Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**	
Title	**ARQ BLOCK SIZE in 802.16**	
Date Submitted	**April 15 2004**	
Source(s)	Radu Selea Redline Communications 302 Town Centre Blvd., Suite 100	Voice: 905 479 8344 (ext 223) Fax: 905 479 7432 mailto:radu@redlinecommunications.com
Re:	802.16REVd/D4 Sponsor Ballot	
Abstract	The document contains suggestion to revision the ARQ_BLOCK_SIZE value range	
Purpose	To Adopt the Change and introduce it on the base document	
Notice	This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.	
Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:chair@wirelessman.org> as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within the IEEE 802.16 Working Group. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.	

ARQ (ARQ_BLOCK_SIZE)

Radu Selea (Redline Communications)

References

[1] IEEE P802.16-REVd/D4-2004, IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems

Background

The way ARQ_BLOCK_SIZE is defined in 11.13.19.8 , creates confusion and impossible scenarios.
The range of this value is from: 1-byte → 2040 bytes.
As specified in the standard the receiver of the request must pick the minimum value from the negotiation between peer entities.
This requirement is correct and useful .The problem here is that any system has to comply with an ARQ_BLOCK_SIZE value that is coming from the peer entity if it is smaller.
In order to make sense we have to avoid possibility of enforcing ridiculous values only because the value is smaller. To do that, we have to limit and to define better the possible values that can be used.

Reasons for change

Reason number 1

The range of ARQ_WINDOW_SIZE values is: [0, (ARQ_BSN_MODULUS/2)] → [0,1024], as ARQ_BSN_MODULUS is 2048.
For 5 Mb/s (Maximum Sustained Rate), the traffic flowing through a connection queue and implicitly an ARQ window is: 5 Mb/s → 625 Bytes/ms at a constant traffic rate (based on the assumption that scheduler shapes the traffic).
If scheduler allows variable grants (averaging the rate over a time period), the situation get worse as for smaller periods of time the traffic entering the queue could exceed the rate.
The time required to get a **FAST** ARQ feedback is on average, minimum 3-4 frames in case that the feedback is sent immediately after receiver process the ARQ information.
Then minimum ARQ_BLOCK_LIFETIME should be an average of at least 4 frames in order to allow at least one retransmission.
Frame duration is defined in a range of 2 to 20 ms.
I will take as reference a frame of duration, 10 ms.
That means the ARQ_WINDOW must keep at minimum, references of:
625 Bytes/ms * 10 ms* 4 → 25000 Bytes
Based on the fact that maximum ARQ_WINDOW_SIZE is 1024 it results without any doubt that the if the ARQ_BLOCK_SIZE it smaller than 25 bytes the connection is strangled.

Even in case of slower connections the need for bigger lifetime is obvious as SS for instance may wait for polling in order to get an opportunity for feedback.
In case that ARQ_BLOCK_SIZE is too small, following things will happen:
ARQ becomes useless as lifetime expires before a feedback can be received QoS cannot be respected on that connection because ARQ settings.
Take in account that the examples are based on maximum ARQ_WINDOW_SIZE and average rates and frame durations.
For the same connection if I take in account, 20 ms frame duration and a lifetime of ~ 6 frames, calculation leads to: 625 Bytes/ms * 20 ms* 6 → 75000 Bytes in a 1024 window.
Then it seems by far that ARQ_BLOCK_SIZE of 64 bytes is a good choice for a minimum threshold.

Reason number 2

The smallest packet we can deal with on a data connection is 64 bytes.
There are strong implementation related reasons for:

- Defining minimum ARQ_BLOCK_SIZE as 64 Bytes
- Defining ARQ_BLOCK_SIZE as a power of 2

An easy and seamless mapping between 'virtual' ARQ_BLOCK_SIZE and physical memory pages (slots)
Easy packet parsing .No complicate calculations in HW to find block boundary. Shifts are easy.
BS that shall have to support multiple block sizes (as per negotiation) has a very difficult task if the broad range stands as it is now.
A reason for smaller granularity requirement does not stand as even in the case a system could fragment a 64 Bytes packet the overall data traffic is not improved, as packet needs to be reassembled. Only a very basic scheduler would require such a thing.

Reason number 3

The present value range of ARQ_BLOCK_SIZE can create a lot of confusion in the future especially on, Interoperability Certification tests (for instance any device should be tested to any block size which is smaller to its default value) in order to certify that ARQ is supported.

Specific change

11.13.19.8 ARQ_BLOCK_SIZE
This value of this parameter specifies the size of ARQ block. This parameter is established by negotiation during the connection creation and connection change dialogs.
The requester includes its desired setting in the REQ message. The receiver of the REQ message shall take the smaller of the value it prefers and value in the REQ message. This minimum value is included in the RSP message and becomes the agreed upon length value.
Absence of the parameter during a DSA dialog shall indicate the originator of the message desires the maximum value. Absence of the parameter during a DSC dialog indicates the current setting shall remain in force.

The size in bytes is obtained using the formula:
ARQ_BLOCK_SIZE=2^ Value → {64,128,256,512,1024,2048}

Type	Length	Value	Scope
[145/146]. 26.1.26	1	0-5 = reserved 6-11 12-255 = reserved	DSA-REQ, DSA-RSP REG-REQ, REG-RSP