors given in Table 12 and assuming an SS speed of 1 m/s.

The MAC level retransmission mechanism is ARQ working in the end-to-end mode. In other words, RS does not take part in the ARQ signalling but just forwards received data. The ARQ parameters are tuned based on our previous research on the ARQ mechanism in the 802.16 networks [47]. The ARQ mechanism also governs the target FEC block error rate of 10^{-1} that we use in the link adaptation module [31].

To gather statistically reliable results, we ran 20 different simulations, each containing 30 SSs placed in random locations. Each simulation run lasted for 10 seconds, which is enough for the TCP protocol to stabilize.

Figure 37 shows the simulation area with SS locations and their associations to the MR-BS or RS node, as indicated by different symbols. As anticipated, an SS associates itself to RS if it observes a stronger DL signal strength coming from the RS node. Note that Figure 37 accumulates 600 different locations from all the

[2] DL relay zone size must be a multiple of 2 OFDM symbols due to the DL PUSC permutation type. Similarly, the UL relay zone size must be a multiple of 3 OFDM symbols due to the UL PUSC structure.

[3] A possible solution is to have two antennas at the RS node: a directional one to exchange data with MR-BS and an omnidirectional one to communicate with associated SSs around RS. This solution is more complicated in implementation and requires more installation efforts due to the antenna direction and tilting.

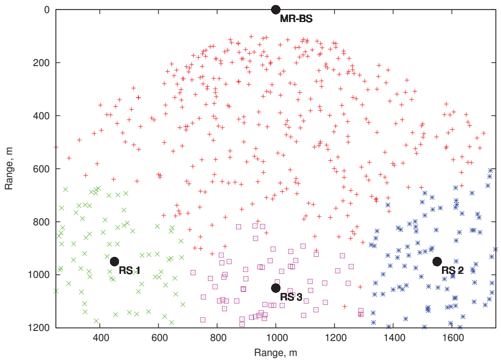


FIGURE 37 Simulation area with SS locations.

simulation runs. Of course, if there is only a single BS node, then all the SSs in Figure 37 are associated with it.

Figure 38 presents the DL *application level* spectral efficiency, i.e., one that excludes any PHY or MAC level management data. We present the minimum, average, and maximum values for a case when there is only the BS (the leftmost bar) and three cases with relays and different DL relay zone sizes (2, 4, and 6 symbols). As can be seen, relays improve the spectral efficiency: the more resources a relay link has, the better an average spectral efficiency becomes because SSs, which are close to an RS, can benefit from a good link between BS and RS.

To provide a better insight on the relay performance, we also present the mean DL connection throughput CDF in Figure 39. The mean throughput is calculated individually for each connection after each simulation run. Firstly, it is noticeable that without relays there are SSs that have no service at all because they are out of the BS coverage area. Once we deploy relays, all the SSs are able to transmit at least at some rate. Secondly, Figure 39 shows that the DL relay zone size of 2 symbols results in a situation when there are SSs that have a lower throughput when compared to the baseline scenario. As explained later, this is due to the fact that a small DL relay zone size becomes a bottleneck.

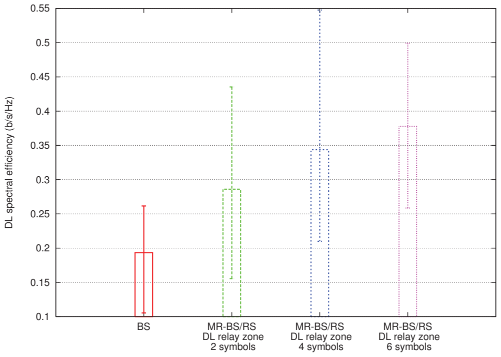


FIGURE 38 DL spectral efficiency.

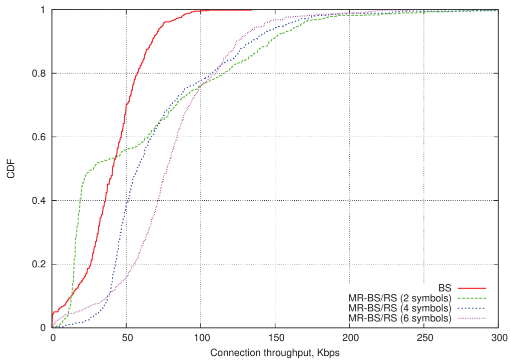
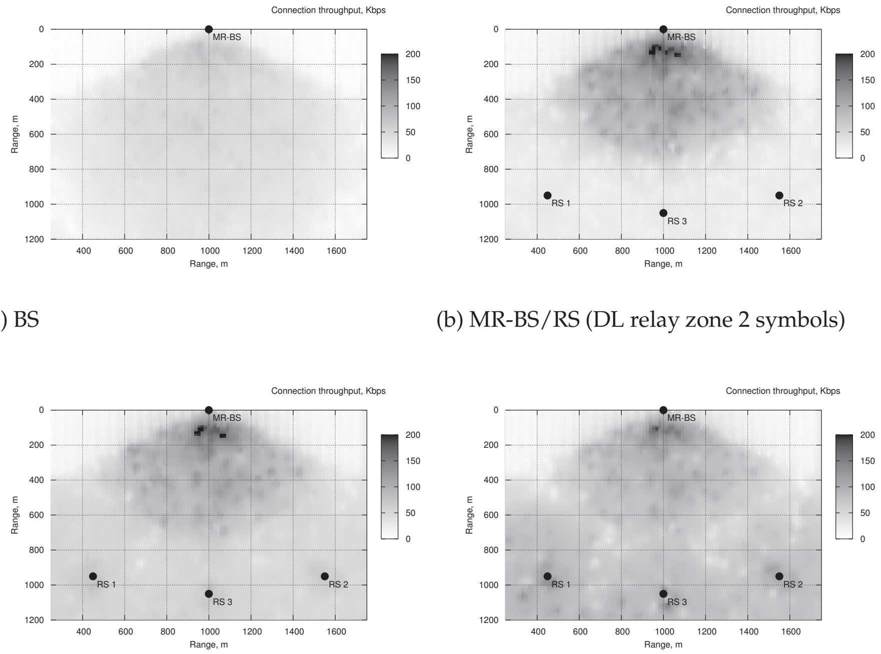


FIGURE 39 DL connection throughput cumulative distribution.

Of course, if there is only the BS node, then it can ensure quite a good fairness because the BS scheduler has a complete control over resource allocation on the access link. Once we deploy RSs, everything the MR-BS scheduler can do is to control resources for the relay link but not the way the RS node will allocate resources between connections on its access link. It can be seen that relays working in the distributed scheduling mode decrease the throughput fairness, especially in case of a badly configured DL relay zone size.



(a) BS

(b) MR-BS/RS (DL relay zone 2 symbols)

(c) MR-BS/RS (DL relay zone 4 symbols)

(d) MR-BS/RS (DL relay zone 6 symbols)

FIGURE 40 DL throughput distribution over the simulation area.

Figure 40 provides a different view on throughput and fairness. Similar to Figure 37, this figure aggregates results from all the simulation runs and presents the throughput distribution over the simulation area under different DL relay zone sizes. As can be seen from Figure 40a, a case when only the BS is deployed results in a low but quite fair throughput distribution over the simulation area. However, cell edge areas have no service at all. If we deploy RSs with a small DL relay zone size (see Figure 40b), then the RS nodes can offload the MR-BS, thus providing a higher throughput to SSs associated with it. However, a small DL relay zone size results in a considerably lower throughput of SSs associated with the RS nodes because the MR-BS to RS link becomes a bottleneck. As we increase the DL relay zone size, we can see that SSs associated with RS nodes start to transmit at much higher throughput, increasing the overall system spectral efficiency. Once the DL relay zone size equals 6 symbols, the throughput fairness starts to decline because now the DL access zone has become a bottleneck.

Figure 40 also presents impact of the non-transparent relaying and, as a result, increased interference level on the throughput distribution. In Figure 40a, low throughputs are observed at the cell edge, where distance increases and directional antenna gain becomes smaller. In Figures 40b-40d, low throughputs are observed at the cell edge and places where signal strength from the MR-BS is as strong as the cumulative interference coming from all the RS nodes. Thus, the fact that the non-transparent relays create additional interference introduces more challenges for the network planning.

Based on the presented results, it is possible to state that the relay zone plays quite a crucial role. It can be treated as a parameter that controls the tradeoff between the overall system spectral efficiency and throughput fairness. For the simulation scenario considered, the DL relay zone size of 4 symbols is a good choice: the spectral efficiency is almost two times higher and the fairness is satisfactory. Of course, a different number of SSs, their location, and/or traffic pattern may yield a different optimal configuration.

4.6 Conclusions of Relays

We have run complex dynamic simulations of the 802.16j non-transparent inband relays working in the distributed scheduling mode. According to the simulation results, relays indeed improve the overall system performance even despite the NLOS link between MR-BS and RSs and the fact that the non-transparent RS nodes and the MR-BS interfere with each other and the MR-BS has to allocate its resources for the relay communication. At the same time, the overall complexity of the whole system makes it quite complicated to achieve simultaneously a high throughput and fairness across all the access links in the system.

The problem of selecting relay zone size becomes even more complex when the traffic pattern and user distribution is different under the relays controlled by the same BS. The split between the relay and access zones is the same for all the relays, but since the traffic pattern is different the decision gets more difficult.

Furthermore, the balance between spectral efficiency and fairness depends heavily on the DL relay zone size, over which the base station exchanges data with relays. Thus, the relay zone sizes must be set up carefully and adjusted dynamically to control the tradeoff between the system performance and fairness. In turn, this creates a need for fast and reliable signalling mechanism to orchestrate relay zone sizes.

5 CONCLUSIONS

While mostly concentrating on improving the spectral efficiency, this dissertation has presented various ways of how to improve the performance of the IEEE 802.16 system. We have shown that the performance depends a lot on how the system is configured. System optimization should be approached from a higher level. Usually the availability of spectrum and terminals govern the duplexing mode used. Then there is the question of whether the relays are used or not. The next step would be a choice between HARQ and ARQ. Once that has been decided, the details of scheduling bursts, PDU size limits etc. can be agreed.

The first topic in this dissertation was the optimization of the IEEE 802.16 MAC PDU size. It was shown that if the channel quality can be estimated then the proposed method can improve the throughput. However, with current the standard the method can be applied to downlink transmission only. The subscriber station does not have information about the uplink channel quality. Secondly, the IEEE 802.16 retransmission mechanisms were presented and their performance evaluated. Finally, the IEEE 802.16 duplexing mode and relay performance was analysed. We presented a novel way to select the group sizes with the H-FDD duplexing mode and how to divide users to them. With the proposed group balancing algorithm, the throughput and its fairness is kept close to TDD and F-FDD.

Although the thesis is about IEEE 802.16 system performance, the future work will not be restricted to WiMAX. Some of the results and ideas can be reused in other wireless technologies. The PDU size optimization methodology can be used in LTE networks since LTE is also using OFDMA PHY in downlink and the MAC layer has a lot of similarities. Also, the LTE-A relay concept is very similar. Now that we have studied several optimization methods on different layers, it will be interesting to see how the optimization can be made cross-layer.

YHTEENVETO (FINNISH SUMMARY)

Tämä väitöskirja, jonka nimi on IEEE 802.16 -järjestelmän fyysisen ja MAC-kerroksen suorituskyvyn optimointi, tutkii IEEE 802.16-verkon suorituskyvyn parantamista eri menetelmillä. Ensimmäinen ongelma on MAC-paketin koon optimointi kun ARQ-virheenkorjaus on käytössä, siten että järjestelmän tiedonsiirtokapasiteetti paranee. Väitöskirja esittää menetelmän optimaalisen paketin koon valinnalle ja arvioi esitetyn menetelmän. Toinen väitöskirjan osa keskittyy ARQ-virheenkorjausmenetelmän eri ominaisuuksien optimaaliseen käyttöön sekä vertaa ARQ- ja HARQ-virheenkorjausmenetelmien suorituskykyä. Väitöskirjan loppu käsittelee kahta IEEE 802.16 -kehyksen käytön optimointialueta. Ensiksi IEEE 802.16 -järjestelmän kolme kanavointimenetelmää esitellään: aikajakoinen (TDD), taajuusjakoinen (F-FDD) ja puolitaajuusjakoinen (H-FDD). Lisäksi esitellään uusi ryhmäjakoalgoritmi puolitaajuusjaolle. Lopuksi kaikkien kanavointimenetelmien ja ryhmäjakoalgoritmin suorituskyky arvioidaan. Toinen esitelty kehyksen optimointimenetelmä on IEEE 802.16 -järjestelmän langattomat välitysasemat, joiden vaikutus verkon suorituskykyyn arvioidaan. Kaikki edellä mainitut suorituskyvyn arvioinnit suoritetaan tietoliikennesimulaatioiden avulla. Simulaattorina käytetään WINSE WiMAX-laajennusta, joka toimii NS-2-simulaattorin päällä.

REFERENCES

[1] "IEEE 802.16m evaluation methodology document (EMD)." IEEE 802.16 Broadband Wireless Access Group, Mar 2008.

[2] "Vocabulary of terms for broadband aspects of ISDN." ITU-T recommendation I.113, 1997.

[3] J. Andrews, A. Ghosh, and R. Muhamed, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*. Prentice Hall PTR.

[4] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 756–769, Dec 1997.

[5] K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *ACM SIGCOMM Computer Communication Review*, vol. 27, pp. 19–43, Oct 1997.

[6] A. V. Bakre and B. R. Badrinath, "Implementation and performance evaluation of indirect TCP," *Mobile Computing, IEEE Transactions on*, vol. 46, pp. 260–278, Mar 1997.

[7] X. Xiao and L. Ni, "Internet QoS: a big picture," *IEEE network*, vol. 13, pp. 8–18, Mar/Apr 1999.

[8] R. Ludwig and R. H. Katz, "The Eifel algorithm: making TCP robust against spurious retransmissions," *ACM SIGCOMM Computer Communication Review*, vol. 30, pp. 30–36, Jan 2000.

[9] Y. Tian, K. Xu, and N. Ansari, "TCP in wireless environments: problems and solutions," *Communications Magazine, IEEE*, vol. 43, pp. 27–32, march 2005.

[10] R. Ludwig, B. Rathonyi, A. Konrad, K. Oden, and A. Joseph, "Multi-layer tracing of TCP over a reliable wireless link," in *Proceedings of the 1999 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pp. 144–154, 1999.

[11] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications." IEEE Standard 802.11, 1999.

[12] 3GPP TS 25.214 v10.6.0, "Physical layer procedures (FDD)," March 2012.

[13] "Air interface for fixed broadband wireless access systems - amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands." IEEE Standard 802.16e, Dec 2005.

[14] A. Ahmad, *Data Communication Principles for Fixed and Wireless Networks*. Kluwer Academic Publishers, 2002.

[15] C. Wang, D. Sklar, and D. Johnson, “Forward error correction coding,” *Crosslink*, vol. 3, pp. 26–29, Dec 2001.

[16] G. Rabbani, J. Kamruzzaman, I. Gondal, I. Ahmad, and R. Hassan, “Dynamic resource allocation for improved QoS in WiMAX/WiFi integration,” *Studies in Computational Intelligence*, vol. 368, pp. 141–156, Dec 2011.

[17] V. Tykhomyrov, *Mitigating the Amount of Overhead Arising from the Control Signaling of the IEEE 802.16 OFDMa System*. PhD thesis, University of Jyväskylä, 2011.

[18] “WiMAX Forum Mobile System Profile, Release 1.0 Approved Specification,” Nov 2007. Revision 1.5.0.

[19] J. Spragins, J. Hammond, and K. Pawlikowski, *Telecommunications: Protocols and Design*. Addison Wesley Publishing Company, 1991.

[20] M. Sarraf, “Optimum packet length in a fading channel environment along with random bit errors and finite buffer size,” in *AT&T Technical Memorandum, no. 140360000*, October 1991.

[21] C. Siew and D. Goodman, “Packet data transmission over mobile radio channels,” *IEEE Transactions On Vehicular Technology*, vol. 38, pp. 95–101, May 1989.

[22] C. Schurgers, M. Srivastava, A. Boulis, and P. Lettieri, “Adaptive control of wireless multimedia links,” in *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, vol. 3, pp. 1498 –1502, Sep 1999.

[23] A. Xhafa, S. Kangude, and L. Xiaolin, “MAC performance of IEEE 802.16e,” in *IEEE Vehicular Technology Conference*, vol. 1, pp. 685–689, Sep 2005.

[24] B. Can, R. Vannithamby, H. Lee, and A. Koc, “MAC-PDU size optimization for OFDMA modulated wireless relay networks,” pp. 1 –6, Dec 2008.

[25] S. Sengupta, M. Chatterjee, S. Ganguly, and R. Izmailov, “Exploiting MAC flexibility in WiMAX for media streaming,” in *Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, pp. 338–343, June 2005.

[26] C. Hoymann, “Analysis and performance evaluation of the OFDM-based metropolitan area network IEEE 802.16,” *Computer Networks*, vol. 49, no. 3, pp. 341–363, 2005.

[27] G. Ciccarese, M. D. Blasi, P. Marra, C. Palazzo, and L. Patrono, “A packet size control algorithm for IEEE 802.16e,” in *IEEE Wireless Communications and Networking Conference*, pp. 1420–1425, Mar 2008.

[28] A. Sayenko, O. Alanen, J. Karhula, and T. Hämäläinen, "Ensuring the QoS requirements in 802.16 scheduling," in *The 9th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 108–117, Oct 2006.

[29] L. Nuaymi, N. Bouida, N. Lahbil, and P. Godlewski, "Headers overhead estimation, header suppression and header compression in WiMAX," in *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on*, p. 17, oct. 2007.

[30] M.-F. Tsai, N. Chilamkurti, and C.-K. Shieh, "An adaptive packet and block length forward error correction for video streaming over wireless networks," *Wirel. Pers. Commun.*, vol. 56, pp. 435–446, Feb. 2011.

[31] A. Puchko, V. Tykhomyrov, and H. Martikainen, "Link adaptation thresholds for the IEEE 802.16 base station," in *Workshop on NS-2 simulator*, Oct 2008.

[32] A. Q. Ansari, D. A. Q. K. Rajput, and D. M. Hashmani, "WiMAX Network Optimization -Analyzing Effects of Adaptive Modulation and Coding Schemes Used in Conjunction with ARQ and HARQ," in *Communication Networks and Services Research Conference*, pp. 6–13, May 2009.

[33] "Air interface for fixed broadband wireless access systems." IEEE Standard 802.16, Jun 2004.

[34] V. Tykhomyrov, A. Sayenko, H. Martikainen, O. Alanen, and T. Hämäläinen, "On ARQ feedback intensity of the IEEE 802.16 ARQ mechanism," in *International Conference on Telecommunications*, Jun 2008.

[35] A. Sayenko, V. Tykhomyrov, H. Martikainen, and O. Alanen, "Performance analysis of the IEEE 802.16 ARQ mechanism," in *The 10th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 314–322, Oct 2007.

[36] J. Lakkakorpi and A. Sayenko, "Uplink VoIP delays in IEEE 802.16e using different ertPS resumption mechanisms," in *Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 157–162, Oct 2009.

[37] "Amendment to IEEE standard for local and metropolitan area networks, part 16: Air interface for broadband wireless access systems - advanced air interface." IEEE Standard 802.16m-2011, Mar 2011.

[38] "WiMAX Forum Mobile System Profile: Release 2.0 Approved Specification," May 2011.

[39] S. Ahmadi, *Mobile WiMAX - A Systems Approach to Understanding IEEE 802.16m Radio Access Technology*. Academic Press, 2011.

[40] V. Tykhomyrov, A. Sayenko, O. Puchko, and T. Hämäläinen, "Decreasing the MAP overhead in the IEEE 802.16 OFDMa system," in *European Wireless Conference*, pp. 63–70, Apr 2010.

[41] "Air interface for broadband wireless access systems." IEEE Standard 802.16 (Rev2), May 2009.

[42] "WiMAX Forum Mobile System Profile Specification: Release 1.5 Approved Specification," Aug 2009.

[43] P. Chan, E. Lo, R. Wang, E. Au, V. Lau, R. Cheng, W. Mow, R. Murch, and K. Letaief, "The evolution path of 4G networks: FDD or TDD?," *IEEE Communications Magazine*, vol. 44, pp. 42–50, Dec 2006.

[44] A. Bacioccola, C. Cicconetti, A. Erta, L. Lenzini, and E. Mingozzi, "Bandwidth allocation with half-duplex stations in IEEE 802.16 wireless networks," *Mobile Computing, IEEE Transactions on*, vol. 6, pp. 1384–1397, Dec 2007.

[45] A. Sayenko, O. Alanen, and T. Hämäläinen, "Scheduling solution for the IEEE 802.16 base station," *Computer Networks*, vol. 52, pp. 96–115, 2008.

[46] J. Lakkakorpi, A. Sayenko, and J. Moilanen, "Comparison of different scheduling algorithms for WiMAX base station: deficit round robin vs. proportional fair vs. weighted round robin," in *IEEE WCNC*, pp. 1991–1996, Mar/Apr 2008.

[47] V. Tykhomyrov, A. Sayenko, H. Martikainen, and O. Alanen, "Analysis and performance evaluation of the IEEE 802.16 ARQ mechanism," *Journal of communications software and systems*, vol. 4, pp. 29–40, Mar 2008.

[48] R. Pabst, B. Walke, D. Schultz, P. Herhold, H. Yanikomeroglu, S. Mukherjee, H. Viswanathan, M. Lott, W. Zirwas, M. Dohler, H. Aghvami, D. Falconer, and G. Fettweis, "Relay-based deployment concepts for wireless and mobile broadband radio," *IEEE Communications Magazine*, vol. 42, pp. 80–89, Sep 2004.

[49] "Air interface for broadband wireless access systems: Multihop relay specification." IEEE Standard 802.16j, Jun 2009.

[50] S. W. Peters and R. W. Heath, "The future of WiMAX: multihop relaying with IEEE 802.16j," *IEEE Communications Magazine*, Jan 2009.

[51] "Harmonized contribution on 802.16j (mobile multi-hop) usage models." IEEE 802.16 Broadband Wireless Access Working Group, Sep 2006.

[52] E. Lang, S. Redana, and B. Raaf, "Business impact of relay deployment for coverage extension in 3GPP LTE-Advanced," in *IEEE International Conference on Communication*, Jun 2009.

[53] “WiMAX Forum Mobile System Profile Specification: Release 1.5 (Revision 0.2.1),” Feb 2009.

[54] “Multi-hop relay system evaluation methodology (channel model and performance metrics).” IEEE 802.16 Broadband Wireless Access Working Group, Feb 2007.

ORIGINAL PAPERS

PI

OPTIMAL MAC PDU SIZE IN IEEE 802.16

by

H. Martikainen,A. Sayenko, O. Alanen and V. Tykhomyrov 2008

4th International Telecommunication NEtworking WorkShop on QoS in
Multiservice IP Networks

Optimal MAC PDU Size in IEEE 802.16

Henrik Martikainen #1, Alexander Sayenko *2, Olli Alanen #3, Vitaliy Tykhomyrov #4

#*Telecommunication laboratory, MIT department,*
University of Jyväskylä, Finland
1henrik.martikainen@jyu.fi
3olli.alanen@jyu.fi
4vitykhom@jyu.fi

**Nokia Research Center,*
Helsinki, Finland
2alexander.sayenko@nokia.com

***Abstract*— In the IEEE 802.16 the number of errors and the MAC PDU size have an impact on the performance of the network. We present a way to estimate the optimal PDU size and we run a number of simulation scenarios to study these parameters and how they impact on the performance of application protocols. The simulation results reveal that the channel bit error rate has a major impact on the optimal PDU size in the IEEE 802.16 networks. Also, the ARQ block rearrangement influences the performance.**

I. Introduction

IEEE 802.16 is a standard for the wireless broadband access network [1] that can provide a high-speed wireless access to the Internet to home and business subscribers. It supports applications and services with diverse Quality-of-Service (QoS) requirements, such as Voice-over-IP (VoIP). The core components of a 802.16 system are a subscriber station (SS) and a base station (BS). The BS and one or more SSs can form a cell with a point-to-multipoint (PMP) structure. On air, the BS controls the activity within a cell, resource allocations to achieve QoS, and network admission based on network security mechanisms. An overview of the key 802.16 features is given in [4].

The automatic repeat request (ARQ) is a mechanism by which the receiving end of a connection can request the retransmission of MAC protocol data unit (PDU), generally as a result of having received it with errors. It is a part of the 802.16 MAC layer and can be enabled on a per-connection basis. The 802.16 specification does not mandate the usage of the ARQ mechanism meaning that it is a provider and a customer specific decision.

Like in all wireless networks, errors happen all the time also in the IEEE 802.16 networks. The channel error rate has a significant impact upon how the ARQ and PDU size values should be chosen. Though 802.16 technology uses forward error correction (FEC) to correct bit level errors it cannot correct all the errors. This paper analyzes these parameters and studies the MAC level performance of the IEEE 802.16 technology. In particular, the optimal MAC PDU size is considered with different FEC block error rate (BLER) values and ARQ block rearrangement on and off.

This paper extends our previous research and simulation work on 802.16 networks. In [6], we presented a scheduling solution for the 802.16 BS. In [5], we analyzed the 802.16 contention resolution mechanism and proposed an adaptive algorithm to adjust the backoff parameters and to allocate a sufficient number of request transmission opportunities. In [7], we presented a general overview of the ARQ mechanism. A choice for the ARQ feedback type, ARQ block rearrangement and ARQ transmission window were considered.

The optimal MAC PDU size in IEEE 802.16 has been studied also in [8], where authors introduce a variable sized CRC and PDU size mechanism. The sizes are changed according to the six level feedback of receptions status. *Other sub-header* bit is used to distinguish important PDUs upon transmission. The adaptive CRC field had a minor impact but the adaptive payload size had a significant impact. However, the introduced method requires changes to the standard and additional feedback of reception status.

The rest of the article is organized as follows. Section II presents the key features and parameters of the 802.16 ARQ mechanism. A model for selecting the optimal PDU size and an upper limit for MAC PDU size when rearrangement is disabled are presented. Next, Section III verifies the proposed model with simulations results. This section also presents a scenario to study the ARQ block rearrangement performance and analyzes the simulation results. Finally, Section IV concludes the article and outlines further research directions.

II. IEEE 802.16 MAC

Table I shows a list of IEEE 802.16 MAC headers and their sizes when ARQ is enabled. When ARQ is disabled PSH and FSH are 1 byte smaller. GMH is always present.

When considering the MAC level performance, fragmentation has a significant impact. If it is not used and a SDU is bigger than the burst size, transmission is not possible. Packing is a useful feature and it does reduce overhead particularly with small SDU sizes, but the absence of it does not cause dramatic performance drop. Specification does not mandate packing, but unpacking, fragmenting and unfragmenting are

TABLE I
IEEE 802.16 MAC HEADERS

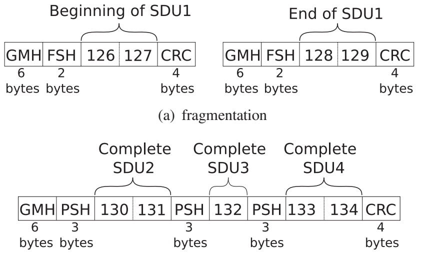
Header	Explanation	Size
GMH	general MAC header	6 bytes
GMSH	grant management subheader	2 bytes
PSH	packing subheader	3 bytes
FSH	fragmentation subheader	2 bytes
CRC	cyclic redundancy check	4 bytes

mandatory features. Both features are completely mandated in the WiMAX Forum Mobile System Profile [3].

A. Basics of the ARQ Mechanism

If ARQ is enabled for a connection, the FSH or the PSH is used. The subheader type is indicated by the extended bit in the GMH. Regardless of the subheader type, there is a block sequence number (BSN) in the subheader that indicates the *first* ARQ block number in the PDU. CRC header is always used when ARQ is enabled. A PDU is considered to comprise a number of ARQ blocks, each of which is of the same constant size except the final block which may be smaller. The ARQ block size is an ARQ connection parameter negotiated between the sender and the receiver upon a connection set-up. Fig. 1 presents ARQ blocks with the fragmentation and packing mechanisms. Block numbers are given with respect to the BSN stored either in the FSH (see Fig. 1(a)) or PSH (see Fig. 1(b)).

It is important to note that while the 802.16d specification [1] defines an ARQ block size as *any* value ranging from 1 to 2040 bytes, the 802.16e specification [2] has limited it to power of two values ranging from 16 to 1024 bytes, e.g. 16, 32, 64 and so on.



(a) fragmentation

(b) packing

Fig. 1. ARQ blocks with packing and fragmentation mechanisms.

While retransmitting a PDU, a connection may face a problem that an allocated data burst is smaller than the PDU size to be retransmitted. This may happen if the BS scheduler allocates data bursts of different sizes, which is usually the case for rtPS, nrtPS, and BE connections. Suppose, that the BS allocates a data burst of three slots for the BE connection and the latter sends a PDU that spans the whole data burst. If this PDU encounters an error, the connection will retransmit it. However, if the BS scheduler allocates later a data bursts of two slots, there is no way to retransmit the original PDU. Fortunately, the connection may rely upon *the retransmission with rearrangement* that allows for fragmenting the retransmitted PDU on the ARQ block size boundaries. If there is a sufficiently small ARQ block size, then the connection may construct a smaller PDU. As an example, Fig. 2 shows the rearranged PDU which originally presented in Fig. 1(b). The original PDU have been divided into two PDUs which contain the same three SDUs as the original PDU did.

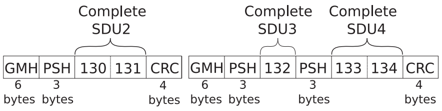


Fig. 2. Rearranged PDU.

The absence of rearrangement can be circumvented by using smaller PDUs. For UGS connections the SS knows the bandwidth connections will receive. The scheduler will allocate constant size burst to the UGS connections. These bursts will be allocated in every frame. The SS can therefore calculate the minimum burst size it will get, which is also the biggest safe PDU size. The SS can trust that it will not get a burst which is smaller than this and therefore retransmissions are always possible even without the rearrangement of retransmissions. It can also be so that UGS connection does not get bandwidth in every frame but less frequently. This means that the bursts will be bigger, but SS can not rely on this, so the upper limit for the PDU size is still the same.

For ertPS connections, the same principal as mentioned above can be used. For other QoS classes the minimum burst size is unknown to the SS. Thus the SS should limit the maximum size of transmitted PDUs to a fairly small value. One solution for finding this PDU size is the monitoring of burst sizes the SS gets. Then the SS should obey (1), where L is the PDU size limit and S_{burst} is the average size of the burst, to make sure that retransmissions are possible.

$$L < S_{burst} \tag{1}$$

B. Optimal PDU Size Estimation

The optimal MAC level PDU size depends on the number of factors. A larger MAC PDU size results in a less MAC overhead because there is the mandatory GMH per a PDU. If a connection utilizes the ARQ mechanism, then the PDU must contain CRC as well as FSH or PSH to hold the BSN. If we assume that application level SDU is bigger than the preferred MAC PDU, there is no need to pack several SDUs into one PDU. Therefore, only FSH is present. We can neglect GMSH because it appears only in one PDU in a data burst. In this case the MAC level overhead can be approximated as follows [6]:

$$O = S_{\text{GMH}} + S_{\text{FSH}} + S_{\text{CRC}}. \tag{2}$$

Assuming the PDU size is L bytes, the amount of user data S in a PDU is:

$$S(L) = L - O. \tag{3}$$

Similar to other wireless technologies, 802.16 uses the forward error correction (FEC) mechanism to improve data transmission reliability. In this paper we do not concentrate on the FEC behaviour but rather assume a certain FEC block error rate (BLER) as a function of the chosen modulation and coding scheme (MCS) and the effective signal-to-noise rate (SNR). If we assume some FEC BLER (E), the PDU error rate (E_P) can be calculated using the following expression, where B stands for the FEC block size measured in bytes:

$$E_P(L) = 1 - (1 - E)^{\frac{L}{B}}. \tag{4}$$

It is important to note that the presented expression can be applied if Hybrid ARQ (HARQ) is *not* enabled. Otherwise, HARQ retransmission gain influences significantly the BLER. Using previous formulas and taking the overhead and error probabilities into account, the optimal PDU size can be estimated. The efficiency (F) is presented as follows:

$$F(L) = \frac{S(L)}{L}(1 - E_P(L)) = \frac{S(L)}{L}(1 - E)^{\frac{L}{B}}. \tag{5}$$

Equation (5) tells how much user data the PDU contains compared to the total PDU size on average taking the error probability into account. The optimal PDU size is the one which gives the best efficiency value.

Fig. 3 shows the results for three different BLER values when the PDU size varies. The FEC block size of 36 bytes comes from the 802.16 OFDMa PHY 16-QAM1/2 MCS that we choose for this particular case. As can be seen from Fig. 3, BLER has a major effect on the efficiency values with different PDU sizes. As anticipated, the optimal PDU size tends to be smaller when there are more errors in the channel. Using a large PDU size of 500 bytes in the most erroneous case where BLER is 10^{-1} causes a very bad performance compared to the optimal PDU size of about 60 bytes.

We can use (5) to find the optimal PDU size for the given channel BLER:

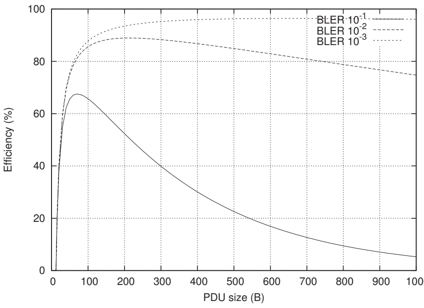


Fig. 3. Theoretical MAC level efficiency from (5) for different BLER ($O =$ 12 bytes, $B = 36$ bytes).

$$F = \frac{L - O}{L}(1 - E)^{\frac{L}{B}}, \tag{6}$$

$$\ln F = \ln(L - O) - \ln L + \frac{L}{B}\ln(1 - E), \tag{7}$$

$$\frac{F\prime}{F} = \frac{1}{L - O} - \frac{1}{L} + \frac{\ln(1 - E)}{B}, \tag{8}$$

$$\frac{F\prime}{F} = \frac{O}{L(L - O)} + \frac{\ln(1 - E)}{B}, \tag{9}$$

$$F\prime = \frac{L - O}{L}(1 - E)^{\frac{L}{B}}\left(\frac{O}{L(L - O)} + \frac{\ln(1 - E)}{B}\right), \tag{10}$$

$$(1 - E)^{\frac{L}{B}}\left(\frac{O}{L^2} + \frac{L - O}{LB}\ln(1 - E)\right) = 0, \tag{11}$$

$$(1 - E)^{\frac{L}{B}} > 0, \frac{O}{L^2} + \frac{L - O}{LB}\ln(1 - E) = 0, \tag{12}$$

$$\frac{\ln(1 - E)}{B}L^2 - \frac{O}{B}\ln(1 - E)L + O = 0, \tag{13}$$

$$L = \frac{O}{2} - \frac{\sqrt{(O\ln(1 - E))^2 - 4BO\ln(1 - E)}}{2\ln(1 - E)}. \tag{14}$$

Fig. 4 shows the results for the optimal PDU size with 16QAM1/2 MCS. By comparing the optimal PDU size in Fig. 3 and Fig. 4 it is possible to arrive at the conclusion that theoretical calculations are correct.

According to the presented above considerations, if a connection resorts to using a constant PDU size, then this value should be sufficiently small. On good channel conditions, small PDUs do not decrease the performance significantly. However, with a bad channel, small PDUs increase the performance greatly. In a real environment, the BLER may vary greatly as a result of changing SNR. So unless reliable information about errors can be gathered in real-time, the PDU size should be small, at least less than 200 bytes.

It is worth noting the BS and the SS measure constantly their carrier-to-interference-and-noise rate (CINR). There is a relationship between CINR-level, MCS, and BLER. Therefore the BS can change the MCS to achieve the target BLER. It is then possible to decide one PDU size limit for each MCS

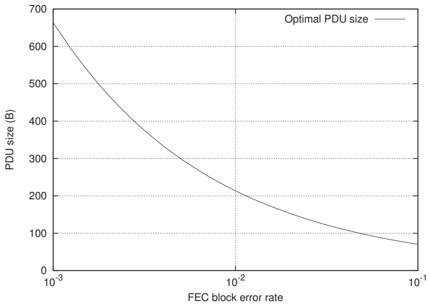


Fig. 4. Theoretical optimal PDU size from (14) with 16QAM1/2 ($O = 12$ bytes, $B = 36$ bytes).

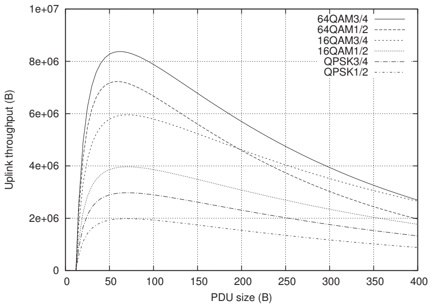


Fig. 5. Theoretical uplink throughput based on (16) with different MCS. BLER = 10^{-1}.

by using equation (14). There could even be several limits per MCS in order to get the PDU size as close to the optimum size as possible. It is important to note the adaptive PDU size approach is feasible only for the downlink transmission because the BS receives reports from SSs about the signal quality. There is no such a report sent by the BS to SSs – the BS just selects the suitable MCS for the uplink transmission based on the received signal strength. Thus, in the uplink direction, SSs should take the worst-case approach and choose smaller PDU sizes.

$$\mathrm{BW} = \mathrm{FPS} \cdot N_{\text{slot}} \cdot S_{\text{slot}} \tag{15}$$

$$F(L) \cdot \mathrm{BW} \tag{16}$$

Equation (15) shows a definition for bandwidth. In this FPS means frames per second, N_{slot} is the number of uplink or downlink slots, and S_{slot} is a single slot size. Equation (16) shows a way to estimate uplink or downlink data for one SS. This formula assumes that all data is user data, management messages are not taken into account. Fig. 5 presents an example how to use (16) to estimate throughput with different PDU sizes. The figure shows results for six different MCS and same BLER. The FEC block size for 64QAM2/3 was 24 bytes, for 64QAM3/4 it was 27 bytes and for others 36 bytes.

III. Simulation

This section presents a simulation analysis of the 802.16 ARQ mechanism and the optimal MAC PDU size. To run simulations, we have implemented the 802.16 MAC and PHY levels in the NS-2 simulator. The implementation is called WINSE (WImax NS-2 Extension) and it contains the main features of the 802.16 standard, such as downlink and uplink transmission, connections, MAC PDUs, packing and fragmentation, the contention and ranging periods, the MAC level management messages, and the ARQ mechanism. The ARQ implementation supports the ARQ blocks, the ARQ transmission window, retransmission with rearrangement, ARQ timers and all the ARQ feedback types. The ARQ implementation also includes the prioritization of the ARQ feedbacks and retransmissions, and the algorithm to select the feedback type. [9] presents ARQ related results and algorithms. The implemented PHY is OFDMa. The simulation results for the OFDM PHY can be found in [6], [5].

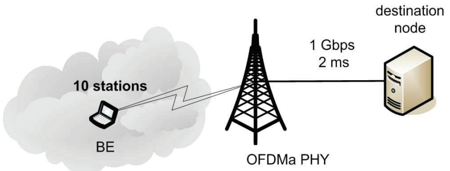


Fig. 6. Network structure.

Fig. 6 shows the network structure we used in the simulation scenarios. There is the BS controlling the WiMAX network, the parameters of which are presented in Table II, ten SSs, and one wired node. The details of the scheduling algorithm at the BS are presented in [6]. In a few words, the BS allocates resources fairly between the SSs based on their bandwidth request sizes. Each SS establishes one uplink and downlink BE connection to the BS (each SS also establishes the basic management connection to exchange management messages with the BS). An SS hosts exactly one FTP-like application that sends data over the TCP protocol to a wired node. The reason we choose such an application type is that it tries to send as much data as possible thus utilizing all the available wireless resources. At the same time, the TCP protocol is very sensible to the packet drops that can occur in the wireless part. The basic ARQ results for the OFDMa PHY can be found from [7].

A. BLER and PDU Size

Fig. 7 shows simulation and theoretical results for different PDU sizes and two different BLER values. If we look simulation results, it can be seen that the error rate has a major impact on optimal PDU size. When using BLER of 10^{-1},

TABLE II

SIMULATION PARAMETERS.

Parameter	**Value**
Simulator	NS2
802.16 extension	WINSE
Simulation length	10 s
Application protocol	FTP
Parameter	**Value**
PHY	OFDMa
Bandwidth	10 MHz
FFT	1024
Cyclic prefix length	1/8
TTG+RTG	464 PS (0.082857 ms)
Duplexing mode	TDD
Frames per second	200 (5 ms per frame)
OFDM symbols	47
DL/UL symbols	26/21
DL/UL subcarrier alloc.	DL FUSC/UL PUSC
DL/UL slots	416/245
MCS	16-QAM1/2 (12 bytes/slot)
FEC block size	3 slots (36 bytes)
Ranging transm. opport.	2
Ranging backoff start/end	2/15
Request transm. opport.	8
Request backoff start/end	4/15
CDMA codes	256
ranging+periodic ranging	64
bandwidth request	192
handover	–
Fragmentation/packing	ON
CRC/ARQ	ON
ARQ feedback	standalone
ARQ feedback types	all
ARQ block size	16 bytes
ARQ window	1024
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ retry timeout	40 ms
ARQ block lifetime	300 ms

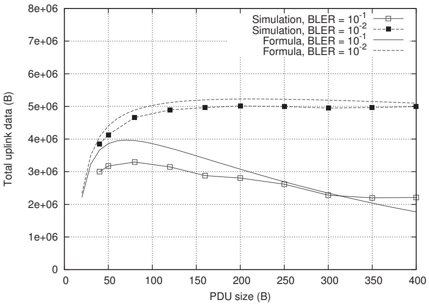


Fig. 7. Total uplink throughput from simulations and theoretical calculations based on (5).

smaller PDUs give better results. It can be seen that in this case the optimal PDU size is about 80 bytes. When BLER is 10^{-2}, larger PDUs give better results and the results with PDU sizes of 160-400 are almost the same.

Fig. 7 also presents a theoretical uplink throughput with the BLER values of 10^{-1} and 10^{-2}. These values are obtained by using (16). The results are optimistic because of the simplified the equation that does not account for all possible sources of overhead. Nevertheless, Fig. 7 shows that (5) provides a good estimation of the optimal PDU size.

B. ARQ Block Rearrangement

The availability of ARQ Block rearrangement feature and the number of SS also have an impact on optimal PDU size. The PDU size should obey equation (1). This is shown in Fig. 8, which shows total uplink throughput with different PDU sizes and rearrangement on and off. When the rearrangement is enabled and there are 10 SSs, the optimal PDU size is about 150 bytes or more. With smaller PDU sizes the overhead from the headers decreases the performance. When the rearrangement is not used, the PDU sizes bigger than 300 bytes start to decrease the performance when compared to the case when the rearrangement is enabled. When the PDU size is less than average burst size (about 300 bytes) per SS, retransmissions are usually possible even without rearrangement. When PDU limit is more than 300 bytes and error occurs, it might be that the PDU cannot be retransmitted, and an SS cannot send anything during that frame. Using small PDUs can compensate the absence of the ARQ block rearrangement feature, but then overhead from headers is bigger. Depending on the channel conditions smaller PDUs might be preferred in any case as shown earlier.

The optimal PDU size without rearrangement also depends much on the network load as can be seen when there are 25 SSs. In this case the average burst for each SS is about 120 bytes. Allowing to use bigger PDUs drops the performance when the rearrangement is disabled. More SSs transfer less bytes in total because the average PDU size is smaller and therefore the MAC header overhead is bigger. Also the MAP-messages are bigger and there are more other management messages.

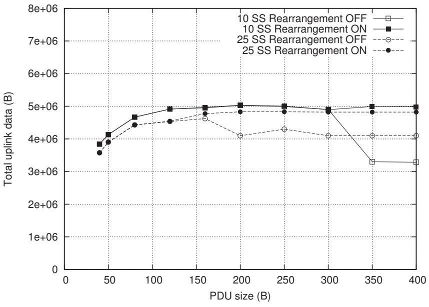


Fig. 8. Total uplink throughput with different PDU sizes and rearrangement on/off. BLER = 10^{-2}.

IV. Conclusions

In this paper, we have analyzed the MAC level performance of the IEEE 802.16. We proposed a method of estimating the optimal PDU size when the error rate for the FEC block is known. The simulation results verified the correctness of this method. The results also showed that if the ARQ block rearrangement is not used, then the average burst size impacts the optimal PDU size.

We proposed to estimate the error rate for FEC blocks through the CINR measurements, and then adjust the MAC PDU size with the equation (14). The proposed method does not require any changes to the specification and can be therefore implemented easily either at the BS or at the SS side. If the error rate cannot be estimated or measured in real-time, the PDU size should be rather small.

In the future we will continue to study the optimal PDU size and error probabilities with QoS-enabled connections like VoIP. We will also study the affect of different error probabilities to other ARQ parameters like ARQ timers. In addition other error correction methods like H-ARQ will be considered.

References

[1] Air interface for fixed broadband wireless access systems. IEEE Standard 802.16, Jun 2004.

[2] Air interface for fixed broadband wireless access systems - amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands. IEEE Standard 802.16e, Dec 2005.

[3] WiMAX Forum Mobile System Profile, Release 1.0 Approved Specification, Apr 2007. Revision 1.4.0.

[4] C. Eklund, R. Marks, K. Stenwood, and S. Wang. IEEE standard 802.16: a technical overview of the Wireless MAN air interface for broadband wireless access. *IEEE Communications*, 40(6):98–107, Jun 2002.

[5] A. Sayenko, O. Alanen, and T. Hämäläinen. Adaptive contention resolution for VoIP services in IEEE 802.16 networks. In *The 8th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Jun 2007.

[6] A. Sayenko, O. Alanen, J. Karhula, and T. Hämäläinen. Ensuring the QoS requirements in 802.16 scheduling. In *The 9th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 108–117, Oct 2006.

[7] A. Sayenko, V. Tykhomyrov, H. Martikainen, and O. Alanen. Performance analysis of the IEEE 802.16 ARQ mechanism. In *The 10th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 314–322.

[8] Shamik Sengupta, Mainak Chatterjee, Samrat Ganguly, and Rauf Izmailov. Exploiting MAC flexibility in WiMAX for media streaming. In *Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, pages 338–343, June 2005.

[9] V. Tykhomyrov, A. Sayenko, H. Martikainen, O. Alanen, and T. Hämäläinen. Performance evaluation of the IEEE 802.16 ARQ mechanism. In *7th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking*, pages 148–161, Sep 2007.

PII

IMPACT OF PORTABLE DEVICE RESTRICTIONS ON IEEE 802.16 PERFORMANCE

by

H. Martikainen, O. Alanen and V. Tykhomyrov 2008

The 2nd IEEE International Interdisciplinary Conference on Portable
Information Devices

Impact Of Portable Device Restrictions On IEEE 802.16 Performance

Henrik Martikainen, Olli Alanen, Vitaliy Tykhomyrov

Telecommunication laboratory, MIT department,
University of Jyväskylä, Finland
{henrik.martikainen, olli.alanen, vitykhom}@jyu.fi

***Abstract*—IEEE 802.16 is a good alternative for fast wireless connection in the next generation portable information devices (PID). However it was not originally designed for portable devices, but mobility was added later to the specification. This means that the lack of resources on portable devices might have a significant impact on which IEEE 802.16 features should be chosen and how they should be used. Proper usage of ARQ can increase the IEEE 802.16 performance but it can also require much resources from PIDs. In this paper we focus on how ARQ features are affected by limited resources on PID and how much the limitations affect on the performance.**

I. Introduction

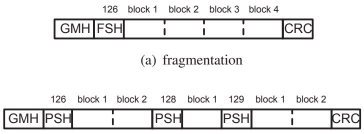
IEEE 802.16 is a standard for the wireless broadband access network [1] that can provide a high-speed wireless access to the Internet to home and business subscribers. It supports applications and services with diverse Quality-of-Service (QoS) requirements, such as Voice-over-IP (VoIP). The core components of a 802.16 system are a subscriber station (SS) and a base station (BS). The BS and one or more SSs can form a cell with a point-to-multipoint (PMP) structure. On air, the BS controls the activity within a cell, resource allocations to achieve QoS, and network admission based on network security mechanisms. An overview of the key 802.16 features is given in [5].

802.16e-2005 [2] is an extension to the original 802.16-2004 -standard, which brings full mobility support. Still not all the features of the of 802.16 can be used on portable information devices (PID). The WiMAX Forum has created WiMAX Forum Mobile System Profile [3], which defines which features have to supported so a PID can receive a WiMAX Forum certificate but even this document does not specify how 802.16 features should be used on PID. This article covers how ARQ efficiency are affected by limited processing power, low memory and limited battery lifetime on PID.

The rest of the article is organized as follows. Section II presents ARQ basics and how the PID restrictions might affect on usage of them. Section III presents simulation results and analysis of these. Finally, Section IV concludes the article and outlines further research directions.

II. IEEE 802.16 ARQ Basics

The IEEE 802.16 technology defines the ARQ mechanism that enables a connection to resend data at the MAC level if an error is detected. When ARQ is enabled for a connection the PDUs are divided into ARQ blocks, which can be retransmitted separately. Fig. 1 presents ARQ blocks with the fragmentation and packing mechanisms. Block numbers are given with respect to the BSN stored either in the FSH (see Fig. 1(a)) or PSH (see Fig. 1(b)). Because ARQ block boundaries are not explicitly marked the ARQ block size is a constant parameter, which can not be changed during the connection. Smaller block sizes provide greater flexibility for retransmissions but consume more resources since every block has a set of ARQ timers.



(a) fragmentation

(b) packing

Fig. 1. ARQ blocks with packing and fragmentation mechanisms.

A. ARQ Feedback Types

To request a retransmission of blocks (NACK) or to indicate a successful reception of blocks (ACK), a connection uses ARQ block sequence numbers. In turn, the sequence numbers are exchanged by means of the ARQ feedback messages. The specification defines the following feedback types: a) selective, b) cumulative, c) cumulative+selective, and d) cumulative+sequence. Fig. 2 presents an example in which every feedback type is applied to the same set of ARQ blocks. A detailed information about ARQ feedbacks can be found from [9].

Creating feedback type a), b) or c) is straightforward but type d) cumulative+sequence is more complicated. It can have 2 or 3 sequences which are 15 or 63 blocks long. The same set of blocks can be therefore acknowledged differently and optimum selection can require some processing. It is possible that a PID might not support the creation of this type.

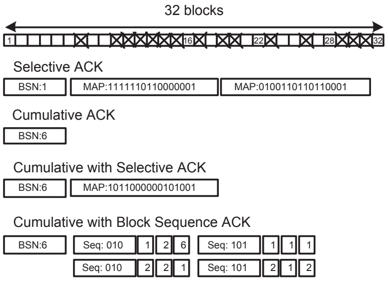


Fig. 2. Example of ARQ feedback types.

B. ARQ Transmission Window and ARQ Block Size

At any time a sender may have a number of outstanding and awaiting acknowledgments ARQ blocks. This number is limited by the ARQ transmission window that is negotiated between an SS and the BS during a connection set-up. A sufficiently large ARQ window allows for a continuous transmission of data. A connection can continue to send ARQ blocks without waiting for each block to be acknowledged. Conversely, a smaller ARQ window causes a sender to pause a transmission of new ARQ blocks until a timeout or the ARQ feedback is received. Though it may seem that a large transmission window is always the best choice, it is worth noting that a large transmission window leads to increased memory consumption and processing load. Every ARQ block must be stored in the retransmission buffer until a positive feedback is received. Taking into account the largest ARQ block size of 1024 bytes and the maximum ARQ transmission window of 1024 blocks, it is possible to arrive at the conclusion that some mobile and portable devices will not have enough resources to handle this amount of data for each frame.

The ARQ transmission window and the ARQ block size parameters depend one on each other. On the one hand, a connection may prefer to work with a small ARQ transmission window that will result in a necessity of choosing a larger ARQ block size because the throughput may be limited by the transmission window size. A large block size requires less resources because a set of the ARQ timers must be associated with a single ARQ block at the sender and the receiver. At the same time, a connection supporting the retransmission with rearrangement may wish to work with a smaller ARQ block size because that will provide a greater flexibility in splitting large PDUs into several smaller ones.

C. ARQ Block Rearrangement

While retransmitting a PDU, a connection may face a problem that an allocated data burst is smaller than the PDU size to be retransmitted. Suppose, that the BS allocates a data burst of three slots for the BE connection and the latter sends a PDU that spans the whole data burst. If this PDU encounters an error, the connection will retransmit it. However, if the BS scheduler allocates later a data bursts of two slots, there is no way to retransmit the original PDU. Fortunately, the connection may rely upon *the retransmission with rearrangement* that allows for fragmenting the retransmitted PDU on the ARQ block size boundaries. As an example, Fig. 3 shows the rearranged PDU presented in Fig. 1(a).

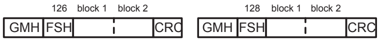


Fig. 3. Rearranged PDU.

It can be that the sender does not support ARQ block rearrangement because rearrangements involve much more complicated actions with PDUs in the retransmission buffer when compared to the PDU construction. A sender must keep a set of the ARQ timers for each ARQ block. If the retransmission with rearrangement is not implemented, then eventually a sender can associate all those timers with a PDU, which requires much less resources. Furthermore, the rearrangement requires a sender to analyze a PDU and to search for block boundaries on which that PDU can be fragmented. In this case the only safe way to ensure smooth transmission is to control the maximum size of transmitted PDUs. More detailed research of optimal PDU size was done in [6].

III. Simulation

We will use a network simulator to show how small ARQ window, bigger ARQ block sizes, absence of ARQ block rearrangement and some ARQ feedback types will decrease the performance of a PID. To run simulations, we have implemented the 802.16 MAC and PHY levels in the NS-2 simulator. The implementation is called WINSE (WImax NS-2 Extension) and it contains the main features of the 802.16 standard, such as downlink and uplink transmission, connections, MAC PDUs, packing and fragmentation, the contention and ranging periods, the MAC level management messages, and the ARQ mechanism. The implemented PHY is OFDMa. The PHY model is based on trace files gathered from the link level multi-cell simulations with the reuse 1-3-3 factor. There are uplink and downlink trace files that are chosen randomly by stations and are read from a random starting index. Then, MAC PDU error generation follows the IEEE 802.16m simulation methodology [4] and is based on the MAC PDU size and FEC BLER, whereas the latter is determined based on the current MCS and the SNR. The simulation results for the OFDM PHY can be found in [8], [7].

Fig. 4 shows the general simulation scenario. There are several subscriber stations (SS) connected to an IEEE 802.16 base station. A file server is connected to the base station using a fast low-latency wired connection. The number of SSs and direction of FTP traffic depends on the particular simulation subcase. Subcases were run five times and presented results are averages from these. Table I [1] shows the network parameters used in the simulations.

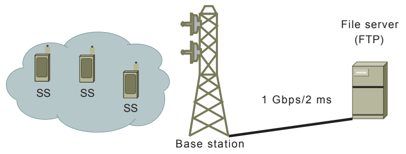


Fig. 4. Network structure.

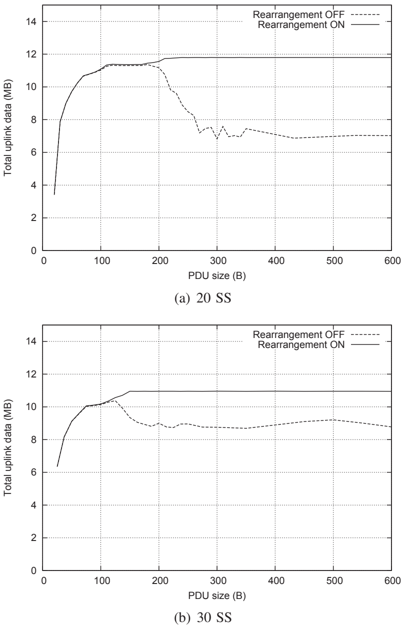
TABLE I
802.16 NETWORK PARAMETERS.

Parameter	**Value**
Simulator	NS2
802.16 extension	WINSE
Simulation length	20 s
Application protocol	FTP
Parameter	**Value**
PHY	OFDMa
Bandwidth	10 MHz
FFT	1024
Cyclic prefix length	1/8
TTG+RTG	464 PS
Duplexing mode	TDD
Frames per second	200 (5 ms per frame)
OFDM symbols	47
DL/UL symbols	32/15
DL/UL subcarrier alloc.	DL PUSC/UL PUSC
DL/UL slots	480/175
DL/UL channel measurements	preamble / data burst
Channel report type / interval	REP-RSP / 20ms
Channel measurement filter	EWMA / α = 0.25
MAP MCS	QPSK1/2
Compressed MAP	ON
Ranging transm. opport.	2
Ranging backoff start/end	1/15
Request transm. opport.	8
Request backoff start/end	3/15
CDMA codes	256
ranging+periodic ranging	64
bandwidth request	192
handover	–
Fragmentation/packing	ON
PDU size	208 B
CRC/ARQ	ON
ARQ feedback	standalone
ARQ feedback types	all
ARQ feedback interval	20 ms
ARQ block size	32 B
ARQ window	1024
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ timers	
retry	50ms
block lifetime	500ms
Rx purge	500ms

A. ARQ Block Rearrangement

In this subsection we study the impact of ARQ block rearrangement on the uplink throughput. Fig. 5(a) shows the total transmitted data with and without ARQ block rearrangement using different MAC level PDU size limit. It can be seen that the best results are obtained by using rearrangement and PDU size of 200 bytes or more. Without rearrangement performance drops significantly if large PDUs are used. This can be prevented by limiting the PDU size so it is less than average burst size. In this case the average burst size is about 200 bytes.

Note that the average burst size depends heavily on the load of the network, type of traffic and scheduler. Fig. 5(b) shows how the PDU size limit is even lower with 30 SS, which will cause more header overhead and decrease the performance. In both cases it is clear that ARQ block rearrangement improves the performance. However if rearrangement is too complicated feature for a PID it can still achieve reasonable performance without it by limiting the MAC level PDU size.



(a) 20 SS

(b) 30 SS

Fig. 5. Impact of ARQ block rearrangement and PDU size on total uplink data.

B. ARQ Feedback Types

In this simulation sub case, we study ARQ feedback types. In all the cases the simulator chooses the appropriate feedback

[1] These parameters conform the WiMAX Forum mobile system profile [3].

from allowed ones using an algorithm presented in [9]. The traffic is downlink FTP traffic because then it's SS responsbility to create ARQ feedbacks which is studied here.

In Fig. 6 can be seen four different simulation cases. In this case downlink FTP-traffic was used and SSes are responsible for creating the ARQ feedback messages. In the first case only selective ARQ feedback type is used. It is clear that this is not efficient since all the blocks have to be acknowledged explicitly. In the second case also standalone cumulative feedbacks are used. This increases the performance significantly since all the correctly received blocks can be acknowledged with a single cumulative message. In the third case a combined cumulative+selective type is also used, which again boosts the performance although not as much as in previous case. The boost is achieved because the combined type can store the same information as separte cumulative and selective types in one message and therefore reduce the overhead. Finally in the fourth case also cumulative+sequence type is allowed. Also this type increases the performance because it can acknowledge more blocks in one message than cumulative+selective.

In conclusion it can be seen at least cumulative and selective types should be used. The extra benefit from cumulative+selective and cumulative+sequence is much smaller. So if creating cumulative+sequence takes too much resources on a PID it can decide no to use that type and still it will not have a major impact on the performance. WiMAX Forum Mobile System Profile [3] mandates that the support for all the types but selective is mandatory. This means that a SS has to able to receive those types but does not mandate the SS to use them. Indeed if cumulative+sequence type is supported our feedback selection algorithm does not select the selective type at all, hence the result is exactly the same as in case four.

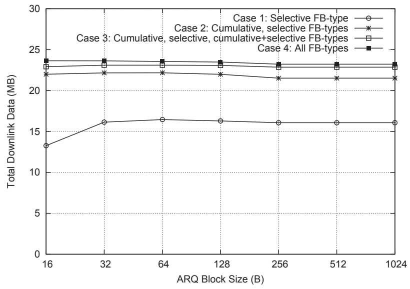


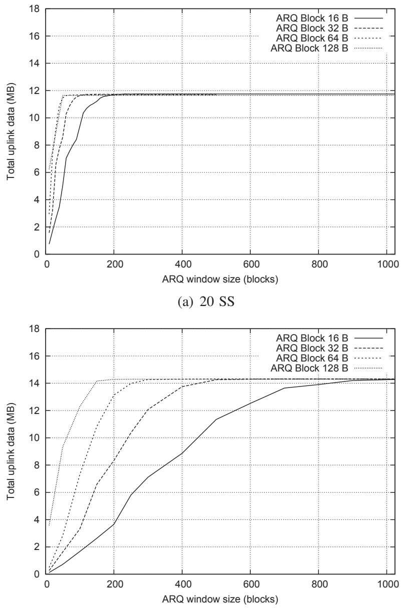
Fig. 6. Impact of different ARQ feedback types and ARQ block sizes on total downlink data (20 SS).

C. ARQ Window and ARQ Block Size

In this simulation scenario we present the simulation results for different ARQ window and ARQ block size values. Fig. 7(a) shows the results when there are 20 SSs receiving FTP traffic. It can be seen that if the ARQ window is more than 200 blocks it does not restrict the performance at all. Also the results for different block sizes of 16-128 are almost the same if the window is more than 200 blocks.

Fig. 7(b) shows the same scenario when there are only 5 SSs present. This case has similar characteristics compared to previous one. If the ARQ window is big enough then the ARQ block size does not matter. However if the ARQ window is for example 300 blocks there is a big difference in total uplink data between the ARQ block sizes of 16-128 bytes.

Also if we analyze also the results from Fig. 6 with different ARQ block sizes it is clear that the smallest block size is not an optimal selection. In practice there is no difference between block sizes of 16 and 128 in performance but the first one requires 8 times more resources for ARQ timers.



(a) 20 SS

(b) 5 SS

Fig. 7. Impact of ARQ window size and ARQ block size on total downlink data.

IV. Conclusions

In this paper we have showed how the absence of resources can limit the performance of IEEE 802.16 on PIDs. Low processing power and limited battery lifetime restrict the usage of sophisticated and powerful algorithms. The lack of memory has an impact on how big the various ARQ buffers can be. However some of these limitations can be circumvented by using appropriate values for parameters. In the future we plan to study what kind of impact these restrictions have to other IEEE 802.16 features.

V. Acknowledgements

We would like to give special thanks to Alexander Sayenko (Nokia) for his valuable input on the topic.

References

[1] Air interface for fixed broadband wireless access systems. IEEE Standard 802.16, Jun 2004.

[2] Air interface for fixed broadband wireless access systems - amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands. IEEE Standard 802.16e, Dec 2005.

[3] WiMAX Forum Mobile System Profile, Release 1.0 Approved Specification, Nov 2007. Revision 1.5.0.

[4] IEEE 802.16m evaluation methodology document (EMD). IEEE 802.16 Broadband Wireless Access Group, Mar 2008.

[5] C. Eklund, R. Marks, K. Stenwood, and S. Wang. IEEE standard 802.16: a technical overview of the Wireless MAN air interface for broadband wireless access. *IEEE Communications*, 40(6):98–107, Jun 2002.

[6] H. Martikainen, A. Sayenko, O. Alanen, and V. Tykhomyrov. Optimal MAC PDU size in IEEE 802.16. In *4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks*, pages 66–71, Feb 2008.

[7] A. Sayenko, O. Alanen, and T. Hämäläinen. Adaptive contention resolution parameters for the IEEE 802.16 networks. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, Aug 2007.

[8] A. Sayenko, O. Alanen, J. Karhula, and T. Hämäläinen. Ensuring the QoS requirements in 802.16 scheduling. In *The 9th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 108–117, Oct 2006.

[9] A. Sayenko, V. Tykhomyrov, H. Martikainen, and O. Alanen. Performance analysis of the IEEE 802.16 ARQ mechanism. In *The 10th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 314–322, Oct 2007.

PIII

ARQ PARAMETERS FOR VOIP IN IEEE 802.16 NETWORKS

by

H. Martikainen, O. Alanen and A. Sayenko 2009

Wireless Telecommunications Symposium

ARQ Parameters for VoIP in IEEE 802.16 Networks

Henrik Martikainen and Olli Alanen
Telecommunication Laboratory, MIT department
University of Jyväskylä, Finland
{henrik.martikainen, olli.alanen}@jyu.fi

Alexander Sayenko, *Member, IEEE*
Research & Technology Platforms
Nokia Siemens Networks, Finland
alexander.sayenko@nsn.com

***Abstract*—IEEE 802.16 standard defines two data retransmission mechanisms. HARQ provides fast retransmissions in cost of slightly increased overhead. On the contrary, ARQ has less overhead in cost of bit longer delays. It is therefore often used with BE connections. In addition to delay tolerant applications, BE can also be used for real-time services. Therefore, in this paper we analyze the ARQ mechanism and provide guidelines on how to set the ARQ parameters to achieve a good balance between the VoIP delay and packet loss.**

***Index Terms*—IEEE 802.16 WiMAX, ARQ, VoIP, BE, NS-2**

I. Introduction

IEEE 802.16 is a standard for the wireless broadband access network [1], [2] that can provide a high-speed wireless access to the Internet to home and business subscribers. It supports applications and services with diverse Quality-of-Service (QoS) requirements. The core components of a 802.16 system are a subscriber station (SS) and a base station (BS). The BS and one or more SSs can form a cell with a point-to-multipoint (PMP) structure. In this case, the BS controls the activity within a cell, the resource allocations to achieve QoS, and admission based on the network security mechanisms.

Voice-over-IP (VoIP) and its support is one of the most crucial features of the IEEE 802.16 system. Otherwise, it cannot compete efficiently with existent 3G and coming 3G LTE technologies, where VoIP is a major service. Since IEEE 802.16, being an IP solution, has no internal VoIP specific carriers, it is important to ensure that all the VoIP requirements are met that usually include throughput and delay. According to [9], one-way transmission delay must not exceed 150 ms. Since one-way delay comprises the wireless part, access network, and the core network delays, it is translated into the wireless part requirement that 95% of all the VoIP packets must experience a delay less than 50 ms [3], [8].

It is understandable that the wireless network is prone to errors. Thus, a key to an efficient VoIP functioning is a fast and reliable retransmission mechanism. IEEE 802.16 provides two mechanisms: ARQ and HARQ. Even though HARQ is considered as a better candidate for VoIP applications due to a faster feedback and the retransmission gain, its usage is not always desirable or even possible. As already considered in [13], HARQ possesses a larger signaling overhead. Furthermore, HARQ is defined only for the OFDMa PHY, whereas some providers may still run the OFDM PHY based 802.16 networks. In addition, in reality it might turn out that real customers of IEEE 802.16 networks will only have a BE subscription, as it is already the case with ADSL and many 802.11 hotspots. Then, a provider for his own internal reasons may use ARQ with such a subscription. Furthermore, a provider will not and cannot differentiate between applications that a customer will use over a single BE connection. It brings VoIP into the worst conditions: BE and ARQ. Even if a provider establishes several transport connections, say extended real-time Priority Service (ertPS) and BE, with a customer to differentiate between VoIP and other applications, it can be the case that a customer VoIP client is not smart to select a proper connection. All these peculiarities bring us to a situation where a need to study VoIP with the ARQ mechanism and BE arises. It can answer a question about the VoIP performance in IEEE 802.16 networks under the worst conditions.

1-4244-2589-1/09/$20.00 © 2009 IEEE

ARQ and VoIP have not been studied thoroughly especially in the IEEE 802.16 networks. In [14], authors studied how different IEEE 802.16 features impact the quality of VoIP calls. They observe that the VoIP quality is more sensitive to loss than delay. In [7], ARQ impact on QoS in 3GPP is studied. A file transfer process and VoIP are considered as typical applications but are analyzed separately. Also, the ARQ model is somewhat simplified and does not correspond to IEEE 802.16 ARQ. Nevertheless, authors come to a conclusion that ARQ can guarantee QoS requirements.

The rest of this paper is organized as follows. Section II provides an insight on the IEEE 802.16 ARQ mechanism. Section III presents the simulation results. Finally, section IV concludes the paper.

II. 802.16 ARQ mechanism

A. Basics of the ARQ Mechanism

A detailed analysis of the ARQ performance is given in [15]. It is worth mentioning that ARQ adds some overhead at the MAC layer. If it is enabled for a connection, then a protocol data unit (PDU) comprises a number of logical ARQ blocks, each of which is of the same constant size except the final block which may be smaller. Then, the extended fragmentation subheader (FSH) or the extended packing subheader (PSH) carries a block sequence number (BSN) that indicates the *first* ARQ block number in the PDU. To request a retransmission of blocks (NACK) or to indicate a successful reception of blocks (ACK), a connection uses ARQ block sequence numbers. In turn, the sequence numbers are exchanged by means of the ARQ feedback messages.

B. ARQ Timers

The IEEE 802.16 specification defines several ARQ timers. Fig. 1 shows how they relate to the ARQ block states. The ARQ block may be in one of the following five states: *done*, *not-sent*, *outstanding*, *discarded*, and *waiting-for-retransmission*. Firstly, as can be seen from Fig. 1, any ARQ block begins as *not-sent*. After it is sent it becomes outstanding for a period of time termed ACK_RETRY_TIMEOUT, which determines the minimum time interval a transmitter can wait before retransmission of an unacknowledged block for retransmission. The interval begins when the ARQ block was last transmitted. While a block is in *outstanding* state, it is either acknowledged and changed to *done*, or transitions to *waiting-for-retransmission* after ACK_RETRY_TIMEOUT or NACK.

An ARQ block can become *waiting-for-retransmission* before the ACK_RETRY_TIMEOUT period expires if it is negatively acknowledged. An ARQ block may also change from *waiting-for-retransmission* to *done* when an ACK message for it is received or to *discarded* after a timeout ARQ_BLOCK_LIFETIME, which determines the maximum time interval an ARQ block can be managed by the transmitter ARQ state machine, once the initial transmission of the block has occurred. If transmission (or subsequent retransmission) of the block is not acknowledged by the receiver before the time limit is reached, the block is *discarded* [16].

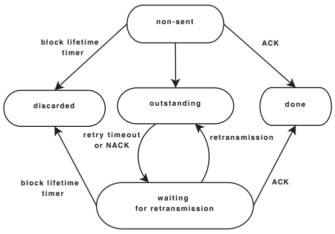


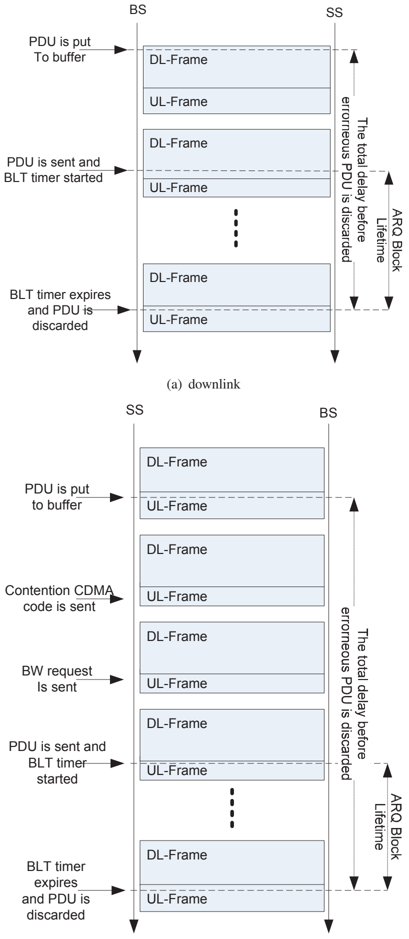
Fig. 1. ARQ transmit block states.

III. VoIP Quality and ARQ Parameters

In this section we present the main QoS parameters which have an impact on the VoIP quality and how the different ARQ parameters relate to them. Then we present guidelines for the presented ARQ parameters.

A. Maximum Latency

According to the IEEE 802.16 specification, *maximum latency* is one of the QoS parameters for the UGS, rtPS and ertPS service flows. There are several issues that influence the latency: request/transmission policy, polling interval, HARQ and ARQ parameters, cotention resolution, etc.


BS
SS
PDU is put To buffer
DL-Frame
UL-Frame
DL-Frame
PDU is sent and BLT timer started
UL-Frame
DL-Frame
BLT timer expires and PDU is discarded
UL-Frame
The total delay before erroneous PDU is discarded
ARQ Block Lifetime
SS
BS
DL-Frame
PDU is put to buffer
UL-Frame
DL-Frame
Contention CDMA code is sent
UL-Frame
DL-Frame
BW request Is sent
UL-Frame
DL-Frame
PDU is sent and BLT timer started
UL-Frame
DL-Frame
BLT timer expires and PDU is discarded
UL-Frame
The total delay before erroneous PDU is discarded
ARQ Block Lifetime

(a) downlink

(b) uplink

Fig. 2. ARQ Block Lifetime.

If ARQ is enabled for a real-time connection, then the ARQ_BLOCK_LIFETIME is the most important parameter to control the maximum latency. Once an ARQ block is transmitted, a sender keeps it in retransmission buffer for the time defined by this parameter. So, it sets implicitly the upper limit for the delay and should always be set based on the preferred delay. However, the sender cannot control the queueing delay because the timer is started only when the first transmission occurs. The queueing delay is in charge of the BS scheduler and the queue size.

Another important thing that increases the delay is the uplink contention. Based on Fig. 2(a), it can be stated that in downlink the ARQ block lifetime should be assigned a value of the desired maximum delay decreased by two frames (usually 10 ms). In uplink the setting of this value is however more tricky. Uplink contention process and sending and receiving of bandwidth requests add some extra delay at least occasionally and absolute definition for correct value of block lifetime is not that easy. In the best case, it takes a sender three frames before a PDU can be transmitted due to the CDMA-based contention resolution in 802.16 OFDMa PHY. Since CDMA codes can collide or the backoff start value can be less than the number of transmission opprtunities in a single frame, the number of frames to wait may increase significantly. The multicast polling can speed up the uplink contention resolution process [5]. Therefore the ARQ block lifetime should at least three frames less than desider maximum latency (usually 15 ms). In addition, fragmentation has a negative impact on the delay since the block lifetime can only be used to restrict the maximum delay of the first fragment.

B. Packet Loss

Limiting the packet drop percentages is not as easy task it is to the maximum delay. Naturally the block lifetime decreases the maximum delay in cost of packet drops and therefore as big as possible value for it should be given. So, if it assumed that a VoIP application has 150 ms end-to-end delay limit, which is recommended by the ITU-T G.114[9] then the end-to-end delay for IEEE 802.16 network should be less e.g. 100 ms. In this case the block lifetime of 80 ms might be the best value, since it can provide the 100 ms delay limit but still provide as little drops as possible.

Another parameters for decreasing the packet drops are the ARQ retry timeout and ARQ feedback intensity. By setting them as low as possible, the packet drops should be decreased. This will also decrease the delay. On the other hand, these parameters can also be used to either increase or decrease the overhead caused by the feedbacks. Therefore, it should be studied which values for these parameters provide the best tradeoff between the drops and the overhead.

C. ARQ Parameter Values

If the ARQ feedback transmission interval T_{feedback} is more than the ARQ retry timer T_{retry}, then the throughput performance starts to decline because a sender will retransmit the same data [6], [16]. If the feedback transmission interval is even more than the ARQ block lifetime T_{life}, then it may result in a very poor performance due to the discarded ARQ blocks. Based on that it is possible to propose the following inequality:

$$T_{\text{feedback}} < T_{\text{retry}} < T_{\text{life}}. \tag{1}$$

It is worth mentioning that the ARQ feedback intensity should not be very close to the ARQ retry timeout. The reason is that the ARQ feedback message can be dropped due to the failed checksum test, as any other PDU. ARQ feedback intensity for non-real-time applications was studied in [16].

IV. Simulation Results

In this section we present the simulation results for the VoIP performance over the IEEE 802.16 ARQ mechanism. The simulation platform is WINSE [12], which is an 802.16 extension for the NS-2 simulator.

The network environment comprises a single *sector* with a base station and a number of subscriber stations with BE and VoIP traffic, as Fig. 3 shows. Table I presents the network parameters that are used in all the simulation scenarios.[1]

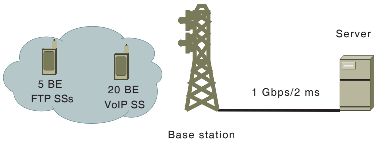


Fig. 3. Network structure.

TABLE I
802.16 NETWORK PARAMETERS.

Parameter	Value
PHY	OFDMa
Bandwidth	10 MHz
FFT	1024
Cyclic prefix length	1/8
TTG+RTG	464 PS
Duplexing mode	TDD
Frames per second	200 (5 ms per frame)
OFDM symbols	47
DL/UL symbols	32/15
DL/UL subcarrier alloc.	DL PUSC/UL PUSC
DL/UL slots	330/280
DL/UL channel measurements	preamble / data burst
Channel report type / interval	CQICH / 20ms
Channel measurements filter	EWMA, α = 0.25
Link adaptation	Fixed target FEC BLER
MAP MCS	Adaptive (QPSK1/2 Rep6 – QPSK1/2)
Compressed MAP	ON
Ranging transm. opport.	3
Ranging backoff start/end	1/15
Request transm. opport.	3
Request backoff start/end	2/15
CDMA codes	256
ranging+periodic ranging	64
bandwidth request	192
handover	–
Fragmentation/packing	ON
PDU size	120 B
CRC/ARQ	ON
ARQ feedback	standalone
ARQ feedback types	all
ARQ feedback interval	20 ms (FTP) 20–40 ms (VoIP)
ARQ block size	32 B (FTP) 16 B (VoIP)
ARQ window	1024
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ timers	
retry	50 ms (FTP) 20–80ms (VoIP)
block lifetime/Rx purge	1300 ms (FTP) 20–80ms (VoIP)

[1] These parameters conform the WiMAX Forum system profile [4].

TABLE II
SYSTEM LEVEL PARAMETERS.

Parameter	Value
Reuse factor	1/3
Inter-site distance	1.5 km
Path loss model	UMTS 30.30
Slow fading std.	8 dB
Fast fading	Ped B (60%), Veh A (40%)
Antenna technique	SISO (1x1)
Antenna pattern BS/MS	3GPP / Omnidirectional
Antenna height BS/MS	32 / 1.5 m
Tx power BS/MS	20 / 0.2 W

TABLE III
VOIP PARAMETERS.

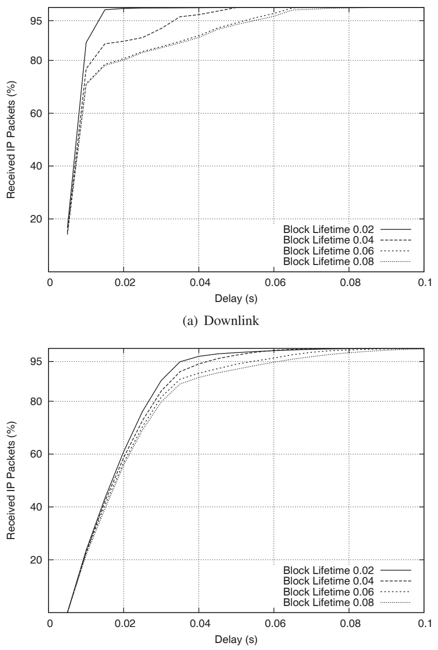
Parameter	Value
Codec	G.729
Aggregation interval	20 ms
Voice payload	0 B (inactive) / 20 B (active)
RTP overhead	12 B
UDP overhead	8 B
IPv4 overhead	20 B
VoIP packet without MAC headers	60 B
Active state VoIP bitrate	24 kbps
Generic MAC header	6 B
FSH	2 B
CRC	4 B
IEEE 802.16 MAC packet size	72 B
Average active time	38%
Average inactive time	62%

To speed up simulations, we do not model all the PHY details but rather rely upon the effective SINR trace files taken from the system level simulator, where we modeled 19 cells with 3 sectors per each cell. Relevant parameters are given in Table II. All the other PHY mechanisms, such as channel measurements, channel reporting, link adaptation, and scheduling are implemented in WINSE. The BS scheduler is a throughput-fair one. The link adaptation model ensures the target forward error correction block error rate of 10^{-2} for the ARQ-enabled connections [11].

The simulation scenario includes 20 BE VoIP connections and 5 BE FTP connections. The length of a single simulation run is 60 s and each simulation case is executed 20 times with different seed values. Each FTP SS has one downlink and one uplink file transfer active at the same time. Each VoIP SS has a bi-directional VoIP transmission with the server. VoIP parameters are given in Table III and they are taken from IEEE 802.16m Evaluation Methodology Document [3]. Active and inactive time distribution is based on function defined in ITU-T recommendation P.59 [10].

A. ARQ Block Lifetime

First, we analyse whether the ARQ block lifetime can ensure a upper limit for the VoIP delay. For these purposes, we run the same simulation scenario with different ARQ block lifetime values. The downlink VoIP E2E delay CDFs with different block lifetime values are presented in Fig. 4(a). It shows that the block lifetime does actually limit the maximum delay. The same results for uplink is shown in Fig. 4(b). However, it can be seen that the block lifetime does not restrict the maximum delay as expected. The reason for it is that the ARQ block lifetime is started only when the ARQ block is transmitted for the first time. Before that, a packet can spend some time in the output buffer waiting for being transmitted. This is especially the case for the uplink BE connection, where an SS has to take part in the uplink contention. Nevertheless, the maximum delay is less than 100 ms and the 95 percentile is less than 60 ms (for uplink) and 50 ms (for downlink).



(a) Downlink

(b) Uplink

Fig. 4. VoIP IP end-to-end delay CDF.

In addition to the VoIP delay distribution, Fig. 5 presents information on the number of dropped VoIP packets. As follows from the results, a small VoIP packet delay is achieved by a significant number of dropped packets due to the expired ARQ block lifetime. At the same time, ARQ block lifetime value of 80 ms provide more than satisfactory number of transmitted packets, where the number of dropped packets is less than 1%.

Based on the presented results, it is possible to arrive at the conclusion that ARQ block lifetime of 80 ms is a good tradeoff between the delay requirements and the number of dropped packets.

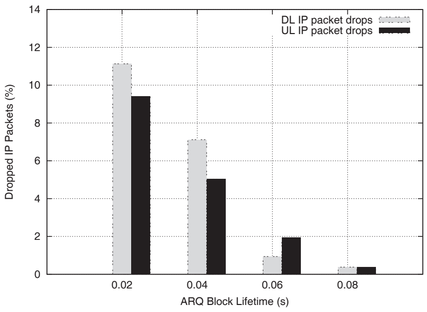
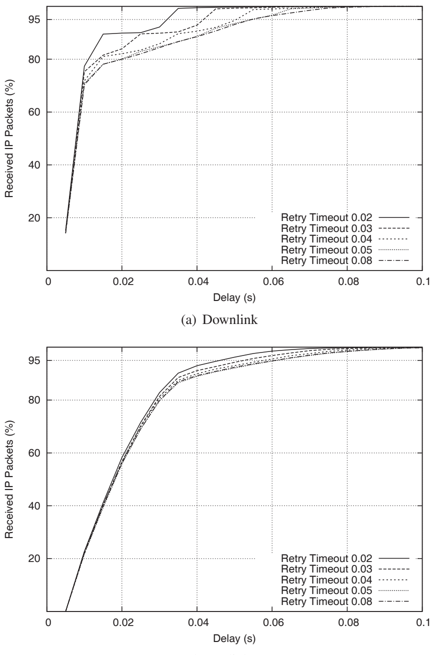


Fig. 5. Dropped VoIP packets with different ARQ Block Lifetime values.



(a) Downlink

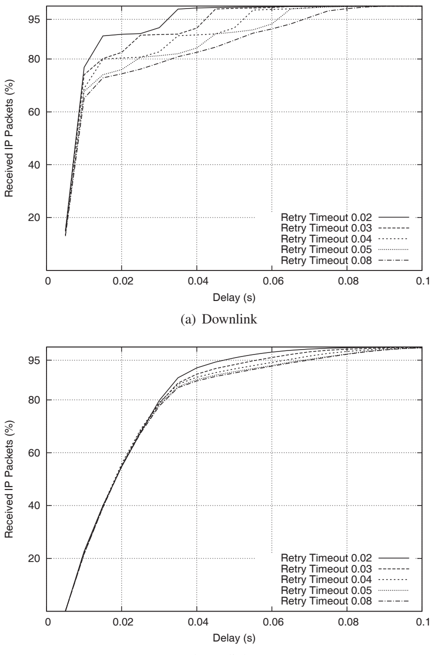
(b) Uplink

Fig. 6. VoIP IP end-to-end delay CDF: ARQ Feedback Intensity 20 ms.

B. ARQ Feedback Intensity and ARQ Retry Timeout

In this subsection we study suitable parameters for the ARQ feedback intensity and the ARQ retry timeout. Since the last simulation campaign resulted in a suggestion to use ARQ block lifetime value of 80 ms, it is chosen for these simulations. Other parameters are the same.

Fig. 6 shows the VoIP packet delay CDF for the ARQ feedback intensity of 20 ms and different ARQ retry timer values. As expected, the larger the ARQ retry timeout is, the larger delay becomes because a sender waits longer for the ARQ feedback to arrive before retransmitting packets. It is especially the case of the DL transmission where no contention occurs. It is interesting to note that increasing the ARQ feedback intensity to 40 ms does not change significantly results. The delays become larger but not much.

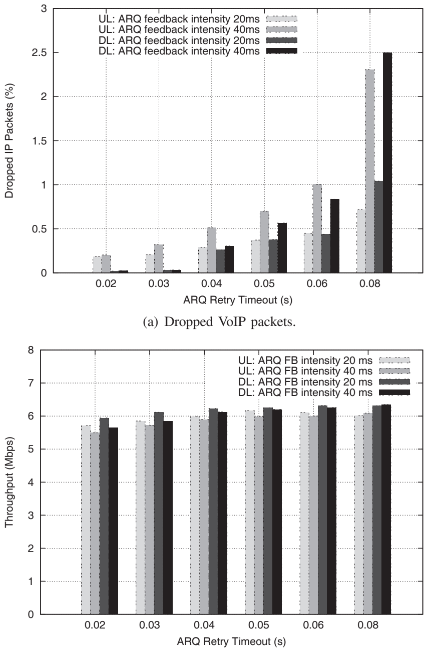


(a) Downlink

(b) Uplink

Fig. 7. VoIP IP end-to-end delay CDF: ARQ Feedback Intensity 40 ms.

Information on the packet drops gives another view on the ARQ feedback intensities and ARQ retry timeouts. Fig. 8(a) shows that as long as the retry timeout is at most 50ms, the drop probability is below 1%. Furthermore, ARQ feedback intensity of 20 ms gives best results from point of view of delays and number of dropped packets. Futher analysis of the ARQ retry timeout reveals that the background BE traffic throughput improves when VoIP retry timeout is set to 40 ms.



(a) Dropped VoIP packets.

(b) Background FTP throughput.

Fig. 8. 20 VoIP SSs: VoIP delay CDFs, drops and background FTP throughput.

As expected, smaller retry timeout values decrease the overall spectral efficiency since too many packets are retransmitted.

Based on the presented simulation results, it is possible to state that a good choice for the ARQ feedback intensity can be 20 ms, which is the VoIP datagram generation interval. The ARQ retry timeout can be of 40ms. The maximum delay is less than 100 ms and the 95-percentile delay is around 50 ms.

V. Conclusions

The main result of this study is that the ARQ mechanism and the BE packet treatment might be used with VoIP connections, even though it is not a typical configuration. However, it is a good possibility when a provider does not know what application a customer will use or when the first wave of network deployment assumes only the BE subscription.

We presented that the ARQ block lifetime can ensure the maximum delay at the expense of dropped packets. A carefully selected ARQ block lifetime of 60–80 ms gives a good tradeoff between the target delay of 50 ms and the number of dropped packets less than 1%. It meets both the IEEE 802.16m and IMT-Advanced requirements. The ARQ feedback intensity should be less than ARQ retry timeout to avoid unnecessary retransmissions due to the ARQ retry timeout expiry. Since a typical VoIP codec outputs data at a rate of one frame per 20 ms, it is quite logical to set the ARQ feedback intensity to 20 ms. Then, the ARQ retry timeout of 40 ms provides a good tradeoff between the network utilization and delays. Even if two consecutive ARQ feedbacks are lost, a VoIP sender still has time to retransmit VoIP packets. The obtained results can also be applied to other wireless broadband technologies, such as 3GPP LTE, where a similar ARQ mechanism functions at the RLC level.

Acknowledgments

We express our acknowledgements to V. Tykhomyrov and A. Puchko (University of Jyväskylä / Telecommunication laboratory) who helped with the WINSE simulator development and testing.

References

[1] Air interface for fixed broadband wireless access systems. IEEE Standard 802.16, Jun 2004.

[2] Air interface for fixed broadband wireless access systems - amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands. IEEE Standard 802.16e, Dec 2005.

[3] IEEE 802.16m evaluation methodology document (EMD). IEEE 802.16 Broadband Wireless Access Group, Mar 2008.

[4] WiMAX Forum Mobile System Profile, Release 1.0 Approved Specification, Apr 2008. Revision 1.6.1.

[5] O. Alanen. Multicast polling and efficient VoIP connections in IEEE 802.16 networks. In *The 10th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 289–294, Oct 2007.

[6] Richard Cam and Cyril Leung. Throughput analysis of some ARQ protocols in the presence of feedback errors. *IEEE Transactions on Communication*, 45(1):35–44, January 1997.

[7] Carla-Fabiana Chiasserini and Michela Meo. Impact of ARQ protocols on QoS in 3GPP systems. *IEE Transactions on Vehicular Technology*, 52(1):205–215, 2003.

[8] Guidelines for evaluation of radio interface technologies for IMT-Advanced. ITU-R M.2135, 2008.

[9] One-way transmission time. ITU-T recommendation G.114, 1996.

[10] Artificial conversational speech. ITU-T recommendation P.59, 1993.

[11] A. Puchko, V. Tykhomyrov, and H. Martikainen. Link adaptation thresholds for the IEEE 802.16 base station. In *Workshop on NS-2 simulator*, Oct 2008.

[12] A. Sayenko, O. Alanen, H. Martikainen, V. Tykhomyrov, O. Puchko, and T. Hämäläinen. WINSE: WiMAX NS-2 extension. In *2nd International Conference on Simulation Tools and Techniques*, Mar 2009.

[13] A. Sayenko, H. Martikainen, and A. Puchko. Performance comparison of HARQ and ARQ mechanisms in IEEE 802.16 networks. In *The 11th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 411–416, Oct 2008.

[14] Shamik Sengupta, Mainak Chatterjee, and Samrat Ganguly. Improving quality of VoIP streams over WiMAX. *IEEE Transactions on Computers*, 57:145–156, Feb 2008.

[15] V. Tykhomyrov, A. Sayenko, H. Martikainen, and O. Alanen. Analysis and performance evaluation of the IEEE 802.16 ARQ mechanism. *Journal of communications software and systems*, 4(1):29–40, Mar 2008.

[16] V. Tykhomyrov, A. Sayenko, H. Martikainen, O. Alanen, and T. Hämäläinen. On ARQ feedback intensity of the IEEE 802.16 ARQ mechanism. In *International Conference on Telecommunications*, Jun 2008.

PIV

ANALYSIS OF DUPLEXING MODES IN THE IEEE 802.16 WIRELESS SYSTEM

by

H. Martikainen 2010

European Wireless

Analysis of Duplexing Modes in the IEEE 802.16 Wireless System

Henrik Martikainen
Telecommunication Laboratory
University of Jyväskylä, Finland
henrik.martikainen@jyu.fi

***Abstract*—The IEEE 802.16 standard defines two main duplexing modes: Time Division Duplexing (TDD) and Frequency Division Duplexing (FDD). The FDD can be divided further into Full-FDD (F-FDD) and Half-FDD (H-FDD). F-FDD requires full duplexing support from subscriber stations and works with two frequency bands. TDD operates a single frequency band, but it does not need full-duplexing support. H-FDD is a combination of these: it works with two frequency bands and does not require full-duplexing support. The cost of this is a more complicated scheduling and added overhead. Still, H-FDD might be the only possible duplexing mode in some occasions. In this paper, these duplexing modes are compared and the H-FDD specific features, such as group balancing, are presented. The simulation results show that H-FDD does not quite match the performance of F-FDD or TDD. In the downlink direction, F-FDD and TDD have similar performance but in the uplink direction F-FDD can benefit from subchannelization gain with fewer bursts per frame.**

I. Introduction

IEEE 802.16 is a standard for the wireless broadband access network [1], [2] which can provide a high-speed wireless access to the Internet to home and business subscribers. It supports applications and services with diverse Quality-of-Service (QoS) requirements. The core components of a 802.16 system are a subscriber station (SS) and a base station (BS). The BS and one or more SSs can form a cell with a point-to-multipoint (PMP) structure. In this case, the BS controls the activity within a cell, the resource allocations to achieve QoS, and admission based on the network security mechanisms.

Similar to 3GPP LTE, the IEEE 802.16 specification defines two major duplexing modes: TDD and FDD. However, unlike the 3GPP LTE basic deployment scenario, WiMAX Forum chose the TDD frequency bands resulting in FDD been excluded from early versions of the system profile. The reason is that the TDD system is simpler in design thus requiring less expensive chipsets at the terminal side. However, lack of a proper FDD support meant that WiMAX could not be deployed to the FDD bands where downlink and uplink reside on different non-adjacent carriers. As a result, recent advances in both the IEEE 802.16 standard [4] as well as the system profile [5] added the FDD duplexing mode. To keep the terminal cost as low as for TDD, a special form of the FDD mode, referred to as H-FDD, was proposed. Thus, the 802.16 system can be considered in three duplexing modes: TDD, F-FDD, and H-FDD.

The goal of this paper is to analyze duplexing modes available in the 802.16 system and compare their performance. In addition, we study the H-FDD group assignment algorithm and group ratio problems from the IEEE 802.16 MAC and PHY perspective. Although this article mentions network planning, frequency planning and system design problems for each duplexing modes, these topics are not considered. The simulation tool used is WINSE (WiMAX NS-2 Extension)[11] which is an 802.16 extension on top of NS-2.

There is very little existing H-FDD study, especially in the 802.16 networks. There is existing study about TDD and F-FDD but they do not take OFDMa specific features like subchannelization gain into account. The authors in [14] propose a hybrid scheme where the users close to basestation would use TDD over OFDMa and users in the cell edge would use FDD over DL-OFDMa/UL-CDMA. The idea is interesting but it does not conform to the current IEEE 802.16 standard.

In [8] the authors introduce the strengths and weaknesses of F-FDD and TDD in general 4G OFDMa context. They show how cross-slot interference can be decreased with TDD by synchronizing the slots across all basestations in the network and adding sectors. Sectors are already present in IEEE 802.16 and the notion of having the same DL/UL ratio over the entire system is important.

In [6] authors introduce Half-Duplex Allocation (HDA) algorithm which does scheduling for half-duplex subscribers with bi-directional traffic with F-FDD frame structure. The authors assume IEEE 802.16 OFDM scheme and they prove that with simple assumptions this allocation of bursts can always be done. They compare VoIP capacity of HDA to the VoIP capacity of F-FDD and the H-FDD theme defined in IEEE 802.16. They find out that HDA can almost match F-FDD and is clearly better than H-FDD. Also web traffic was analyzed and with similar results. The HDA requires that time division multiple access (TDMA) can be utilized in the uplink. However this is not possible in uplink with the mobile IEEE 802.16e OFDMa scheme[2] so HDA is not feasible with mobile WiMAX devices. In the presented IEEE 802.16 H-FDD scheme the subscribers were placed randomly to the two H-FDD groups which is not the optimal solutions which we will show later in this paper.

The rest of the paper is organized as follows. Section II provides an overview of duplexing modes and analysis their advantages and drawbacks. Section III presents the simulation

978-1-4244-6001-4/10/$26.00 ©2010 IEEE

results. Finally, section IV concludes the paper and outlines further research directions.

II. Duplexing modes

A. TDD

Time Division Duplexing (TDD) frame structure is presented in Fig. 1. One of the main benefits of TDD is that the ratio between the downlink and uplink subframes can be adjusted flexibly thus adapting the system throughput to an operator demands. Of course, the whole operator network should have the same ratio. Otherwise, the downlink data from neighboring cells will cause severe interference to the uplink data transmission in the other cell. The radio implementation in TDD is simpler since there is no need to send and receive simultaneously. It is enough to have a single radio interface with a single encoding/decoding chain. This is particularly important for subscriber stations because it simplifies the hardware design and therefore makes the devices cheaper [8].

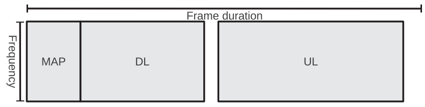


Fig. 1. TDD frame structure.

One of the drawbacks of TDD is a relatively small uplink subchannelization gain, especially if the downlink sub-frame is considerably larger than the uplink one. It may become a limiting factor for many services and makes operation at the cell edge much more challenging.

B. F-FDD

Fig. 2 presents the frame structure for Full Duplex Frequency Division Duplexing (F-FDD). The uplink and downlink subframes reside on different frequencies, where the frequency bands do not need to be adjacent. This is also a benefit of FDD modes and it means that an operator can use two narrow frequency bands where TDD could use only one of them. The downside of F-FDD is that subscriber stations must be able to send and receive at the same time. This makes the radio implementation more expensive. Also the antenna design might become more complicated if the downlink and uplink carriers reside at considerably different frequencies. Because the frequency bands given to the operator are fixed, it is impossible to change the ratio between the downlink and uplink subframes. This is not a desired feature if the network is used as a last-mile data connection where the traffic nature is very asymmetric. However, even though a typical FDD frequency allocation assumes two bands of the same size, the WiMAX Forum system profile allows for having bands of different size, e.g., 10 MHz for downlink and 5 MHz for uplink. Nevertheless, it is less flexible than in the TDD system.

One of the tempting features of FDD in OFDMA is subchannelization gain. If a subscriber uses fewer subchannels, it can use more power per a subchannel and thus increase the received signal-to-noise ratio (SNR). A typical F-FDD system with 5+5 MHz frequency bands has twice the number of slots per subchannel when compared to a typical TDD system with 10 MHz frequency band. This means that subscribers in the F-FDD system use less uplink subchannels for the same amount of data as in TDD and thus might get better uplink throughput.

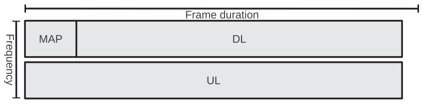


Fig. 2. F-FDD frame structure.

C. H-FDD

The introduction of the H-FDD duplexing mode aimed at solving a few problems of F-FDD. To transmit and receive simultaneously, a terminal must have two antennas with two related radio processing chains. Theoretically, a terminal with one radio chain might have announced itself as a one that can either receive or transmit, but then the scheduling becomes much more complicated [6]. The BS must ensure that downlink and uplink bursts are not scheduled for the same terminal at the same moment of time. The H-FDD divides each subframe into two groups in such a way that downlink and uplink transmissions occur at different moment of times. The group 1 downlink subframe and group 2 uplink subframe start the frame. A subscriber station belongs logically to one of the aforementioned groups.

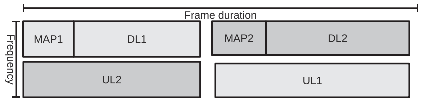


Fig. 3. H-FDD frame structure.

There are a number of drawbacks in the H-FDD duplexing mode. Firstly, due to the MAP messages transmitted in both groups, the downlink overhead doubles. Secondly, there are more gaps on the uplink carrier to allow a terminal to switch from a transmitting to a receiving mode. From the network point of view, H-FDD creates a need to implement and run a few additional resource management algorithms that are not present in other duplexing modes.

D. H-FDD Group Ratio

One of the key questions in the H-FDD system is how to partition the H-FDD frame into two groups. The group ratio plays a crucial role in the system performance as it allows for achieving a subchannelization gain for stations that reside at the cell edge.

A group ratio of 1:1 is the simplest solution for partitioning the H-FDD frame. However, when compared to the TDD mode, it does not provide a significant gain since each group has approximately 21 symbols. It is very close to the typical TDD system configuration where symmetrical services are supported. Of course, if the TDD system has a much larger DL frame, then H-FDD starts to provide an additional gain. Nevertheless, the gain is not as high as for the F-FDD system where the UL subframe spans all the symbols. It is also worth mentioning that the group ratio of 1:1 simplifies significantly the scheduling process because all two groups are similar from the resource allocation point of view.

An interesting approach is to have unequal group sizes in the H-FDD system where the first group downlink sub-frame is always of the smaller size. The first group is the default group that new subscribers use. Having the uplink subframe bigger in the first group means that stations joining the network at the cell edge can benefit from subchannelization gain as much as possible. At the same time, a group with a small UL subframe might be allocated stations with a very good uplink performance.

The biggest problem with an unequal group ratio is that it imposes significant constraints on the BS scheduler. As an example, if too many stations with a poor uplink performance are assigned to group 2 that has a large UL subframe, then the DL subframe may run out of resources. Similarly, if a station with high uplink and downlink bandwidth requirements is assigned to group 1, then the BS scheduler may fail to ensure its uplink bandwidth needs. Due to a small UL subframe, an UL allocation will span too many subchannels thus causing problems at the cell edge.

We treat the H-FDD group ratio as a network design parameter that an operator sets based on the environment and supported services. In the following sub-section we assume that it is fixed and does not change in the course of time.

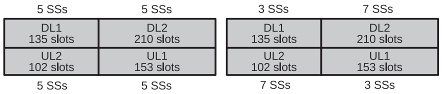
E. H-FDD Group Balancing

The 802.16 specification[1], [2] or system profile[5] do not say anything about how or how often the H-FDD group balancing should be done. We have implemented two H-FDD group balancing algorithms which, hence, will be referred to as basic and adaptive fair.

In the basic algorithm the subscriber ratio between groups is kept as close to 1:1 as possible. The subscribers' SNR or type of traffic etc. do not have any impact on the decision; neither the H-FDD group ratio is taken into account. An uneven group ratio means that the users in the group with a bigger UL subframe will get more UL resources and vice versa. Since the subscribers are selected randomly it probably means that these users also get more bandwidth. An example of group balancing for 10 subscribers with the basic algorithm can be seen in Fig. 4(a). It is already evident that the basic algorithm cannot provide a good fairness across *all* the stations. The only achievable thing is a good fairness *within* a particular group.

The objective of the adaptive fair group balancing algorithm is to provide a good fairness between all the subscribers in two groups. Adaptive fair consists of two main stages. In the first stage, the traffic direction in the cell is detected. If most of the users have mostly downlink traffic, then downlink fair balancing will be used, and if the traffic is mostly in the uplink direction, then uplink fair balancing is utilized. In the second stage users are sorted based on the SNR of detected direction. Then, the users with the worst SNR are placed to H-FDD group 1. The premise idea if this solution is that users with bad SNR can benefit from the uplink subchannelization gain since group 1 has a larger uplink subframe size. The balancing between groups is based either on the uplink or downlink fairness criterion so that the bandwidth for every subscriber in the cell should be roughly same. An example of group balancing done with the adaptive fair algorithm in the downlink case is shown in Fig. 4(b).

In both cases the groups are balanced periodically. H-FDD group switch information is passed to the subscriber in the DL-MAP. When a subscriber changes the group it loses one uplink subframe because information on UL-MAP message always refers to the next frame. Thus, rapid switching impacts negatively uplink performance and, in particular, the HARQ functioning.



(a) Basic (b) Adaptive Fair - DL optimized

Fig. 4. Example of H-FDD group balancing

III. Simulations

A. Simulation environment & setup

Fig. 5 shows a network that we use in a simulation scenario. It comprises a single BS controlling its sector, a FTP server with which data is exchanged, and 32 subscriber stations. In the uplink scenarios only uplink traffic is present; similarly, there is only downlink traffic in scenarios where the downlink performance is analyzed. The traffic model is the TCP full buffer FTP transmission over the 802.16 BE connection. Thus, regardless of the scenario, there is also traffic in the opposite direction caused by the TCP acknowledgements. Each simulation is ran 12 times and the application level data is measured at a wired link between the FTP server and the base station.

TABLE I
System level parameters.

Parameter	Value
Reuse factor	1/3
Path loss model	802.16m urban macro cell
Fast fading	Jakes model, K=0
Interference level DL/UL	-157 / -159 dBm/Hz
Antenna technique	SISO (1x1)
Antenna pattern BS/SS	3GPP / Omnidirectional
Antenna gain BS/SS	17 / 0 dBi
Antenna height BS/SS	32 / 1.5 m
Tx power BS/SS	5W(TDD), 2.5W(FDD) / 0.2 W

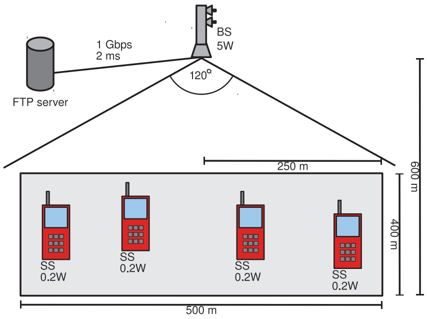


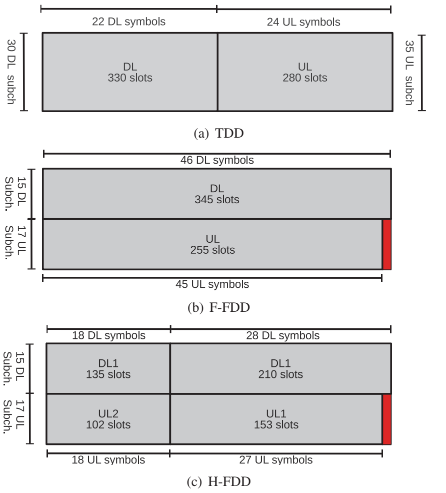
Fig. 5. General Network Structure.

Even though we concentrate on simulations of a single sector, we assume the presence of other cells that generate interference. Table I provides information on the relevant parameters. We use the 802.16 UMa propagation [3] with the Jakes fast fading model. For the sake of simulation simplicity, we use a constant interference level. The BS and SS use different antenna patterns, antenna heights. The BS preserves the same transmission power density regardless of the duplexing mode, i.e., 5W for the 10 MHz TDD frame and 2.5W for the 5 MHz FDD mode DL sub-frame. The SS *maximum* transmission power is 0.2W, whereas the instantaneous transmission power is governed by the BS power control algorithm.

TABLE II
TDD/F-FDD/H-FDD PHY PARAMETERS.

Parameter	TDD	F-FDD	H-FDD
Center frequency	2.5 GHz		
PHY	OFDMa		
Cyclic prefix length	1/8		
Frames per second	200 (5 ms/frame)		
Long preamble	1 symbol		
Bandwidth	10 MHz	5+5 MHz	
FFT	1024	512	
TTG+RTG	296+168 PS	0+168 PS	
DL/UL subchannels	30/35	15/17	
DL/UL subcarrier alloc.	DL PUSC / UL PUSC		
OFDM symbols	47		
DL/UL symbols	22 / 24	46 / 45	18+28 / 27+18
DL slots	330	345	135+210
UL slots	280	255	102+153
Ranging backoff start/end	1/15		0/15
Ranging transm. opport.	2		1
Request backoff start/end	3/15		2/15
Request transm. opport.	8		4

The 802.16 network parameters are given in Table II (duplexing mode specific PHY parameters) and Table III (common MAC level parameters). In FDD, there are two 5 MHz bands that impact a choice for the number of FFT points, which ensures the same sub-carrier spacing also for TDD that spans one 10 MHz band. It explains the number of subchannels we have in DL and UL directions in different duplexing modes. The TDD DL/UL ratio is chosen in such a way that the number of slots is comparably the same as in the FDD modes. Fig. 6(a) shows number of slots, symbols and subhannels for TDD. For H-FDD, an unequal group ratio is utilized to benefit from the UL subchannelization gain. Fig. 6(b) clarifies the number of symbols, subchannels, and slots used in F-FDD mode and Fig. 6(c) in H-FDD mode. It must be noticed that there are unused symbols in the uplink sub-frame in FDD modes due to the UL PUSC structure that mandates to use 3 symbols per one uplink slot.[1] It is also worth mentioning the initial ranging and bandwidth request parameters. Since there are two independent groups in H-FDD, the number of transmission opportunities per each H-FDD group is reduced two times and the backoff parameters are adjusted accordingly.



(a) TDD

(b) F-FDD

(c) H-FDD

Fig. 6. Number of slots, symbols and subchannels used in the simulations.

The BS runs the throughput-fair scheduling algorithm, a simple, yet efficient, solution that is capable of allocating slots based on the connection QoS requirements and bandwidth request sizes. It is based conceptually on the deficit round-robin; details of the algorithm are presented in [10]. The reason we did not choose the proportional-fair scheduler [7], which may improve the overall spectral efficiency, is the fact that it tends to decrease fairness, thus making an overall analysis more challenging. In the H-FDD mode, the BS scheduler runs two independent scheduling entities that are responsible for resource allocation in both H-FDD groups. Furthermore, the

[1] The real 802.16 system may benefit from those leftovers and use them for other purposes, such as UL sounding. Another option is to allocate there an UL ranging/request channel that should not be aligned on the 3 OFDM symbol boundary.

TABLE III
COMMON MAC PARAMETERS.

Parameter	**Value**
DL/UL channel measurements	preamble / data burst
Channel report type / interval	CQICH / 20ms
Channel measurements filter	EWMA, α = 0.25
UL Power Control	Closed loop
Link adaptation model	target FEC BLER, 10^{-1}
H-FDD group balancing algorithm	Basic / Adaptive fair
H-FDD group balancing interval	500 ms
MAP MCS	QPSK1_2
Compressed MAP	ON
sub-MAPs	OFF
CDMA codes	256
ranging+periodic ranging	64
bandwidth request	192
handover	–
Fragmentation/packing	ON
PDU size	140 B
CRC	ON
ARQ feedback	standalone
ARQ feedback types	all
ARQ feedback interval	20 ms
ARQ block size / window	16 B
ARQ window	1024
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ timers	
retry	100ms
block lifetime	500ms
Rx purge	500ms

BS scheduler takes as two independent parameters a preferred number of bursts that should be allocated in the DL and UL directions. It allows for studying the performance of duplexing modes under different scheduling configurations.

In addition to the scheduling algorithm, it is crucial to mention basics of the UL power control module because the specification does not define an exact algorithm. We implemented a simple closed-loop power control that works in coordination with the BS scheduler. Firstly, every time the BS scheduler makes an UL allocation, it ensures that an SS power budget is not exceeded. If the BS scheduler allocates small UL data grants, then the UL power control increases gradually SS transmission power to benefit from the subchannelization gain. As opposed to that, if the BS scheduler tends to allocate larger UL data grants, the UL power control instructs an SS to decrease the transmission power per a sub-carrier so that an SS can transmit in more subchannels. All the UL power control commands are carried in the DL broadcast FPC management message. All the SSs also report periodically their UL transmission power values via signaling headers.

The MAC level retransmission mechanism is ARQ, parameters of which are tuned based on our previous research on the ARQ mechanism in the 802.16 networks [12]. The ARQ mechanism also governs the target FEC block error rate of 10^{-1} that is used in the link adaptation module [9]. Even though our simulator supports so-called sub-MAPs that can improve dramatically the performance [13], we decided to turn them off as they also may impact fairness.

While running simulations, we consider a few cases with a different number of preferred bursts per a sub-frame. As is presented later, it helps to explain a difference between TDD and FDD performance and uplink subchannelization gain.

B. TDD & F-FDD

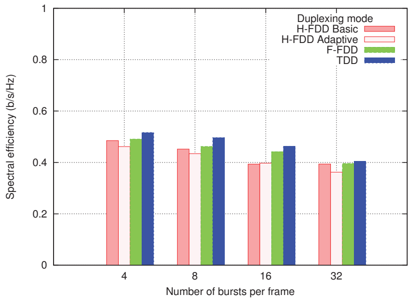


Fig. 7. Downlink spectral efficiency for different duplexing modes. TDD bars have been scaled up to match the slot count of F-FDD and H-FDD.

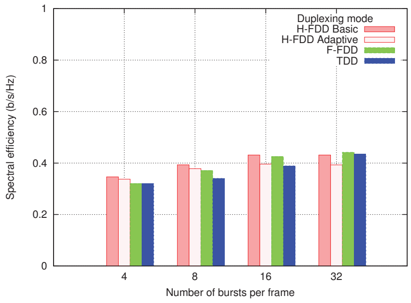
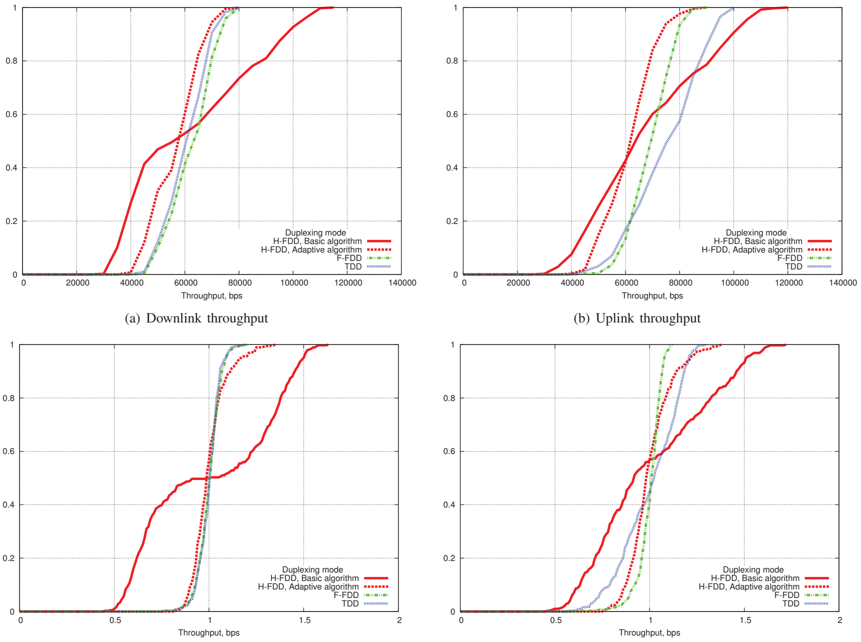


Fig. 8. Uplink spectral efficiency for different duplexing modes. TDD bars have been scaled down to match the slot count of F-FDD and H-FDD.

First we analyze the difference between TDD and F-FDD. Fig. 7 shows the downlink spectral efficiency for F-FDD and TDD. For both duplexing modes the spectral efficiency gets worse when there are more bursts per a subframe. This is because more bursts means that there are more entries in DL-MAP which creates additional overhead. Also with TCP traffic it means that more TCP acknowledgements are sent in the uplink direction, which also increases the UL-MAP size residing in the DL sub-frame. So, regardless of the duplexing mode, the BS scheduler can benefit from the full-buffer traffic by allocating only a few downlink bursts. Downlink fairness for 32 bursts can be seen in Fig. 9(c) and for 4 bursts in Fig. 10(c). It can be seen that there is no fairness difference between TDD and F-FDD. Furthermore, the BS scheduler can achieve a good fairness in the downlink direction even for a few data bursts.

Fig. 8 shows the uplink spectral efficiency for F-FDD and TDD. It can be seen that with both duplexing modes the



(a) Downlink throughput

(b) Uplink throughput

(c) Downlink fairness

(d) Uplink fairness

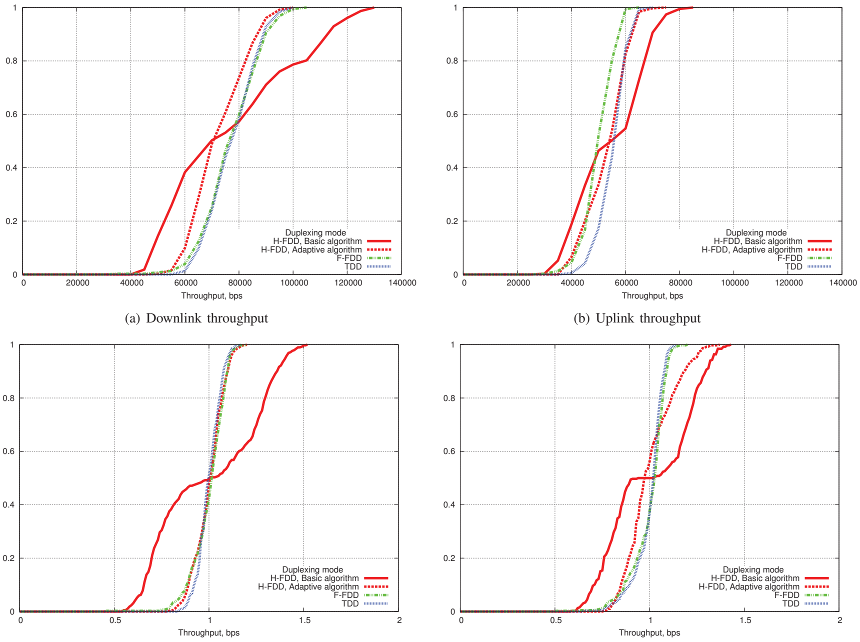
Fig. 9. Duplexing mode comparison throughput and fairness CDFs - 32 bursts per frame.

performance improves when the number of the bursts per frame increase. This is due to increased subchannelization gain. F-FDD gets the biggest gain already with 16 bursts per frame because there are 17 subchannels in the uplink subframe in F-FDD. It means that there is roughly one subchannel per subscriber where it can concentrate all transmission power. With TDD there are 35 subchannels in the uplink direction, which means that TDD still benefits significantly when the number of bursts is increased from 16 to 32. With 32 bursts per frame the performance difference between TDD and F-FDD is not big. Also it has to be remembered that in the uplink direction more bursts add to the MAP message size but MAP message is carried in the downlink subframe. The MAP message sizes are not measured here but it is clear that MAP messages are smaller with 16 uplink bursts than with 32 uplink bursts. F-FDD can therefore benefit from full subchannelization gain with less MAP message overhead than TDD. Uplink fairness for 32 bursts can be seen in Fig. 9(d) and for 4 bursts in Fig. 10(d). With 4 bursts per subframe there is no difference between TDD and F-FDD. With 32 bursts per subframe F-FDD is providing better fairness than TDD.

C. H-FDD Group Balancing Algorithms

In this subsection we analyze the difference between two proposed H-FDD group balancing algorithms. Fig. 7 shows the downlink spectral efficiency for basic and adaptive fair H-FDD group balancing algorithms. In all the cases the basic algorithm provides better spectral efficiency than adaptive fair. This can be explained by the bad fairness which is seen in Fig. 9(c) and Fig. 10(c). The basic balancing assigns randomly 16 subscribers to the first group and 16 to the second one. The downlink subframe of the first group is smaller than the downlink subframe of second group which means that there are fewer slots per subscriber in the first group. This phenomenon can also be seen from throughput CDFs in Fig. 9(a) and Fig. 10(a).

In the uplink direction the difference between basic and adaptive fair is similar to that in downlink. Spectral efficiency for basic is better (Fig. 8) but fairness (Fig. 9(d) and Fig. 10(d)) and throughput distribution (Fig. 9(b) and Fig. 10(b)) are worse. The uplink contention mechanism is equalizing the difference between the two groups with basic balancing so



(a) Downlink throughput

(b) Uplink throughput

(c) Downlink fairness

(d) Uplink fairness

Fig. 10. Duplexing mode comparison throughput and fairness CDFs - 4 bursts per frame.

it is not as clear as in the downlink case.

D. H-FDD Adaptive Fair, TDD and F-FDD

Because H-FDD with adaptive fair group balancing can provide good fairness, we compare it to the other duplexing modes. In the downlink direction H-FDD with adaptive fair balancing does not reach the spectral efficiency of F-FDD or TDD, which is seen in Fig. 7. This is because H-FDD has more overhead from MAP message headers. Still, Fig. 9(c) shows how fairness with H-FDD adaptive fair is almost as good as with F-FDD and TDD.

In Fig. 8 it can be seen that in the uplink direction with 4 and 8 bursts per frame the H-FDD has a better spectral efficiency than F-FDD or TDD which is explained by worse fairness. With 16 bursts F-FDD gets the full benefit from subchannelization gain and has the best efficiency while H-FDD and TDD perform equally. With 32 bursts TDD benefits from subchannelization gain and outperforms H-FDD.

IV. Conclusions

In this paper, we have analyzed the different duplexing modes in the IEEE 802.16 wireless system. In addition, we have considered the H-FDD specific features, such as group ratio and group balancing, and analyzed two ways to balance users between the H-FDD groups.

The simulation results show that, as expected, there is no big difference in performance between the TDD and FDD modes in the downlink direction. In the uplink direction, F-FDD gets the full benefit from subchannelization gain already with a fewer number of bursts per a frame. In general, longer uplink sub-frame makes cell edge performance better under F-FDD. H-FDD has the worst performance, mostly due to two groups that introduce a number of small transmission gaps and increase the DL signaling overhead. While comparing proposed H-FDD group balancing methods, one can notice that the basic H-FDD balancing algorithm always achieves better spectral efficiency than the adaptive fair balancing, which is explained by bad fairness. The adaptive fair balancing can achieve as good fairness as TDD and F-FDD except the uplink direction with 4-8 bursts per frame. At the same time, the spectral efficiency is 0-10% behind of TDD and F-FDD when fairness is similar.

There is no simple choice between TDD and F-FDD since they both have their benefits. TDD has an adjustable DL/UL ratio and F-FDD can utilize better the uplink subchannelization gain. However, it is usually the case that an operator does not select the duplexing mode freely, but rather adapts to available frequency bands. Then, if the F-FDD subscriber stations are not available or they are too expensive, H-FDD must be used. It is worth mentioning that the scheduling for H-FDD is more complicated which creates a burden for the network side. Furthermore, bidirectional traffic mixture will create additional fairness problems. Still, we have shown that in simple cases fairness can be guaranteed with the proposed adaptive fair balancing where H-FDD has only slightly worse spectral efficiency than TDD or F-FDD.

In the future we plan to study how the H-FDD group ratio can be adjusted jointly with the H-FDD group balancing. Also, delay sensitive traffic will be analyzed by studying VoIP capacity with different duplexing modes.

V. Acknowledgements

I express my acknowledgments to Alexander Sayenko (Nokia Siemens Networks) for his feedback and contribution to this H-FDD study. Special thanks to Andrea Bacioccola and Zexian Li (Nokia) for providing an insight on technical aspects of the duplexing modes and comments on the simulation results. I also would like to thank Vitaliy Tykhomyrov, Oleksandr Puchko, Vesa Hytönen (Jyväskylä University) and Olli Alanen (Magister Solutions) for their contribution to the implementation of the 802.16 module for the NS-2 simulator.

References

[1] Air interface for fixed broadband wireless access systems. IEEE Standard 802.16, Jun 2004.

[2] Air interface for fixed broadband wireless access systems - amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands. IEEE Standard 802.16e, Dec 2005.

[3] IEEE 802.16m evaluation methodology document (EMD). IEEE 802.16 Broadband Wireless Access Group, Mar 2008.

[4] Air interface for broadband wireless access systems. IEEE Standard 802.16 (Rev2), May 2009.

[5] WiMAX Forum Mobile System Profile Specification: Release 1.5 Approved Specification, Aug 2009.

[6] A. Bacioccola, C. Cicconetti, A. Erta, L. Lenzini, and E. Mingozzi. Bandwidth allocation with half-duplex stations in IEEE 802.16 wireless networks. *Mobile Computing, IEEE Transactions on*, 6(12):1384–1397, Dec 2007.

[7] J. Lakkakorpi, A. Sayenko, and J. Moilanen. Comparison of different scheduling algorithms for WiMAX base station: deficit round robin vs. proportional fair vs. weighted round robin. In *IEEE WCNC*, pages 1991–1996, Mar/Apr 2008.

[8] Ray R. Wang Edward K. S. Au Vincent K. N. Lau Roger S. Cheng Wai Ho Mow Ross D. Murch Peter W. C. Chan, Ernest S. Lo and Khaled Ben Letaief. The evolution path of 4G networks: FDD or TDD? *IEEE Communications Magazine*, 44(12):42–50, Dec 2006.

[9] A. Puchko, V. Tykhomyrov, and H. Martikainen. Link adaptation thresholds for the IEEE 802.16 base station. In *Workshop on NS-2 simulator*, Oct 2008.

[10] A. Sayenko, O. Alanen, and T. Hämäläinen. Scheduling solution for the IEEE 802.16 base station. *Computer Networks*, 52:96–115, 2008.

[11] A. Sayenko, O. Alanen, H. Martikainen, V. Tykhomyrov, O. Puchko, and T. Hämäläinen. WINSE: WiMAX NS-2 Extension. In *2nd International Conference on Simulations Tools and Techniques*, Mar 2009.

[12] V. Tykhomyrov, A. Sayenko, H. Martikainen, and O. Alanen. Analysis and performance evaluation of the IEEE 802.16 ARQ mechanism. *Journal of communications software and systems*, 4(1):29–40, Mar 2008.

[13] V. Tykhomyrov, A. Sayenko, O. Puchko, and T. Hämäläinen. Decreasing the MAP overhead in the IEEE 802.16 OFDMA system. In *European Wireless Conference*, 2010. accepted for publication.

[14] Sangboh Yun, Seung Young Park, Yeonwoo Lee, Daeyoung Park, Yungsoo Kim, Kiho Kim, and Chung Gu Kang. Hybrid division duplex system for next-generation cellular services. *Vehicular Technology, IEEE Transactions on*, 56(5):3040–3059, Sep 2007.

PV

PERFORMANCE COMPARISON OF HARQ AND ARQ MECHANISMS IN IEEE 802.16 NETWORKS

by

A. Sayenko, H. Martikainen and O. Puchko 2008

The 11-th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems

Performance Comparison of HARQ and ARQ Mechanisms in IEEE 802.16 Networks

Alexander Sayenko
Nokia Research Center
Nokia, Helsinki
alexander.sayenko@nokia.com

Henrik Martikainen, Alexander Puchko
Telecommunication laboratory, MIT department
University of Jyvaskyla, Finland
{henrik.martikainen,olpuehko}@jyu.fi

ABSTRACT

The IEEE 802.16 technology defines several link level mechanisms to retransmit erroneous data. In this paper we compare the performance of the ARQ and HARQ mechanisms in the IEEE 802.16 networks. Our simulations results show that in general HARQ provides a better performance. However, ARQ can compete successfully with it due to a smaller signaling overhead. Furthermore, since ARQ does not require a dedicated uplink signaling channel for the acknowledgments messages, it results in better resources utilization in the uplink direction.

Keywords

IEEE 802.16 WiMAX, ARQ, HARQ, NS-2

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Performance

1. INTRODUCTION

IEEE 802.16, called WiMAX, is a standard for the wireless broadband access network [1] that can provide a high-speed wireless access to home and business subscribers. It supports applications and services with diverse Quality-of-Service (QoS) requirements. The core components of a 802.16 system are a subscriber station (SS) and a base station (BS). The BS and one or more SSs can form a cell with a point-to-multipoint (PMP) structure. In this case, the BS controls the activity within a cell, resource allocations to achieve QoS and admission based on the network security mechanisms.

A major concern in any wireless communication system is the control of transmission errors caused by channel noise so that error free data can be delivered to the user. Modern broadband wireless systems provide a number of mechanisms to achieve the error-free data transmission. In particular, IEEE 802.16 provides two mechanisms: automatic repeat query (ARQ) and hybrid automatic repeat query (HARQ). Both two mechanisms are available in the OFDMa PHY, which serves as a basis for the mobile 802.16 networks [2]. Both ARQ and HARQ rely on an integrity check to detect channel errors, and uses a retransmission process to retransmit lost (i.e., missing or corrupted) data. However, unlike ARQ that works as a part of the upper MAC layer, HARQ requires a more complicated signaling to report ACKs and request retransmissions.

In this paper we provide an accurate comparison of ARQ and HARQ mechanisms in IEEE 802.16 networks by means of extensive network simulations. The comparison of ARQ and HARQ mechanisms has not been studied thoroughly yet. In most cases, authors either compare different HARQ flavors or compare HARQ against non-ARQ data transmission by means of link level simulations. While the link level simulations provide an accurate link performance estimation, usually they are not capable of accounting for the system level overhead or even higher level transport protocol behavior. We combine the link level simulation results with the 802.16 network simulations where we have a complete MAC implementation with upper layers. Since our module is based on the NS-2 simulator, there is also an accurate modeling of transport protocols. To compare ARQ and HARQ, we conduct the TCP data transmission because it is usually the case that the real-time UDP based services, such as VoIP, either rely upon HARQ or do not use any retransmission mechanism at all asking the link adaptation model to choose a more robust modulation and coding scheme (MCS). Another reason for using TCP is that it is much more sensitive to packet drops.

The rest of this paper is organized as follows. Section II considers theoretically the HARQ and ARQ mechanisms and delves into the details of their functioning. Section III presents the simulation results. Section IV discusses about the results and outlines performance enhancements. Finally, section V concludes the paper.

2. ANALYSIS OF ARQ AND HARQ MECHANISMS

In subsequent subsections, we analyze theoretically ARQ and HARQ mechanisms. We consider the following comparison parameters:

MSWiM'08, October 27–31, 2008, Vancouver, BC, Canada.
Copyright 2008 ACM 978-1-60558-235-1/08/10 ...$5.00.

1. MAP messages (DL-MAP and UL-MAP), downlink and uplink signaling overhead
2. MAC PDU overhead
3. PDU error rate
4. Scheduling constraints

For the sake of brevity we omit the detailed description of ARQ and HARQ mechanisms. A good general overview of ARQ and HARQ is given in [6]. Technical description of ARQ with simulation results is presented in [13]. A detailed description of HARQ mechanism with simulation results can be found in [7, 5].

2.1 ARQ

The ARQ mechanism does not introduce any additional MAP signalling overhead when compared to the non-ARQ-enabled functioning. ARQ enabled data and the ARQ feedbacks are transmitted at the MAC level as normal PDUs. Thus, the MAP messages encode normal data bursts for the ARQ-enabled connections. However, the MAC PDU overhead is larger for the ARQ enabled connection. First, the CRC-32 field must be added to each PDU. Secondly, the size of the fragmentation and packing subheaders are larger since they carry the ARQ block sequence number (BSN). Finally, even if a PDU carries a non-fragmented SDU, the fragmentation subheader (FSH) is still mandatory because it encodes the ARQ BSN. It is also worth noting that the typical ARQ connection should have a limited PDU size, otherwise a connection may experience a poor performance if large PDUs are constructed [9]. Fig. 1 presents a sample data burst with two MAC PDUs inside (for the sake of clarify, a data burst is presented as a one-dimensional allocation).

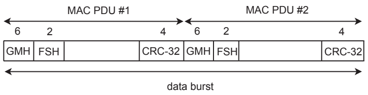


Figure 1: ARQ PDU.

ARQ mechanism does not introduce significant constraints for the BS scheduler. As mentioned above, the ARQ works on the top of the basic MAC functionality. Thus, once the BS allocates a data burst, the ARQ mechanism starts to utilize it in excatly the same way as the non-ARQ connection does. The only serious constraint the BS scheduler must account for is the ARQ block size. To avoid ARQ transmission stalls, the minimum data burst size should be larger than a single ARQ block size negotiated between the SS and BS [11].

The only serious ARQ limitation is the absence of the retransmission gain as in the case of HARQ. If there is a high FEC BLER either due to the decreased channel conditions or incorrectly working link adaptation model, then subsequent retransmissions will experience the same drop probability as the initial one. It is also important to note that ARQ feedback messages can be dropped as any other MAC PDU. Then, the resulting performance depends heavily on the ARQ feedback intensity and ARQ retransmission timers [14]. At the same time, it is important to note that the ARQ mechanism retransmits only *erroneous* PDUs, while HARQ retransmits the whole data burst. Furthermore, ARQ retransmissions can use a different MCS than the original transmission while HARQ is obliged to use the same MCS for all retransmissions.

2.2 HARQ

The MAP signaling overhead introduced by the HARQ mechanism depends a lot on how the scheduler allocates resources. It makes sense to mention that in 802.16 networks, HARQ data is allotted in a form of HARQ sub-bursts, where a sub-burst is a *one-dimensional* entity that occupies slots in the frequency-first order. Multiple HARQ sub-bursts can be placed into one burst that is a two-dimensional allocation. If we assume that all the HARQ sub-bursts are located in a single burst, then the MAP overhead is not large. Fig. 2 illustrates a data burst with several HARQ sub-bursts. Conversely, placing a HARQ sub-burst in an independent data bursts creates a large overhead because the MAP message has to encode a data burst and clarify the HARQ sub-burst configuration, e.g., the number of slots, MCS, HARQ mode etc.

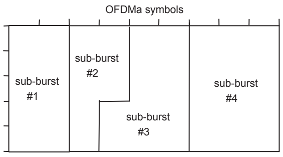


Figure 2: HARQ sub-bursts.

In addition to HARQ enabled data bursts, the MAP signalling overhead comprises information on downlink slots to transmit ACKs for the uplink bursts, and uplink slots for ACKs sent by SSs for the downlink bursts. Downlink ACKs do not create a significant overhead – there is just a bitmap, where each bit corresponds to a particular burst. The uplink HARQ ACK channel is more demanding. Each HARQ downlink sub-burst requires an uplink transmission opportunity to transmit the HARQ ACK message. Though a single UL ACK message requires only half of an uplink data slot, the resulting overhead may reduce noticeably the amount of available uplink resources. Besides, the HARQ UL ACK channel is a two-dimensional uplink allocation that poses additional constraints for the scheduler.

The MAC level PDU overhead is somewhat less or comparable to the ARQ mechanism. Firstly, the sender must reserve 2 bytes at the end of HARQ sub-burst to include the HARQ CRC-16 field. Though there is no need carry the per-PDU CRC-32 field, there is the PDU sequence number (SN) extended subheader that occupies 4 bytes.[1] It should be noted that the PDU SN is optional. However, it is anticipated that it is turned on for most services. Otherwise, PDUs can arrive in the wrong order to the upper MAC causing SDU reassembly problems. Furthermore, if SDUs are delivered in the wrong order to a receiver, it may result

[1] The extended subheader encoding and its type occupy 2 bytes, thus leaving 2 bytes for the sequence number field. Another option is to use a short sequence number that needs only 1 byte.

in a decreased performance at the application level. Fig. 3 illustrates a HARQ sub-burst with one MAC PDU. It is anticipated that there will be one MAC PDU per a HARQ sub-burst because the whole HARQ sub-burst is retransmitted if an error is detected. Though it is possible to have a number of PDUs in a single HARQ sub-burst, it results in a larger MAC overhead.

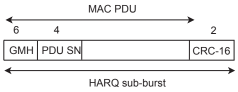


Figure 3: HARQ PDU.

The HARQ mechanism introduces significant constraints to the BS scheduler. While the initial HARQ transmission can be of any size, the subsequent HARQ retransmissions must be of exactly the same size. Taking into account two-dimensional data allocations in the OFDMa frame, it is easy to imagine the complexity of this problem. Furthermore, as mentioned above, there is also the HARQ UL ACK channel that must be placed in the uplink part.

As mentioned in the previous subsection, HARQ retransmits the whole data burst when an error is detected. Since large data bursts have a higher probability of being dropped, the BS scheduler should consider making smaller allocations. Such a requirement may conflict with certain scheduling policies, such as proportional fair, where the scheduler tends to allot slots when an SS has a good channel performance or where large allocations are made at large time intervals.

2.3 ARQ on top of HARQ

This case combines properties of both the ARQ and HARQ mechanisms. The reason an operator may resort to using such a configuration is the fact that HARQ has a limited number of retransmissions, whereas the ARQ mechanism can retry until the ARQ block life timer expires. It creates an appealing scenario for the TCP based services that suffer from packet drops. If HARQ detects errors and copes with retransmissions, then the ARQ levels sees a clear channel thus resulting in small cumulative ARQ feedback messages. The ARQ feedback overhead is very small in this case [13]. If HARQ fails, the ARQ mechanism can request a retransmission by means of negative ACK, which can be encoded efficiently by the cumulative+sequence feedback type [13].

To avoid double MAC PDU overhead caused by the HARQ and ARQ mechanisms, it is possible to disable the PDU SN. Indeed, to ensure a correct ordering of PDUs at receiver, it is enough to use ARQ BSN. If a certain PDU is missing as a result of failed HARQ transmission (or retransmission), the ARQ mechanism will buffer all the subsequent ARQ blocks and wait until a PDU with expected ARQ blocks is received. It should be noted that it is possible to keep both PDU SN and ARQ BSN. However, then the ARQ *block rearrangement* must be disabled so as not to damage the PDU SN numbering.

In any case, enabling both ARQ and HARQ mechanisms results in the largest MAC overhead: there are CRC fields in a HARQ sub-burst and in each MAC PDU, and the increased packing/fragmentation subheaders due to the ARQ BSN. Fig. 4 illustrates an example of the ARQ-enabled MAC PDU carried in the HARQ sub-burst. Similar to the HARQ case, there is one PDU that spans the whole sub-burst. At the same time, the PDU has the FSH and CRC-32 field.

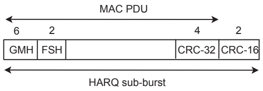


Figure 4: HARQ PDU.

From the scheduling point of view, the majority of the scheduling limitations come from the HARQ mechanism because ARQ works on top of it. At the same time, the ARQ block size mandates implicitly the minimum HARQ burst size that the scheduler should allocate.

3. SIMULATION RESULTS

To run simulations, we have implemented the 802.16 MAC and PHY levels in the NS-2 simulator. The PHY implementation includes both OFDM and OFDMa. The MAC implementation contains the main features of the IEEE 802.16 standard, such as downlink and uplink transmission, connections, MAC PDUs, packing and fragmentation, the contention and ranging periods, the MAC level management messages, and the ARQ mechanism. The ARQ implementation supports the ARQ blocks, the ARQ transmission window, retransmission with rearrangement, and all the ARQ feedback types. The ARQ implementation also includes the prioritization of the ARQ feedbacks and retransmissions. All the related algorithms are presented in [13]. The HARQ implementation supports Type I, i.e., chase combining (CC).

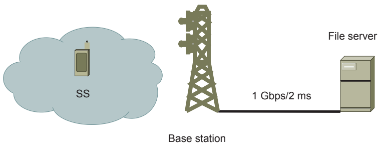


Figure 5: Network structure.

Fig. 5 shows the network structure we use in the simulation scenarios. The simulation environment includes one wired node that is connected with a high-speed link to the BS. Such a choice is motivated by a desire to diminish any impact the wired medium may have on the simulation results. Also, there are SSs each of which establishes one uplink and downlink BE connection to the BS. Each SS also establishes the basic management connection to exchange the management messages with the BS. An SS hosts exactly one FTP-like application that downloads data over the TCP protocol from a wired node. The reason we choose such an application type is that it tries to send as much data as possible thus utilizing all the network resources. At the same time, the TCP protocol is very sensible to the packet drops that can occur in the wireless part.

There is the BS controlling the 802.16 network, the parameters of which are presented in Table 1.[2] They are the

[2] These parameters conform the WiMAX Forum recommen-

Table 1: **802.16 network parameters.**

Parameter	Value
PHY	OFDMa
Bandwidth	10 MHz
FFT	1024
Cyclic prefix length	1/8
TTG+RTG	464 PS
Duplexing mode	TDD
Frames per second	200 (5 ms per frame)
OFDM symbols	47
DL/UL symbols	32/15
DL/UL subcarrier alloc.	DL PUSC/UL PUSC
DL/UL slots	480/175
DL/UL channel measurements	preamble / data burst
Channel report type / interval	CQICH / 20ms
Channel measurements filter	EWMA, $\alpha = 0.25$
MAP MCS	QPSK1/2
Compressed MAP	ON
Ranging transm. opport.	2
Ranging backoff start/end	1/15
Request transm. opport.	8
Request backoff start/end	3/15
CDMA codes	256
ranging+periodic ranging	64
bandwidth request	192
handover	–
HARQ	Type I (CC)
HARQ channels	16
HARQ buffer size	2048 B (per channel)
HARQ shared buffer	ON
HARQ max. retransmissions	4
HARQ ACK delay	1 frame
PDU SN	ON (HARQ case) OFF (HARQ & ARQ case)
PDU SN type	long (2 bytes)
Fragmentation/packing	ON
PDU size	140 B (ARQ case) unlimited (HARQ & ARQ case)
CRC/ARQ	ON
ARQ feedback	standalone (ARQ case) piggy-backed (HARQ & ARQ)
ARQ feedback types	all
ARQ feedback interval	20 ms (ARQ case) 60 ms (HARQ & ARQ case)
ARQ block size	32 B
ARQ window	1024
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ timers	
retry	100ms
block lifetime	500ms
Rx purge	500ms

same except a few cases. First, we limit the PDU size for a pure ARQ connection to avoid excess PDU drops. In the case of ARQ on top of HARQ, we set a larger ARQ feedback interval so that ARQ waits for HARQ to finish its retransmissions and disable the PDU SN to avoid the double MAC overhead. Depending on a simulation case, the ARQ feedback messages are of slightly different format. If there are pure ARQ connections, then the ARQ feedback messages are transmitted in standalone PDUs to decrease their drop probability. Otherwise, when ARQ works on top of HARQ, it is more efficient to piggy-back them to PDUs with user data. The HARQ parameters are in effect only when HARQ is turned on.

The BS scheduler reserves two transmission opportunities for the initial ranging purposes (as in real life, an SS has to join the network in our simulator) and eight transmission opportunities for the bandwidth request contention resolution. The backoff parameters are given in Table 1. The distribution of the CDMA codes is also given in the table (since we do not simulate any mobility, there are no CDMA handover codes).

The PHY model is based on trace files gathered from the link level multi-cell simulations with the reuse 1-3-3 factor. There are uplink and downlink trace files that are chosen randomly by stations and are read from a random starting index. Then, the MAC PDU error generation follows the IEEE 802.16m simulation methodology [4] and is based on the MAC PDU size and FEC BLER, whereas the latter is determined based on the current MCS and the SNR HARQ PDU error generation also accounts for the HARQ retransmission gain. To adapt to the varying channel, the BS runs the link adaptation model with the target FEC BLER for HARQ of $10^{-1.5}$ and for ARQ of 10^{-2} [10].

The downlink broadcast messages, such as DL-MAP and UL-MAP, use a robust QPSK1/2 MCS; they are never dropped in our simulations. The BS runs the scheduling algorithm, details of which are presented in [12]. In a few words, if there are only the BE connections, then the BS allocates resources fairly between the SSs based on their uplink bandwidth request and downlink queue sizes. In addition, we apply the ARQ aware scheduling to the BS scheduler. The BS HARQ scheduler always prioritizes HARQ retransmissions over normal data.

To obtain confidence intervals, we perform each simulation run 10 times with different initial random seed values. Each simulation run lasts for 20 seconds.

3.1 Downlink performance

In this sub-section we compare the downlink performance of ARQ and HARQ mechanisms. Since the downlink performance depends a lot on how many data bursts the BS scheduler allocates per a single frame, we conduct several simulation cases where we vary that parameter.

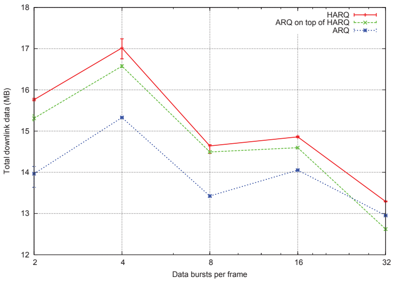


Figure 6: Amount of transferred data in the downlink direction.

Fig. 6 presents the total amount of data transferred in the downlink direction during the simulation run with different retransmission mechanisms and number of bursts. As can be seen from the figure, the confidence intervals indicate that almost the same performance is achieved for a particular configuration regardless of an initial seed value. In general, HARQ outperforms the ARQ retransmission mechanism which is especially the case for large data bursts. The reason is that even though a large data burst has a higher

dations [3].

probability of being dropped, the HARQ retransmission gain helps when the same data is retransmitted. On the contrary, ARQ suffers a lot because a significant amount of PDUs are dropped when the channel quality declines. ARQ has a poor performance when the downlink channel varies because the BS link adaptation cannot track it as good as for the uplink data. As a result, large ARQ data bursts are very vulnerable to bad channel conditions and seldom channel reports. As the burst size becomes smaller, both ARQ and HARQ improves the performance. The reason is that smaller HARQ bursts have a less probability of being dropped and a number of ARQ bursts exploit better the time-varying properties of the wireless channel. As can be seen from the figure, when there are 4 data bursts per a frame, the best performance can be achieved. Further increase in the number of data bursts per frame results in better ARQ performance due to the HARQ signaling overhead. Since the downlink broadcast messages are transmitted in the downlink subframe, the network wastes more resources on transmitting the HARQ signaling information rather than sending the actual data.

It makes sense to mention separately the performance of ARQ on top of HARQ. As expected, it provides somewhat worse results than HARQ due to the increased MAC level overhead. At the same time, it outperforms ARQ when HARQ provides better results. Thus, ARQ on top of HARQ should be considered only in very particular cases, e.g., the equipment does not support HARQ PDU SN and HARQ receive buffer PDU ordering. Section 4 presents a few more cases when a provider may resort to using such a configuration.

3.2 Uplink performance

In this subsection we analyze the uplink performance of HARQ and ARQ mechanisms. We consider exactly the same simulation scenario as in the previous subsection, but all the connections send data in the *uplink* direction. Since the TCP connection tries to occupy all the available uplink resources, it is a good way to estimate the uplink performance.

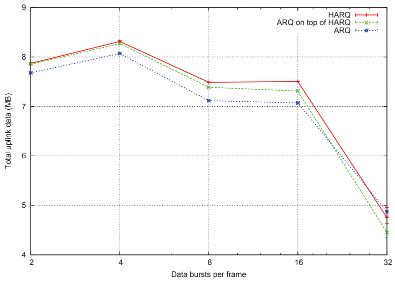


Figure 7: Amount of transferred data in the uplink direction.

Fig. 7 presents the total amount of data transferred in the uplink direction from all the connections. The ARQ on top HARQ behaves similar to the downlink case – its performance is a little bit worse than HARQ. At the same time, ARQ performance is much closer to HARQ and even outperforms it when there are many data bursts per a frame. The reason is that each downlink HARQ sub-burst needs a dedicated uplink transmission opportunity to report ACK or NACK. Thus, the more downlink connections a networks has, the less resources remain in the uplink sub-frame. Even uplink TCP transmission sends TCP ACKs in the downlink direction thus requiring HARQ uplink ACK transmission opportunity. For exactly these reasons ARQ performs better. Even though ARQ does not have the retransmission gain, it is capable of transmitting more data just because the uplink sub-frame has more free slots.

It is worth mentioning a reduced performance when there are 32 bursts per a frame. Since SSs use the CQICH channel to report downlink channel performance, each of them requires a dedicated uplink allocation. Thus, the more stations we have, the less free slots remain for user data.

4. DISCUSSIONS

Based on the presented simulation results it is possible to state that the ARQ mechanism can compete with HARQ. Of course, still HARQ outperforms ARQ in most cases. A key to the efficient HARQ functioning is a low signaling overhead. Otherwise, the significant amount of HARQ signaling data diminishes its gain over the ARQ mechanism. Thus, the BS scheduler should consider a few useful mechanisms that can reduce the HARQ overhead: packing of several HARQ sub-bursts into one data burst and HARQ sub-maps that allow for encoding different parts of the downlink MAP messages with different MCSs. However, these mechanisms can reduce the signaling overhead only in the *downlink* subframe. HARQ mechanism creates a significant overhead in the *uplink* sub-frame because each downlink HARQ sub-burst needs a separate allocation in the HARQ ACK channel. Unfortunately, the IEEE 802.16 specification does not provide any solution to alleviate this signaling burden.

To benefit from the best features of HARQ and ARQ mechanisms, it is possible to exploit the fact the 802.16 connections are always unidirectional. We can apply HARQ only for the uplink connections: a) no HARQ ACK transmission opportunity is necessary and b) HARQ helps to cope with a low uplink transmission power and interference from other cells. At the same time, by applying ARQ for the downlink connections we eliminate a need to allocate the HARQ ACK transmission in the uplink sub-frame. The BS high transmission power and other mechanisms, such as MIMO and beamforming, can mitigate absence of HARQ.

Another, and a more flexible approach, is to disable periodically the HARQ mechanism. Unlike ARQ, the BS may inform an SS that HARQ is disabled for a certain sub-burst. Such a behavior requires that both ARQ and HARQ mechanisms are turned on simultaneously. As presented in section 3, being configured correctly, ARQ on top of HARQ provides almost the same performance as HARQ. If the BS detects good channel conditions, then it can disable HARQ on fly thus leaving only the ARQ mechanism that can cope easily with random errors. Of course, this solution requires quite a sophisticated algorithm at the BS side that can activate and disable HARQ at right moments of time.

It is worth mentioning that we considered HARQ CC scheme, while there is another Incremental Redundancy (IR) mode that is supported by the IEEE 802.16 specification. It is anticipated that IR scheme can provide a larger gain over

HARQ CC and ARQ [5]. However, as presented in [8, 7], the IR scheme provides a significant gain only with the moderately varying channel; in the worst case it can have even a negative gain.

5. CONCLUSIONS

In this paper, we have analyzed the ARQ and HARQ mechanisms and presented their performance by means of extensive network simulations. According to our results, HARQ outperforms ARQ in the downlink direction because the BS is not capable of knowing the exact channel performance perceived by an SS. At the same time, ARQ can outperform HARQ when there is a significant amount of data bursts that create a large signaling overhead, especially in the uplink sub-frame. Based on this, the BS scheduler should consider placing only a limited amount of HARQ sub-bursts per a single frame. Otherwise, ARQ can be a more suitable option especially when the network has to support symmetrical services. Since HARQ mechanism can use two options for ordering the received PDUs – PDU SN and ARQ BSN – we have analyzed both of them. As follows from the results, the HARQ PDU SN provides the best results. At the same time, ARQ on top of HARQ results only in marginal performance degradation and is a feasible option especially when the ARQ mechanism is configured properly.

Our future research will aim at optimizing the HARQ performance and reducing its signaling overhead. Besides, further improvements in the ARQ mechanism are possible.

Acknowledgements

We express our acknowledgments to Vitaliy Tykhomyrov and Olli Alanen (University of Jyväskylä, Telecommunication laboratory) for contributing the development of the 802.16 extension for the NS-2 simulator. Special thanks to Jani Lakkakorpi (Nokia) and Jani Moilanen (Nokia Siemens Networks) for valuable discussions on the HARQ and ARQ functioning. Thanks to Thomas Höhne (Nokia Siemens Networks) for providing the PHY trace files.

6. REFERENCES

[1] Air interface for fixed broadband wireless access systems. IEEE Standard 802.16, Jun 2004.

[2] Air interface for fixed broadband wireless access systems - amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands. IEEE Standard 802.16e, Dec 2005.

[3] WiMAX Forum Mobile System Profile, Release 1.0 Approved Specification, Nov 2007. Revision 1.5.0.

[4] IEEE 802.16m evaluation methodology document (EMD). IEEE 802.16 Broadband Wireless Access Group, Mar 2008.

[5] Jung-Fu Cheng. Coding performance of Hybrid ARQ schemes. *IEEE Transactions on Communication*, 54(6):1017–1029, Jun 2006.

[6] C. Eklund, R. Marks, K. Stenwood, and S. Wang. IEEE standard 802.16: a technical overview of the Wireless MAN air interface for broadband wireless access. *IEEE Communications*, 40(6):98–107, Jun 2002.

[7] F. Frederiksen and T.E. Kolding. Performance and modeling of WCDMA/HSDPA transmission/H-ARQ schemes. In *IEEE Vehicular Technology Conference*, pages 472–476, Sep 2002.

[8] P. Frenger, S. Parkvall, and E. Dahlman. Performance comparison of HARQ with chase combining and incremental redundancy in HSDPA. In *IEEE Vehicular Technology Conference*, pages 1829–1833, Oct 2001.

[9] H. Martikainen, A. Sayenko, O. Alanen, and V. Tykhomyrov. Optimal MAC PDU size in IEEE 802.16. In *4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks*, pages 66–71, Feb 2008.

[10] A. Puchko, V. Tykhomyrov, and H. Martikainen. Link adaptation thresholds for the IEEE 802.16 base station. In *Workshop on NS-2 simulator*, 2008. Accepted for publication.

[11] A. Sayenko, O. Alanen, and T. Hämäläinen. ARQ aware scheduling for the IEEE 802.16 base station. In *IEEE International Conference on Communication*, pages 2667–2673, May 2008.

[12] A. Sayenko, O. Alanen, J. Karhula, and T. Hämäläinen. Ensuring the QoS requirements in 802.16 scheduling. In *The 9th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 108–117, Oct 2006.

[13] A. Sayenko, V. Tykhomyrov, H. Martikainen, and O. Alanen. Performance analysis of the IEEE 802.16 ARQ mechanism. In *The 10th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 314–322, Oct 2007.

[14] V. Tykhomyrov, A. Sayenko, H. Martikainen, O. Alanen, and T. Hämäläinen. On ARQ feedback intensity of the IEEE 802.16 ARQ mechanism. In *International Conference on Telecommunications*, Jun 2008.

PVI

ANALYSIS AND PERFORMANCE EVALUATION OF THE IEEE 802.16 ARQ MECHANISM

by

V. Tykhomyrov, A. Sayenko, H. Martikainen and O. Alanen 2008

Journal of Communications Software and Systems, vol 4, issue 1

Analysis and Performance Evaluation of the IEEE 802.16 ARQ Mechanism

Vitaliy Tykhomyrov, Alexander Sayenko, Henrik Martikainen, and Olli Alanen

***Abstract*—The IEEE 802.16 standard defines the ARQ mechanism as a part of the MAC layer. The functioning of the ARQ mechanism depends on a number of parameters. The IEEE 802.16 specification defines them but it does not provide concrete values and solutions. This paper studies the key features and parameters of the 802.16 ARQ mechanism. In particular, we consider a choice for the ARQ feedback type, an algorithm to build block sequences, the ARQ feedback intensity, a scheduling of the ARQ feedbacks and retransmissions, the ARQ block rearrangement, ARQ transmission window and the ARQ block size. We run simulation scenarios to study these parameters and how they impact the performance of application protocols. The simulation results reveal that the ARQ mechanism and its correct configuration play an important role in achieving reliable data transmission.**

***Index Terms*—IEEE 802.16 WiMAX, ARQ, NS-2**

I. Introduction

IEEE 802.16 is a standard for the wireless broadband access network [1] that can provide a high-speed wireless access to the Internet to home and business subscribers. It supports applications and services with diverse Quality-of-Service (QoS) requirements. The core components of a 802.16 system are a subscriber station (SS) and a base station (BS). The BS and one or more SSs can form a cell with a point-to-multipoint (PMP) structure. In this case, the BS controls the activity within a cell, resource allocations to achieve QoS and admission based on the network security mechanisms. An overview of the key 802.16 features is given in [6].

The automatic repeat request (ARQ) is the mechanism by which a receiving end of a connection can request the retransmission of MAC protocol data unit (PDU), generally as a result of having received it with errors. It is a part of the 802.16 MAC layer and can be enabled on a per-connection basis. The 802.16 specification does not mandate the usage of the ARQ mechanism meaning that it is a provider and a customer specific decision.

The 802.16 ARQ mechanism is controlled by a number of parameters. The specification defines them but it does not provide concrete values and solutions. The 802.16 ARQ configuration parameters have not been studied sufficiently, especially by means of extensive simulations. In [7], an analysis of the ARQ feedback types is presented. However, the UDP traffic, which is not sensitive to packet drops, is considered. Furthermore, no algorithm to select the feedback is presented. In [9], the ARQ mechanism is analyzed in the context of real-time flows of small packets. Authors estimate the bandwidth needed for the ARQ feedback messages. However, a simple simulation environment is used that does not capture any of the ARQ configuration parameters. This paper analyzes ARQ parameters and studies their impact on the performance of the ARQ mechanism. In particular, the following parameters are considered: ARQ feedback type, scheduling of ARQ feedbacks and retransmissions, ARQ feedback intensity, ARQ transmission window size, ARQ block size, ARQ block rearrangement. Though the 802.16 specification defines the Hybrid ARQ mechanism, we focus on ARQ because it is applicable to all the PHY types.

This paper extends our previous research and simulation work on 802.16 networks. In [13], [11], we presented a scheduling solution for the 802.16 BS and extensions for the ARQ aware scheduling. In [10], we analyzed the 802.16 contention resolution mechanism and proposed an adaptive algorithm to adjust the backoff parameters and to allocate a sufficient number of the request transmission opportunities.

The rest of the article is organized as follows. Section II presents key features and parameters of the 802.16 ARQ mechanism. We consider their impact on performance and propose a set of solutions. Next, Section III presents a number of simulation scenarios to study the ARQ performance. This section also analyzes the simulation results. Finally, Section IV concludes the article and outlines further research directions.

II. 802.16 ARQ mechanism

A. Basics of the ARQ mechanism

If ARQ is enabled for a connection, the extended fragmentation subheader (FSH) or the extended packing subheader (PSH) is used, which is indicated by the extended bit in the general MAC header (GMH). Regardless of the subheader type, there is a block sequence number (BSN) in the subheader that indicates the *first* ARQ block number in the PDU. A PDU is considered to comprise a number of ARQ blocks, each of which is of the same constant size except the final block which may be smaller. The ARQ block size is an ARQ connection parameter negotiated between the sender and the receiver upon a connection setup. It is worth mentioning that the ARQ block is a logical entity – the block boundaries are not marked explicitly. The remaining block numbers in a PDU can be derived easily on the basis of the ARQ block size, the overall

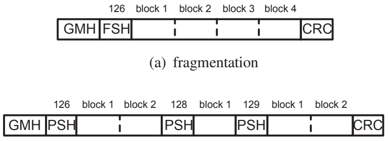
Manuscript received December, 2007 and revised February, 2008.

This paper was presented as part at the Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN) 2007

A. Sayenko is with Nokia Research Center, Helsinki, Finland.
(e-mail: alexander.sayenko@nokia.com).

V. Tykhomyrov, H. Martikainen, and O. Alanen are with Telecommunication laboratory, MIT department, University of Jyväskylä, Finland.
(e-mail: {vitykhom,henrik.martikainen,olli.alanen}@jyu.fi)

PDU size, and the first block number. Precisely for these reasons the ARQ block size is a constant parameter. Fig. 1 presents ARQ blocks with the fragmentation and packing mechanisms. Block numbers are given with respect to the BSN stored either in the FSH (see Fig. 1(a)) or PSH (see Fig. 1(b)).



(a) fragmentation

(b) packing

Fig. 1. ARQ blocks with packing and fragmentation mechanisms.

It is important to note that while the 802.16d specification [1] defines an ARQ block size as *any* value ranging from 1 to 2040 bytes, the 802.16e specification [2] has limited it to power of two values ranging from 16 to 1024 bytes, e.g., 16, 32, 64 and so on.

B. ARQ feedback types

To request a retransmission of blocks (NACK) or to indicate a successful reception of blocks (ACK), a connection uses ARQ block sequence numbers. In turn, the sequence numbers are exchanged by means of the ARQ feedback messages. The specification defines the following feedback types: a) selective, b) cumulative, c) cumulative+selective, and d) cumulative+sequence.

The selective feedback type acknowledges ARQ blocks received from a transmitter with a BSN and up to four 16-bit selective ACK maps. The BSN value refers to the first block in the first map. The receiver sets the corresponding bit of the selective ACK map to zero or one according to the reception of blocks with or without errors, respectively. The cumulative type can acknowledge any number of the ARQ blocks. The BSN number in the ARQ feedback means that all ARQ blocks whose sequence number is equal to or less than BSN have been received successfully. The cumulative+selective type just combines the functionality of the cumulative and selective types explained above. The last type, cumulative+sequence, combines the functionality of the cumulative type with the ability to acknowledge reception of ARQ blocks in the form of block sequences. A block sequence, whose members are associated with the same reception status indication, is defined as a set of ARQ blocks with consecutive BSN values. A bit set to one in the sequence ACK map entity indicates that a corresponding block sequence has been received without errors and the sequence length indicates the number of block that are members of the associated sequence.

When the ARQ feature is declared to be supported, a transmitting side, i.e., a receiver of the ARQ feedbacks, must support all the feedback types described by the 802.16 specification. The sender of the ARQ feedbacks has the ability to choose whatever format it will use. The WiMAX Forum recommendations [4] mandate the support of all the types except the selective ACK.

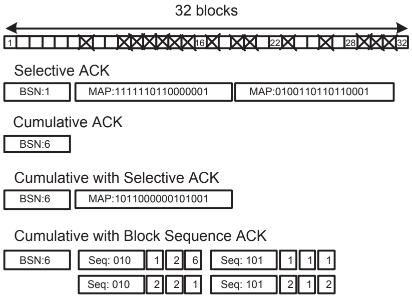


Fig. 2. Example of ARQ feedback types.

Fig. 2 presents an example in which every feedback type is applied to the same set of ARQ blocks. Selective ACK can acknowledge these 32 blocks in two maps. Cumulative ACK cannot acknowledge all the blocks because there are negative acknowledgements. Thus, only six blocks are encoded. Cumulative+selective ACK can send both positive and negative acknowledgements. However, since there should be 16 blocks per one selective map, some blocks remain unacknowledged. For this particular example, cumulative+sequence ACK can acknowledge only 28 blocks; one message can hold four sequence maps at most, whereas each map can have either two or three sequences. This type does not work effectively in this case because the block sequences are very short.

C. Choosing the feedback type

Each feedback type has its advantages depending on the ARQ feedback transmission frequency, the error disturbance patterns, and the computational complexity. From the implementation point of view, the selective feedback type does not require much processing resources because a connection simply puts information on the received blocks into the bitmap. On the other hand, a connection should try to rely upon the cumulative+sequence feedback type if resource utilization is of greater importance. However, it is more complex in implementation because block sequences must be detected. It could form an obstacle for a low power and low capacity mobile device.

In this section, we do not analyze the feedback types from the implementation complexity point of view, but rather propose an algorithm to choose an ARQ feedback type to achieve a good resource utilization. Our algorithm is based on the following assumptions: a) it is always more efficient to send positive acknowledgements by means of the cumulative type, and b) the sequence map can encode more blocks than the selective one. Indeed, the cumulative type can encode *any* number of ARQ blocks by using just one BSN number. Consequently, four sequence maps, each of which can have two sequences of 63 blocks, encode 504 blocks. If a map contains three short sequences, each of which can keep up to 15 blocks, then 180 blocks can be encoded. The proposed algorithm, simplified form of which is shown in Fig. 3, comprises the following three stages:

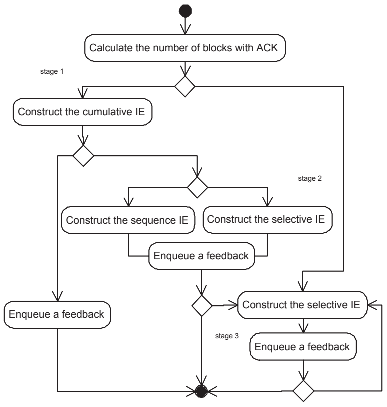

Calculate the number of blocks with ACK
stage 1
Construct the cumulative IE
stage 2
Construct the sequence IE
Construct the selective IE
Enqueue a feedback
Enqueue a feedback
Construct the selective IE
stage 3
Enqueue a feedback

Fig. 3. Algorithm to choose ARQ feedback types.

1) If there are positive acknowledgements in the beginning of the ARQ transmission window, construct the cumulative part. If there are no negative acknowledgements, then a single cumulative feedback message is created.
2) If there are remaining negative acknowledgements (optionally followed by positive and other negative acknowledgements), which we cannot send by using the cumulative part, then we have to choose a map type. To make a decision, we construct the sequence maps and calculate whether the selective maps can acknowledge more blocks. The maximum number of blocks to acknowledge selectively is 64 and it should be a multiple of 16. As for the sequence part, there is a limit for a sequence length and the number of sequences we can send in one message. Eventually, we will have either the cumulative+sequence or cumulative+selective feedback type. As a choice is made, we ”attach” map(s) to the cumulative part constructed at the previous stage.
3) Note that we can reach this stage in two cases. The first one is when there are no positive acknowledgements in the beginning of the ARQ transmission window and there is no way to create cumulative, cumulative+selective, or cumulative+sequence types. The second case to reach this stage is when neither cumulative+selective nor cumulative+sequence feedback types encode all the blocks. Though it is a rare case it can happen, because both the cumulative+selective and cumulative+sequence types have technical limitations. It is important to note that we cannot create and send several consecutive cumulative+. . . feedbacks because the cumulative part of the second message will re-acknowledge positively those ARQ blocks that are acknowledged negatively in the first message. Regardless of the situation, we just create as many selective feedback types as necessary to acknowledge the remaining blocks. As mentioned above, four selective maps can acknowledge up to 64 blocks. It is important to note that due to the clarifications in [3], it is feasible to construct and send the cumulative+sequence feedback type when there are negative acknowledgements in the beginning of the ARQ window. It is possible to put out of the Tx window BSN field in the cumulative part so that it is ignored at the sender (receiver of the ARQ feedback). Such a solution eliminates the need for the selective type when there are errors in the beginning and improves the MAC overhead.

It is worth noting that the presented algorithm scales well to the SS capabilities. If the selective type is not supported, then stage 3 is never executed. If there is no support for one of the cumulative types, then stages 1 and 2 are simplified.

Referring back to stage 2, it is worth mentioning an algorithm to create sequences for the cumulative+sequence ARQ feedback type. The specification does not define it thus allowing alternative implementations. As mentioned before, it is more complex in implementation because block sequences must be detected and correct sequence lengths must be constructed. To simplify this process, the algorithm uses two steps. On the first step, the algorithm parses all blocks and constructs sequences without checking any lengths. On the second step, the algorithm chooses sequence formats and, if necessary, splits large sequences into smaller ones so that they conform to the specification. The algorithm analyzes the current and the next sequence length to decide which sequence format should be used. As the sequence format is chosen, sequences are put into a map. If the sequence length exceeds the technical limit (63 for the format 0 and 15 for the format 1), then it is truncated and the remaining part is written into the input list so that it is processed at the next iteration. The simplified form of this algorithm is presented in Fig. 4. The algorithm stops when either all the sequences are processed or four maps are built. If there are not enough sequences to fill a single map, then zero lengths are put.

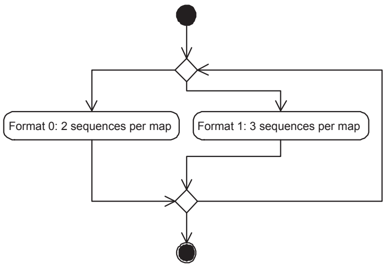


Fig. 4. Algorithm to construct sequence maps.

As mentioned above, two sequence formats are available. The algorithm uses a simple, yet powerful, condition to select an appropriate sequence format:

$$\text{Format} = \begin{cases} 0, & (S_i > 15) \text{ OR } (S_{i+1} > 15), \\ 1, & \end{cases} \tag{1}$$

where S_i is the ith sequence length. The idea behind (1) is that it is more resource conserving to switch to the sequence format 0 if either current or the next sequence is a large one. Otherwise, it is more efficient to use format 1 to encode more short sequences. As an example, Fig. 5 presents the input sequences and constructed sequence maps (the first number is the sequence status while a number in the square brackets is a sequence length). As can be seen, the algorithm and proposed conditions split efficiently sequences into the maps according to the 802.16 specification.

```
0[65] 1[17] 0[3] 1[11] 0[4] 1[67] 0[40]

   map #1    |   map #2    |     map #3       |    map #4
  format 0   |  format 0   |    format 1      |   format 0
 0[63] 0[2]  | 1[17] 0[3]  | 1[11] 0[4] 1[15] | 1[52] 0[40]
```

Fig. 5. Input sequence lengths and built maps

To illustrate that the proposed algorithm selects efficiently the required sequence format, we present two cases for the same initial input sequences when only one particular format is in effect. As can be seen from Fig. 6, both cases fail to encode all the blocks (due to space limitations, only three maps are presented for the sequence format 1).

```
   map #1    |   map #2    |   map #3    |   map #4
  format 0   |  format 0   |  format 0   |  format 0
 0[63] 0[2]  | 1[17] 0[3]  | 1[11] 0[4]  | 1[63] 1[4]
                 (a) format 0 maps only

       map #1       |      map #2       |     map #3      |
      format 1      |     format 1      |    format 1     |
 0[15] 0[15] 0[15]  | 0[15] 0[5] 1[15]  | 1[2] 0[3] 1[11] |
                 (b) format 1 maps only
```

Fig. 6. Built maps (only one particular format).

The resulting computational complexity of the proposed algorithm to construct sequence maps is $O(2N)$. We need to make two passes: the first one is to calculate the initial sequence lengths and the second one is to split sequences between the maps. The computational complexity of the selective map is $O(1)$.

D. Ordering of feedbacks and retransmissions

While sending normal packets, retransmissions, and ARQ feedback messages, a connection should determine their order. Indeed, as a scheduler at the BS allocates resources to a connection, either uplink or downlink, a connection's internal priority mechanism should decide upon which message is of more importance.

We propose to send first the ARQ feedbacks, then retransmissions, and finally the normal user PDUs. The reason we assign the highest priority to the ARQ feedbacks is that they do not require much space and they have a huge impact on the ARQ performance. As a sender receives a feedback, it knows the blocks that were received successfully and the blocks that are to be retransmitted. The successfully transmitted blocks can be removed from the retransmission buffer and the associated resources are cleared (see section II-H). Furthermore, the sender adjusts the ARQ transmission window that, in turn, influences the performance, because a connection cannot send more blocks than the ARQ window allows.

The reason we assign a higher priority to retransmissions is that a receiver can reconstruct a MAC service data unit (SDU) from fragments and forward it to the upper level only once all the fragments are received. Furthermore, if the *ARQ deliver in order* option is turned on,[1] then a receiver is obliged to forward SDUs in the same order in which a sender transmits them. This means that even though a receiver reconstructs successfully an SDU from all the fragments, it has to wait for all the previous SDUs.

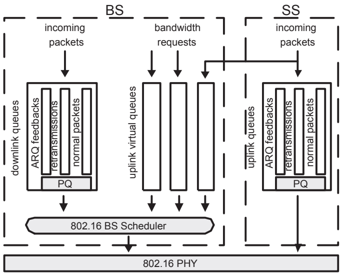


Fig. 7. Queue structure to prioritize feedbacks and retransmissions.

The simplest way to organize these priorities is to introduce several internal subqueues within a connection queue, as Fig. 7 illustrates. It is an extended version of the 802.16 QoS architecture considered in [5], [13]. Every time a PDU arrives to the connection queue, it will be checked and depending on its type it will be placed into an appropriate subqueue. When PDUs are dequeued, the queue can check first the subqueue with the ARQ feedbacks, then the subqueue with retransmissions, and only then the subqueue with normal PDUs. In other words, a connection queue implements internally the strict priority queuing.

An appealing feature of this approach is that it is completely transparent to the BS scheduler. Everything the BS scheduler needs to know to allocate resources is connection QoS requirements, if any, and the queue size [13]. If there are several internal subqueues, then the BS scheduler will be informed about the *aggregated* queue size. It is especially the case for the uplink virtual queues that are maintained through bandwidth requests sent by SSs. An SS cannot inform about the size of each subqueue but rather about the aggregated size. When a connection is allotted slots, first it will send ARQ feedbacks. If there are remaining bytes in a data burst, the connection will send retransmissions, and only then normal PDUs will be sent.

[1] It is anticipated that this option will be turned on for most services. Indeed, there is no sense in turning this option off for the UDP based applications, such as VoIP. The VoIP receiver will just discard packets that arrive in the wrong order unless some sufficiently larger input buffer is utilized, which is not typical for interactive applications. In the case of the TCP based services, an absence of a packet can be treated as a packet drop. It will trigger a retransmission of this packet though it can arrive later.

E. ARQ feedback intensity

Though IEEE 802.16 specification defines ARQ feedback types, it does not specify how often a receiver should send them. We considered the ordering of the ARQ feedbacks and retransmissions in section II-D , however, it does not provide an answer *when* a receiver should construct the ARQ feedback message and place it into the output queue.

Intuitively, it is understandable that the ARQ feedback intensity is a tradeoff between the MAC overhead and the robustness of the ARQ state machine. On the one hand, we may delay sending ARQ feedbacks to decrease the MAC overhead. On the other hand, failing to send the ARQ feedback on time may result in a very bad performance because ARQ blocks will be discarded by the ARQ timers. If the ARQ feedback transmission period T_{feedback} is less than the ARQ retry timer T_{retry}, then the performance starts to decline because a sender will retransmit the same data. If the feedback intensity is even less than the ARQ block lifetime T_{life}, then it may result in a very poor performance due to the discarded ARQ blocks. Based on that it is possible to propose the following inequality:

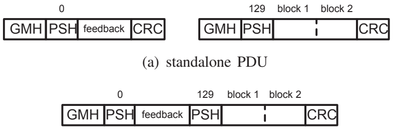
$$T_{\text{feedback}} < T_{\text{retry}} < T_{\text{life}}. \tag{2}$$

Since the ARQ retry and life timers are the connection specific parameters, the receiver can always adapt its ARQ feedback intensity on a per-connection basis. Since it is usually the case that the retry timeout is less than the life timeout, it is enough to analyze the retry timer value to choose a suitable ARQ feedback intensity.

It is worth mentioning that the ARQ feedback intensity should not be very close to the ARQ retry timeout. The reason is that the ARQ feedback message can be dropped due to the failed checksum test, as any other PDU.

F. Standalone and piggy-backed feedbacks

While sending the ARQ feedback message, a connection has an option whether to send it as a standalone message or piggy-back it to a PDU with user data (see Fig. 8). The former approach has somewhat larger MAC overhead of 12 bytes because the ARQ feedback resides in a separate PDU with mandatory GMH and PSH headers, and the CRC field. At the same time, the piggy-backed transmission is less reliable when compared to a standalone message. The reason is that being piggy-backed to a large PDU, the ARQ feedback has a higher probability of being dropped [8] because the whole PDU is discarded when an error is detected. If a sender does not receive any feedback before the ARQ retry timer expires, then correspondent ARQ blocks will be retransmitted. No need to say that a loss of the ARQ feedback message will lead to the retransmission of all ARQ blocks, even of those ones that have been received correctly. As mentioned in section II-E, if a sender does not receive any ARQ feedback before the ARQ block life timeout, then blocks will be discarded completely. Thus, to achieve a more reliable transmission of the ARQ feedbacks, it makes sense to rely upon the standalone feedbacks.



(a) standalone PDU

(b) piggy-backed to a PDU with user data

Fig. 8. ARQ feedback transmission.

G. ARQ block rearrangement

While retransmitting a PDU, a connection may face a problem that an allocated data burst is smaller than the PDU size to be retransmitted. This may happen if the BS scheduler allocates data bursts of different sizes, which is usually the case for real-time Polling Service (rtPS), non-real-time Polling Service (nrtPS), and Best Effort (BE) connections. Suppose, that the BS allocates a data burst of three slots for the BE connection and the latter sends a PDU that spans the whole data burst. If this PDU encounters an error, the connection will retransmit it. However, if the BS scheduler allocates later a data bursts of two slots, there is no way to retransmit the original PDU. Fortunately, the connection may rely upon *the retransmission with rearrangement* that allows for fragmenting the retransmitted PDU on the ARQ block size boundaries. If there is a sufficiently small ARQ block size, then the connection may construct a smaller PDU. As an example, Fig. 9 shows the rearranged PDU presented in Fig. 1(a). There are two PDUs with two blocks per each PDU.

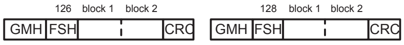


Fig. 9. Rearranged PDU.

In this subsection we do not focus on the optimal ARQ block size, but rather consider a solution for a case where a sender retransmission policy is not to use the ARQ block rearrangement. The reason this functionality can be absent is the fact that rearrangements involve much more complicated actions with PDUs in the retransmission buffer when compared to the PDU construction. A sender must keep a set of the ARQ timers for each ARQ block. If the retransmission with rearrangement is not implemented, then eventually a sender can associate all those timers with a PDU, which requires much less resources.[2] Furthermore, the rearrangement requires a sender to analyze a PDU and to search for block boundaries on which that PDU can be fragmented.

It is important to note that this problem concerns merely the uplink connections, because having the bandwidth request size, the BS does not know whether it is one big PDU or several smaller ones. In the case of the downlink transmission, the BS can always look inside the queue. Besides, this problem

[2] Practically, a sender can associate a timer with a whole PDU even if the ARQ block rearrangement is turned on. However, then it has to perform quite complicated actions with ARQ timers when the retransmitted PDU is partitioned into several PDUs because certain ARQ blocks are retransmitted while the other ones remain in the output buffer.

would not be so critical if the BS knew that a connection does not support rearrangements. However, there is no such QoS parameter that would indicate it. On the one hand, the BS can guess that a connection does not rearrange PDUs by monitoring bandwidth request sizes and the number of received bytes. On the other hand, a connection should not rely much upon this functionality because it is not mandated by the specification. Thus, the only safe way is to control the maximum size of transmitted PDUs. It is not a complicated task for the rtPS and nrtPS connections that should be always allocated such a number of slots that their minimum bandwidth requirements are ensured [13]. Thus, the maximum PDU size can be limited by the minimum data burst size allocated by the BS scheduler. The BE scheduling class is a more challenging task since the BS scheduler can allocate a data burst of any size. A connection may monitor allocated data burst sizes to control the maximum PDU size. Another possible solution is to send as large PDU as the size of one slot. However, such an approach may be unacceptable due to the increased MAC overhead and very small slot size of robust MCSs. As a result, regardless of an approach taken, the BE connection, which does not support retransmissions with rearrangements, should avoid sending large PDUs.

H. ARQ transmission window and ARQ block size

At any time a sender may have a number of outstanding and awaiting acknowledgements ARQ blocks. This number is limited by the ARQ transmission window that is negotiated between an SS and the BS during a connection set-up. A sufficiently large ARQ window allows for a continuous transmission of data. A connection can continue to send ARQ blocks without waiting for each block to be acknowledged. Conversely, a smaller ARQ window causes a sender to pause a transmission of new ARQ blocks until a timeout or the ARQ feedback is received. Though it may seem that a large transmission window is always the best choice, it is worth noting that a large transmission window leads to increased memory consumption and processing load. Every ARQ block must be stored in the retransmission buffer until a positive feedback is received. Taking into account the largest ARQ block size of 1024 bytes and the maximum ARQ transmission window of 1024 blocks, it is possible to arrive at the conclusion that some mobile and portable devices will not have enough resources to handle this amount of data for each frame.

If we assume a continuous errorless data transmission, then the maximum throughout a connection can achieve is limited by the following expression:

$$\frac{S^{\text{ARQ}}\,W\,\text{FPS}}{\text{DF}}, \tag{3}$$

where S^{ARQ} is the ARQ block size, W is the ARQ transmission window size, FPS is the number of frames per second and DF is the delay factor. In the case of the downlink transmission, the delay factor is always 1 because the BS can allocate a downlink data burst whenever it wants. In the case of the uplink transmission, the delay factor depends on PHY and whether a polling is in effect. If the BS polls a connection in *every* frame, then the delay factor is also 1. Otherwise, like in the case of the BE connections, the delay factor is 2 for OFDM and 3 for OFDMa PHY. The reason is that in OFDM PHY, the uplink bandwidth request carries the request size, while in the OFDMa PHY, special CDMA codes are used that do not carry any request size. As a result, once the BS receives the CDMA code, it puts a special uplink CDMA allocation where an SS can transmit the request size.

The ARQ transmission window and the ARQ block size parameters depend one on each other. On the one hand, a connection may prefer to work with a small ARQ transmission window that will result in a necessity of choosing a larger ARQ block size because the throughput may be limited by the transmission window size. A large block size requires less resources because a set of the ARQ timers must be associated with a single ARQ block at the sender and the receiver. At the same time, a connection supporting the retransmission with rearrangement may wish to work with a smaller ARQ block size because that will provide a greater flexibility in splitting large PDUs into several smaller ones. Furthermore, the choice for the ARQ block size can be dictated by the device peculiarities, such as the memory page size. These various requirements introduce a cyclic dependency between these two parameters.

We anticipate that the ARQ block size should be the governing parameter, while the ARQ transmission window size should be adapted. The reason is that the ARQ block size has a set of discrete values, while the ARQ transmission window can accept any value within the specified range.

III. Simulation

This section presents a simulation analysis of the 802.16 ARQ mechanism. To run simulations, we have implemented the 802.16 MAC and PHY levels in the NS-2 simulator. The implementation is called WINSE (WiMAX NS-2 Extension). The MAC implementation contains the main features of the 802.16 standard, such as frames, bursts, downlink and uplink transmission, connections, MAC PDUs, packing and fragmentation, the contention and ranging periods, the MAC level management messages, dynamic size of the MAP messages, and the ARQ mechanism. The ARQ implementation supports the ARQ blocks, the ARQ transmission window, retransmission with rearrangement, all the ARQ feedback types, and the ARQ timers. The ARQ implementation also includes the prioritization of the feedbacks and retransmissions, and the algorithm to select the feedback type and to build block sequences. The implemented PHY is OFDMa. The simulation results for the OFDM PHY can be found in [13].

Fig. 10 shows the network structure we use in the simulation scenarios. There is the BS controlling the 802.16 network, the parameters of which are presented in Table I.[3] To compare results fairly, we run somewhat simplified PHY model with a fixed signal to noise ratio of 2 dB, which corresponds to QPSK3/4 MCS, and forward error correction (FEC) block error rate of 1%. The downlink broadcast messages, such as DL-MAP and UL-MAP, use a more robust QPSK1/2 MCS;

[3] These parameters conform the WiMAX Forum mobile system profile [4].

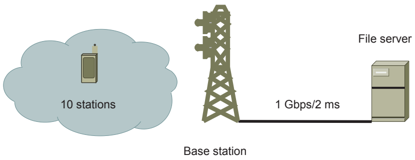


Fig. 10. Network structure.

TABLE I
802.16 NETWORK PARAMETERS.

Parameter	Value
PHY	OFDMa
Bandwidth	10 MHz
FFT	1024
Cyclic prefix length	1/8
TTG+RTG	464 PS
Duplexing mode	TDD
Frames per second	200 (5 ms per frame)
OFDM symbols	47
DL/UL symbols	26/21
DL/UL subcarrier alloc.	DL FUSC/UL PUSC
DL/UL slots	416/245
MAP MCS	QPSK1/2 (6 B/slot)
MCS	QPSK 3/4 (9 B/slot)
FEC BLER	1%
Ranging transm. opport.	2
Ranging backoff start/end	2/15
Request transm. opport.	8
Request backoff start/end	4/15
CDMA codes	256
ranging+periodic ranging	64
bandwidth request	192
handover	–
Fragmentation/packing	ON
PDU size	unlimited
CRC/ARQ	ON
ARQ feedback	standalone
ARQ feedback types	all
ARQ feedback intensity	5 ms
ARQ block size	16 B
ARQ window	1024
ARQ discard	ON
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ timers	
retry	50 ms
block lifetime	200 ms
Rx purge	200 ms

they are never dropped in our simulations. The BS runs the scheduling algorithm, details of which are presented in [13], [12]. In a few words, if there are only the BE connections, then the BS allocates resources fairly between the SSs based on their bandwidth request sizes. In addition, the ARQ aware scheduling is deployed to the BS station scheduler [11].

The BS scheduler also reserves two transmission opportunities for the initial ranging purposes (as in real life, an SS has to join the network in our simulator) and eight transmission opportunities for the bandwidth request contention resolution. The backoff parameters are given in Table I. The distribution of the CDMA contention codes is also given in Table I (since we do not simulate any mobility, there are no CDMA handover codes).

The simulation environment includes one wired node and ten SSs. Each SS establishes the basic management connection to exchange the management messages with the BS. In addition, to exchange user data, an SS establishes one uplink and downlink BE connection. An SS hosts exactly one FTP-like application that downloads data from a wired node over the TCP protocol. The reason we choose such an application type is that it tries to send as much data as possible thus utilizing all the network resources. At the same time, the TCP protocol is very sensible to packet drops that can occur in the wireless part. Each simulation run lasts for 10 seconds. The actual data transmission starts at the 1.5th second of the simulation run because first SSs has to enter the cell and register at the BS.

A. General ARQ results

In this simulation scenario we present general results concerning the ARQ performance. Fig. 11 presents the downlink throughput when neither ARQ nor errors are enabled. The throughput is calculated at the upper MAC level of the SS wireless interface, i.e., when the SS reconstructs original packets from received PDUs. As can be seen, all the BE connections have almost identical throughput. Since there are no QoS requirements, the BS scheduler allocates resources fairly between them.

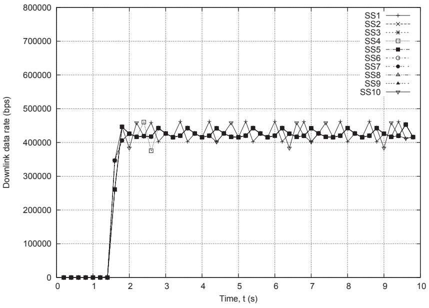


Fig. 11. Downlink throughput (no errors, no ARQ).

If we enable errors at the PHY level but keep the ARQ mechanism disabled for the transport connections, then there will be no smooth transmission anymore. As Fig. 12 illustrates, there is quite a bursty downlink data transmission. Some SSs even do not send data for some periods of time. Such a behavior is explained by the fact that the receiver does not test whether there is an erroneous PDU or not – it passes all the reconstructed SDUs to the wired node. Thus, the error detection and retransmission occurs at the transport layer which affects greatly the throughput.[4] It is worth mentioning that Fig. 12 presents even somewhat optimistic results because there is a small round-trip delay between the source subscriber stations and the destination wired node. As it becomes larger, the throughput would decline appropriately.

[4] Practically, a connection may include the CRC field into the PDU without enabling the ARQ mechanism. It will prevent a receiver from forwarding erroneous PDUs. However, a retransmission will still occur at the transport level.

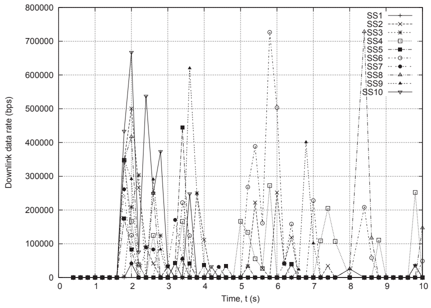


Fig. 12. Downlink throughput (errors, no ARQ).

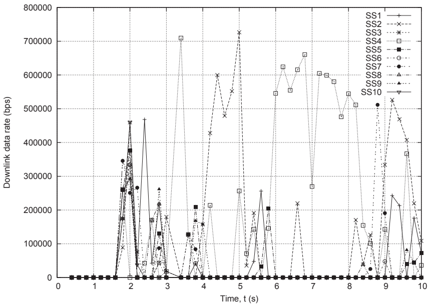


Fig. 14. Downlink throughput (errors, ARQ, no ARQ priority).

Fig. 13 shows the connection throughput when errors at the PHY level and the MAC ARQ mechanism are enabled. As follows from the figure, each BE connection achieves a smooth data transmission. Since there are errors in the PHY channel, the mean connection throughput is less than in Fig. 11. Nevertheless, the ARQ mechanism ensures extremely good resource utilization. The fluctuations are explained by the fact that PDUs are dropped and retransmitted.

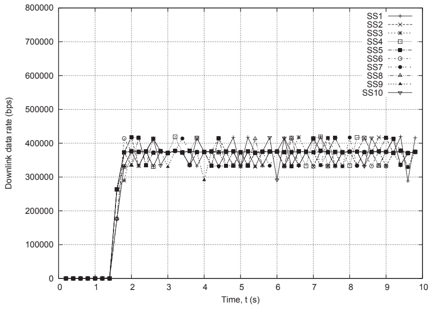


Fig. 13. Downlink throughput (errors, ARQ).

Fig. 14 presents the simulation results when the ARQ *priority* is absent, i.e., the ARQ feedbacks and retransmissions are transmitted in exactly the same order as they are put into the connection output queue. As can be seen from the figure, there are bursty changes in the uplink connection throughput, similar to the case when the ARQ mechanism is completely disabled. As considered in II-D, failing to prioritize ARQ feedbacks and retransmissions leads to a situation when the sender does not receive immediately information on ARQ blocks to retransmit thus resulting in a low performance.

Table II provides a comparison of these subcases by using another criterion, the total amount of downlink data. The amount of uplink data is much less and, due to the TCP behavior, is proportional to the downlink data. As follows from the results, an absence of the ARQ mechanism when there are errors in the transmission channel (which is usually a case for the wireless networks) results in a very low resource utilization. Table II also presents an interesting subcase when the ARQ is turned on, but errors are turned off. Its purpose is to show that ARQ introduces some overhead to the MAC level. Finally, absence of the priority for the ARQ retransmissions and the ARQ feedback messages decreases significantly the overall performance.

TABLE II
AMOUNT OF TRANSFERRED DATA.

ARQ	ARQ priority	errors	Downlink data (MB)
–	–	–	4.296
√	√	–	4.097
–	–	√	0.392
√	√	√	3.718
√	–	√	0.592

B. ARQ block rearrangement

In this subsection, we study the ARQ retransmission with rearrangement. The network parameters are the same as presented in Table I. There are ten SSs that download from the wired node through the BS. To demonstrate the ARQ block rearrangement importance, we turn on/off this feature and adjust the PDU size.

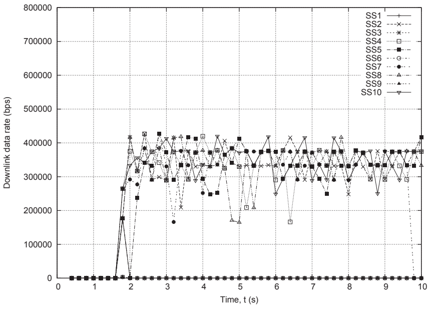


Fig. 15. Downlink throughput (no ARQ block rearrangement, large PDU).

Fig. 15 shows the throughput of downlink transmission when the ARQ block rearrangement is *turned off*. As can be seen from the figure, uplink connection throughputs are not smooth but rather change drastically. This is a result of the insufficient size of the uplink data burst when a connection retransmits a PDU. As explained earlier, while a connection may transmit a large PDU, an attempt to retransmit the same PDU may fail if the BS allocates later a data burst of a smaller size.

If a connection does not support the ARQ block rearrangement, then a possible solution is to use a smaller PDU size. Fig. 16 shows the downlink throughput for exactly the same case, but now all the connections have the maximum PDU size of 108 bytes, the ARQ block rearrangement is turned off.

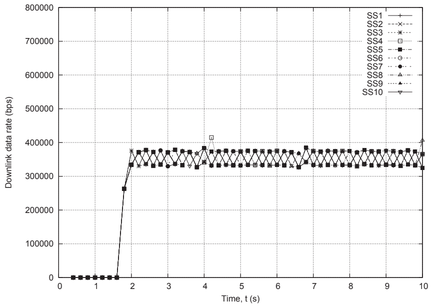


Fig. 16. Downlink throughput (no ARQ block rearrangement, PDU size is 108 bytes).

If we compare Fig. 16 (small PDU, no ARQ block rearrangement) and Fig. 13 (unlimited PDU size, ARQ block rearrangement), we may notice that the ARQ block rearrangement has an impact on the performance. Connections can use large PDUs of any size thus decreasing the MAC level overhead. At the same time, all the connection achieve a smooth data transmission. It is noticeable that an average connection throughput in Fig. 16 is less than in Fig. 13, which is explained by the MAC overhead caused by the small PDU size.

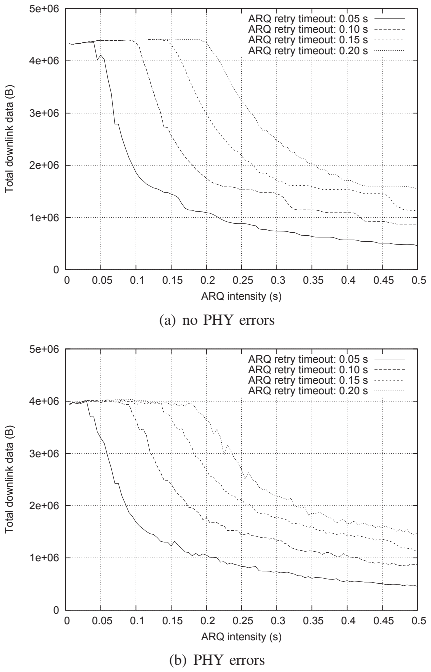
TABLE III
AMOUNT OF TRANSFERRED DATA.

Rearrangement	PDU size (B)	Downlink data (MB)
√	unlimited	3.718
–	unlimited	2.062
–	108	3.581

Table III also shows the amount of downlink data for this simulation scenario. As follows from the results, a connection should consider smaller PDU sizes if the ARQ block rearrangement functionality is not supported.

C. *ARQ feedback intensity*

In this simulation subcase, we study the impact of the ARQ feedback intensity on the network resource utilization. The network parameters are the same as in the previous simulations scenarios, the only difference is that we use the ARQ block size of 128 bytes. Otherwise, with a low ARQ feedback intensity the ARQ transmission window may get full and the transmission will stall. Also, the ARQ block lifetime timer is increased to 1.3 seconds to prevent ARQ blocks from being discarded when the low ARQ feedback intensity is in effect.



(a) no PHY errors

(b) PHY errors

Fig. 17. Total downlink data

Fig. 17 presents the simulations results for different ARQ feedback intensity and ARQ retry timeout values. According to Fig. 17(a), less frequent ARQ feedback messages allow connections to achieve better throughput due to the decreased MAC overhead. However, as the ARQ feedback transmission interval value is close to the ARQ retry timeout value, the MAC utilization starts to decline because a sender retransmits the same data. If there are errors in the wireless channel, then a lower ARQ feedback intensity results only in a marginal improvement, as Fig. 17(b) shows. At the same time, the MAC utilization starts to degrade even earlier than the ARQ retry timeout value. The reason for this is that ARQ feedback messages can be dropped, as any other PDU. Thus, a higher ARQ feedback intensity introduces a redundancy into the ARQ feedback mechanism – even if one feedback message is lost, the next one will duplicate the information.

Based on the presented simulation results, it is possible to arrive at the conclusion that the ARQ feedback transmission interval must be at least two times less than the ARQ retry timer. A higher ARQ feedback intensity results only in a slightly increased MAC overhead. At the same time, it improves the robustness of the ARQ feedback mechanism. More results including the ARQ feedback intensity over the HARQ enabled connections can be found in [14]

D. ARQ feedback types

In this simulation subcase, we study the ARQ feedback types. The network parameters are the same as in the previous simulations scenarios, the only difference is that we use different ARQ block size values and the ARQ feedback transmission interval is set to 40 ms.

TABLE IV
THE ARQ FEEDBACK TYPES STATISTICS.

ARQ block (B)	Feedback type percentage (%)				Num. of msg.	Downlink data (MB)
	Sel	Cum	Cum+ sel	Cum+ seq		
16	0	92.025	0	7.975	3586	2,769
32	0	91.910	0	8.091	3535	2,773
64	0	92.125	0	7.875	3543	2,769
128	0	91.556	0	8.444	3541	2,759
256	0	92.033	0	7.967	3477	2,817
512	0	92.312	0	7.688	3278	2,757
1024	0	90.547	0	9.453	2941	2,635

Table IV shows the results for these simulation runs. The total number of ARQ feedback messages sent in each simulation run and the percentage of each ARQ feedback type are presented. As can be seen, neither selective nor cumulative+selective feedback messages are sent during the simulation runs. As explained earlier, it is almost always more efficient to send acknowledgments by means of the cumulative+sequence type that can encode more blocks than the cumulative+selective. If there are only positive acknowledgments, then the cumulative feedback type is used. As follows from the table, the majority of the ARQ feedback messages are of this type. As explained earlier, due to the clarifications in [3], it is feasible to construct and send the cumulative+sequence feedback type when there are negative acknowledgements in the beginning of the ARQ window. It is possible to put out of the Tx window BSN field in the cumulative part so that it is ignored at the sender (receiver of the ARQ feedback). Such a solution eliminates the need for the selective type and improves the MAC level utilization.

It was anticipated that as we increase the ARQ block size, the number of the ARQ feedback messages should decline. As follows from Table IV, it is indeed so. It is also important to note that the best performance is achieved for the ARQ block size of 256 bytes. Smaller ARQ block sizes create a larger MAC overhead, while larger ARQ block sizes result in a higher PDU error rate [8] because the minimum PDU size should be large enough to carry at least one ARQ block [11]. Besides, as mentioned earlier, large ARQ block size values may prevent a connection from utilizing all the burst size because the PDU is fragmented and retransmitted on the ARQ block boundaries. If a connection uses a large ARQ block size then it is less flexible in retransmitting PDUs. Thus, the optimal ARQ block size is a tradeoff between the PDU error rate and the number of the ARQ feedback messages, which cause the ARQ overhead at the MAC level.

TABLE V
THE ARQ FEEDBACK TYPES STATISTICS.

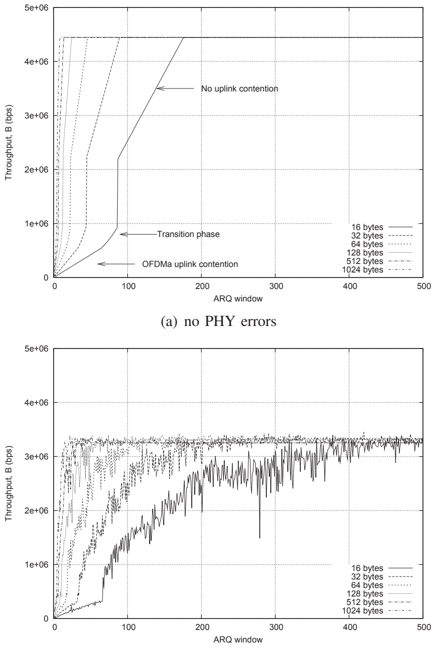
ARQ block (B)	Feedback type percentage (%)				Num. of msg.	Downlink data (MB)
	Sel	Cum	Cum+ sel	Cum+ seq		
16	100	0	0	0	5639	2,680
32	100	0	0	0	3621	2,644
64	100	0	0	0	3382	2,356
128	100	0	0	0	3351	2,394
256	100	0	0	0	3411	2,758
512	100	0	0	0	3221	2,697
1024	100	0	0	0	2853	2,596

Table V shows the results where only the selective ARQ feedback type is enabled. As expected, there are more ARQ feedback messages, especially for small ARQ block sizes, such as 16 bytes. If we compare the amount of transferred data in Table V and Table IV, then we can arrive at the conclusion that the selective ARQ feedback type does not result in a severe performance degradation. Thus, being combined with larger ARQ block sizes, it can be a valid choice for certain mobile devices with limited computational resources.

E. ARQ transmission window

In this subsection we study the impact of the ARQ transmission window on the throughput. The network parameters are the same as in the previous simulations scenarios, the only difference is that we vary the ARQ transmission window and block sizes. There is only one SS, otherwise it would be difficult to present an analysis of the throughput of all the SSs. We run a separate simulation for each ARQ transmission window value and ARQ block size. Since an SS throughput fluctuates during a simulation run, it is averaged by using the exponentially weighted moving average algorithm.

Fig. 18 presents the simulations results for this case with the PHY errors turned off and on. The figure indicates that large ARQ block sizes allow a connection to achieve its maximum throughput even for small ARQ transmission window values. Conversely, a small ARQ block value needs a large ARQ transmission window to achieve a high throughput. In the case of the errorless transmission, as the ARQ transmission window grows, the throughput increases linearly regardless of the ARQ block size. Of course, it grows faster for larger ARQ block sizes. When the ARQ transmission window reaches a certain value, its further growth does not have an impact on the throughput because the latter is limited by the overall network capacity. It is noticeable that regardless of the ARQ block size value, there are several phases in how the throughput increases. (see Fig. 18(a)). In the beginning, it grows very slow due to the fact the stations have to take part in the uplink connection resolution to send to the BS TCP acknowledgements and the ARQ feedback messages. In this case, the throughput is approximated accurately by (3) with the delay factor of 3. When a certain point is reached, there is a continuous uplink transmission due to the increased downlink traffic. Stations do not take part in the uplink contention resolution anymore as



(a) no PHY errors

(b) PHY errors

Fig. 18. Downlink throughput and ARQ window.

they piggy-back their bandwidth requests to user data. In this case, the throughput is approximated accurately by (3) with the delay factor of 1. A similar behavior is observed when there are PHY errors (see Fig. 18(b)). The only difference is that throughput increases much slower due to the PDU retransmissions.

Fig. 18 illustrates clearly that small ARQ transmission window values may prevent a connection from sending data even if it has slots allocated by the BS scheduler. Though it is not a huge problem for the BE connections, one should account for it if there is a QoS connection with the minimum bandwidth requirements.

IV. Conclusions

In this paper, we have analyzed the performance of the 802.16 ARQ mechanism. We have shown that the ARQ mechanism can improve significantly a performance of the TCP based applications. Since a probability for an erroneous transmission in the wireless channel is much higher when compared to the wired medium, the ARQ mechanism should be enabled for the TCP connections if a provider wants to ensure better QoS and to maximize the network utilization. Though we did not present simulation results for the UDP protocol, it is clear that its performance would not be affected by the absence of the ARQ mechanism because the UDP transmission does not depend on packet drops.

We have proposed a solution on how to prioritize normal PDUs, ARQ feedbacks, and retransmissions. The simulation results have also revealed the importance of the ARQ block rearrangement functionality. If an SS does not support it, then an additional care must be taken. An SS should choose smaller PDU sizes to achieve a smooth data transmission. We have also demonstrated that a connection must choose a sufficiently large ARQ transmission window size to utilize the allocated resources. While large ARQ blocks can utilize resources even with a small ARQ window, small ARQ blocks, such as those of 16 and 32 bytes, require much larger ARQ window. We proposed lightweight, yet efficient, algorithms to select the ARQ feedback type and to build block sequences for the cumulative+sequence feedback type. Besides, the selective ARQ feedback type does not result in a severe performance degradation; mobile devices with scarce computational resources may rely safely upon it. If a receiver can adjust the ARQ feedback intensity, then it better to rely upon a higher ARQ feedback intensity to avoid retransmissions activated by the ARQ retry timer. In any case, the ARQ feedback transmission interval must not be less than the ARQ retry timer.

Our future research works will aim to study the optimal parameters of the ARQ mechanism, which is especially the case for the ARQ-enabled QoS connections. It is also important to compare the results provided by the ARQ mechanism and the HARQ mechanism available in the OFDMa PHY.

Acknowledgments

We would like to express our acknowledgments to Jani Lakkakorpi (Nokia) and Jani Moilanen (Nokia Siemens Networks) for their valuable discussions on the IEEE 802.16 ARQ mechanism functioning. Special thanks to Jan Suumäki (Nokia) for providing a valuable insight on the ARQ mechanism implementation and reviewing the paper.

References

[1] Air interface for fixed broadband wireless access systems. IEEE Standard 802.16, Jun 2004.

[2] Air interface for fixed broadband wireless access systems - amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands. IEEE Standard 802.16e, Dec 2005.

[3] Air interface for fixed and mobile broadband wireless access systems. IEEE Standard 802.16 (Corrigendum 2/D4), May 2007.

[4] WiMAX Forum Mobile System Profile, Release 1.0 Approved Specification, Nov 2007. Revision 1.5.0.

[5] C. Cicconetti, L. Lenzini, and E. Mingozi. Quality of service support in IEEE 802.16 networks. *IEEE Networks*, 20(2):50–55, Mar/Apr 2006.

[6] C. Eklund, R. Marks, K. Stenwood, and S. Wang. IEEE standard 802.16: a technical overview of the Wireless MAN air interface for broadband wireless access. *IEEE Communications*, 40(6):98–107, Jun 2002.

[7] Min-Seok Kang and Jaeshin Jang. Performance evaluation of IEEE 802.16d ARQ algorithms with NS-2 simulator. In *IEEE Asia-Pacific Conference on Communications*, pages 1–5, Aug 2006.

[8] H. Martikainen, A. Sayenko, O. Alanen, and V. Tykhomyrov. Optimal MAC PDU size in IEEE 802.16. In *4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks*, pages 66–71, Feb 2008.

[9] S. Perera and H. Sirisena. Contention based negative feedback ARQ for VoIP services in IEEE 802.16 networks. In *4th IEEE International Conference on Networks*, volume 2, pages 1–6, Sep 2006.

[10] A. Sayenko, O. Alanen, and T. Hämäläinen. Adaptive contention resolution parameters for the IEEE 802.16 networks. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, Aug 2007.

[11] A. Sayenko, O. Alanen, and T. Hämäläinen. ARQ aware scheduling for the IEEE 802.16 base station. In *IEEE International Conference on Communication*, pages 2667–2673, May 2008.

[12] A. Sayenko, O. Alanen, and T. Hämäläinen. Scheduling solution for the IEEE 802.16 base station. *Computer Networks*, 52:96–115, 2008.

[13] A. Sayenko, O. Alanen, J. Karhula, and T. Hämäläinen. Ensuring the QoS requirements in 802.16 scheduling. In *The 9th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 108–117, Oct 2006.

[14] V. Tykhomyrov, A. Sayenko, H. Martikainen, O. Alanen, and T. Hämäläinen. On ARQ feedback intensity of the IEEE 802.16 ARQ mechanism. In *International Conference on Telecommunications*, Jun 2008.

Vitaliy Tykhomyrov has obtained the M.Sc. degree from Kharkov National University of RadioElectronics(Ukraine) in 2006. Currently, he is a Ph.D. student at the University of Jyväskylä. His current postgraduate research interests include Quality-of-Service in wireless networks, and in particular IEEE 802.16.

Alexander Sayenko has obtained the B.Sc. degree from the Kharkov State University of RadioElectronics (Ukraine) in 2001. He has obtained the M.Sc. degree from the University of Jyväskylä (Finland) and the Ph.D. degree from the same university in 2002 and 2005, respectively. Currently, he works for the Nokia Research Center, where he is responsible for the resource and power management solutions. His research interests are QoS, resource management and scheduling in the wireless networks.

Henrik Martikainen is a Ph.D. student at University of Jyväskylä, Finland. He received his M.Sc. in Computer Sciences from University of Jyväskylä in 2006. Since that he has been studying IEEE 802.16 MAC level performance and optimization.

Olli Alanen is a researcher working in University of Jyväskylä, Finland. He has been studying QoS issues in IEEE 802.16 and IP based networks for the past years. He received his M.Sc. in Computer Sciences from University of Jyväskylä in 2004 and the Ph.D. degree from the same university in 2007.

PVII

WINSE: WIMAX NS-2 EXTENSION

by

A. Sayenko, O. Alanen, H. Martikainen, V. Tykhomyrov, O. Puchko, V. Hytönen,
T. Hämäläinen 2011

Simulation: Society for Computer Simulation International

Simulation

Simulation: Transactions of the Society of Modeling and Simulation International
87(1-2) 24–44
© The Author(s) 2010
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0037549710373334
sim.sagepub.com

WINSE: WiMAX NS-2 extension*

Alexander Sayenko[1], Olli Alanen[2], Henrik Martikainen[2], Vitaliy Tykhomyrov[2], Oleksandr Puchko[2], Vesa Hytönen[2] and Timo Hämäläinen[2]

Abstract

The IEEE 802.16 standard defines a wireless broadband technology called WiMAX. When compared with other wireless technologies, it introduces many interesting advantages at PHY, MAC, and QoS layers. Heavy simulations are needed to study the performance of IEEE 802.16 and propose further enhancements to this standard. Link-level simulations are not always sufficient, while system-level simulators are not always accurate enough to capture MAC and transport protocol details. We implemented an 802.16 extension for the NS-2 network simulator. It includes upper PHY modeling, almost all of the features of the 802.16 MAC layer, as well as the QoS framework. This article describes the implemented features and simulation methodology, while sharing our experience that can be used with other NS-2 modules. There is also an overview of the past and ongoing research.

Keywords

IEEE 802.16 WiMAX, NS-2

1. Introduction

IEEE 802.16, also known WiMAX, is a standard for the wireless broadband access network[1,2] that can provide high-speed wireless access to home and business subscribers. It can satisfy diverse quality-of-service (QoS) requirements which makes IEEE 802.16 a scalable platform for many services.[3] The core components of a 802.16 system are a subscriber station (SS) and a base station (BS). The BS and one or more SSs can form a cell with a point-to-multipoint (PMP) structure. In this case, the BS controls the activity within a cell, resource allocations to achieve QoS, and admission based on the network security mechanisms.

As any other wireless technology, IEEE 802.16 emerged from an enormous number of technical contributions from various vendors. This technology continues to evolve through technical corrections, best practices, and more radical proposals. All of these changes are supported by heavy simulations performed by researchers and engineers at various architectural levels. It is quite difficult, or even impossible, to accept a contribution if it is not based on reliable simulation results.

Traditionally, wireless network simulations are performed with two different types of simulators. The first type is the link-level simulator, in which a link between a base station and a subscriber station is modeled with many details. The output from these simulators is usually a bit, block and packet error probability under different parameters and conditions.

The second large group of simulators, which is widely adopted by industry, consists of the system-level simulators. They model a number of geographical cells where base stations provide service to subscriber stations. Usually, the system simulators do not model all of the link-level aspects, but rather rely on the results obtained from link-level simulations. Nevertheless, their level of detail includes sub-carriers and a particular permutation type. System simulators are classified

[1] CTO, Nokia Siemens Networks, Espoo, Finland.
[2] Telecommunication laboratory, MIT Department, University of Jyväskylä, Finland.

*This paper is an extension of our publication in the 2nd International Conference on Simulation Tools and Techniques.

Corresponding author:
A Sayenko, CTO, Nokia Siemens Networks, Espoo, Finland
Email: alexander.sayenko@nsn.com

further as either static or dynamic. Static simulators assume that subscriber stations do not move and the whole system is modeled with several ''snapshots'' of time, where the positions of stations are randomly distributed over the simulation area. In dynamic system simulations, stations can move over the simulation area performing network entries and handovers between the base stations. Therefore, the dynamic system simulators capture simulation results as a function of time. Their only disadvantage is their high complexity and long simulation running time.

A third option for wireless simulations is packet-level simulators, such as NS-2.[4] When compared with the dynamic system simulators, the packet-level simulators are very similar in terms of features provided. However, protocol stacks and application behavior are modeled more accurately. Many PHY aspects are abstracted by means of simpler interfaces and models. The packet-level simulators allow the access service networks to be simulated because it is possible to define a network topology where base stations, routers, gateways, and clients send or receive data. This makes it possible to obtain true end-to-end simulation results.

In this paper we present the 802.16 extension for the NS-2 simulator, which is a dynamic packet-level simulator. The 802.16 extensions on top of the NS-2 packet core result in a simulator that combines the properties of both the system-level and packet-level simulators and allows us to concentrate on the application-level performance and study truly end-to-end results. Nevertheless, it is worth mentioning that the system-level simulators are used to model PHY in greater detail.

The rest of this paper is organized as follows. Section 2 gives a brief overview of other NS-2 802.16 modules. Section 3 provides details of our 802.16 extension. We do not delve into specific implementation details, but rather elaborate on simulation methodology and the trade-off between complexity and simulation time. Section 4 gives an overview of past topics studied with our 802.16 module, while Section 5 presents an overview of ongoing research. Finally, Section 6 concludes the paper.

2. Previous work

In this section we give a brief overview of existing 802.16 modules for NS-2 that we are aware of. We refrain from any evaluation: our aim is to provide an insight on typical features that other wireless broadband packet-level simulators may or may not have. A more practical comparison and analysis of different 802.16 modules can be found in Bohnert et al.[5]

2.1. NIST module

NIST module is definitely one of the first 802.16 extensions for the NS-2 simulator. Even though the project was merely to study the inter-network handovers, its results include also 802.16 MAC, handover, and scheduling extensions. The list of supported features as well as the general description is given by Rouil.[6] As a brief summary: there is almost no proper PHY with a correct error generation; there is a OFDM PHY emulation, whereas WiMAX is based on OFDMa PHY; absence of the ARQ mechanism makes it complicated to deploy the error model; and the MAC level lacks several important features, such as packing.

2.2. NDSL module

This module is a result of a joint work between Chan Gung University and the Institute for Information Industry. The module description and features are presented in Chen et al.[7] Somewhat similarly to the NIST module, it focuses mostly on MAC leaving the PHY level unattended. However, the MAC-level operational parameters correspond to the OFDMa PHY. In addition, the MAC-level implementation includes both fragmentation and packing; special attention is paid to the management messages and network entry procedure. ARQ and HARQ retransmission mechanisms are not implemented. Despite a good set of features, further development of this module has stopped.

2.3. WiMAX Forum module

The WiMAX Forum has been developing its NS-2 802.16 extension that emerged from the NIST module. Since this module is available only for the WiMAX Forum members, only a brief overview is given here. Unlike the module from NIST, it focuses on OFDMa PHY and tends to emulate the PHY behavior at a sub-carrier level. However, several important PHY features, such as channel reporting and link adaptation, were not introduced. The MAC level lacks a full support for ARQ; the HARQ retransmission is not implemented at all. Unfortunately, the WiMAX Forum announced that the development process will stop and the future of this module is not clear. Nevertheless, this module is very important in a sense that this was one of the first attempts to introduce PHY at the sub-carrier-level granularity.

2.4. INRIA NS-3 WiMAX module

This is an interesting 802.16 extension written for the NS-3 simulator, a technical description of which is given by Farooq and Turletti.[8] The module focuses mostly on the MAC layer and, unlike many other modules, has support for features, such as scanning, initial ranging, and network entry. At the same time, many important MAC-level features are not implemented, e.g., fragmentation and packing. Furthermore, the absence of ARQ and HARQ retransmission mechanisms does not allow for the error-generation module to be applied. The PHY layer is OFDM, whereas WiMAX relies upon OFDMa.

2.5. Other modules

There is an 802.16 extension for NS-2 from the Eurecom Institute. This is a relatively new module, features of which are described by Msadaa et al.[9] Even though the authors present it as a novel module with an integrated QoS architecture, it does not differ in principle from the NIST implementation. The PHY model is not revised at all and the MAC timing works in accordance with OFDM PHY parameters.

There is also a module from KAIST University[10] and 802.16 an extension for the NS-2 MIRACLE framework.[11]

2.6. Pisa University 802.16d Mesh module

This module implements 802.16 Mesh that is an alternative to the PMP mode. However, the IEEE 802.16 working group discontinued the 802.16 Mesh mode removing it completely from the IEEE 802.16 evolution.[12] The absence of standardization and industry support makes the competition with other *ad-hoc* technologies, such as 802.11s, quite complicated.

Table 1. Features supported by WINSE

PHY	OFDM and OFDMa PHY (FUSC and PUSC)
	FEC blocks
	HARQ: Type I, UL ACK channel
	Channel reports: REP-RSP and CQICH
	Link adaptation
	Closed-loop power control
MAC	Duplexing modes: TDD, FDD, H-FDD
	Zones: static and dynamic size adjustment
	DL broadcast messages: DL-MAP, UL-MAP, DCD, UCD
	Compressed MAP, sub-MAPs
	Connections: DL broadcast, basic management, transport
	PDU construction, fragmentation, packing
	Bandwidth requests: standalone and piggy-backed
	ARQ: blocks, feedbacks, timers, transmission window
	Uplink contention: OFDM and CDMA-based for OFDMa
	Network entry
	Handover: SS-initiated, automatic & manual
	Sleep mode: class I, II, and III
	2-hop TTR non-transparent relays
QoS & scheduling	UGS, ertPS, rtPS, nrtPS, BE
	BS scheduler
	SS uplink scheduler
Access service network	ASN-GW
	R4, R6, and R8 interfaces
	ASN-anchored mobility

3. WINSE

3.1. Overview

WINSE is a WiMAX extension for the NS-2 simulator. It was started as a small student project and then evolved into a powerful simulation tool that several companies now use to study MAC and QoS in the 802.16 system. Table 1 gives a short overview of features supported in WINSE.

3.2. Core principles

Before delving into the technical details, it is worth mentioning the core principles behind our 802.16 module.

- *Preserving the existing framework*. While introducing the 802.16 extensions, we do not change the NS-2 core classes, but rather introduce functionality by creating new ones. As considered in many papers, the NS-2 core lacks many features and sometimes suffers from a bad internal design. However, our approach is that researchers should be able to combine different modules on a common simulation platform to study more complex scenarios. Radical changes should be addressed by major revisions, e.g., NS-3.
- *C++ modularity*. This emerged implicitly from the previous principle: there should be a class hierarchy with well-defined class responsibilities. Figure 1 shows the UML diagram with the top-level classes that constitute the core of our 802.16 module. Each functional block is contained in an independent C++ class that allows for further virtualization and abstraction.
- *OTcl modularity*. The most critical and fundamental C++ classes are mapped to the correspondent OTcl

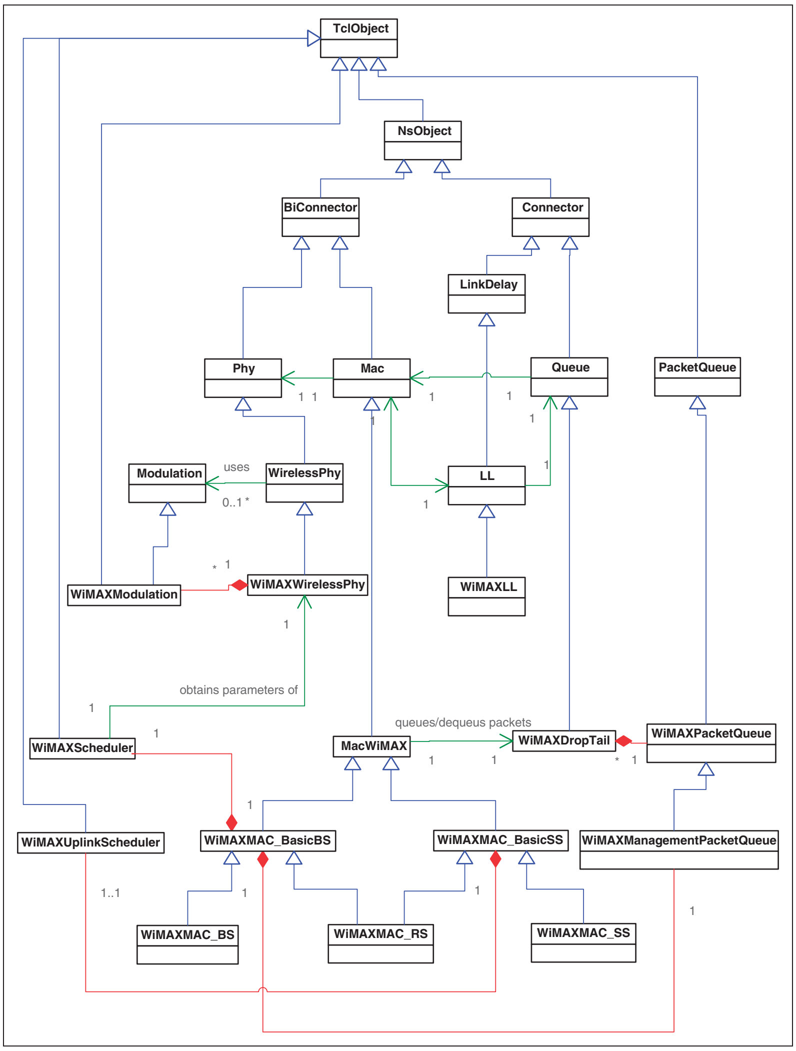


Figure 1. UML diagram for WINSE core classes.

classes. As can be seen from Figure 1, all of the major classes are derived, either explicitly or implicitly, from **TclObject**. It allows a script designer to change easily parameters and switch between different modules just by selecting their OTcl class names.

- *Balance between the PHY and MAC features*. The power of the NS-2 simulator is in the transport and application level. It is unreasonable to go deep into PHY modeling as it will significantly increase the computational burden. We carefully selected the PHY features to model and the ways to model them to ensure that NS-2 does not turn into a link-level simulator.

3.3. PHY layer

3.3.1. PHY abstraction. A properly designed PHY level must introduce an abstraction that is general enough to hide particular PHY details from other architectural components, such as MAC and the scheduler. All of the 802.16 modules mentioned in Section 2 have PHY-specific MAC implementation resulting in either an OFDM or OFDMa 802.16 simulator. Even though the industry chose OFDMa PHY as a basis for WiMAX networks, there are a number of OFDM devices. Furthermore, an upcoming 802.16m PHY also creates a need for supporting several PHYs simultaneously.

Based on our experience, it is possible to provide such an abstraction if the PHY layer exposes at least the following parameters: duplexing mode, OFDM symbol duration, zone parameters (a number of OFDM symbols in a single slot and a number of channels in one symbol), TTG and RTG gaps, effective slot size for each modulation and coding scheme (MCS).

Having this information, the MAC level can work without knowing the particular PHY details. This abstraction is also important for the BS scheduler. As an example, our implementation has a common MAC-level scheduler that works correctly on top of both OFDM and OFDMa. As can be seen from Figure 1, all of the core components have an association to the **WiMAXWirelessPhy** class that abstracts particular PHY sub-classes.

3.3.2. Effective signal-to-interference noise ratio. One of the most critical issues for a packet-level simulator is how to obtain a valid effective signal-to-interference noise ratio (SINR). As we show later, once the packet-level simulator has an effective SINR, it can model upper PHY functionalities accurately, such as protocol data unit (PDU) errors, HARQ operations, channel reports, link adaptation etc.

Many NS-2 researchers and developers assume that the NS-2 wireless framework already addresses all of the necessary PHY aspects, and with 802.16 PHY it is just a question of changing the existing parameters or choosing slightly different models.[13] Unfortunately, NS-2 is far from capturing all of the necessary 802.11 PHY aspects.[14] Its logical design corresponds to a single-carrier case, to which path loss, and optionally antenna and fading models are applied. In reality, both 802.11[15] and 802.16 PHY rely upon the OFDM technology with multiple sub-carriers. This means that each sub-carrier can experience a different path loss, fading, interference and so on. On the one hand, in 802.11 and in 802.16 OFDM PHY we can assume that all of the sub-carriers have exactly the same behavior, thus working with a single carrier that will represent an effective SINR. To some extent, it is a valid approach for 802.16 OFDM where a slot always maps to all of the sub-carriers in the OFDM symbol; the same holds for 802.11. On the other hand, sub-carriers may have quite different SINR values due to partially overlapping bandwidth (a typical 802.11 case), a small guard band or interfering cells (a typical 802.16 case). Furthermore, the fast fading process is different for each sub-carrier. Another important PHY aspect, which NS-2 researchers usually do not account for is the UL sub-channelization gain. Since an SS transmission power must be distributed evenly between all of the UL allocation sub-carriers, the size of the allocation has an impact on the received signal strength.

In practice, it is not very complicated to introduce sub-carriers in NS-2. They can be modeled quite easily over the existent **Channel** class without even changing the core framework. On top of that, one can add path loss, antenna pattern, shadowing/slow fading, fast fading, etc. All of the related models and algorithms are well known and defined.[16,17] The problem is that we need the effective SINR that is calculated based on individual sub-carrier SINR values. The more sub-carries we have, the more computational resources are needed. The 802.16 OFDM PHY has 256 sub-carriers; in 802.16 OFDMa, there can be up to 2,048 sub-carriers, e.g., in a 20 MHz channel. Furthermore, in OFDMa PHY, sub-carriers can be either adjacent or distributed over the whole bandwidth. The problem becomes even more computationally expensive if we start to model multiple cells. It is not reasonable to turn NS-2 into another link-level simulator as its power is in accurate modeling of higher layers. It is also worth mentioning that interference calculation involves heavy computations. Even system-level simulators use simplifications here.

Based on these considerations, a possible approach is to obtain an effective SINR from trace files generated with dynamic system simulators. It allows for a good trade-off between computational complexity and accuracy. However, this approach has obvious limitations. The trace files are generated from a particular environment, particular node locations, particular traffic mixes, loads and so on. As an example, it is impossible to study the channel aware scheduling with SINR taken from the trace files. Another scenario where an approach with trace files fails is mobility. As an SS moves across a simulation area, path loss, slow fading and the interference level vary. Also, if relays are introduced into the simulation scenario, both the signal strength and the interference levels change.

Thus, an alternative solution is to have a simple PHY model. To keep the trade-off between accuracy and computational complexity, we calculate the effective SINR for a single slot and use the same value in all of the slots that a received PDU comprises. Furthermore, we assume the same path loss and slow fading for all of the sub-carriers in a slot, only the fast fading varies. Then, we use the exponential effective SINR mapping (EESM) given by

$$\mathrm{SINR_{eff}} = -\beta \ln\left(\frac{1}{N}\sum_{i=1}^{N} e^{-\frac{\mathrm{SINR}_i}{\beta}}\right), \tag{1}$$

where SINR_i is an individual sub-carrier SINR, N is the total number of sub-carriers, and β is an MCS-dependent tuning parameter. As mentioned above, we run EESM only for a single slot that always has 24 data sub-carriers regardless of the permutation type.

To hide implementation details and abstract the core PHY components in a particular way to obtain the effective SINR, we derive new classes from the NS-2 **Propagation** class.

3.3.3. Path loss framework.

One of the disadvantages of the NS-2 PHY framework is the fact that the same path loss model is assumed between all of the communicating nodes. This is a valid approach for simple scenarios where all of the nodes are the same by nature, e.g., an *ad-hoc* network, or all of the nodes exchange data only with a base station. Once we start to model more complicated scenarios, such as relays, it becomes evident that a more complicated framework is needed. The reason is that there might be different path loss models between a node, a relay, and a base station.

We extended the NS-2 framework in such a way that one can specify the default path loss model for all of the nodes (for the backward compatibility) and, in addition, a different path loss model for a certain communication pair. The latter solution is used in relay simulations to account for a different propagation environment between a base station and a relay.

3.3.4. Slow Fading and Fast Fading.

The slow fading or shadowing reflects the fact that certain obstacles, such as buildings, have an impact on the signal loss. It is modeled as a fixed value that is generated randomly once upon a simulation startup. Since shadowing depends on a particular location, we generate a so-called shadowing map that keeps slow fading values for the whole simulation environment. Furthermore, since an SS experiences different shadowing from different BS sites, an appropriate number of shadowing maps is created. Figure 2 shows a sample map for a small simulation area. All of the shadowing maps are kept in the **WiMAXTopography** class which is derived from the NS-2 **Topography** class.

In addition to the slow fading, one has to model the fast fading that varies in the course of time. Along with complicated models similar to that defined by 3GPP,[18] one can use a simpler Jakes model that combines efficiency and realistic behavior. Figure 3 illustrates the fast fading component generated by using the Jakes model with two different Doppler shift and K values.

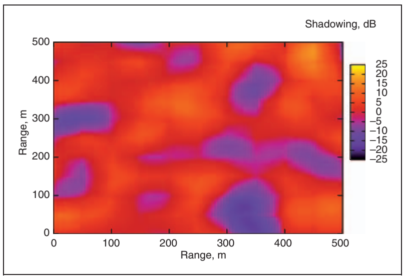


Figure 2. Slow fading variations.

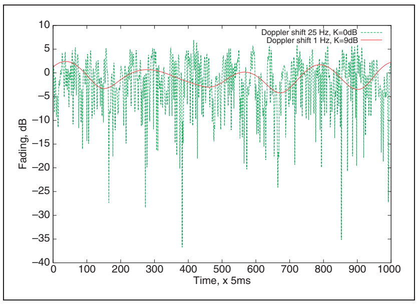


Figure 3. Fast fading variations.

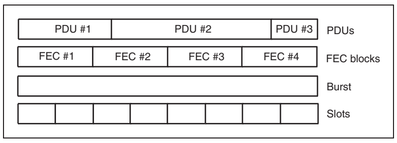


Figure 4. Slots, burst, FEC blocks, and PDUs.

As can be seen, a higher Doppler shift and non-line-of-sight (NLOS) conditions result in quite high channel variations. For each transmitter–receiver pair, a separate class instance is created that returns a different fast fading value.

3.3.5. Error generation. The error-generation model is responsible for answering a question whether a received PDU is erroneous or not. To model it, we account for the way 802.16 encodes and sends data. Each data allocation, i.e. a burst, is an integer number of contiguous slots. On top of that, slots are grouped into forward error correction (FEC) blocks, where the FEC block size depends on a particular MCS. A PDU can start and end on any *byte* within a data burst. All of these layers are presented in Figure 4. Thus, to model errors correctly, we have to map a received PDU to the FEC blocks it spans.

To accomplish a correct PDU to FEC block mapping, we instruct the PHY layer to build a FEC block list whenever a new burst starts. Then, inside the **sendUp()** function, we track the number of received bytes so that whenever a new PDU arrives, we can determine the starting and ending FEC block. Once the list of FEC blocks, to which the received PDU maps, is known, it is passed to the error-generation module. Each FEC block carries information on its size and SINR. The error generator applies the following formula

$$\mathbf{E} = 1 - \prod_i (1 - E_i), \qquad (2)$$

where E_i is an individual FEC block error probability determined based on the FEC BLER curves, as Figure 5 shows.

It is important to note that FEC BLER curves presented in Figure 5 are not hardcoded but rather specified at the OTcl level. This allows easy switching between different link-level simulation results and coding schemes, e.g., convolutional coding and convolutional turbo coding.

3.3.6. HARQ. The HARQ Type I, i.e. chase combining (CC), implementation and modeling follows the 802.16 simulation methodology.[16] Every time a new HARQ retransmission is made, the FEC block SINR from all of the previous (re)transmissions is summed and submitted to the error-generation module considered earlier. The HARQ Type II, incremental redundancy (IR), is more complicated to model with NS-2 without going into coding and decoding details. One approach is to model it on top of Type I as a positive or negative gain based on retransmissions' SINR.[19–21] Another approach is to have different FEC BLER curves for the first retransmission, for the second, and so on.[22] Anyway, HARQ IR mode is not so important for WiMAX networks as it is not mandated by the system profile.[23]

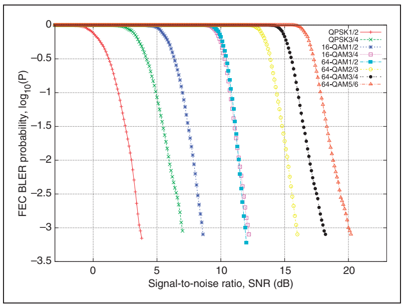


Figure 5. CTC FEC BLER curves (only the largest FEC block sizes).

In the case of UL HARQ transmission, the BS always knows the burst reception status. In the case of DL HARQ burst, an SS reports back the HARQ status via the HARQ ACK channel. Once the BS scheduler knows the burst reception status, it can decide whether to schedule a HARQ retransmission or continue with allocating data on the next free HARQ channels.

Our HARQ implementation fully conforms to the 802.16 specification in a sense of supported and adjustable parameters. It is possible to specify the maximum number of HARQ channels (16 by default), maximum number of HARQ retransmissions (4 by default), HARQ buffer mode (shared by default), and UL HARQ ACK delay (1 frame by default).

3.3.7. Repetition factors. It is quite crucial to model the repetition factors defined in the 802.16 specification. The reason for this is that quite a few DL management messages use MCSs with a repetition factor to ensure a sufficient cell edge performance. Otherwise, an SS may be out of service. The repetition factor is modeled similarly to HARQ Type I. The received packet power is just multiplied by the repetition factor and then passed to the error-generation model.

3.3.8. Channel Reports and Link Adaptation. When the BS receives data in UL, it can always estimate the channel to switch to a more suitable UL MCS in the next frame. When an SS receives data, it has to estimate the channel and report it back to the BS so that the BS link adaptation can also choose a suitable DL MCS. We support two reporting mechanisms: REP-RSP messages and the channel quality indication channel (CQICH). While the former is available in both OFDM and OFDMa PHY, it is less reliable because REP-RSP is an ordinary MAC-level management message that can be dropped easily. The CQICH channel, which is defined only for OFDMa PHY, provides a more robust way to report channel status.

The implementation of the REP-RSP message is quite straightforward. Six-bit CQICH messages are modeled with a special NS-2 packet type, the payload of which carries the necessary information.

Once the BS scheduler link adaptation module (see Figure 1) has information on all of the SSs DL and UL SINRs, it can adjust MCS to achieve the target FEC BLER for each connection. Refer to Section 4.7 for more information.

3.4. MAC layer

3.4.1. Queue system. The general structure of the queue system is presented in Figure 6. From the BS point of view, the air interface is a bottleneck which creates a need to buffer DL packets. Similarly, an SS needs queues to buffer UL packets. To differentiate between service flows and ensure QoS, each connection is allocated a separate queue. In addition, the BS keeps so-called UL virtual queues maintained through the bandwidth requests transmitted by SSs.

Each transport connection is equipped with several internal sub-queues where it keeps initial transmissions, retransmissions, and ARQ feedbacks. The internal priority queuing (PQ) scheduler ensures that first a connection will send ARQ feedbacks, then retransmissions, and only then normal PDUs. By default, a sub-queue for initial transmissions relies upon a simple drop tail method. Alternatively, some Active Queue Management (AQM) mechanism can be applied there. Refer to Section 4.3 for more information on AQM in 802.16.

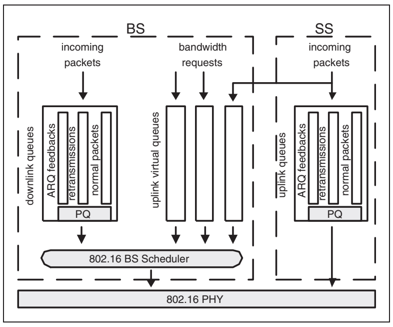


Figure 6. The queue system.

At the BS side, there is also a special DL queue where the scheduler puts generated DL-MAP, UL-MAP, DCD, and UCD messages. In addition, certain management messages, such as MOB_TRF-IND, are also placed into this queue, as they are designated to all of the SSs.

3.4.2. Support for duplexing modes. To support different duplexing modes, e.g., time-division duplexing (TDD), full frequency division duplexing (F-FDD) and half frequency division duplexing (H-FDD), we developed a scalable design for the MAC-level timers. Figure 7 shows that there is a top-level timer that elapses whenever a new frame should start. Then, a burst timer ensures a transition from one burst to another. Whenever a new burst starts, the MAC level prepares PHY and resets certain MAC-level state variables. The third level is a PDU timer that elapses whenever data transmission (or reception) ends.

Figure 7 shows a simplified timer functioning for the TDD mode where DL bursts are followed by UL bursts. A transition from the DL sub-frame to the UL sub-frame is done through the sub-frame timer that elapses when the UL sub-frame starts. In the FDD mode, two burst timers function in parallel to support simultaneous data transmission and reception. Of course, the sub-frame timer is not started at all. The burst and PDU timers abstract the core MAC functionality from a particular duplexing mode.

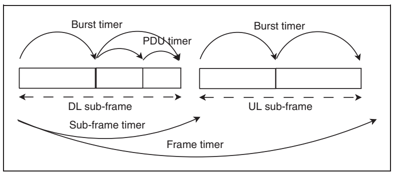


Figure 7. MAC-level timers.

3.4.3. Packet header suppression. The packet header suppression is implemented in a simple way. A script designer just specifies a constant suppression size. Whenever a packet is placed in the queue, its size is decremented to emulate suppression. Then, when a packet is passed to LL, the same constant size is added back. This functionality can be enabled on the per-connection basis thus providing a different number of bytes to suppress, which may depend on the traffic type.

3.4.4. Packing and fragmentation.

While implementing packing and fragmentation, we faced the biggest problems with the NS-2 framework. There is no internal support for such a simple operation as fragmentation. This functionality must be implemented on top of the NS-2 **Packet** class. Even worse is that a bad C++ design of the **Packet** class prevents from applying virtualization and abstraction. Packing caused even bigger problems as it implies that several NS-2 packets, or even packet fragments, are transmitted as one entity. Our solution for packing is to have a NS-2 packet with a special payload that keeps all of the embedded packets or packet fragments.

3.4.5. ARQ mechanism.

The ARQ mechanism does not contain any NS-2 specific features. Since it works on top of MAC, it is a straightforward implementation from the 802.16 specification. Once the MAC level detects missing ARQ blocks, it activates the ARQ mechanism and informs the sender to retransmit data. ARQ timers take care of situations when ARQ feedbacks are lost constantly or when retransmission attempts expire.

It makes sense to mention only one important optimization that we use in the ARQ implementation. Instead of creating and running ARQ timers for each ARQ block that a PDU has, we associate them with the whole PDU. This is an approach that also certain WiMAX products use. Of course, special care must be taken later when a retransmitted PDU is fragmented or packed.

The ARQ mechanism implementation supports the following features: ARQ blocks, ARQ block rearrangement, ARQ feedbacks (standalone and piggy-backed), ARQ window, ARQ timers (retry, block lifetime, Rx purge), and ARQ discard.

3.4.6. Contention resolution.

Depending on the underlying PHY, the 802.16 contention resolution works differently. Even though the top-level backoff mechanism is identical to both OFDM and OFDMa PHY, the former just sends a 6-byte PDU header with the bandwidth request size. In OFDMa PHY, there are 256 144-bit pseudo-orthogonal code division multiple access (CDMA) codes. An SS sends first a CDMA contention code; once the BS detects the code, it allocates a special uplink CDMA allocation where an SS can send a bandwidth request. All of these differences are abstracted through the MAC-level contention resolution class.

The collision in OFDM PHY is detected and handled quite easily: if the BS detects transmission and the previous transmission has not yet ended, then all of the packets are dropped. The OFDMa PHY uplink contention is trickier to model in NS-2 because it is a tradeoff between accuracy and computational complexity. When several SSs transmit CDMA codes simultaneously, the BS tries to detect each of the transmitted codes. Several approaches for modeling a CDMA receiver in NS-2 are available.

1. *Optimistic*. A CDMA code is modeled with a special NS-2 packet type, where the packet payload carries just the CDMA code index. Since CDMA codes are pseudo-orthogonal, we can assume that there is a high probability that all of the transmitted codes are detected. The only case when a code collision occurs is when two or more identical codes are sent. Thus, the BS CDMA receiver just analyzes CDMA code indexes to decide whether they are detected or not.
2. *Simple*. This receiver is similar to the previous one with one addition: it tries to account for inter-code interference based on the link simulation or other simulation results.[24] Even if two or more received codes have different indexes, there is a chance that a code is not detected.
3. *Single-code correlator*. This approach tries to model the way the CDMA codes are transmitted and detected by the CDMA receiver. The NS-2 CDMA packet payload carries 144 bits. If there are several codes sent during the same transmission opportunity, then the received bit sequences are summed to obtain a so-called interference pattern. Since each bit is binary phase shift keying (BPSK) modulated, there is a non-zero probability that its value changes during transmission. Thus, a receiver models it by applying the BPSK error curve to each bit. Then, the BS CDMA receiver tries to detect individual codes by applying the dot product operation and comparing the correlator output to the predefined threshold, e.g., 75%.[25] Of course, this approach does not account for the fact that codes may arrive with different power levels. On the other hand, a properly functioning UL power control algorithm tries to ensure that received signal strength from all of the SSs is approximately at the same level.

Depending on the simulation accuracy and speed, an appropriate CDMA receiver can be chosen. As an example, an accurate voice over IP (VoIP) delay analysis, where uplink ertPS connections rely upon the contention, definitely requires a complicated CDMA receiver. On the other hand, a simple CDMA receiver suffices for a basic analysis of the TCP performance over 802.16.

3.4.7. Scanning and handover. In a simple single-sector simulation, there is no need for the scanning functionality as there is only one BS which an SS will exchange data with. In this case, the SS is provided with an explicit BS ID where the network entry must be performed. Otherwise, in multi-cell or relay simulations, an SS first listens for a DL channel and then enters the network via the BS that has the strongest DL signal. The scanning time is a configurable parameter; usually, 50–100 ms is enough to filter out the fast fading component and obtain an average DL channel performance.

Handover is a part of the basic MAC functionality and includes support for all of the necessary management messages to initiate and control the handover process. Our implementation supports two basic handover methods: manual and automatic. In the former case, at a particular moment of time an SS is instructed to handover to the specified BS. In a case of automatic handover, an SS measures periodically DL preambles from neighboring BSs and initiates a handover process once the strongest neighboring BS signal strength exceeds the *hysteresis* margin and stays for more than *time-to-trigger* seconds. If for some reason the handover fails, an SS falls back to the initial network entry mode and starts to perform scanning. This allows us to simulate large mobility scenarios containing multiple BSs and random movement of SSs.

3.5. LL layer

The LL layer has the fewest extensions when compared with MAC or PHY. Following the general principles, there is a derived class that changes the behavior of the virtual **recv()** function. Firstly, we disable Address Resolution Protocol (ARP) because it does not make much sense in the 802.16 PMP mode. Second, the LL layer has to stamp a packet with a correct CID so that the queue system can place it into a correct queue. Practically, the NS-2 LL layer performs functions of the 802.16 convergence sublayer. This functionality is identical for an SS and BS as both of them have to classify incoming packets. The current implementation classifies packets based on the following parameters: (a) source address, (b) destination address, and (c) flow ID. The reason we also account for flow ID is that we have to differentiate between incoming packets that belong to different applications from the same node. If they belong to different application types, e.g. VoIP and BE, then they should be placed into different connection queues to obtain an appropriate treatment. On the other hand, by not specifying the flow ID we will put all of the incoming packets into one connection queue, which is also a valid case. A script designer decides which option to use.

3.6. QoS and scheduling

The internal QoS architecture of our 802.16 module does not differ significantly from QoS architectures presented in other papers and 802.16 implementations.[7,26] Thus, for the sake of brevity, we focus only on the distinctive features that we adopted in our module.

3.6.1. BS scheduler. We omit the description of the BS scheduler details as more information is given in Section 4.1. However, we mention briefly that to study different schedulers in 802.16, we introduced a common interface between the BS MAC and the BS scheduler, as shown in Table 2.

The input parameters are the status of the DL physical and UL virtual queues (see Figure 6) that are maintained and managed by the MAC layer. In addition, we also pass HARQ ACKs because they arrive at the BS via the MAC level. The result of the scheduling decision is two lists with DL and UL bursts for the BS MAC. The scheduler also constructs the DL-MAP, UL-MAP, DCD, and UCD messages that the BS MAC transmits later to all of the SSs.

3.6.2. SS uplink scheduler. The SS uplink scheduler is as important as the BS scheduler. The reason is that once the BS makes an UL allocation, it is per a whole SS, not per an individual connection. Then, it is the SS responsibility to partition this UL allocation between multiple transport connections, if any. Such a solution aims at reducing the signaling overhead and allowing an SS to gain a better control on the allocation size, which can start and end at any byte within the burst. Figure 8 shows an example of how an UL burst can be shared between several connection types.

The default uplink scheduler in our implementation is PQ. Its simple yet efficient design allows for supporting triple-play services at mobile terminals. In other words, first the uplink scheduler allocates space for the management data, if any, then UGS, then ertPS

Table 2. BS scheduler interface

Input	Output
DL queue sizes	DL burst list
UL queue sizes	UL burst list
HARQ ACKs	DL-MAP, UL-MAP, DCD, UCD

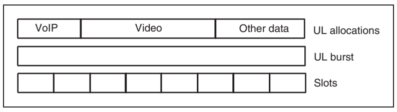


Figure 8. UL burst partitioning.

Table 3. Uplink scheduler interface

Input	Output
Time	UL allocation list
UL allocation size	
UL queue sizes	

(e.g., VoIP), then rtPS (e.g. video), and then nrtPS (e.g. Web) with BE.

Of course, the PQ scheduler cannot address more complicated and general cases when, as an example, there are several nrtPS connections with diverse QoS requirements, such as the 802.16 CPE that serves a small local network. To experiment with different uplink schedulers, we developed a common interface that allows abstracting from a particular allocation algorithm (see Table 3). The input parameters are time, UL allocation size, and the queue sizes. Output information consists of a list with UL allocation sizes that tell how many bytes each connection may occupy. It is the MAC-level responsibility to enforce this decision.

3.6.3. Uplink power control.

The uplink power control mechanism is as important as the link adaptation and the scheduling. On the one hand, it is possible to avoid implementing the uplink power control by assuming that an SS uses a certain fixed transmission power. On the other hand, since an SS has a limited transmission power, the BS can instruct the SS to increase or decrease its power to ensure efficient functioning at both the cell edge and cell center.

Since the uplink power control algorithm is not defined by the IEEE 802.16 specification, we provide here details of a simple closed-loop algorithm that works in a coordinated way with the BS scheduler. There are a few basic principles:

1. Ensure that the uplink signal-to-noise ratio (SNR) is higher than QPSK1/2 by increasing SS transmission power. While increasing the transmission power, make sure that the SS power budget is not exceeded. Furthermore, the algorithm also ensures that an SS can always transmit at least within two sub-channels.
2. Lower SS transmission power if the uplink SNR is higher than that needed to support the highest uplink MCS, e.g. 64-QAM 5/6. It allows an SS to occupy more UL sub-channels and thus transmit more data.

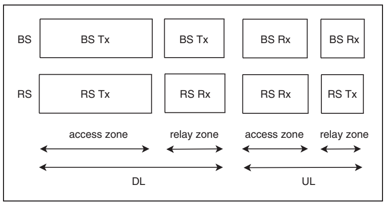


Figure 9. TDD frame with access and relay zones.

In addition, the BS scheduler provides hints to the power control module regarding the UL allocation size. In particular, the power control module tries to decrease the transmission power if it notices that the scheduler tends to allocate larger UL data grants. On the other hand, it increases the transmission power if an SS can transmit within more sub-channels than a typical UL data grant occupies. However, the aforementioned principles always override the decision made that is based on a hint from the BS scheduler.

3.7. 802.16j multi-hop extensions

Multi-hop extensions[27] were added recently to the baseline 802.16e specification and allow for deployment of various relay types with different operational modes. In this paper we refrain from describing all of the 802.16j technical details, a good overview of the key 802.16j features can be found in Peters and Heath's work on Multihop relaying.[28]

The key concept behind implementing a scalable support for the multi-hop functionality is a zone. Being part of the basic 802.16 specification, it allows the DL/UL sub-frame to be partitioned into smaller logical parts by using the time division concept. Originally, this approach was introduced to support different permutation types within the same sub-frame, e.g., PUSC, FUSC, AMC. In the 802.16j extensions, a zone also specifies whether it is used for communication with SSs or with relay stations (RSs). Figure 9 shows the TDD frame with access and relay zones in the DL and UL sub-frames.

The WINSE module supports two-hop non-transparent time-division transmit relays as a main solution for the coverage extension problem.[29] Transparent relays, which are used merely for the throughput enhancement within a cell coverage area, are left for the next stage. Even though the specification allows for more than two hops, such a possibility is not supported due to the resulting system complexity;[30] the same approach is also taken in other technologies, such as 3GPP LTE-A relays.[31] As a result, there is no need to support more than one DL/UL relay zone in the 802.16 frame. Further WINSE development may include a support for an unlimited number of hops.

3.7.1. PHY, MAC, queue system.

Since the 802.16j specification does not define any new PHY features, we rely upon the PHY concepts and principles described earlier. The only enhancement we introduced is a set of propagation models defined by the 802.16j simulation methodology.[32] All of the other PHY mechanisms, such as channel measurements, reporting, and link adaptation, are exactly the same as for SS and BS.

Obviously, the MAC level has the largest number of extensions. Owing to the scalable architecture, the RS MAC level is a combination of the BS and SS functionality, as Figure 1 illustrates. Indeed, RS behaves like a full-featured BS while scheduling resources for its access link, sending and receiving data there. At the same time, the RS node behaves like an SS when it follows the scheduling decision made by BS on the relay link.

The queue system is identical to the MAC-level queue with one small extension: it is capable of buffering both the DL and UL PDUs, because depending on the channel conditions RS may have to buffer both DL and UL traffic.

3.7.2. Scheduling.

Scheduling was one of the most challenging tasks while implementing support for 802.16j. In addition to providing resources for SSs, as in a basic 802.16 network, the BS has to allocate slots on a relay link to exchange data with RSs. To accomplish this task, we introduced the concept of *scheduling group*, where each group is an independent set of resources managed by BS. Referring back to Figure 9, DL and UL relay zones form a separate scheduling group, whereas DL and UL resources for an access link belong to a different scheduling group.

In addition to the DL and UL scheduler that assigns resources on the RS access link, the RS node has to run a so-called UL allocation scheduler. Similarly to SS, BS grants an UL allocation to a whole RS by using its basic management CID; it is RS responsibility to share it between all of the associated stations. In Section 3.6.2, we presented the architecture of the SS UL scheduler and described a simple resource allocation mechanism based on PQ. Even though it suffices for an SS, it is not flexible enough for the RS node. The biggest obstacle is that due to the PQ nature, one BE TCP connection can easily monopolize all of the resources. Instead, we implemented a PQ-RR scheduler that prioritizes management, UGS and ertPS connections by using strict priorities. Connections that belong to the remaining classes, i.e. rtPS, nrtPS, BE, are shared with a round-robin scheduler. Such a scheme combines simplicity and efficiency assuming that the BS UL scheduler accounts for the QoS requirements of the RS connections and allocates sufficiently large UL grants.

3.8. Routing

While working in the PMP mode, packets are always exchanged between SSs and the BS belonging to the same cell: there is no way an SS can communicate with another one bypassing the BS. As a result, there is no need for an *ad-hoc* routing protocol because all of the forwarding information becomes available during a connection set up. Thus, we rely upon the NOAH module that just disables *ad-hoc* routing.

3.9. Tracing

Following one of the core principles, we rely upon the trace format provided by the NS-2 framework. We do not change the trace file format as it will break existent scripts or start to conflict with other extensions. However, there is 802.16-specific information that one would like to see in the trace file for further analysis and testing.

3.9.1. 802.16-specific MAC fields.

The NS-2 wireless trace file format defines four `-Mxxx` fields, the content of which is specific to a particular MAC level. We redefine them to include 802.16-specific information, as shown in Table 4. The `-Mt` field always have the same value, 802.16, which allows differentiation from other MAC types. The `-Mc` field keeps the CID value. It helps to differentiate between several connections belonging to the same SS or just track down a particular connection. The `-Mm` field specifies the management message type if a transmitted PDU belongs to the management connection. Having the management message type in the trace file, one can gather plenty of important information. As examples, there are network entry delay (the time between the first RNG-REQ and the last

Table 4. New trace fields

Field	Description
–Mt	802.16
–Mc	CID
–Mm	management message type
–Mb	bandwidth request size

REG-RSP messages), connection setup delay (the time between the first DSA-REQ and the last DSA-ACK messages), handover delay (the time between the first MOB_MSHO-REQ and the last REG-RSP messages), and so on. The -Mb field specifies the bandwidth request size of a standalone or piggy-backed bandwidth request.

3.9.2. Burst entry. Even though the NS-2 trace file contains information on all of the transmitted PDUs, sometimes it is not enough to analyze the resulting 802.16 behavior. First, the NS-2 format does not provide enough -Mxxx fields to display other useful information, such as MCS or the HARQ (re-)transmission status. Second, 802.16 data transmission occurs in a form of a data burst that comprises one or more PDUs. Thus, we also add a support for a so-called "burst" entry.

Figure 10 shows the format of the "burst" entry that appears in the trace file whenever a new burst starts. The entry includes information on a burst direction, its index, type, e.g. data burst or a contention region, CID and ACID values, burst MCS, size in slots and bytes, as well as the HARQ status for the HARQ-enabled data burst. It allows statistics to be gathered, such as MCS distribution, burst size distribution, distribution of HARQ retransmissions, etc.

3.9.3. Contention entry. We add a special "c" entry to the trace file. It specifies when an SS takes part in the uplink contention resolution. The reason we need this entry is that when the SS starts the uplink contention, it may defer for a number of frames due to the backoff start value. As a result, the first bandwidth request is sent after the actual uplink contention has begun. Thus, the "c" entry helps to measure the medium access delay accurately. Since the SS performs the uplink contention on behalf of all of the connections it has, the format of this entry is very simple, as shown in Figure 11. There is time when a contention starts, SS node ID that originates the contention, and BS node ID to which the contention request is sent.

3.9.4. Drop reason. There are many reasons why a packet can be dropped inside the MAC layer. We rely upon the existing NS-2 drop reasons (see Table 5) to put appropriate information in the trace file.

```
b -t <time> -Hs <bs_id>
-Bd <direction> -Bi <index> -Bt <type>
-Bc <cid> -Ba <acid>
-Bm <MCS> -Bs <size_slots> -Bb <size_bytes>
-Bh <HARQ_status>
```

Figure 10. Format of the burst entry.

```
c -t <time> -Hs <ss_id> -Hd <bs_id>
```

Figure 11. Format of the contention entry.

Table 5. Drop reasons

Drop reason	Description
IFQ	The queue is full
ERR	Packet error
COL	Uplink contention collision (OFDM only)
RET	Contention attempts exceeded
NRTE	No classification rule

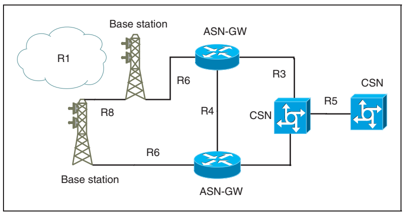


Figure 12. WiMAX access network components.

3.10. Access network

Along with the 802.16 radio interfaces, the performance of the WiMAX network is affected by the wired part. In fact, wired network components, such as access service network gateway (ASN-GW) and connectivity service network (CSN) node (see Figure 12), may play a crucial role during network entry, connection setup, and handover processes because the BS contacts them at various stages.[33] Failing to model the access network, one can obtain too optimistic results. Apart from time needed to send a signaling message and wait for a response, there are also delays that may come from the user data preempting management traffic inside the access network.

Access network modeling combined with 802.16 PHY and MAC is a perfect task for a simulator, such as NS-2, where wired networking has been present for a long time. In addition, NS-2 provides a good framework to develop new protocols, in particular signaling protocols.

We already developed an extension to the WINSE module that aims at adding support for the 802.16 access network entities in the NS-2 simulator. Our primary goal is to support R4, R6, and R8 interfaces to obtain more accurate results for network entry and handover procedures. The description of the module is given by Mendieta[34] with simulation results presented by Hytönen et al.[35]

4. Research done with WINSE

In this section we give a brief overview of past and ongoing research topics where the WINSE module is used.

4.1. Scheduling and resource allocation

Scheduling and resource allocation was the first research topic studied with the WINSE extension.[36,37] It was shown that the classical fair resource allocation is applicable but is too complicated for the 802.16 networks, where the basic allocation unit is a slot of a fixed size and duration. Nevertheless, a simpler yet efficient algorithm is possible that is based conceptually on the round-robin approach. At the same time, there were proposed extensions that allowed the 802.16 service class, QoS parameters, UL bandwidth request or DL queue size to be accounted for. The simulation results showed that the proposed scheme satisfies all of the QoS requirements, protects service flows, and ensures fair resource allocation across the BE connections. All of our subsequent research papers relied upon the proposed scheduler that was adopted easily for the OFDMa PHY. Its simple but scalable nature also allowed introducing an extension for ARQ[38] and a support for the HARQ scheduler.

Owing to a highly modular internal architecture, other schedulers, such as proportional fair, were studied with the WINSE module.[39]

4.2. Uplink contention performance

Even though the 802.16 system is based on the demand assigned multiple access (DAMA) concept, there is still an uplink contention created by connections that are either not polled regularly, e.g., nrtPS, or are not polled at all, such as BE. The ertPS class can also take part in the uplink contention if so allowed by the BS. Previously,[40] we studied the 802.16 uplink contention resolution and proposed a scheme about how to adapt dynamically backoff start/end parameters, and the number of the contention transmission opportunities. Furthermore, we also showed[41] that by adjusting dynamically the uplink contention parameters one can achieve a good tradeoff between resource utilization and delay requirements of the VoIP connections.

Another 802.16 feature to achieve the tradeoff between delay guarantees and resource utilization is multicast polling. A study was performed[42] to research the applicability of it with VoIP connections. It was shown that the multicast polling can be used to provide a maximum delay guarantee and even several separate delay limits for separate connection types. As a continuation of the uplink contention and VoIP topics, we also compared different resumption mechanisms in OFDMa PHY.[43]

Our recent paper studied the OFDMa PHY CDMA contention code performance and how they can be optimized for future broadband wireless systems.[24]

4.3. AQM

From the SS and BS point of view, the 802.16 network is a bottleneck because it is highly anticipated that a wired connection behind BS (or SS working as a gateway) will always be higher than the maximum achievable throughput on the air interface. Thus, the queue sizes tend to grow, especially when spectrum efficiency declines. We studied[44] AQM mechanisms in which it is possible to apply to 802.16 to reduce queuing delays. The results showed that when applied to the BS DL queues, the AQM mechanism is capable of reducing TCP delays.

4.4. Optimal PDU size

While transmitting data, incoming packets, i.e. SDUs, are fragmented or packed into PDUs, the size of which is not governed by the specification. On the one hand, the probability of erroneous MAC PDU increases when the PDU size grows. On the other hand, a small PDU has a larger overhead. We studied[45] the optimal MAC-level PDU size under different channel conditions. We showed that it is possible to find the optimal PDU size if we know the channel conditions. Because the 802.16 system has an advanced link adaptation mechanism, the error probability is known and the optimal PDU size can be selected. It was also shown that if the error probability is unknown, then it is better to rely on smaller PDU sizes of around 100–120 bytes.

4.5. ARQ

The ARQ retransmission mechanism is available in all of the major PHYs of the 802.16 system and plays a key role in improving the system performance, especially the application-level throughput. We previously studied[46] the properties of the 802.16 ARQ mechanism and proposed solutions on how to choose an ARQ feedback type, and how to prioritize feedbacks, retransmissions, and normal PDUs. We showed the importance of the ARQ block rearrangement and a correctly set up ARQ transmission window. We also analyzed the impact of the ARQ feedback intensity in the case when ARQ works standalone and on top of HARQ.[47]

We also studied[48] the performance of the ARQ mechanism with the real-time VoIP traffic. We argued that even though HARQ is considered to be a better candidate for VoIP, it is possible to tune the ARQ algorithm so that all of the VoIP performance requirements are met.

4.6. ARQ and HARQ performance

Along with ARQ, the 802.16 OFDMa PHY provides a possibility to run the HARQ retransmission mechanism. We previously[49] made a preliminary comparison of these two mechanisms and also studied ARQ on top of HARQ. Even though, as expected, HARQ outperforms ARQ in most cases due to a retransmission gain, there are cases when ARQ provides a better performance due to less signaling information.

4.7. Link adaptation thresholds

We have analyzed the link adaptation model and MCS transition thresholds for the IEEE 802.16 BS.[50] We have shown that the optimal transition threshold for the ARQ connections is between 10^{-2} and $10^{-2.5}$, while for the HARQ-enabled connections it is from $10^{-1.5}$ to 10^{-2}. It fully conforms to the theoretical expectations that HARQ should outperform ARQ due to the retransmission gain. An interesting outcome of the paper is that the optimal thresholds depend on the number of data bursts per frame. It requires a coordinated function between the BS link adaptation model and the scheduler.

4.8. Sub-MAPs

One of the performance bottlenecks of the 802.16 system is DL broadcast control MAP messages that specify data grants in the DL and UL sub-frames. To reduce the signalling overhead, the 802.16 standard introduces so-called sub-MAPs. However, similar to the scheduling, the algorithm to build sub-MAPs is not defined by the specification. We proposed[51] an efficient and computationally friendly algorithm to build sub-MAPs and presented that by means of extensive simulations that sub-MAPs can improve significantly the spectral efficiency of the system. We analyzed[52] the impact of sub-MAPs on the VoIP capacity. According to the results, the VoIP capacity can be improved up to 100%.

5. Ongoing research topics

In this section we provide some simulation results of our ongoing research topics, which are a comparison of different duplexing modes in 802.16 and 802.16j relay performance.

5.1. TDD versus F-FDD versus H-FDD

In this section we present simulation results for different duplexing modes that our simulator supports. We consider the TDD and FDD, where the frequency division is further classified into full mode and half-duplex modes. In the half-duplex mode, the 802.16 DL/UL sub-frames are partitioned into two groups to support stations that cannot transmit and receive simultaneously.

In the simulation scenario, there is a single BS sector and 32 SS that are placed randomly in the BS serving area. There are two separate simulation cases. In the DL case, each SS downloads over the FTP BE TCP connection and in the UL case, all the SSs upload data. In each simulation case, we vary the number of bursts per sub-frame, both DL and UL, to present how it impacts the final system performance. Each simulation lasts for 10 seconds and is run 12 times with a different random seed value.

Table 6 presents the PHY parameters for this simulation. It is worth mentioning that the 10 MHz frequency band used in TDD is partitioned into two 5 MHz bands used by DL and UL carriers in FDD. Further, depending on the FDD mode, either all of the OFDM symbols are used for transmission or they are partitioned into two groups in the H-FDD mode. The TDD DL/UL ratio is chosen in such a way that the number of slots for DL and UL sub-frames is as close to FDD as possible. Also note that the uplink contention resolution parameters are slightly adjusted for H-FDD to account for the fact roughly two times less SSs reside in each H-FDD contention group. Table 7 presents the common MAC-level parameters that do not depend on a particular duplexing mode.

Figure 13 shows the DL and UL spectral efficiency for the considered duplexing modes. It can be seen that in the DL case the spectral efficiency decreases as the number of bursts grows. This is explained by the increased MAP message overhead when more bursts are scheduled per single sub-frame. Also H-FDD always has a larger overhead when compared with the other duplexing modes, which explains its worse spectral efficiency.

In the UL direction, the FDD duplexing mode is as good as TDD and even better. The reason for this is the better UL sub-channelization gain for FDD. Owing to a longer UL sub-frame size in FDD, an SS can transmit more data in one sub-channel when compared with TDD. In addition, an SS can focus its transmission

Table 6. TDD/F-FDD/H-FDD PHY parameters

	Duplexing modes		
Parameter	TDD	F-FDD	H-FDD
Center frequency	2.5 GHz		
PHY	OFDMa		
Cyclic prefix length	1/8		
Frames per second	200 (5 ms/frame)		
Long preamble	1 symbol		
Bandwidth	10 MHz	5+5 MHz	
FFT	1,024	512	
TTG+RTG	296+168 PS	0+168 PS	
DL/UL subchannels	30/35	15/17	
DL/UL subcarrier alloc.	DL PUSC/UL PUSC		
OFDM symbols	47		
DL/UL symbols	22/24	46/45	18+28/27+18
DL slots	330	345	135+210
UL slots	280	255	102+153
BS/SS Tx power	10/0.25 W		
BS/SS antenna	3GPP/omni		
BS/SS antenna gain	17/0 dBi		
BS/SS antenna height	32/1.5 m		
Path loss	.16m UMa		
Ranging backoff start/end	1/15		0/15
Ranging transm. opport.	2		1
Request backoff start/end	3/15		2/15
Request transm. opport.	8		4

Table 7. Common MAC parameters

Parameter	Value
DL/UL channel measurements	preamble/data burst
Channel report type/interval	CQICH/20 ms
Channel measurements filter	EWMA, $\alpha = 0.25$
UL Power Control	Closed loop
Link adaptation model	target FEC BLER, 10^{-1}
H-FDD group balancing algorithm	Adaptive fair
H-FDD group balancing interval	500 ms
MAP MCS	QPSK1_2
Compressed MAP	ON
CDMA codes	256
Ranging+periodic ranging	64
Bandwidth request	192
Handover	—
Fragmentation/packing	ON
PDU size	140 B
CRC	ON
ARQ feedback	Standalone
ARQ feedback types	all
ARQ feedback interval	20 ms
ARQ block size	16 B
ARQ window	1,024
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ timers	
Retry	100 ms
Block lifetime	500 ms
Rx purge	500 ms

power on a fewer channels, thus having a more efficient MCS. This is seen clearly from the spectral efficiency figure. In fact, all of the duplexing modes improve as the number of bursts increases, but F-FDD is performing better. The reason why H-FDD outperforms F-FDD and TDD with four and eight bursts per frame is because the BS scheduler is less flexible in assigning resources for SSs that are associated with different H-FDD groups. This increases the spectral efficiency at the cost of fairness. More simulation results with a description of the H-FDD group balancing algorithm can be found in Martikainen.[53]

5.2. 802.16j Non-transparent relays

Figure 14 shows the relay simulation scenario. It is assumed that there is a single BS controlling its sector. To serve an area denoted by a dashed line, an operator may deploy additional BSs to cover two more sectors denoted by dotted lines. However, a more cost efficient solution might be to deploy a few relay nodes,[54] as shown in the figure. For the sake of brevity, we compare a case with a single BS and a case with the BS and RS nodes.

Table 8 presents the key parameters used in the simulation. We consider DL TCP transmission over the BE connections as it is a good way to analyze the resulting system throughput and the spectral efficiency. It is worth mentioning that to study the relay performance, we consider different *fixed* DL relay zone sizes, i.e. two, four, and six OFDM symbols. The UL zone size is also fixed and has the constant size of three symbols. Unlike the BS, RS uses an omnidirectional antenna, has a lower Tx power of 5 W, a smaller antenna height and gain. We assume the sub-urban macro-cell scenario and thus choose the 802.16m SMa propagation model for an access link and 802.16j TypeD model for the relay link. The latter is used for the above rooftop line-of-sight communication between a RS and a BS. The retransmission mechanism is ARQ working in the end-to-end mode. In other words, RS does not take part in the ARQ signaling and just forwards received data. The interference modeling accounts for the fact that the BS

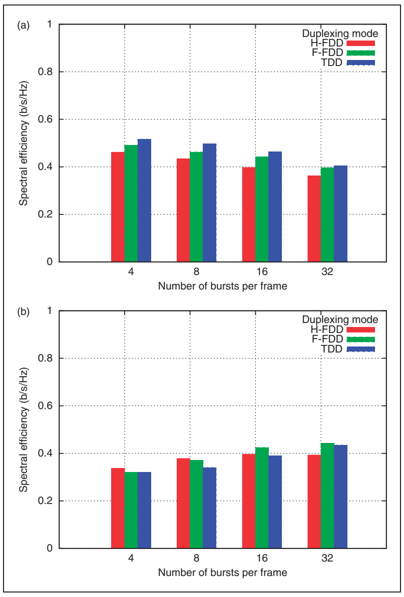


Figure 13. Spectral efficiency (TDD bars are scaled to match the number of slots count in H-FDD and F-FDD): (a) DL; (b) UL.

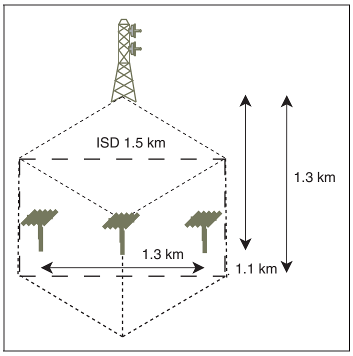


Figure 14. Simulation scenario.

Table 8. 802.16 network parameters

Parameter	Value
Center frequency	2.5 GHz
PHY	OFDMa
Bandwidth	10 MHz
Duplexing mode	TDD
Frame duration	5 ms
CP length	1/8 symbol
TTG+RTG	296+168 PS
OFDM symbols	47
DL/UL symbols	30/15
DL/UL relay zone size	2, 4, 6/3 symbols
DL/UL subcarrier alloc.	DL PUSC/UL PUSC
Channel report type/interval	CQICH/20 ms
Channel measurements DL/UL	preamble/data burst
Channel measurements filter	EWMA, $\alpha = 0.25$
MAP MCS	QPSK1/2 Rep6. . . QPSK1/2
Compressed MAPs	ON
sub-MAPs	ON, maximum 3
BS/RS/SS Tx power	10/5/0.25 W
BS/RS/SS antenna	3GPP/omni/omni
BS/RS/SS antenna height	32/12/1.5 m
BS/RS/SS antenna gain	17/5/0 dBi
A-link/R-link path loss	.16m SMa/.16j TypeD
A-link/R-link fast fading K factor	0/9 dB
Ranging transm. opport.	2
Ranging backoff start/end	1/15
Request transm. opport.	8
Request backoff start/end	3/15
CDMA codes	256
Ranging+periodic ranging	64
Bandwidth request	192
Handover	—
PDU size	140 B
Fragmentation	ON
ARQ feedback	standalone
ARQ feedback types	all
ARQ feedback intensity	20 ms
ARQ block size	64 B
ARQ window	1024
ARQ discard	ON
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ timers	
Retry	100 ms
Block lifetime/Rx purge	500 ms

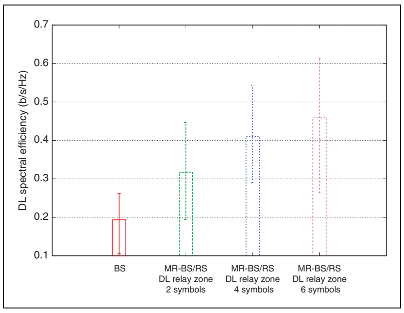


Figure 15. DL spectral efficiency.

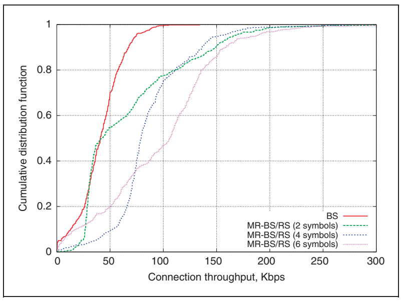


Figure 16. DL throughput cumulative distribution.

and the non-transparent RS transmit simultaneously, thus impacting each other.

To gather statistically reliable results, we ran 20 different simulations, where each simulation run contained 30 nodes placed in random locations. Each simulation run lasted for 10 seconds. It must be mentioned that we did not put SSs close the BS as they experience very good SNR there and thus are not so interesting for this relay study.

Figure 15 presents the DL spectral efficiency for a case when there is only the BS (the leftmost bar) and three cases with relays and different DL relay zone sizes (two, four, and six symbols). As can be seen, relays improve the spectral efficiency: the more resources a relay link has, the better the average spectral efficiency becomes because SSs, which are close to RSs, can benefit from a good link between BS and RS.

To provide a better insight on the relay performance, we also present the DL throughput cumulative distribution function (CDF) in Figure 16. It shows that relays improve performance, but there is a certain number of SSs that has worse throughput. The reason is that these SSs reside in very bad channel conditions, e.g., at a cell edge or BS/RS coverage area intersection. Their performance gets worse due to the increased interference level when we deploy non-transparent relays. In addition, throughput values become more diverse because certain SSs experience very good channel conditions if they are close to RS.

In addition to the spectral efficiency, we present results for the DL SDU delay. The delay is calculated as the time from the first transmission of SDU or its first fragment until the reception of the whole SDU at the receiving end after possible retransmissions. Figure 17 shows that RS nodes can improve significantly the DL delay. It is anticipated that with a higher DL throughput we can spend less time on transmitting an SDU and/or retransmitting its fragments.

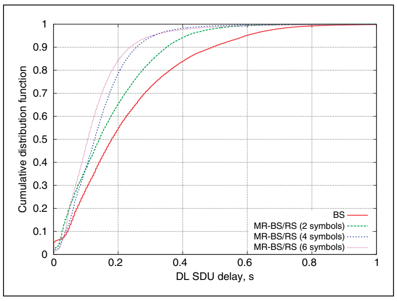


Figure 17. DL SDU transmission delay.

As a small summary, relays indeed improve the overall system performance at the expense of the overall throughput fairness that is quite difficult to achieve in the multi-hop environment. Furthermore, non-transparent relays increase the interference level that have a negative effect on the performance of SSs that reside in very bad channel conditions. The relay zone sizes must be set up carefully as they control the trade-off between the system performance and fairness. For the simulation scenario presented above, the DL relay zone size of four symbols is a good compromise between the increased spectral efficiency and decreased throughput fairness.

A deeper analysis is given in Sayenko et al.[55] for a more challenging case with NLOS communication between a RS and a BS.

6. Conclusions

In this paper we have presented WINSE, the 802.16 module for the NS-2 simulator. We introduced a support for the upper PHY level, MAC, QoS and scheduling, as well as the basic support for the access service network. Our implementation has demonstrated that NS-2 can be used to model complicated wireless broadband technologies, such as IEEE 802.16 and its multi-hop extensions. We have used the WINSE module in many research papers where we studied the MAC and QoS aspects of the IEEE 802.16 system. A number of technical contributions have also been submitted.

Based on our experience, it is possible to state that a truly powerful and scalable NS-2 module should not aim at a particular solution, but rather provide as many abstractions as possible to allow for different alternative implementations without changing the main core. The presented 802.16 module is an example of how it is possible to abstract from a particular PHY, BS scheduler, SS UL scheduler, contention resolution mechanism, etc. A highly modular architecture must also allow for flexible testing of various radio resource management algorithms. It creates unlimited possibilities for future research in areas, such as 802.16m.[56]

Acknowledgments

We express our gratitude to the following people, who helped during development, took part in testing and provided valuable comments on implementation and simulation techniques: Juha Karhula, Levente Sajo, Timo Joensuu, Jani Lakkakorpi, Jani Moilanen, Thomas Höhne, Lauri Kuuru, Yanqun Le, Wu Yi, Zhang Dongmei, Jan Suumäki, Michael Gundlach, Milton Mendieta, Alexey Dudkov, and Nenad Veselenovic. Special thanks go the WiMAX Forum Application Working Group members Tom Tofigh, Shayem Parekh, and Raj Jain for supporting the simulation methodology work item.

References

1. IEEE. Air interface for fixed broadband wireless access systems. IEEE Standard 802.16, 2004.
2. IEEE. Air interface for fixed broadband wireless access systems—amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands. IEEE Standard 802.16e, 2005.
3. Lu K, Qian Y, Chen H-H and Fu S. WiMAX neworks: From access to service platform. *IEEE Network* 2008; 22(3): 38–45.
4. UCB/LBNL/VINT. Network simulator ns-2. http://www.isi.edu/nsnam/ns
5. Bohnert T, Jakubiak J, Katz M, Koucheryavy Y, Monteiro E, Borcoci E. On evaluating a WiMAX access network for isolated research and data networks using NS-2. In *7th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking*. 2007, pp. 133–147.
6. Rouil R. *The Network Simulator NS-2 NIST Add-on: IEEE 802.16 Model (MAC+PHY)*. Technical Report, NIST, 2007.
7. Chen J, Wang C-C, Tsai FC-D, Chang C-W, Liu S-S, Guo J, et al. The design and implementation of WiMAX module for NS-2 simulator. In *Workshop on NS-2,* 2006.
8. Farooq J, Turletti T. An IEEE 802.16 WiMAX module for the NS-3 simulator. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 2009.
9. Msadaa IC, Filali F and Kamoun F. An 802.16 model for NS-2 simulator with an integrated QoS architecture. In *Simutools* 2008.
10. Kim S and Yeom I. IEEE 802.16 simulator 2006. http://cnlab.kaist.ac.kr/802.16/ieee802.16.html.
11. Baldo N, Maguolo F, Miozzo M, Rossi M and Zorzi M. NS-2 MIRACLE: A modular framework for multi-technology and cross-layer support in network simulator 2. In *NS tools* 2007.
12. IEEE. Air interface for broadband wireless access systems. IEEE Standard 802.16 (Rev2), 2009.
13. Betancur L, Hincapie RC and Bustamante R. WiMAX channel – PHY model in network simulator 2, In *Workshop on NS-2,* 2006.
14. Chen Q, Schmidt-Eisenlohr F, Jiang D, Torrent-Moreno M, Delgrossi L and Hartenstein H. Overhaul of IEEE 802.11 modeling and simulation in NS-2. In *The 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems,* 2007.
15. IEEE. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Standard 802.11, 1999.
16. IEEE. IEEE 802.16m evaluation methodology document (EMD). IEEE 802.16 Broadband Wireless Access Group, 2008.
17. Punnoose RJ, Nikitin PV, Stancil DD. Efficient simulation of ricean fading within a packet simulator. In *IEEE Vehicular Technology Conference*, 2000, pp. 764–767.
18. 3GPP. *Spatial Channel Model for Multiple Input Multiple Output (MIMO) Simulations*. Technical Report 25.996 v6.1.0, 3GPP, 2003.
19. Frederiksen F, Kolding T. Performance and modeling of WCDMA/HSDPA transmission/H-ARQ schemes. In *IEEE Vehicular Technology Conference*, 2002, pp. 472–476.
20. Cheng JF. Coding performance of Hybrid ARQ schemes. *IEEE Transactions on Communication* 2006; 54: 1017–1029.
21. Frenger P, Parkvall S, Dahlman E. Performance comparison of HARQ with chase combining and incremental redundancy in HSDPA. In *IEEE Vehicular Technology Conference*, 2001, pp. 1829–1833.
22. Kwon T, Lee H, Choi S, Kim J, Cho D-H, Cho S, et al. Design and implementation of a simulator based on a cross-layer protocols between MAC and PHY layers in a WiBro compatible IEEE 802.16e OFDMa system. *IEEE Communications Magazine* 2005; 43(12): 136–146.
23. WiMAX Forum. WiMAX Forum Mobile System Profile Specification: Release 1.5 (Revision 0.2.1), 2009.
24. Dudkov A and Sayenko A. Orthogonal bandwidth request codes for IEEE 802.16 networks. In *The 11th*

International Symposium on Wireless Personal Multimedia Communications, 2008.

25. Trees HLV. *Detection, Estimation and Modulation Theory. Part 1: Detection, Estimation, and Linear Modulation Theory*. New York: John Wiley and Sons, 2001.
26. Cicconetti C, Lenzini L and Mingozi E. Quality of service support in IEEE 802.16 networks. *IEEE Networks* 2006; 20(2): 50–55.
27. IEEE. Air interface for broadband wireless access systems: Multihop relay specification. IEEE Standard 802.16j, 2009.
28. Peters SW and Heath RW. The future of WiMAX: Multihop relaying with IEEE 802.16j. *IEEE Communications Magazine* 2009; 47(1): 104–111.
29. IEEE. Harmonized contribution on 802.16j (mobile multi-hop) usage models. IEEE 802.16 Broadband Wireless Access Working Group, 2006.
30. WiMAX Forum. Relay SG features. WiMAX Forum Technical Working Group, 2009.
31. 3GPP. Further advancements for E-UTRA; physical layer aspects. 3GPP, 2009.
32. IEEE. Multi-hop relay system evaluation methodology (channel model and performance metrics). IEEE 802.16 Broadband Wireless Access Working Group, 2007.
33. WiMAX Forum. WiMAX Forum Network Architecture, Release 1, Version 1.3.0, 2008.
34. Mendieta M. Modelling of test specifics for multi-vendor WiMAX networks. *Master's thesis*. Mannheim University of Applied Sciences, 2008.
35. Hytönen V, Sayenko A, Martikainen H and Alanen O. Handover performance in the IEEE 802.16 mobile networks. In *SIMUTools,* 2010.
36. Sayenko A, Alanen O, Karhula J and Hämäläinen T. Ensuring the QoS requirements in 802.16 scheduling. In *The 9th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems,* 2006, pp. 108–117.
37. Sayenko A, Alanen O and Hämäläinen T. Scheduling solution for the IEEE 802.16 base station. *Computer Networks* 2008; 52: 96–115.
38. Sayenko A, Alanen O, Hämäläinen T. ARQ aware scheduling for the IEEE 802.16 base station. In *IEEE International Conference on Communication*, 2008, pp. 2667–2673.
39. Lakkakorpi J, Sayenko A and Moilanen J. Comparison of different scheduling algorithms for WiMAX base station: Deficit round robin vs. proportional fair vs. weighted round robin. In: *IEEE WCNC,* 2008; 1991–1996.
40. Sayenko A, Alanen O, Hämäläinen T. Adaptive contention resolution parameters for the IEEE 802.16 networks. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, 2007.
41. Sayenko A, Alanen O and Hämäläinen T. Adaptive contention resolution for VoIP services in the IEEE 802.16 networks. In *WOWMOM. IEEE* 2007, pp. 1–7.
42. Alanen O. Multicast polling and efficient VoIP connections in IEEE 802.16 networks. In *The 10th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems,* 2007, 289–294.
43. Lakkakorpi J, Sayenko A. Uplink VoIP delays in IEEE 802.16e using different ertPS resumption mechanisms. In: *UBICOMM.* , 2009.
44. Lakkakorpi J, Sayenko A, Karhula J, Alanen O, Moilanen J. Active queue management for reducing downlink delays in WiMAX. In *IEEE 66th Vehicular Technology Conference, 2007 (VTC-2007)*. 2007, pp. 326–330.
45. Martikainen H, Sayenko A, Alanen O, Tykhomyrov V. Optimal MAC PDU size in IEEE 802.16. In *4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks*. 2008, pp. 66–71.
46. Tykhomyrov V, Sayenko A, Martikainen H and Alanen O. Analysis and performance evaluation of the IEEE 802.16 ARQ mechanism. *Journal of Communications Software and Systems* 2008; 4: 29–40.
47. Tykhomyrov V, Sayenko A, Martikainen H, Alanen O, Hämäläinen T. On ARQ feedback intensity of the IEEE 802.16 ARQ mechanism. In *International Conference on Telecommunications*, 2008.
48. Martikainen H, Alanen O and Sayenko A. ARQ parameters for VoIP in IEEE 802.16 networks. In *Wireless Telecommunications Symposium,* 2009.
49. Sayenko A, Martikainen H, Puchko A. Performance comparison of HARQ and ARQ mechanisms in IEEE 802.16 networks. In *The 11th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2008, pp. 411–416.
50. Puchko A, Tykhomyrov V and Martikainen H. Link adaptation thresholds for the IEEE 802.16 base station. In *Workshop on NS-2 Simulator,* 2008.
51. Tykhomyrov V, Sayenko A, Puchko O, Hämäläinen T. Decreasing the MAP overhead in the IEEE 802.16 OFDMA system. In *European Wireless Conference*, 2010.
52. Tykhomyrov V. Increasing the VoIP capacity through MAP overhead reduction in the IEEE 802.16 OFDMa systems. In *MOBILIGHT,* 2010.
53. Martikainen H. Analysis of duplexing modes in the IEEE 802.16 wireless system. In *European Wireless Conference*, 2010.
54. Lang E, Redana S, Raaf B. Business impact of relay deployment for coverage extension in 3GPP LTE-Advanced. In IEEE International Conference on Communication, 2009.
55. Sayenko A, Alanen O and Martikainen H. Analysis of the non-transparent in-band relays in the IEEE 802.16 multi-hop system. In *IEEE WCNC,* 2010.
56. IEEE. Air interface for broadband wireless access systems—advanced air interface. IEEE standard 802.16m (D3), 2009.

Alexander Sayenko has obtained the B.Sc degree from the Kharkov State University of RadioElectronics (Ukraine) in 2001. He has obtained the M.Sc degree from the University of Jyväskylä (Finland) and the PhD degree from the same university in 2002 and 2005, respectively. In 2007, he joined the Nokia

Research Center, where he was responsible for the resource and power management solutions. In 2008, he joined Nokia Siemens Networks where he concentrated on simulations, performance analysis, QoS, resource management and scheduling in the wireless networks. He participated in the WiMAX Forum member meetings and currently represents Nokia Siemens Networks in 3GPP RAN2 working group.

Olli Alanen received his M.Sc. and Ph.D. degrees from the MIT department of in University of Jyväskylä, Finland in 2004 and 2007, respectively. His research interests include the management of radio resources and quality of service in wireless broadband networks like WiMAX and LTE.

Henrik Martikainen received his M.Sc. in Computer Science from University of Jyväskylä in 2006. Since that he has been studying IEEE 802.16 MAC level performance and optimization. Lately he has also been studying cooperative MIMO techniques in 3GPP's HSDPA context. At the moment, he is a Ph.D. student at University of Jyväskylä, Finland.

Vitaliy Tykhomyrov received the M.Sc. degree in Computer Science from Kharkov National University of Radio Electronics, Ukraine, in 2006. From 2007 to 2010, he was a researcher at University of Jyväskylä, Finland, where he involved in the design, performance evaluation, and development of the IEEE 802.16 system. Currently, he is a Ph.D. student at University of Jyväskylä, Finland. His recent work focuses scheduling algorithms in HSDPA networks, radio resource management, QoS in wireless networks, and the IEEE 802.16 and HSDPA networks.

Oleksandr Puchko is a Ph.D. student at University of Jyväskylä, Finland. He received the M.Sc degree in Telecommunication from Kharkov National University of RadioElectronics, Ukraine in 2007. His current postgraduate research interest is PHY level in wireless networks.

Vesa Hytönen received his M.Sc. degree in Computer Science from University of Jyväskylä, Finland, in 2009. For the recent year he has been studying user mobility performance in the IEEE 802.16 networks. His current research topics include mobility in relay networks as well as signaling issues for cooperative MIMO in HSDPA.

Timo Hämäläinen received the B.Sc in automation engineering from the Jyväskylä Institute of Technology in Finland on 1991 and the M.Sc and Ph.D degrees in telecommunication from Tampere University of technology and University of Jyväskylä, Finland in 1996 and 2002, respectively. Currently, he is Professor of Telecommunications at the University of Jyväskylä. His current research interests include traffic engineering and Quality of Service in wired and wireless networks.

PVIII

ANALYSIS OF THE NON-TRANSPARENT IN-BAND RELAYS IN THE IEEE 802.16 MULTI-HOP SYSTEM

by

A. Sayenko, O. Alanen and H. Martikainen 2010

IEEE Wireless Communications & Networking Conference

Analysis of the Non-transparent In-band Relays in the IEEE 802.16 Multi-hop System

Alexander Sayenko
Research, Technology & Platforms,
Nokia Siemens Networks,
Espoo, Finland
alexander.sayenko@nsn.com

Olli Alanen and Henrik Martikainen
Telecommunication laboratory, MIT department,
University of Jyväskylä,
Finland
{olli.alanen, henrik.martikainen}@jyu.fi

***Abstract*—This paper presents extensive dynamic simulations of the non-transparent in-band relays working in the distributed scheduling mode. The simulation results show that in-band relays can improve noticeably the spectral efficiency without acquiring an additional radio spectrum. Also, packet transmission delays become smaller. An important outcome of the dynamic simulations is that it is very crucial to choose a correct relay zone size where the base station and relay nodes exchange data. Otherwise, throughput fairness of the whole system declines. It indicates an importance of the relay zone size adjustment algorithm that the base station must run.**

***Index Terms*—IEEE 802.16j, relays, scheduling, NS-2.**

I. Introduction

The very high data rate demands for wireless communication systems create a need for more fundamental enhancements, other than just increasing the transmission bandwidth or introducing higher order modulation and coding schemes. Along with technologies, such as MIMO and cooperative multi-point transmission, relaying is seen as a quite promising solution [1].

There are a number of reasons why a provider would decide to deploy a relay instead of a full-featured base station. Firstly, a relay does not need any connection to the mobile access network wired backhaul. As a result, a relay can be deployed to a location where the wired connection does not exist at all or where it is too expensive and/or complicated to install. It is worth noting that not all the base stations are connected directly to the wired medium; quite many of them use microwave links to forward data via other sites. However, a microwave link needs a larger site, requires a strong line of sight (LOS) path and needs a license to use the microwave spectrum. What is also important is that there is no standard for microwave links, meaning that a provider has no choice but to deploy equipment from the same vendor on both microwave link endpoints. When compared to a full-featured base station site, relays are faster and simpler in deployment and may even use omni-directional antennas, which simplifies a lot the hardware design and reduces the cost. Even though this paper does not consider all the design options, it is necessary to mention that a relay can integrate the system module with an antenna thus getting rid of a feeder cable where a considerable power loss occurs. It is also worth emphasizing that a transition to the relay networks consists purely of a software upgrade of the base station that is backward compatible with the legacy network and terminals.

At the moment, there are two major wireless technologies where multi-hop relays have gained a significant interest from vendors and operators: 3GPP LTE-A [2], [3] and IEEE 802.16j extensions [4] of the baseline 802.16 system [5], [6]. In this paper, we refrain from comparing these two technologies, but rather limit ourselves to IEEE 802.16j as a more mature solution for the multi-hop communication. In any case, relaying concepts are very similar in these technologies and we believe strongly that results obtained from the IEEE 802.16j system can be applied to 3GPP LTE-A relays, and vice versa.

To promote deployment and usage of relays, we need extensive simulation results that can indicate clearly performance gains, as well as possible disadvantages. Due to the overall system complexity, it is often the case that researchers either rely upon pure theoretical approach or use simple static simulators that account only for the path loss and do not consider dynamics of the whole system, not mentioning any related system and management overhead. At the same time, many dynamic simulators adopted for relay purposes simulate the PHY and MAC levels without capturing the transport and application level performance. However, an accurate modeling of higher layers becomes quite important in the multi-hop environment due to delays, packet drops and retransmissions that may cause easily performance degradation. Furthermore, there is a non-trivial amount of signaling and management information passed between the base station and relay, which also consumes system resources.

In this paper, we present the performance analysis of the 802.16j non-transparent in-band relays working in the distributed scheduling mode. We elaborate more on our choice for this relay type in Section II. We use a dynamic system simulator that accounts for PHY, MAC, and for the radio resource management algorithms, such as scheduling and link adaptation. On top of that, we run transport protocols to simulate accurately the application level performance. We focus on the performance analysis of relay solutions and abstain from cost analysis or optimal relay deployment. Research results on these topics can be found in [7], [8].

The rest of this paper is organized as follows. Section II provides a brief overview of 802.16j relays, available configuration options and related challenges. Section III presents the simulation results for the non-transparent in-band relays working in the distributed scheduling mode. Finally, section IV concludes the paper.

II. IEEE 802.16 RELAYS

For the sake of brevity, we will not divulge into extensive description of the relaying functionality defined in 802.16j. The technical specification is given in [4] and a good technical overview is presented in [9]. However, it is worth to mention available relaying options. From the viewpoint of the spectrum usage, relays can be either in-band (TTR) or out-band (STR). From the viewpoint of the downlink (DL) management signaling, they can be either transparent or non-transparent. These combinations also define the possible scheduling modes – either centralized or distributed – that are summarized in Table I.

TABLE I
AVAILABLE RELAY MODES IN 802.16J.

	transparent	non-transparent
in-band (TTR)	centralized	centralized/distributed
out-band (STR)	–	distributed

Hence, following the 802.16j terminology, BS will refer to the base station, while MR-BS will denote multi-hop relay BS; RS stands for the relay station.

There is a strong motivation to consider the non-transparent in-band relays working in the distributed scheduling mode. Firstly, an in-band relay reuses the *existent* spectrum instead of requiring a new frequency band. Thus, it is an appealing option for operators that do not have or cannot acquire additional radio resources. Secondly, a non-transparent relay can enhance both *coverage and throughput*, while a transparent relay can improve only throughput within the *existent* cell boundaries. Finally, the distributed scheduling mode makes scheduler implementation simpler and allows for reusing an existent BS software at RS without implementing a complicated centralized scheduling. It also makes the overall scheduling process faster because both MR-BS and RS schedulers work as two independent entities.

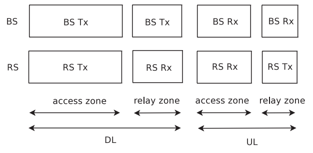


Fig. 1. Non-transparent in-band relay frame structure.

For the sake of further clarity, Fig. 1 presents the frame structure of the non-transparent in-band relay. The relay zone is a one where the MR-BS and RS exchange data. It must be noted that both BS and RS transmit simultaneously in the DL access zone to the associated subscriber stations (SS) thus mutually interfering. A similar situation occurs in the uplink (UL) access link when SSs associated with RS and BS start to interfere with each other. Thus, the non-transparent relays reuse the existent spectrum at the cost of increasing the interference level.

III. SIMULATION

A. Simulation environment

We use the 802.16 extension to the NS-2 simulator called WINSE, detailed description of which is given in [10]. On top of that, we provide a support for two-hop non-transparent in-band relays as a main solution for the coverage extension problem [11]. Even though the specification allows for more than two hops, such a possibility is not supported due to the resulting system complexity [12]; the same approach is also taken in other technologies, such as 3GPP LTE-A. As a result, there is no need to support more than one DL/UL relay zone in the 802.16 frame.

The non-transparent RS node runs the same radio resources management mechanisms as a normal BS does, e.g., scheduling [13], channel estimation, link adaptation [14] etc. The scheduler is a *throughput fair* one that is based conceptually on deficit round robin. Even though the proportional fair scheduler might provide a better spectral efficiency [15], we choose the throughput fair scheduler to show the impact of relaying on the throughput fairness.

B. Simulation results

Fig. 2 shows a simulation scenario. It is assumed that there is a single BS controlling its sector. To serve an area limited by a dashed line, an operator may deploy additional BSs to cover two more sectors denoted by dotted lines. However, a more cost efficient solution might be to deploy a few relay nodes, as shown in the figure. It is understandable that deploying additional BSs will bring a better performance at the expense of increased deployment cost: installation and support of two macro BS with microwave links or wired backhaul connections will cost more than three RS nodes [8]. Thus, for the sake of brevity, we compare a case with a single BS and a case with the MR-BS and RS nodes.

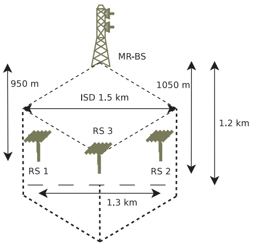


Fig. 2. Simulation scenario.

The choice for the number of RSs was motivated based on the cost analysis in [8] that stated that it is better to have a few high-power RS nodes rather than a number of very low-power ones. While placing three RS nodes, we account for the MR-BS directional antenna gain and its coverage area. While RS3 is placed at the MR-BS main antenna lobe direction, RS1 and RS2 are placed closer to the MR-BS and to the cell edge. Furthermore, since MR-BS and the non-transparent RSs interfere with each other, we do not put the RS nodes too close to the MR-BS to avoid mutual interference. The final RSs coordinates were tuned after a few simulation runs. However, we do not claim in this paper that they are the optimal ones.

TABLE II
802.16 NETWORK PARAMETERS.

Parameter	**Value**
Center frequency	2.5 GHz
Bandwidth	10 MHz
PHY	OFDMa
Reuse factor	1/3
Duplexing mode	TDD
Frame duration	5 ms
CP length	1/8 symbol
TTG+RTG	296+168 PS
OFDM symbols	47
DL/UL symbols	30/15
DL/UL relay zone size	2, 4, 6 / 3 symbols
DL/UL subcarrier alloc.	DL PUSC / UL PUSC
Channel report type / interval	CQICH / 20ms
Channel measurements DL/UL	preamble / data burst
Channel measurements filter	EWMA, α = 0.25
Link adaptation model	target FEC BLER, 10^{-1}
Antenna technique	SISO
BS / RS / SS Tx power	10 / 5 / 0.25 W
BS / RS / SS antenna pattern	3GPP / omni / omni
BS / RS / SS antenna gain	17 / 5 / 0 dBi
BS / RS / SS antenna height	32 / 7 / 1.5 m
access / relay link path loss	.16m SMa / .16j TypeA
access / relay link fast fading K factor	0 / 0 dB
DL MAP MCS	QPSK1/2 Rep6…QPSK1/2
Compressed MAPs	ON
sub-MAPs	ON, max. 3
Ranging transm. opport.	1
Ranging backoff start/end	0/15
Request transm. opport.	2
Request backoff start/end	1/15
CDMA codes	256
ranging+periodic ranging	64
bandwidth request	192
handover	–
PDU size	140 B
Fragmentation	ON
ARQ feedback	standalone
ARQ feedback types	all
ARQ feedback intensity	20 ms
ARQ block size	64 B
ARQ window	1024
ARQ discard	ON
ARQ block rearrangement	ON
ARQ deliver in order	ON
ARQ timers	
retry	60 ms
block lifetime/Rx purge	500 ms

Table II presents the key 802.16 parameters used in the simulation, which conform to [16]. We consider the DL FTP-like continuous TCP transmission over 802.16 BE connections, where the IP level service data unit (SDU) size is 1000 bytes. It is a good way to analyze the resulting application level throughput and the spectral efficiency. Of course, there is also UL traffic caused by the TCP protocol functioning. It is worth mentioning that to study the relay performance, we consider different *fixed* DL relay zones size (see Fig. 1) of 2, 4, and 6 OFDM symbols. The UL zone size is also fixed and has the constant size of 3 symbols.[1] Unlike the MR-BS, RS uses an omni-directional antenna, has a lower Tx power of 5W and a smaller antenna height. The motivation is that a lower Tx power requires a simpler and a less expensive amplifier chain. The omni-directional antenna simplifies the design and the installation efforts. Furthermore, an omni-directional antenna at the RS node allows for communicating efficiently to any SS around the RS node.[2]

We assume the sub-urban macro-cell scenario and thus choose the 802.16m SMa propagation model for an access link and the 802.16j TypeA model for the relay link [17]. The latter one is for the NLOS communication between MR-BS and RS nodes because otherwise, an operator might deploy a microwave link between two BS sites. The interference modeling accounts for the fact that the MR-BS and the non-transparent RSs transmit simultaneously thus impacting each other. The interference from the neighboring cells is also taken into account assuming the reuse 1/3 factor and full load traffic. The fast fading is generated based on the Jakes model with the K factors given in Table II assuming an SS speed of 1 m/s.

The MAC level retransmission mechanism is ARQ working in the end-to-end mode. In other words, RS does not take part in the ARQ signaling but just forwards received data. The ARQ parameters are tuned based on our previous research on the ARQ mechanism in the 802.16 networks [18]. The ARQ mechanism also governs the target FEC block error rate of 10^{-1} that we use in the link adaptation module [14].

To gather statistically reliable results, we ran 20 different simulations, where each of them contained 30 SSs placed in random locations. Each simulation run lasted for 10 seconds, which is enough for the TCP protocol to stabilize.

Fig. 3 shows the simulation area with SS locations and their associations to the MR-BS or RS node, as indicated by different symbols. As anticipated, an SS associates itself to RS if it observes a stronger DL signal strength coming from the RS node. Note that Fig. 3 accumulates 600 different locations from all the simulation runs. Of course, if there is only a single BS node, then all the SSs in Fig. 3 are associated with it.

Fig. 4 presents the DL *application level* spectral efficiency, i.e., the one that excludes any PHY or the MAC level management data. We present the minimum, average, and maximum values for a case when there is only the BS (the leftmost bar) and three cases with relays and different DL relay zone sizes (2, 4, and 6 symbols). As can be seen, relays improve the spectral efficiency: the more resources a relay link has, the better an average spectral efficiency becomes because SSs,

[1] DL relay zone size must be a multiple of 2 OFDM symbols due to the DL PUSC permutation type. Similarly, the UL relay zone size must be a multiple of 3 OFDM symbols due to the UL PUSC structure.

[2] A possible solution is to have two antennas at the RS node: a directional one to exchange data with MR-BS and an omnidirectional one to communicate with associated SSs around RS. This solution is more complicated in implementation and requires more installation efforts due to the antenna direction and tilting.

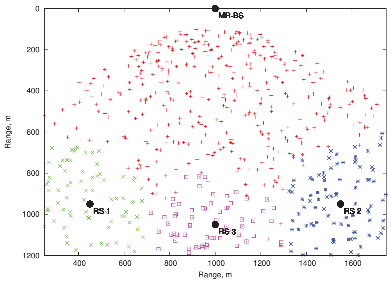


Fig. 3. Simulation area with SS locations.

which are close to an RS, can benefit from a good link between BS and RS.

To provide a better insight on the relay performance, we also present the mean DL connection throughput CDF in Fig. 5. The mean throughput is calculated individually for each connection after each simulation run. Firstly, it is noticeable that without relays there are SSs that have no service at all because they are out of the BS coverage area. Once we deploy relays, all the SSs are able to transmit at least at some rate. Secondly, Fig. 5 shows that the DL relay zone size of 2 symbols results in a situation when there are SSs that have a lower throughput when compared to the baseline scenario. As explained later, this is due to the fact that a small DL relay zone size becomes a bottleneck.

Fig. 6 presents an analysis of the throughput fairness.[3] Of course, if there is only the BS node then it can ensure quite a good fairness because the BS scheduler has a complete control over resource allocation on the access link. Once we deploy RSs, everything the MR-BS scheduler can do is to control resources for the relay link, but not the way the RS node will allocate resources between connections on its access link. It can be seen that relays with working in the distributed scheduling mode decrease the throughput fairness, especially in case of a badly configured DL relay zone size.

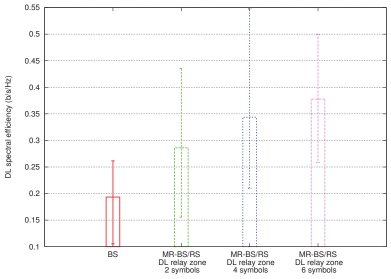


Fig. 4. DL spectral efficiency.

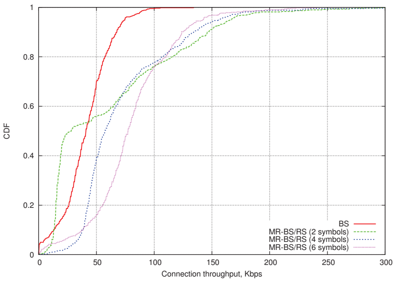


Fig. 5. DL connection throughput cumulative distribution.

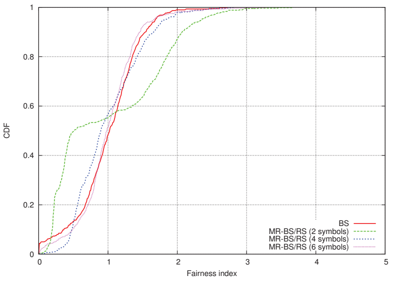
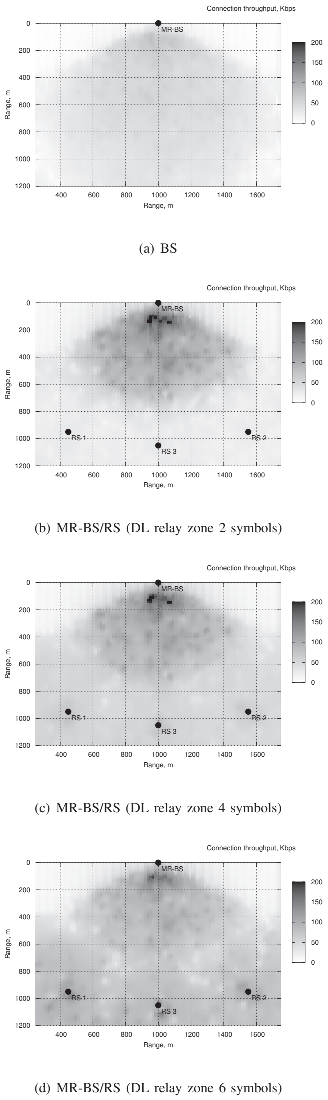


Fig. 6. DL connection throughput fairness.

Fig. 7 provides a different view on throughput and fairness. Similar to Fig. 3, this figure aggregates results from all the simulation runs and presents the throughput distribution over the simulation area under different DL relay zone sizes. As can be seen from Fig. 7(a), a case when only the BS is deployed results in a low but quite fair throughput distribution over the simulation area. However, cell edge areas have no service at all. If we deploy RSs with a small DL relay zone size (see Fig. 7(b)), then the RS nodes can offload the MR-BS, thus providing a higher throughput to SSs associated with it. However, a small DL relay zone size results in a considerably lower throughput of SSs associated with the RS nodes because the MR-BS to RS link becomes a bottleneck. As we increase the DL relay zone size, we can see that SSs associated with RS nodes start to transmit at much higher throughput, increasing the overall system spectral efficiency. Once the DL relay zone size equals 6 symbols, the throughput fairness starts to decline

[3] We use the fairness index defined in [19], which is a CDF of the normalized per user throughputs. An absolute fair throughput allocation yields a vertical line with the x coordinate of 1.



(a) BS

(b) MR-BS/RS (DL relay zone 2 symbols)

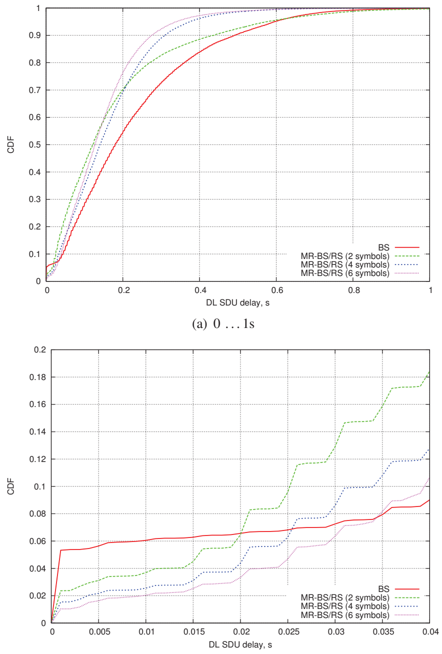
(c) MR-BS/RS (DL relay zone 4 symbols)

(d) MR-BS/RS (DL relay zone 6 symbols)

Fig. 7. DL throughput distribution over the simulation area.

because now the DL access zone has become a bottleneck.

Fig. 7 also presents the impact of the non-transparent relaying and, as a result, increased interference level on the throughput distribution. In Fig. 7(a), low throughputs are observed at the cell edge where distance increases and directional antenna gain becomes smaller. In Fig. 7(b)-Fig. 7(d), low throughputs are observed at the cell edge and places where signal strength from the MR-BS is as strong as the cumulative interference coming from all the RS nodes. Thus, the fact that the non-trasparent relays create additional interference introduces more challenges for the network planning.



(a) 0 . . . 1s

(b) 0 . . . 40ms

Fig. 8. DL SDU delay cumulative distribution.

In addition to the spectral efficiency and fairness analysis, we present results for the DL SDU delay. The delay is calculated as a time from the first transmission of SDU or its first fragment till the reception of the whole SDU at the receiving end after possible retransmissions. Fig. 8(a) shows that RS nodes can improve significantly the DL delay. It is anticipated that with a higher DL throughput we can spend less time on transmitting an SDU and/or retransmitting its fragments. At the same time, Fig. 8(b) illustrates eloquently that while decreasing delays on average, relays also increase the delay for certain packets. If there is only the BS in a cell, then a certain amount of packets have a delay of less than 5 ms. These are unfragmented SDUs that are delivered to SSs without a retransmission within a single frame. Once we introduce RS nodes, we add an additional transmission hop, thus increasing the overall transmission time. Even though it is not a significant delay for non-critical data, such as BE TCP, it might be noticeable for VoIP traffic, especially if more than two hops exist. As can be seen from Fig. 8(b), the switching point is around 20…40 ms, which is a typical frame generation rate of many VoIP codecs.

Based on the presented results, it is possible to state that the relay zone plays quite a crucial role. It can be treated as a parameter that controls the tradeoff between the overall system spectral efficiency and the throughput fairness. For the simulation scenario considered, the DL relay zone size of 4 symbols is a good choice: the spectral efficiency is almost two times higher and the fairness is satisfactory. Of course, a different number of SSs, their location, and/or traffic pattern may yield a different optimal configuration.

It is worth noting that the presented simulation results meet the initial requirements for the relay performance set in [20]. In particular, the 95 percentile throughput CDF is 1.5 times larger than the baseline scenario with a single BS. Even though BE traffic delays are not mentioned explicitly in [20], the 95 percentile delay CDF also indicates a noticeable improvement. Furthermore, if relays are deployed to places with a high user concentration or a LOS link between MR-BS and RS nodes exists, then even a better gain is anticipated.

IV. Conclusions

In this paper, we have run complex dynamic simulations of the 802.16j non-transparent in-band relays working in the distributed scheduling mode. According to the simulation results, relays indeed improve the overall system performance even despite the NLOS link between MR-BS and RSs and the fact that the non-transparent RS nodes and the MR-BS interfere with each other and the MR-BS has to allocate its resources for the relay communication. At the same time, the overall complexity of the whole system makes it quite complicated to achieve simultaneously a high throughput and fairness across all the access links in the system. Furthermore, the balance between the spectral efficiency and fairness depends heavily on the DL relay zone size over which the base station exchanges data with relays. Thus, the relay zone sizes must be set up carefully and adjusted dynamically to control the tradeoff between the system performance and fairness. In turn, this creates a need for fast and reliable signaling mechanism to orchestrate relay zone sizes. We believe strongly that the presented results can be reused in the 3GPP LTE-A technology that has the relay concepts similar to IEEE 802.16j.

Our future research topics will aim at analyzing the VoIP capacity of the relay systems where a continuous bi-directional data transmission and tight timing requirements pose even more challenges.

V. Acknowledgements

We express our acknowledgements to the following Nokia Siemens Networks colleagues: Jani Moilanen and Lauri Kuru for valuable and helpful comments on the relay results and simulation methodology, Simone Redana for providing an analysis on relays cost and deployment tradeoffs, Risto Mononen and Timo Naalisvaara for discussions on hardware design and cost issues, Mika Perhonen for a deep insight on the BS microwave links and their parameters.

Special thanks to the WINSE development team at the University of Jyväskylä (Telecommunication laboratory) for the 802.16 extensions to the NS-2 simulator.

References

[1] R. Pabst, B. Walke, D. Schultz, P. Herhold, H. Yanikomeroglu, S. Mukherjee, H. Viswanathan, M. Lott, W. Zirwas, M. Dohler, H. Aghvami, D. Falconer, and G. Fettweis, “Relay-based deployment concepts for wireless and mobile broadband radio,” *IEEE Communications Magazine*, vol. 42, no. 9, pp. 80–89, Sep 2004.

[2] 3GPP TR 36.814 v1.4.2, “Further advancements for E-UTRA; physical layer aspects,” Sep 2009.

[3] 3GPP TR 36.806 v0.1.0, “Relay architecture for E-UTRA (LTE-Advanced),” Sep 2009.

[4] “Air interface for broadband wireless access systems: Multihop relay specification,” IEEE Standard 802.16j, Jun 2009.

[5] “Air interface for fixed broadband wireless access systems - amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands,” IEEE Standard 802.16e, Dec 2005.

[6] “Air interface for broadband wireless access systems,” IEEE Standard 802.16 (Rev2), May 2009.

[7] D. Shultz and B. Waalke, “Fixed relays for cost efficient 4G network deployments: an evaluation,” in *The 18th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications*, Sep 2007, pp. 1–5.

[8] E. Lang, S. Redana, and B. Raaf, “Business impact of relay deployment for coverage extension in 3GPP LTE-Advanced,” in *IEEE International Conference on Communication*, Jun 2009.

[9] S. W. Peters and R. W. Heath, “The future of WiMAX: multihop relaying with IEEE 802.16j,” *IEEE Communications Magazine*, Jan 2009.

[10] A. Sayenko, O. Alanen, H. Martikainen, V. Tykhomyrov, O. Puchko, V. Hytönen, and T. Hämäläinen, “WINSE: WiMAX NS-2 Extension,” *Special Issue of Simulation: Software Tools, Techniques and Architectures for Computer Simulation*, accepted for publication.

[11] “Harmonized contribution on 802.16j (mobile multi-hop) usage models,” IEEE 802.16 Broadband Wireless Access Working Group, Sep 2006.

[12] “Relay SG features,” WiMAX Forum, Technical Working Group, Jul 2009.

[13] A. Sayenko, O. Alanen, and T. Hämäläinen, “Scheduling solution for the IEEE 802.16 base station,” *Computer Networks*, vol. 52, pp. 96–115, 2008.

[14] A. Puchko, V. Tykhomyrov, and H. Martikainen, “Link adaptation thresholds for the IEEE 802.16 base station,” in *Workshop on NS-2 simulator*, Oct 2008.

[15] J. Lakkakorpi, A. Sayenko, and J. Moilanen, “Comparison of different scheduling algorithms for WiMAX base station: deficit round robin vs. proportional fair vs. weighted round robin,” in *IEEE WCNC*, Mar/Apr 2008, pp. 1991–1996.

[16] “WiMAX Forum Mobile System Profile Specification: Release 1.5 (Revision 0.2.1),” Feb 2009.

[17] “Multi-hop relay system evaluation methodology (channel model and performance metrics),” IEEE 802.16 Broadband Wireless Access Working Group, Feb 2007.

[18] V. Tykhomyrov, A. Sayenko, H. Martikainen, and O. Alanen, “Analysis and performance evaluation of the IEEE 802.16 ARQ mechanism,” *Journal of communications software and systems*, vol. 4, no. 1, pp. 29–40, Mar 2008.

[19] “IEEE 802.16m evaluation methodology document (EMD),” IEEE 802.16 Broadband Wireless Access Group, Mar 2008.

[20] “Requirements and recommendations for multi-hop relay profile for WiMAX networks,” WiMAX Forum, Service Provider Working Group, Jun 2008.

JYVÄSKYLÄ STUDIES IN COMPUTING

1 Ropponen, Janne, Software risk management - foundations, principles and empirical findings. 273 p. Yhteenveto 1 p. 1999.

2 Kuzmin, Dmitri, Numerical simulation of reactive bubbly flows. 110 p. Yhteenveto 1 p. 1999.

3 Karsten, Helena, Weaving tapestry: collaborative information technology and organisational change. 266 p. Yhteenveto 3 p. 2000.

4 Koskinen, Jussi, Automated transient hypertext support for software maintenance. 98 p. (250 p.) Yhteenveto 1 p. 2000.

5 Ristaniemi, Tapani, Synchronization and blind signal processing in CDMA systems. - Synkronointi ja sokea signaalinkäsittely CDMA järjestelmässä. 112 p. Yhteenveto 1 p. 2000.

6 Laitinen, Mika, Mathematical modelling of conductive-radiative heat transfer. 20 p. (108 p.) Yhteenveto 1 p. 2000.

7 Koskinen, Minna, Process metamodelling. Conceptual foundations and application. 213 p. Yhteenveto 1 p. 2000.

8 Smolianski, Anton, Numerical modeling of two-fluid interfacial flows. 109 p. Yhteenveto 1 p. 2001.

9 Nahar, Nazmun, Information technology supported technology transfer process. A multi-site case study of high-tech enterprises. 377 p. Yhteenveto 3 p. 2001.

10 Fomin, Vladislav V., The process of standard making. The case of cellular mobile telephony. - Standardin kehittämisen prosessi. Tapaustutkimus solukkoverkkoon perustuvasta matkapuhelintekniikasta. 107 p. (208 p.) Yhteenveto 1 p. 2001.

11 Päivärinta, Tero, A genre-based approach to developing electronic document management in the organization. 190 p. Yhteenveto 1 p. 2001.

12 Häkkinen, Erkki, Design, implementation and evaluation of neural data analysis environment. 229 p. Yhteenveto 1 p. 2001.

13 Hirvonen, Kullervo, Towards better employment using adaptive control of labour costs of an enterprise. 118 p. Yhteenveto 4 p. 2001.

14 Majava, Kirsi, Optimization-based techniques for image restoration. 27 p. (142 p.) Yhteenveto 1 p. 2001.

15 Saarinen, Kari, Near infra-red measurement based control system for thermo-mechanical refiners. 84 p. (186 p.) Yhteenveto 1 p. 2001.

16 Forsell, Marko, Improving component reuse in software development. 169 p. Yhteenveto 1 p. 2002.

17 Virtanen, Pauli, Neuro-fuzzy expert systems in financial and control engineering. 245 p. Yhteenveto 1 p. 2002.

18 Kovalainen, Mikko, Computer mediated organizational memory for process control. Moving CSCW research from an idea to a product. 57 p. (146 p.) Yhteenveto 4 p. 2002.

19 Hämäläinen, Timo, Broadband network quality of service and pricing. 140 p. Yhteenveto 1 p. 2002.

20 Martikainen, Janne, Efficient solvers for discretized elliptic vector-valued problems. 25 p. (109 p.) Yhteenveto 1 p. 2002.

21 Mursu, Anja, Information systems development in developing countries. Risk management and sustainability analysis in Nigerian software companies. 296 p. Yhteenveto 3 p. 2002.

22 Seleznyov, Alexandr, An anomaly intrusion detection system based on intelligent user recognition. 186 p. Yhteenveto 3 p. 2002.

23 Lensu, Anssi, Computationally intelligent methods for qualitative data analysis. 57 p. (180 p.) Yhteenveto 1 p. 2002.

24 Ryabov, Vladimir, Handling imperfect temporal relations. 75 p. (145 p.) Yhteenveto 2 p. 2002.

25 Tsymbal, Alexey, Dynamic integration of data mining methods in knowledge discovery systems. 69 p. (170 p.) Yhteenveto 2 p. 2002.

26 Akimov, Vladimir, Domain decomposition methods for the problems with boundary layers. 30 p. (84 p.). Yhteenveto 1 p. 2002.

27 Seyukova-Rivkind, Ludmila, Mathematical and numerical analysis of boundary value problems for fluid flow. 30 p. (126 p.) Yhteenveto 1 p. 2002.

28 Hämäläinen, Seppo, WCDMA Radio network performance. 235 p. Yhteenveto 2 p. 2003.

29 Pekkola, Samuli, Multiple media in group work. Emphasising individual users in distributed and real-time CSCW systems. 210 p. Yhteenveto 2 p. 2003.

30 Markkula, Jouni, Geographic personal data, its privacy protection and prospects in a location-based service environment. 109 p. Yhteenveto 2 p. 2003.

31 Honkaranta, Anne, From genres to content analysis. Experiences from four case organizations. 90 p. (154 p.) Yhteenveto 1 p. 2003.

32 Raitamäki, Jouni, An approach to linguistic pattern recognition using fuzzy systems. 169 p. Yhteenveto 1 p. 2003.

33 Saalasti, Sami, Neural networks for heart rate time series analysis. 192 p. Yhteenveto 5 p. 2003.

34 Niemelä, Marketta, Visual search in graphical interfaces: a user psychological approach. 61 p. (148 p.) Yhteenveto 1 p. 2003.

35 You, Yu, Situation Awareness on the world wide web. 171 p. Yhteenveto 2 p. 2004.

36 Taatila, Vesa, The concept of organizational competence – A foundational analysis. - Perusteanalyysi organisaation kompetenssin käsitteestä. 111 p. Yhteenveto 2 p. 2004.

JYVÄSKYLÄ STUDIES IN COMPUTING

37 Lyytikäinen, Virpi, Contextual and structural metadata in enterprise document management. - Konteksti- ja rakennemetatieto organisaation dokumenttien hallinnassa. 73 p. (143 p.) Yhteenveto 1 p. 2004.

38 Kaario, Kimmo, Resource allocation and load balancing mechanisms for providing quality of service in the Internet. 171 p. Yhteenveto 1 p. 2004.

39 Zhang, Zheying, Model component reuse. Conceptual foundations and application in the metamodeling-based systems analysis and design environment. 76 p. (214 p.) Yhteenveto 1 p. 2004.

40 Haarala, Marjo, Large-scale nonsmooth optimization variable metric bundle method with limited memory. 107 p. Yhteenveto 1 p. 2004.

41 Kalvine, Viktor, Scattering and point spectra for elliptic systems in domains with cylindrical ends. 82 p. 2004.

42 Dementieva, Maria, Regularization in multistage cooperative games. 78 p. 2004.

43 Maaranen, Heikki, On heuristic hybrid methods and structured point sets in global continuous optimization. 42 p. (168 p.) Yhteenveto 1 p. 2004.

44 Frolov, Maxim, Reliable control over approximation errors by functional type a posteriori estimates. 39 p. (112 p.) 2004.

45 Zhang, Jian, Qos- and revenue-aware resource allocation mechanisms in multiclass IP networks. 85 p. (224 p.) 2004.

46 Kujala, janne, On computation in statistical models with a psychophysical application. 40 p. (104 p.) 2004.,

47 Solbakov, Viatcheslav, Application of mathematical modeling for water environment problems. 66 p. (118 p.) 2004.

48 Hirvonen, Ari P., Enterprise architecture planning in practice. The Perspectives of information and communication technology service provider and end-user. 44 p. (135 p.) Yhteenveto 2 p. 2005.

49 Vartiainen, Tero, Moral conflicts in a project course in information systems education. 320 p. Yhteenveto 1p. 2005.

50 Huotari, Jouni, Integrating graphical information system models with visualization techniques. - Graafisten tietojärjestelmäkuvausten integrointi visualisointitekniikoilla. 56 p. (157 p.) Yhteenveto 1p. 2005.

51 Wallenius, Eero R., Control and management of multi-access wireless networks. 91 p. (192 p.) Yhteenveto 3 p. 2005.

52 Leppänen, Mauri, An ontological framework and a methodical skeleton for method engineering – A contextual approach. 702 p. Yhteenveto 2 p. 2005.

53 Matyukevich, Sergey, The nonstationary Maxwell system in domains with edges and conical points. 131 p. Yhteenveto 1 p. 2005.

54 Sayenko, Alexander, Adaptive scheduling for the QoS supported networks. 120 p. (217 p.) 2005.

55 Kurjenniemi, Janne, A study of TD-CDMA and WCDMA radio network enhancements. 144 p. (230 p.) Yhteenveto 1 p. 2005.

56 Pechenizkiy, Mykola, Feature extraction for supervised learning in knowledge discovery systems. 86 p. (174 p.) Yhteenveto 2 p. 2005.

57 Ikonen, Samuli, Efficient numerical methods for pricing American options. 43 p. (155 p.) Yhteenveto 1 p. 2005.

58 Kärkkäinen, Kari, Shape sensitivity analysis for numerical solution of free boundary problems. 83 p. (119 p.) Yhteenveto 1 p. 2005.

59 Helfenstein, Sacha, Transfer. Review, reconstruction, and resolution. 114 p. (206 p.) Yhteenveto 2 p. 2005.

60 Nevala, Kalevi, Content-based design engineering thinking. In the search for approach. 64 p. (126 p.) Yhteenveto 1 p. 2005.

61 Katasonov, Artem, Dependability aspects in the development and provision of location-based services. 157 p. Yhteenveto 1 p. 2006.

62 Sarkkinen, Jarmo, Design as discourse: Representation, representational practice, and social practice. 86 p. (189 p.) Yhteenveto 1 p. 2006.

63 Äyrämö, Sami, Knowledge mining using robust clustering. 296 p. Yhteenveto 1 p. 2006.

64 Ifinedo, Princely Emili, Enterprise resource planning systems success assessment: An integrative framework. 133 p. (366 p.) Yhteenveto 3 p. 2006.

65 Viinikainen, Ari, Quality of service and pricingin future multiple service class networks. 61 p. (196 p.) Yhteenveto 1 p. 2006.

66 Wu, Rui, Methods for space-time parameter estimation in DS-CDMA arrays. 73 p. (121 p.) 2006.

67 Parkkola, Hanna, Designing ICT for mothers. User psychological approach. – Tieto- ja viestintätekniikoiden suunnittelu äideille. Käyttäjäpsykologinen näkökulma. 77 p. (173 p.) Yhteenveto 3 p. 2006.

68 Hakanen, Jussi, On potential of interactive multiobjective optimization in chemical process design. 75 p. (160 p.) Yhteenveto 2 p. 2006.

69 Puttonen, Jani, Mobility management in wireless networks. 112 p. (215 p.) Yhteenveto 1 p. 2006.

70 Luostarinen, Kari, Resource , management methods for QoS supported networks. 60 p. (131 p.) 2006.

71 Turchyn, Pavlo, Adaptive meshes in computer graphics and model-based simulation. 27 p. (79 p.) Yhteenveto 1 p.

72 Zhovtobryukh, Dmytro, Context-aware web service composition. 290 p. Yhteenveto 2 p. 2006.

JYVÄSKYLÄ STUDIES IN COMPUTING

73 Kohvakko, Nataliya, Context modeling and utilization in heterogeneous networks. 154 p. Yhteenveto 1 p. 2006.

74 Mazhelis, Oleksiy, Masquerader detection in mobile context based on behaviour and environment monitoring. 74 p. (179 p). Yhteenveto 1 p. 2007.

75 Siltanen, Jarmo, Quality of service and dynamic scheduling for traffic engineering in next generation networks. 88 p. (155 p.) 2007.

76 Kuuva, Sari, Content-based approach to experiencing visual art. - Sisältöperustainen lähestymistapa visuaalisen taiteen kokemiseen. 203 p. Yhteenveto 3 p. 2007.

77 Ruohonen, Toni, Improving the operation of an emergency department by using a simulation model. 164 p. 2007.

78 Naumenko, Anton, Semantics-based access control in business networks. 72 p. (215 p.) Yhteenveto 1 p. 2007.

79 Wahlstedt, Ari, Stakeholders' conceptions of learning in learning management systems development. - Osallistujien käsitykset oppimisesta oppimisympäristöjen kehittämisessä. 83 p. (130 p.) Yhteenveto 1 p. 2007.

80 Alanen, Olli, Quality of service for triple play services in heterogeneous networks. 88 p. (180 p.) Yhteenveto 1 p. 2007.

81 Neri, Ferrante, Fitness diversity adaptation in memetic algorithms. 80 p. (185 p.) Yhteenveto 1 p. 2007.

82 Kurhinen, Jani, Information delivery in mobile peer-to-peer networks. 46 p. (106 p.) Yhteenveto 1 p. 2007.

83 Kilpeläinen, Turo, Genre and ontology based business information architecture framework (GOBIAF). 74 p. (153 p.) Yhteenveto 1 p. 2007.

84 Yevseyeva, Iryna, Solving classification problems with multicriteria decision aiding approaches. 182 p. Yhteenveto 1 p. 2007.

85 Kannisto, Isto, Optimized pricing, QoS and segmentation of managed ICT services. 45 p. (111 p.) Yhteenveto 1 p. 2007.

86 Gorshkova, Elena, A posteriori error estimates and adaptive methods for incompressible viscous flow problems. 72 p. (129 p.) Yhteenveto 1 p. 2007.

87 Legrand, Steve, Use of background real-world knowledge in ontologies for word sense disambiguation in the semantic web. 73 p. (144 p.) Yhteenveto 1 p. 2008.

88 Hämäläinen, Niina, Evaluation and measurement in enterprise and software architecture management. - Arviointi ja mittaaminen kokonais- ja ohjelmistoarkkitehtuurien hallinnassa. 91 p. (175 p.) Yhteenveto 1 p. 2008.

89 Ojala, Arto, Internationalization of software firms: Finnish small and medium-sized software firms in Japan. 57 p. (180 p.) Yhteenveto 2 p. 2008.

90 Laitila, Erkki, Symbolic Analysis and Atomistic Model as a Basis for a Program Comprehension Methodology. 321 p. Yhteenveto 3 p. 2008.

91 Nihtilä, Timo, Performance of Advanced Transmission and Reception Algorithms for High Speed Downlink Packet Access. 93 p. (186 p.) Yhteenveto 1 p. 2008.

92 Setämaa-Kärkkäinen, Anne, Network connection selection-solving a new multiobjective optimization problem. 52 p. (111p.) Yhteenveto 1 p. 2008.

93 Pulkkinen, Mirja, Enterprise architecture as a collaboration tool. Discursive process for enterprise architecture management, planning and development. 130 p. (215 p.) Yhteenveto 2 p. 2008.

94 Pavlova, Yulia, Multistage coalition formation game of a self-enforcing international environmental agreement. 127 p. Yhteenveto 1 p. 2008.

95 Nousiainen, Tuula, Children's involvement in the design of game-based learning environments. 297 p. Yhteenveto 2 p. 2008.

96 Kuznetsov, Nikolay V., Stability and oscillations of dynamical systems. Theory and applications. 116 p. Yhteenveto 1 p. 2008.

97 Khriyenko, Oleksiy, Adaptive semantic Web based environment for web resources. 193 p. Yhteenveto 1 p. 2008.

98 Tirronen, Ville, Global optimization using memetic differential evolution with applications to low level machine vision. 98 p. (248 p.) Yhteenveto 1 p. 2008.

99 Valkonen, Tuomo, Diff-convex combinations of Euclidean distances: A search for optima. 148 p. Yhteenveto 1 p. 2008.

100 Sarafanov, Oleg, Asymptotic theory of resonant tunneling in quantum waveguides of variable cross-section. 69 p. Yhteenveto 1 p. 2008.

101 Pozharskiy, Alexey, On the electron and phonon transport in locally periodical waveguides. 81 p. Yhteenveto 1 p. 2008.

102 Aittokoski, Timo, On challenges of simulation-based globaland multiobjective optimization. 80 p. (204 p.) Yhteenveto 1 p. 2009.

103 Yalaho, Anicet, Managing offshore outsourcing of software development using the ICT-supported unified process model: A cross-case analysis. 91 p. (307 p.) Yhteenveto 4 p. 2009.

104 K ollanus, Sami, Tarkastuskäytänteiden kehittäminen ohjelmistoja tuottavissa organisaatioissa. - Improvement of inspection practices in software organizations. 179 p. Summary 4 p. 2009.

105 Leikas, Jaana, Life-Based Design. 'Form of life' as a foundation for ICT design for older adults. - Elämälähtöinen suunnittelu. Elämänmuoto ikääntyville tarkoitettujen ICT tuotteiden ja palvelujen suunnittelun lähtökohtana. 218 p. (318 p.) Yhteenveto 4 p. 2009.

JYVÄSKYLÄ STUDIES IN COMPUTING

106 Vasilyeva, Ekaterina, Tailoring of feedback in web-based learning systems: Certitude-based assessment with online multiple choice questions. 124 p. (184 p.) Yhteenveto 2 p. 2009.

107 Kudryashova, ElenaV., Cycles in continuous and discrete dynamical systems. Computations, computer assisted proofs, and computer experiments. 79 p. (152 p.) Yhteenveto 1 p. 2009.

108 Blackledge, Jonathan, Electromagnetic scattering and inverse scattering solutions for the analysis and processing of digital signals and images. 297 p. Yhteenveto 1 p. 2009.

109 Ivannikov, Andriy, Extraction of event-related potentials from electroencephalography data. - Herätepotentiaalien laskennallinen eristäminen EEG-havaintoaineistosta. 108 p. (150 p.) Yhteenveto 1 p. 2009.

110 kalyakin, Igor, Extraction of mismatch negativity from electroencephalography data. - Poikkeavuusnegatiivisuuden erottaminen EEG-signaalista. 47 p. (156 p.) Yhteenveto 1 p. 2010.

111 Heikkilä, Marikka, Coordination of complex operations over organisational boundaries. 265 p. Yhteenveto 3 p. 2010.

112 Fekete, Gábor, Network interface management in mobile and multihomed nodes. 94 p. (175 p.) Yhteenveto 1 p. 2010.

113 Kujala, Tuomo, Capacity, workload and mental contents - Exploring the foundations of driver distraction. 146 p. (253 p.) Yhteenveto 2 p. 2010.

114 Lugano, Giuseppe, Digital community design - Exploring the role of mobile social software in the process of digital convergence. 253 p. (316 p.) Yhteenveto 4 p. 2010.

115 Kampylis, Panagiotis, Fostering creative thinking. The role of primary teachers. - Luovaa ajattelua kehittämässä. Alakoulun opettajien rooli. 136 p. (268 p.) Yhteenveto 2 p. 2010.

116 Toivanen, Jukka, Shape optimization utilizing consistent sensitivities. - Muodon optimointi käyttäen konsistentteja herkkyyksiä. 55 p. (130p.) Yhteenveto 1 p. 2010.

117 Mattila, Keijo, Implementation techniques for the lattice Boltzmann method. - Virtausdynamiikan tietokonesimulaatioita Hila-Boltzmann -menetelmällä: implementointi ja reunaehdot. 177 p. (233 p.) Yhteenveto 1 p. 2010.

118 Cong, Fengyu, Evaluation and extraction of mismatch negativity through exploiting temporal, spectral, time-frequency, and spatial features. - Poikkeavuusnegatiivisuuden (MMN) erottaminen aivosähkönauhoituksista käyttäen ajallisia, spektraalisia, aika-taajuus - ja tilapiirteitä. 57 p. (173 p.) Yhteenveto 1 p. 2010.

119 Liu, Shenghua, Interacting with intelligent agents. Key issues in agent-based decision support system design. 90 p. (143 p.) Yhteenveto 2 p. 2010.

120 Airaksinen, Tuomas, Numerical methods for acoustics and noise control. - Laskennallisia menetelmiä akustisiin ongelmiin ja melunvaimennukseen. 58 p. (133 p.) Yhteenveto 2 p. 2010.

121 Weber, Matthieu, Parallel global optimization Structuring populations in differential evolution. - Rinnakkainen globaalioptimointi. Populaation rakenteen määrittäminen differentiaalievoluutiossa. 70 p. (185 p.) Yhteenveto 2 p. 2010.

122 Väärämäki, Tapio, Next generation networks, mobility management and appliances in intelligent transport systems. - Seuraavan sukupolven tietoverkot, liikkuvuuden hallinta ja sovellutukset älykkäässä liikenteessä. 50 p. (111 p.) Yhteenveto 1 p. 2010.

123 Viukari, Leena, Tieto- ja viestintätekniikkavälitteisen palvelun kehittämisen kolme diskurssia. - Three discourses for an ICT-service development . 304 p. Summary 5 p. 2010.

124 Puurtinen, Tuomas, Numerical simulation of low temperature thermal conductance of corrugated nanofibers. - Poimutettujen nanokuitujen lämmönjohtavuuden numeerinen simulointi matalissa lämpötiloissa . 114 p. Yhteenveto 1 p. 2010.

125 Hiltunen, Leena, Enhancing web course design using action research . - Verkko-opetuksen suunnittelun kehittäminen toimintatutkimuksen keinoin . 192 p. Yhteenveto 2 p. 2010.

126 Aho, Kari, Enhancing system level performance of third generation cellular networks through VoIP and MBMS services. 121 p. (221 p.). Yhteenveto 2 p. 2010.

127 Häkkinen, Markku, Why alarms fail. A cognitive explanatory model. 102 p. (210 p.). Yhteenveto 1 p. 2010.

128 Pennanen, Anssi, A graph-based multigrid with applications. - Graafipohjainen monihilamenetelmä sovelluksineen. 52 p. (128 p.). Yhteenveto 2 p. 2010.

129 Ahlgren, Riikka, Software patterns, organizational learning and software process improvement. 70 p. (137 p.). Yhteenveto 1 p. 2011.

130 Nikitin, Sergiy, Dynamic aspects of industrial middleware architectures 52 p. (114 p.). Yhteenveto 1 p. 2011.

131 Sindhya, Karthik, Hybrid Evolutionary Multi-Objective Optimization with Enhanced Convergence and Diversity. 64 p. (160 p.). Yhteenveto 1 p. 2011.

JYVÄSKYLÄ STUDIES IN COMPUTING

132 Mali, Olli, Analysis of errors caused by incomplete knowledge of material data in mathematical models of elastic media. 111 p. Yhteenveto 2 p. 2011.

133 Mönkölä, Sanna, Numerical Simulation of Fluid-Structure Interaction Between Acoustic and Elastic Waves. 136 p. Yhteenveto 2 p. 2011.

134 Puranen, Tuukka, Metaheuristics Meet Metamodels. A Modeling Language and a Product Line Architecture for Route Optimization Systems. 270 p. Yhteenveto 1 p. 2011.

135 Mäkelä, Jukka, Mobility Management in Heterogeneous IP-networks. 86 p. (145 p.) Yhteenveto 1 p. 2011.

136 Savolainen, Paula, Why do software development projects fail? Emphasising the supplier's perspective and the project start-up. 81 p. (167 p.) Yhteenveto 2 p. 2011.

137 Kuznetsova, Olga, Lyapunov quantities and limit cycles in two-dimensional dynamical systems: analytical methods, symbolic computation and visualization. 80 p. (121 p.) Yhteenveto 1 p. 2011.

138 Kozlov, Denis, The quality of open source software and its relation to the maintenance process. 125 p. (202 p.) Yhteenveto 1 p. 2011.

139 Iacca, Giovanni, Memory-saving optimization algorithms for systems with limited hardware. 100 p. (236 p.) Yhteenveto 1 p. 2011.

140 Isomöttönen, Ville, Theorizing a one-semester real customer student software project course. 189 p. Yhteenveto 1 p. 2011.

141 Hartikainen, Markus, Approximation through interpolation in nonconvex multiobjective optimization. 74 p. (164 p.) Yhteenveto 1 p. 2011.

142 Mininno, Ernesto, Advanced optimization algorithms for applications in control engineering. 72 p. (149 p.) Yhteenveto 1 p. 2011.

143 Tykhomyrov, Vitaliy, Mitigating the amount of overhead arising from the control signaling of the IEEE 802.16 OFDMa System. 52 p. (138 p.) Yhteenveto 1 p. 2011.

144 Maksimainen, Johanna, Aspects of values in human-technology interaction design—a content-based view to values. - Ihmisen ja teknologian vuorovaikutussuunnittelun arvoulottuvuudet—sisältöperustainen lähestymistapa arvoihin. 111 p. (197 p.) Yhteenveto 2 p. 2011.

145 Juutinen, Sanna, Emotional obstacles of e-learning. 97 p. (181 p.) Yhteenveto 3 p. 2011.

146 Tuovinen, Tero, Analysis of stability of axially moving orthotropic membranes and plates with a linear non-homogeneous tension profile. 104 p. Yhteenveto 1 p. 2011.

147 Hilgarth, Bernd, The systemic cognition of e-Learning success in internationally operating organizations. - Kokonaisvaltainen käsitys e-oppimisen menestyksestä kansainvälisissä organisaatioisssa. 100 p. (181 p.) Yhteenveto 1 p. 2011.

148 Jeronen, Juha, On the mechanical stability and out-of-plane dynamics of a travelling panel sub-merged in axially flowing ideal fluid. A study into paper production in mathematical terms. - Ideaali virtaukseen upotetun, aksiaalisesti liikkuvan paneelin mekaani-sesta stabiilisuudesta ja dynamiikasta. Tutkimus paperintuotannosta matemaattisin käsittein. 243 p. Yhteenveto 3 p. 2011.

149 Finne, Auvo, Tanzanit - Towards a comprehensive quality meta-model for information systems: Case studies of information system quality modelling in East Africa. 209 p. Yhteenveto 2 p. 2011.

150 Kankaanpää, Irja, IT Artefact Renewal: Triggers, Timing and Benefits. 79 p. (164 p.) Yhteenveto 1 p. 2011.

151 Kotilainen, Niko, Methods and Applications for Peer-to-Peer Networking. 46 p. (133 p.) Yhteenveto 1 p. 2011.

152 Skrypnyk, Iryna, Unstable feature relevance in classification tasks. - Epävakaiden ominaisuuksien merkitys luokittelutehtävissä. 232 p. Yhteenveto 1 p. 2011.

153 Zaidenberg, Nezer Jacob, Applications of virtualization in systems design. 297 p. Yhteenveto 1 p. 2012.

154 Martikainen, Henrik, PHY and MAC Layer Performance Optimization of the IEEE 802.16 System. 80 p. (150 p.)Yhteenveto 1 p. 2012.