Overhead of ARQ mechanism in IEEE 802.16 networks

Zdenek Becvar, Robert Bestak

Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Telecommunication Engineering, Technicka 2, 16627 Prague, Czech Republic

+420 2 2435 5994

+420 2 2333 9810

becvaz1@fel.cvut.cz

http://www.cvut.cz/en

Different techniques for error correction such as automatic repeat request, forward error correction or combination of both methods also known as hybrid automatic repeat request are utilized in today's wireless networks. All these techniques require an implementation of control and management mechanisms to ensure its proper work. These mechanisms increase the management overhead and consume a part of network capacity. It leads to the reduction of data throughput dedicated to users. Besides, the control mechanism can negatively influence a packet delay. This paper proposes three alternatives of automatic repeat request mechanism to minimize the management overhead without a negative impact on the packet delay. Description of all proposed mechanisms, their evaluation and comparison with conventional techniques used in the IEEE 802.16 networks are presented in this paper.

ARQ, error correction, MAC, overhead, WiMAX

1 Introduction

Generally, errors occur during data transmission in wireless networks. Erroneous packets cannot be used for further processing without a method of correction. For this purpose, wireless networks commonly use techniques based either on Automatic Repeat reQuest (ARQ) or Forward Error Correction (FEC) method. The ARQ mechanism uses a feedback channel for the confirmation of error-free packet delivery or for packet retransmission request. This method can increase a network throughput if radio channel conditions are getting worse [1]. On the other hand, the ARQ method increases the delay of packets by time spent for the retransmission of erroneous packets. The FEC method allows an increase in user's data throughput within an impaired channel quality by adding redundant coding information on the transmitter side. Both methods can be combined to Hybrid ARQ (HARQ) used for example in WiMAX (Worldwide Interoperability Microwave Access) [2] networks according to IEEE 802.16e [3], IEEE 802.16j [4], IEEE 802.16m [5] or in LTE (Long Term Evolution) [6] networks.

All the above mentioned methods should implement a mechanism on link layer that control and manage these techniques to achieve optimal performance on a wireless link. The ARQ in WiMAX networks is mandatory supported by all Base Stations (BS) as well as by all Mobile Stations (MS) according to [7].

The performance of ARQ defined in [3] depends on the setting of parameters such as size of user data carried in a frame, size of ARQ block, size of PDU (Packet Data Unit), value of retransmission timeout timer or on the type of packet acknowledgement [8]. Evaluation of the type of packet acknowledgment for different channel condition is presented in [9] by Kang and Jang. Tykhomyrov et. al. evaluate the ARQ performance for different ARQ parameters [10]. This work is later on enhanced about analysis of the impact of PDU size on IEEE 802.16e networks performance while ARQ mechanism is used by Martikainen et. al. [11]. Sayenko et. al. provide a comparison of ARQ and HARQ performance in IEEE 802.16 networks [12]. This paper also compares the overhead size generated by ARQ and HARQ. The optimal PDU size and MAC (Medium Access Control) overhead due to the packets retransmission is analyzed by Hoymann [13]. Sengupta et. al. [14] propose to adjust the MAC PDU size depending on the channel state to achieve the best ARQ performance. The paper is extended for analysis of a combination of error correction techniques such as ARQ, FEC or MAC PDU aggregation on the VoIP speech quality [15]. Authors proofs, the improvement of VoIP speech quality by using this techniques. Chen and De Marca [16] investigate an optimization of ARQ parameter setting from the link throughput point of view. Kliazovich et. al. propose a cross-layer ARQ mechanism, which substitutes the transmission of TCP acknowledgment packet with a short request transmitted on the MAC layer of the wireless link [17]. The cross-layer approach is investigated also in [18] by Krishnamachari et. al. The authors proposes a novel adaptive cross-layer protection strategy for video transmission. Combination of applicagtion layer FEC and MAC layer ARQ and optimum setting of parameters of those techniques is also investigated in this paper.

Most of the above mentioned papers focus on the investigation of optimal ARQ parameters setting, but not too much research has been done on possible optimization of the ARQ procedure currently used on MAC layer. This paper discusses the optimization of ARQ mechanism used on WiMAX MAC layer. The optimization is done from the overhead reduction point of view while the packet delay is not negatively affected in comparison to conventional IEEE 802.16e ARQ.

The remainder of this paper is organized as follows. The next section provides an overview of ARQ mechanism according to IEEE 802.16e. The second section also describes all proposed schemes on ARQ improvement. The third section describes the simulation scenario and parameters of wireless link. Following section discusses the results and presents an efficiency of proposed ARQ schemes. Last section presents our conclusions and future work plans.

2 ARQ mechanism

The ARQ method is used for the retransmission of data units (packets, blocks) received with errors that cannot be corrected by error correction mechanisms (e.g., FEC). This method assumes a segmentation of user data into blocks and their transmission to the receiver. The receiver checks received data for errors and reply to the transmitter with request on retransmission of erroneous blocks. The channel quality can be expressed by a parameter representing the ratio between all transmitted blocks and blocks received with errors – Block Error Rate (BLER). The method of transmission of retransmission requests used by ARQ is different in different wireless networks. The conventional IEEE 802.16e ARQ and the proposed methods are described in the following subsections.

2.1 Conventional ARQ according to IEEE 802.16e

The conventional IEEE 802.16e ARQ mechanism works on general basis described above. Each burst from a user carried in a frame is segmented into PDUs. A PDU consists of several blocks N_{block}. The number of blocks is given by following equation:

$$N_{blocks} = \frac{S_{data}}{S_{ARQ\_block}} \qquad (1)$$

where S_{data} is a total size of data in one frame by one user in bytes, parameter S_{ARQ_block} represents a block size defined by parameter denoted in the standard as ARQ_Block_Size in bytes [3]. This parameter is carried in TLV (Type/Length/Value) section of registration messages (REG-REQ/RSP) exchanged between BS and MS during a process of a dynamic service addition or modification (see [3]). The parameter ARQ_Block_Size can take values from the following range: 16, 32, 64, 128, 256, 512 and 1024 bytes. The ARQ block is smallest unit considered in case of ARQ transmission. A sequence of consecutive blocks is transmitted in the MAC PDU. The receiver checks the received data and sends an acknowledgment (ACK) feedback message to the transmitter. Each ARQ block is acknowledged individually. The feedback is sent in the subsequent frame after the data transmission. The feedback message contains 1 byte (8 bits) field Message ID and field ARQ_Feedback_Payload. The ARQ payload can be carried either by using standalone ARQ feedback message or by piggybacking the ARQ payload to the user's data block. The payload is always carried in a single PDU. The ARQ_Feedback_Payload consists of one or more Information Elements (IE) carried by ARQ_Feedback_IE (see Fig. 1).

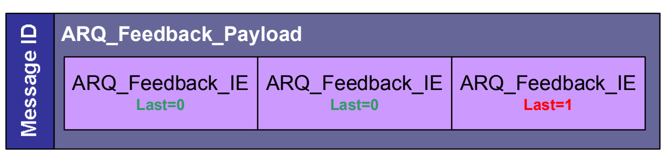

Fig. 1. Transmission of ARQ feedback including several ARQ_Feedback_IE

The structure and content of ARQ_Feedback_IE is presented in Table 1. Every ARQ_Feedback_IE is related to just one CID (Connection ID). The IEs in payload are not numbered. The last IE is identified by setting up the field "Last" equal to 1. All ARQ blocks are consecutively numbered by BSN (Block Sequence Number) to identify the position of transmission error. WiMAX enables to use four different types of acknowledgment (ACK Type) and two kinds of acknowledgment maps (ACK Map). The principle of individual ACK Types and ACK Maps are explained later in this section.

Table 1. Structure of ARQ_Feedback_IE [3].

Syntax	Size	Notes
CID	16 bits	Connection ID
Last	1 bit	Identify the last IE in ARQ_Feedback
ACK Type	2 bits	0x0...Selective ACK 0x1...Cumulative ACK 0x2...Cumulative with Selective 0x3...Cumulative with Block Sequence
BSN	11 bits	Block Sequence Number (0...2047)
Number of ACK Map	2 bits	Number of Maps (M) = 1,2,3 or 4
Maps	M x 16 bits	Selective (16 blocks) or Cumulative maps (2 x 64 blocks / 3 x 16 blocks) Cumulative maps: 1 bit sequence format (2 or 3 blocks), 2/3bits Sequence ACK (ACK/NACK of sequence), (2x6) / (3x4) bits Sequence length

The size of IE of each ARQ feedback message (in bits) can be calculated according to the next equation:

$$Size_{ARQ\_FB\_IE} = 32 + \left(M \times 16\right) \quad (2)$$

where M represents a number of maps carried in one ARQ_Feedback_IE (see Table 1). The overall size (in bits) of whole feedback message is given in (3).

$$Size_{ARQ\_FB} = 8 + \sum^{N_{IE}} Size_{ARQ\_FB\_IE_{N_{IE}}} \quad (3)$$

where N_{IE} is a number of information elements carried in one ARQ Feedback message and the number *8* (bits) represents the ARQ feedback message overhead (Message ID field). The overhead transmitted in all considered frames (N_{frame}) is equal to the sum of partial overheads over the N_{frame}:

$$OH_{ConvARQ} = \sum^{N_{frame}} Size_{ARQ\_FB_{Nframe}} \quad (4)$$

The $OH_{ConvARQ}$ is presented in bits.

The principle of conventional ARQ method according to the IEEE 802.16e standard and the structure of user's information carried in the frame are depicted in Fig.2.

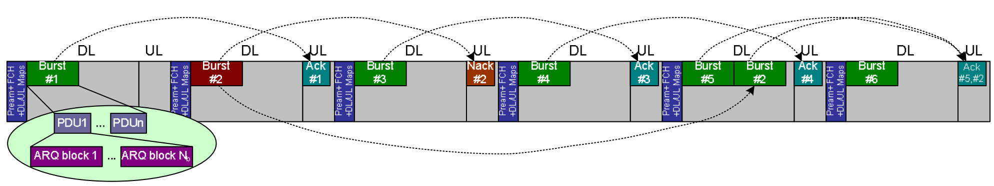

Fig. 2. Principle of conventional ARQ

All transmitted blocks have to be confirmed by ACK or by negative ACK (NACK) even if all blocks are received without errors. Therefore, the size of overhead per frame and per user depends especially on the number of blocks transmitted in one frame by one user. The IEEE 802.16e standard defines four types of acknowledgments: Selective ACK entry, Cumulative ACK entry, Cumulative with Selective ACK entry and Cumulative with Block Sequence ACK entry. The first type of acknowledgment (ACK Type = 0x0) uses selective maps to provide feedback to the transmitter. Each bit set to "1" in the selective map indicates error-free receiving of the corresponding ARQ block. The BSN corresponds to the most significant bit in the map. The second type (ACK Type = 0x1), Cumulative ACK entry, is based on the utilization of sequence maps. A sequence map defines a group of consecutive blocks where each group includes a sequence of only erroneous blocks or sequence of only error free blocks. The sequence maps can contain two or three sequences with a length of 64 or 16 blocks respectively. The third type of ACK (ACK Type = 0x2) combines the previous two types. Finally, the last type (ACK Type - 0x3) combines the second type with ability to acknowledge ARQ blocks in the form of block sequences.

The retransmission of erroneous blocks cannot be provided sooner than in the third block after the first transmission since the transmitter receives NACK in the following frame after transmission (2nd frame). Hence a request for additional resources can be created earliest at the next frame (3rd frame). Therefore, the dedicated resources are not available before the 4th frame. The retransmitted data (burst #2 in Fig. 2) can be transported either together with normally ordered data (burst #5 in Fig. 2) or the fresh data (burst #5 in Fig. 2) can be delayed by one frame. It causes a delay of retransmitted packets with duration that corresponds at least 3 times frame duration (e.g. if the frame duration is 10 ms, the packet delay is at least 30 ms).

Fresh data and retransmitted data are sent in one frame only if the requested capacity (fresh data + retransmitted data) is available. The WiMAX technology implements Stop-and-Wait mechanism that requests confirmation of the previous block before transmitting subsequent blocks. The number of blocks that can be unconfirmed before transmission of the consequent blocks is defined in the standard by parameter ARQ_Window_Size.

2.2 Proposed ARQ schemes

A number of blocks received with errors increases as decrease the link quality between transmitter and receiver. Thus, if the BLER increases, the amount of NACK blocks also increases. We can assume that the major part of links with enabled ARQ has an enough high quality to transfer most of blocks without errors (confirmed by ACK) than the number of blocks with errors (confirmed by NACK). In such case, the transmission of ACK blocks appears more often than NACK blocks. If we only consider the transmission of request for retransmission (NACK), the ARQ overhead can be significantly reduced. The above mentioned assumption is a basis for all following proposals.

2.2.1 ARQ Scheme I – Only Negative ACK

The first proposed scheme assumes ARQ feedback message and ARQ_Feedback_IEs with the same structure as the conventional IEEE 802.16 ARQ feedback message. However in this proposal, the ARQ feedback is only sent if a received PDU contains at least one erroneous block.

If all blocks in the PDU are error free, no feedback is sent (see Fig. 3). The PDU is assumed to be correctly transferred if the transmitter receives no feedback in the following W frames after the transmission (W=1 in Fig. 3). To define W, a new ARQ parameter, called in this paper ACK_Window, is introduced. If the block is received with errors, the ARQ feedback message is transmitted in the same way and with the same content as in the conventional ARQ. An error-free block is acknowledged after W frames at the latest. It means that if no NACK is received in the one of following W frames, the block is supposed correctly received, i.e. as it would be confirmed by ACK. Hence, the request for the retransmission of blocks with at least one error should be sent within one of the subsequent W frames. If the feedback with NACK is lost, the data belonging to the delay sensitive services (e.g. VoIP) are assumed lost since the delay caused by repeated ARQ retransmission is very significant. Therefore the proposal has no negative impact on packet delay. In case of services not sensitive to delay, data belonging to lost NACK can be retransmitted using upper layer protocols, e.g., TCP [19–21]. As the probability of lost packet together with the NACK feedback is very low, the increase of overhead due to upper layer protocols is negligible.

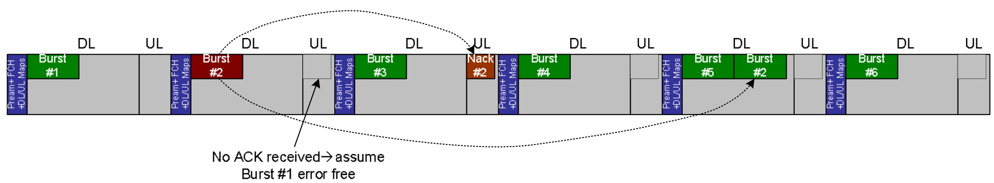

Fig. 3. Principle of proposed ARQ scheme I

This proposed scheme does not need any modification of ARQ MAC management messages, besides specification of the ACK_Window parameter. The description of this parameter is shown in Table 2. This parameter is carried in the registration messages (REG-REQ/RSP) and in messages related to the dynamic service management (DSx-REQ/RSP) [3]. These messages are only transmitted during a registration of a MS to the network and during change of dynamic services. Therefore, the increase of overhead is insignificant in comparison with the overhead generated by ARQ acknowledgements.

Table 2. Definition of the parameter ACK_Window

Length	Value	Description
1 byte	0...64	The number of frames after which the belonging block is considered as an error free (as it could be confirmed by ACK).

The size of ACK feedback message can be calculated according to equation (3). The overhead saving is achieved since not all transmissions have to be acknowledged by ARQ feedback message. The overall overhead of proposed ARQ scheme I is a sum of ARQ overhead created in each frame over number of frames. It can be calculated according to the next equation:

$$OH_{SchemeI} = \sum^{N_{frame}} Size_{ARQ\_FB_{Nframe}} \quad (5)$$

2.2.2 ARQ Scheme II – BSN of blocks with errors

The second proposed scheme is based on the same assumptions as the first one. The ACK feedback is likewise transmitted only if there is at least one erroneous block. A block is assumed to be error-free if no feedback is received in one of the following W frames after the transmission of appropriate data frame. Both proposals differ among each other in the structure of retransmission request. The ARQ scheme II slightly modifies the structure of ARQ_Feedback_IE. The conventional ARQ feedback message carries a set of ACK maps in conventional ARQ (see Table 1). Instead of these maps, the ARQ scheme II carries the set of BSNs related to erroneous blocks. Therefore, the ARQ feedback message contains only one IE field and the fields “Last” and “ACK Type” can be omitted. The format of modified ARQ feedback message is shown in Table 3.

Table 3. Structure of modified ARQ_Feedback_IE according to the proposal of ARQ scheme II

Syntax	Size	Notes
CID	16 bits	Connection ID
Number of BSNs	10 bits	Number of BSNs (B) = 1...1024
Set of BSN	B x 11 bits	Set of Block Sequence Numbers (0...2047)
reserve	0-8	Align a message length to bytes

The maximum number of BSNs can be 1024 since ARQ_Window_Size is the half of range of BSN (2^{11}=2048) [3]. In case of the ARQ scheme II, the size of ARQ feedback message is given by equation:

$$Size_{ARQ\_FB\_II} = 8 + 26 + (B \times 11) + res \quad (5)$$

where *B* is the number of BSNs included in a message and *res* is the number of bits used for alignment of the message length to bytes. Only one IE filed is always carried in an ARQ feedback message since IE can carry BSNs of all erroneous blocks.

The total overhead due to ARQ scheme II is a sum over the overhead in all frames within the transmission:

$$OH_{SchemeII} = \sum^{N_{frame}} Size_{ARQ\_FB\_II\ N_{frame}} \quad (6)$$

The ARQ scheme II reduces the overhead especially for low values of BLER. For high value of BLER can be assumed opposite idea – transmit only confirmation of error free packets (ACK). It will be profitable only for very high level of BLER (over approx. 80%) and it is almost impossible to reach this state in the real networks since the network would be overloaded by retransmitted packets.

2.2.3 ARQ Scheme III – Combination of ARQ Scheme I & II

The last proposed scheme, ARQ scheme III, is the combination of previous two. This scheme dynamically selects the best one from the ARQ scheme I, ARQ scheme II and conventional IEEE 802.16e ARQ.

The ARQ scheme III introduces a new field (denoted ARQ Scheme) in ARQ_Feedback_IE that is used to decide which of the schemes should be applied in given moment. The choice is based on the per frame calculation of minimum overhead generated in each frame. The modified structure of ARQ_Feedback_IE message is presented in Table 4.

Table 4. Structure of ARQ_Feedback_IE according to proposal III

Syntax	Size	Notes
CID	16 bits	Connection ID
ARQ Scheme	2 bits	0x0…Conventional ARQ (802.16e) 0x1…ARQ Scheme I 0x2…ARQ Scheme II 0x3…Reserve
if ARQ Scheme =0x0 or 0x1 {		
Last	1 bit	Identify the last IE in ARQ_Feedback
ACK Type	2 bits	0x0...Selective ACK 0x1...Cumulative ACK 0x2...Cumulative with Selective 0x3...Cumulative with Block Sequence
BSN	11 bits	Block Sequence Number (0...2047)
Number of ACK Map	2 bits	Number of Maps (M) = 1,2,3 or4
Maps	M x 16 bits	Selective (16 blocks) or Cumulative maps (2 x 64 blocks / 3 x 16 blocks) Cumulative maps: 1 bit sequence format (2 or 3 blocks), 2/3bits Sequence ACK(ACK/NACK of sequence), (2x6) / (3x4) bits Sequence length
}		
if ARQ Scheme =0x2 {		
Number of BSNs	10 bits	Number of BSNs (B) = 1...1024
Set of BSN	B x 11 bits	Set of Block Sequence Numbers (0...2047)
}		
reserve	0-8	Align a message length to bytes

With regards to the above mentioned structure of ARQ_Feedback_IE, the overhead generated by ARQ scheme III by a user in one frame can be calculated according to the following equation:

$$Size_{ARQ\_FB\_III} = 8 + 18 + min\left\{ \sum^{N_{IE}} 16 + 16 \times M_{N_{IE}}, 10 + B \times 11 \right\} + res \quad (8)$$

where N_{IE} is the number of IEs carried in one ARQ feedback message, M_{NIE} corresponds to the number of ACK maps in ARQ_Feedback_IE, B is the number of BSNs included in one message and res is the number of bits used for an alignment of feedback message length to bytes. The overhead generated by ARQ scheme III is given by the following equation:

$$OH_{SchemeIII} = \sum^{N_{frame}} Size_{ARQ\_FB\_III_{N_{frame}}} \quad (9)$$

All ARQ schemes are proposed with respect to need no additional hardware modifications of WiMAX equipments (MSs, BSs…) currently available at the market. All changes only implicate the MAC layer software modification.

3 Simulations

The following sections are focused on the comparison of conventional IEEE 802.16e ARQ and all proposals; therefore the delay of packets is not investigated since it is not affected by proposed techniques. The packet delay is not negatively influenced for delay sensitive services by all proposals since exchange of no additional MAC management messages in comparison to IEEE 802.16e is required (see description of proposals).

The link level calculation (developed in MATLAB) focuses on the evaluation of overhead generated by ARQ procedure in the uplink direction by one user (see Fig. 4).

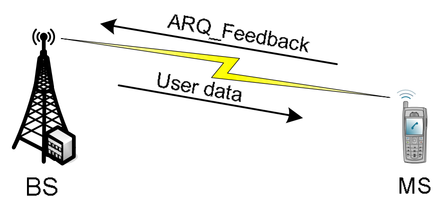

Fig. 4. Link level simulation scenario

At the beginning, all simulation and link parameters are set up according to values mentioned in Table 5.

Table 5. Simulation parameters

Parameter	Value
Number of frames	5 000
BLER [%]	0 – 20
ARQ_Block_Size [bytes]	16 / 64 / 256 / 1024
PDU size [blocks]	1 / 2 / 4 / 16 / 32
ACK Types	Selective, Cumulative
Size of data in each DL frame [bytes]	1024 / 4096

The number of frames represents an amount of frames transmitted from the BS to the MS. The overhead size is evaluated per all transmitted frames. A frame consists of one or several PDUs and a PDU itself contains one or several ARQ blocks. The frames are subsequently sent by the BS to the MS. A vector indicating positions of blocks with/without errors is created for each frame based on the given value of BLER. The MS responds to the BS by sending an ARQ feedback message that includes selected ARQ scheme, ACK Type and a vector of errors in the transmission. According to the feedback message, the BS retransmits erroneous blocks as soon as possible, but not sooner than in the third frame after the original transmission. The size of user's data in a downlink frame is kept same within the whole simulation run (1024 bytes or 4096 bytes).

This procedure is repeated until all frames are sent to the MS and the MS confirms error-free reception of all blocks. The same vectors indicating positions of blocks with/without errors are considered in all ARQ schemes. Parameters such as frame duration, ARQ_Window_Size and ACK_Window need not to be specified since they have no impact on the results.

4 Results

The results obtained from simulations are presented in Fig. 5 – Fig. 8. All figures describe dependence of the ARQ overhead to BLER. The *y* axes in figures indicate a recalculation of absolute values of the overhead into the relative values. As the reference value is taken the overhead of conventional ARQ with selective ACK and BLER= 0%. The total absolute overhead value is highlighted in figures with bold font. The total absolute overhead of ARQ with selective maps at BLER=0% only depends on the number of blocks per a frame; just this value influences the number of ACK maps carried in the ARQ feedback message of conventional ARQ. The reference value is about 508 kB for ARQ_Block_Size=16 B and the user data size equal to 1024 B. The corresponding referential value is about 273 kB while ARQ_Block_Size ≥64 B and the size of user data is equal to 1024 B. This value is independent on other parameters since the number of blocks per frame is always equal or less than 16. Therefore all ACKs/NACKs can be carried in one selective ACK map. The overhead of conventional ARQ varies according to BLER since the length of maps depends on the number of blocks in a belonging frame. The amount of blocks in the frames differs due to the retransmission of erroneous blocks.

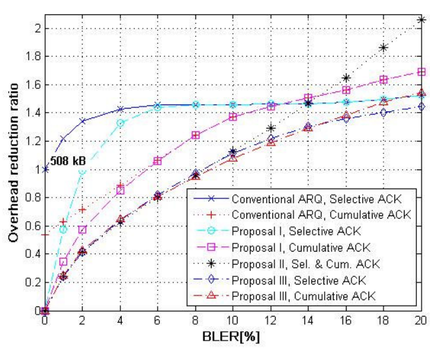

Fig. 5. ARQ Overhead vs. BLER for ARQ_Block_Size = 16 B, PDU Size = 1 block and Size of user data = 1024 B/ frame

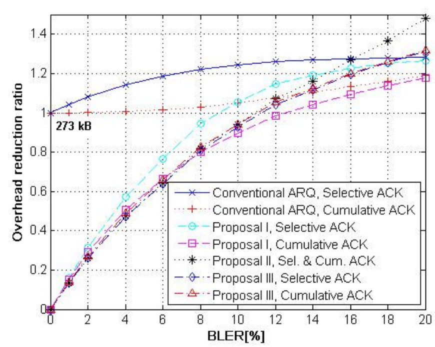

Fig. 6. ARQ Overhead vs. BLER for ARQ_Block_Size = 64 B, PDU Size = 1 block and Size of user data = 1024 B/ frame

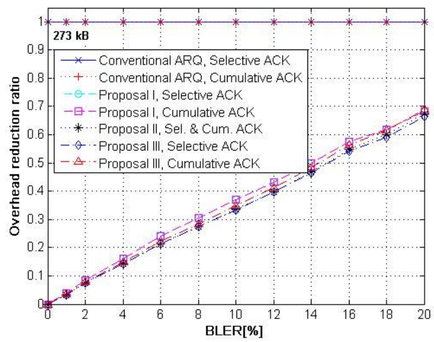

Fig. 7. ARQ Overhead vs. BLER for ARQ_Block_Size = 256 B, PDU Size = 1 block and Size of user data = 1024 B/ frame

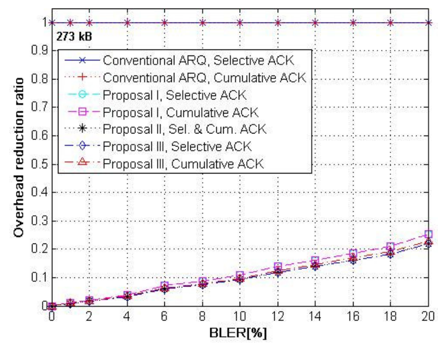

Fig. 8. ARQ Overhead vs. BLER for ARQ_Block_Size = 1024 B, PDU Size = 1 block and Size of user data = 1024 B/ frame

Fig. 5 – Fig. 8 show the impact of BLER on the size of ARQ overhead for different values of ARQ_Block_Size. Each figure compares the conventional ARQ using selective and cumulative ACK (solid line with cross mark and dotted line with plus mark respectively; both lines overlaps in Fig. 7 and Fig. 8) with three versions of ARQ proposals. As it could be assumed the cumulative ACK shows better performance for low BLER since the low BLER leads to the longer consecutive sequences of blocks with/without errors. Hence less ACK maps is required to confirm all blocks by cumulative maps in comparison with selective maps. The difference between selective and cumulative ACK becomes negligible as the value of ARQ_Block_Size increasing (compare same lines among Fig. 5 – Fig. 8).

The decrease of ARQ overhead using ARQ scheme I (dash line with circle or square marker) can be observed from all figures. The size of ARQ overhead decreases significantly for low values of BLER. The overhead of conventional and proposed ARQ scheme I converge with rising BLER. The ARQ overhead saving is more significant as simultaneously BLER and ARQ_Block_Size values become higher since the bigger block size leads to the lower number of blocks in a PDU. Therefore the number of retransmitted blocks is also decreased due to the lower number of block contained in one PDU (all blocks belonging to one PDU are retransmitted if there is at least one erroneous block). In case of ARQ scheme II (dotted line with asterisk marker), the overhead reduction is a more significant for the lower BLER in comparison with the ARQ scheme I. As BLER grows, the ARQ scheme II produces higher ARQ overhead than the conventional ARQ or ARQ scheme I. The impact is more significant when using high values of ARQ_BLOCK_Size. The ARQ scheme III (dash line with diamond or triangle marker) gives the best results since it selects the best ARQ scheme (from the overhead minimization point of view) for each block. In exceptional case, the overhead created by proposal III can be higher than overhead of the other proposals due to 2-bits field ACK_Scheme in IE (see Table 4).

From Fig. 5 – Fig. 8 can be observed reduction of absolute overhead for all proposed ARQ schemes as the value of ARQ_Block_Size increasing. Additionally, the efficiency of overhead reduction of all proposed ARQ schemes become more similar as the ARQ_Block_Size increasing.

The assumption of only 1 block per PDU was considered in all previous figures. The impact of different number of blocks per PDU on the overhead is shown in Fig. 9 – Fig. 12 (ARQ_Block_Size = 16 bytes in all figures). The results indicate a very significant overhead increase as the PDU Size and BLER grow. This increase is significant especially in case of ARQ scheme II. This is due to the retransmission behavior when an erroneous block results in retransmission of all blocks included in the same PDU. Hence, the ARQ scheme II is more suitable for low values of BLER and low number of blocks per PDU. The ratio of total ARQ overhead among conventional ARQ and the proposed schemes significantly depends on the value of BLER. When comparing the selective and cumulative ACK, the cumulative ACK is more suitable for very low values of BLER, with regards to PDU Size. In case of ARQ scheme III, the overhead is reduced in the whole simulated BLER range.

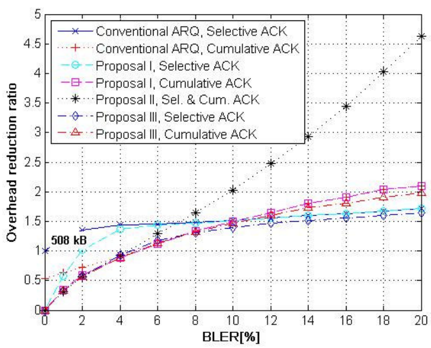

Fig. 9. ARQ Overhead vs. BLER for ARQ_Block_Size = 16 B, PDU Size = 2 blocks and Size of user data = 1024 B/ frame

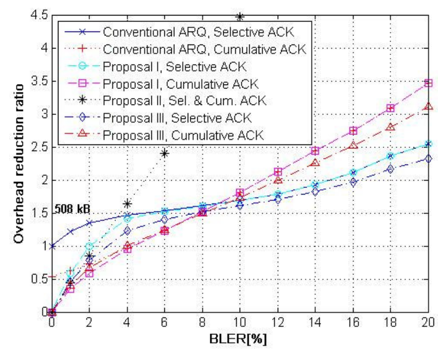

Fig. 10. ARQ Overhead vs. BLER for ARQ_Block_Size = 16 B, PDU Size = 4 blocks and Size of user data = 1024 B/ frame

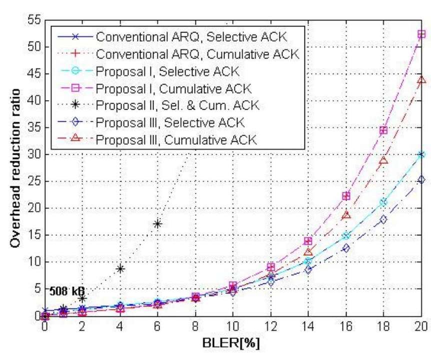

Fig. 11. ARQ Overhead vs. BLER for ARQ_Block_Size = 16 B, PDU Size = 16 blocks and Size of user data = 1024 B/ frame

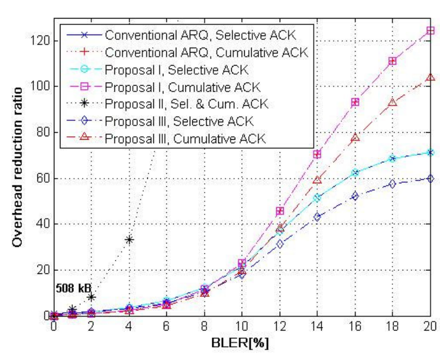

Fig. 12. ARQ Overhead vs. BLER for ARQ_Block_Size = 16 B, PDU Size = 32 blocks and Size of user data = 1024 B/ frame

The absolute total overhead, represented by overhead bitrate (OHBR), is given by the following equation:

$$OHBR = \frac{OH_{ABS}}{N_{frames} \times FD} \qquad (10)$$

where OH_{ABS} presents the absolute value of overhead over all frames, N_{frames} is the total number of considered frames and *FD* is the frame duration (WiMAX defines values [3]: FD ∈ {2; 2.5; 4; 5; 8; 10; 12.5 ; 20 ms}).

The absolute total overhead for BLER=0% per 5000 frames corresponds to the value marked in figures with bold font. This value is approximately 508 kbytes (4064 kbits) per 5000 frames while ARQ_Block_Size is 16 bytes. Hence, if we assume FD = 2ms we can achieve an increase in uplink throughput approximately about 406 kbps. The maximum downlink throughput for that scenario (1024 bytes/8192 bits per frame and 2 ms frame duration) is approximately 4096 kbps. Assuming a symmetrical service with 1024 bytes granted in each frame in uplink as well as in downlink, all proposals leads to the overhead reduction about 10% of the data throughput. If we consider the size of each ARQ block equal or higher than 64 bytes, the byte saving is about 273 kbytes (2184 kbits). In this case it corresponds to 218 kbps bitrate. The maximum user data throughput is about 4000 kbps (1024 bytes/8192 bits per frame and 2 ms frame) which results in 5.5% saving of user's data throughput.

As the above mentioned results show, proposal III outperforms conventional ARQ as well as the proposal I and proposal II in an absolute majority of scenarios for both: selective and cumulative acknowledgment. Therefore, subsequent text is focused on the analysis of ARQ overhead reduction by proposal III in comparison to the conventional ARQ according to IEEE 802.16e.

The overhead reduction of ARQ scheme III in comparison with the conventional ARQ for different block sizes is depicted in Fig. 13 (selective ACK) and Fig. 14 (cumulative ACK). Both figures present the results while user sent 1024 bytes in each frame and PDU contains one block. In case of Fig. 15 and Fig. 16, the frame size is set to 4096 bytes and PDU = 4 blocks. The proposed ARQ scheme III provides a significant overhead reduction (up to 100%) comparing to the conventional ARQ. The proposal can cause a small increase of overhead, but only for extremely high values of BLER and specific ARQ_Block_Size. However in these cases the overhead increase is marginal.

For the scenario with selective ACK, the overhead reduction shows better performance for the lower BLER and high ARQ_Block_Size except the combination of very high BLER (over 10%) and low ARQ_Block_Size. In this case the proposal getting slightly worse results than the conventional ARQ. The better performance of proposal III is apparent especially for very low BLER and high ARQ_Block_Size.

The cumulative ACK results show the highest overhead reduction for low BLER level. The increasing ARQ_Block_Size decreases ARQ overhead reduction effect of proposal III in case of combination of low BLER level and low ARQ_Block_Size. With an increase in the BLER level, the proposal shows better results for higher ARQ_Block_Size. The results of the overhead reduction for different BLER converge together for higher ARQ_Block_Size. The positive impact of proposal is decreasing with the increasing number of blocks in a PDU (compare Fig. 13 with Fig. 15 or Fig. 14 with Fig. 16).

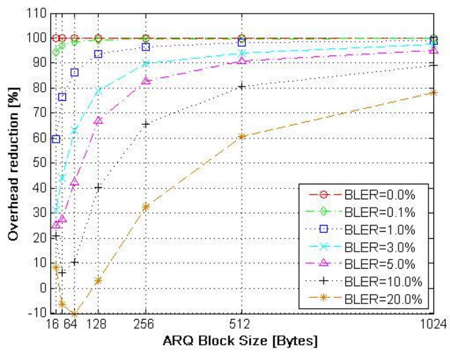

Fig. 13. Overhead reduction by ARQ scheme III vs. ARQ_Block_Size using selective ACK, PDU Size = 1 block and Size of user data = 1024 B/ frame

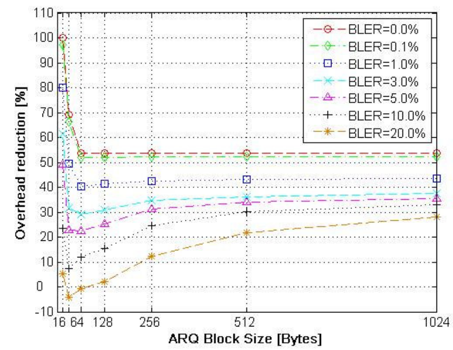

Fig. 14. Overhead reduction by ARQ scheme III vs. ARQ_Block_Size using cumulative ACK, PDU Size = 1 block and Size of user data = 1024 B/ frame

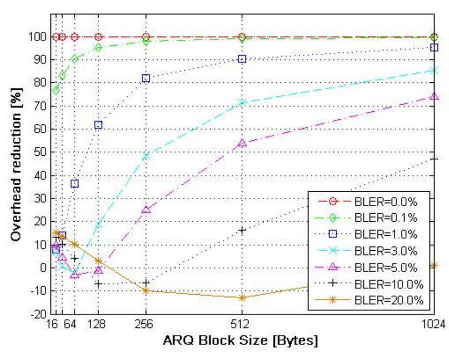

Fig. 15. Overhead reduction by ARQ scheme III vs. ARQ_Block_Size using selective ACK, PDU Size = 4 block and Size of user data = 4096 B/ frame

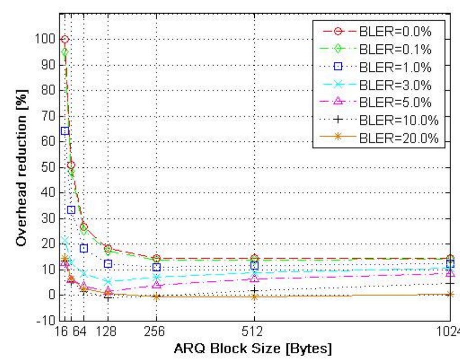

Fig. 16. Overhead reduction by ARQ scheme III vs. ARQ_Block_Size using cumulative ACK, PDU Size = 1 block and Size of user data = 4096 B/ frame

5 Conclusions and future work

The paper presents three types of proposal on reduction of the overhead caused by ARQ mechanism in WiMAX networks. The first proposed scheme transmitting only negative acknowledgment (NACK) to inform the transmitter about status of received blocks. The second proposed scheme is based on the first one and in addition it utilizes different type of block acknowledgment. Finally, the last proposal combines features of the conventional ARQ (according to IEEE 802.16e) and the fist two proposed schemes. All proposals reduce the overhead without negatively influence packet delay. The proposal III outperforms conventional ARQ as well as proposal I and II for selective and cumulative acknowledgment in absolute majority of scenarios. The overhead saving depends on several parameters such as BLER, ARQ_Block_Size parameter, PDU size and ACK type. The ARQ overhead reduction can reach up to 100% in comparison to the conventional IEEE 802.16e ARQ. Since the ARQ overhead influences the uplink channel throughput in dependence on the downlink channel quality, the throughput in the uplink can be enhanced up to about 10% of downlink throughput (e.g. the uplink throughput can be increase about 400 kbps if the user uses 4 Mbps in the downlink) when considering symmetric services. All proposals do not influence the throughput in downlink.

Our future work analyses the impact of relay station (new network entity introduced in IEEE 802.16j) deployment on the ARQ overhead. Another way of investigation is focused on the minimization of packet delay caused by ARQ since ARQ packet delay significantly increases when the user's requirements on throughput are getting closer to the maximum throughput offered by the BS.

ACKNOWLEDGMENT

This work has been performed in the framework of the FP7 project ROCKET IST-215282 STP, which is funded by the European Community. The Authors would like to acknowledge the contributions of their colleagues from ROCKET Consortium (http://www.ict-rocket.eu).

REFERENCES

[1] Sambale, K., Becvar, Z., Ulvan, A. (2008). Identification of the MAC/PHY key reconfiguration parameters, ICT ROCKET project milestone 5M2, ICT-215282 STP.
[2] Nuaymi, L. (2007). *WiMAX: Technology for Broadband Wireless Access*. West Sussex: Wiley&son Ltd.
[3] Standard IEEE 802.16e-2005. Air Interface for Fixed and Mobile Broadband Wireless Access Systems: Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands. February 2006
[4] Standard IEEE 802.16j Relay Task Group: IEEE 802.16j proposals.
[5] Standard IEEE802.16m Task Group: IEEE 802.16m proposals. Resource documents. http://www.ieee802.org/16/tgm/. Accessed 12 January 2009.
[6] 3GPP TS 36.321. LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification, 3GPP (version 8.2.0 Release 8). November 2008.
[7] WiMAX Forum. WiMAX ForumTM Mobile System Profile Specification (Release 1.5 Common Part), August 2009.
[8] Lee, B.G., Choi,S. (2008). *Broadband Wireless Access and Local Networks: Mobile WiMAX and WiFi*. USA: Artech House.
[9] Kang, M.-S., Jang, J. (2006). Performance evaluation of IEEE 802.16d ARQ algorithms with NS-2 simulator. Asia-Pacific Conference on Communications. doi:10.1109/APCC.2006.255785.
[10] Tykhomyrov, V., Sayenko, A., Martikainen, H., Alanen, O., Hämäläinen, T. (2007). *Performance Evaluation of the IEEE 802.16 ARQ Mechanism*. Next Generation Teletraffic and Wired/Wireless Advanced Networking (pp. 148-161). Berlin: Springer.
[11] Martikainen, H., Sayenko, A., Alanen, O., Tykhomyrov, V. (2008). Optimal MAC PDU size in IEEE 802.16. 4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks (pp. 66-71).
[12] Sayenko, A., Martikainen, H., Puchko, A. (2008). Performance comparison of HARQ and ARQ mechanisms in IEEE 802.16 networks. International symposium on Modeling, analysis and simulation of wireless and mobile systems. Vancouver, Canada.
[13] Hoymann, C. (2005). *Analysis and performance evaluation of the OFDM-based metropolitan area network IEEE 802.16*. Computer Networks, 49, 341-363.
[14] Sengupta, S., Chatterjee, M., Ganguly, S., Izmailov, R. (2005). Exploiting MAC Flexibility in WiMAX for Media Streaming. World of Wireless Mobile and Multimedia Networks, doi: 10.1109/WOWMOM.2005.41.
[15] Sengupta, S., Chatterjee, M., Ganguly, S. (2008). Improving Quality of VoIP Streams over WiMax. IEEE Transactions on Computers, 57(2), 145-156.
[16] Chen, K.C, De Marca, J.R.B. (2008). *Mobile WiMAX*. West Sussex: Wiley&son Ltd.

[17] Kliazovich, D., Tommaso, B., Dalsass, S., Serrelli, F., Redana, S., Granelli,F. (2008). Cross-Layer Error Control Optimization in WiMAX. Global Telecommunication Conference 2008, New Orleans, USA, 2008.
[18] Krishnamachari, S., van der Schaar, M., Choi, S., Xu, X. (2003) Video Streaming over Wireless LANs: A Cross-layer Approach. IEEE Packet Video Workshop, Nantes, France.
[19] RFC 793. Transmission Control Protocol, September 1981.
[20] RFC 791. Internet Protocol, September 1981.
[21] RFC 2460. Internet Protocol, Version 6 (IPv6) , December 1998.