Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**FEC Proposal**
Date Submitted	**2000-06-07 (Revision 0: 2000-06-06)**
Source(s)	Moshe Ran TelesciCOM, LTD 6 Hamachtesh st. Holon ISRAEL, 58810 Voice: +972-3-5589595/208 Fax: +972-3-5589091 mailto:Mran@telescicom.com David Williams Advanced Hardware Architectures 2365 NE Hopkins CT. Pullman,WA 99163 USA Voice:509 334 1000 Fax:509 334 9000 mailto:davew@aha.com
Re:	This is response to a Call for Contribution for BWA FEC solutions (IEEE802.16p-00/06)
Abstract	This contribution combines efforts from two independent sources to present BWA FEC solutions based on Block Turbo Codes (BTC).
Purpose	This document should be considered as a guideline for evaluation of BTC and Higher Order Modulations for both MODE A and MODE B of the current PHY draft (IEEE802.16.1pc-00/29r1)
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 text 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 (Version 1.0) <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, if there is technical justification in the opinion of the standards-developing committee and provided the IEEE receives assurance from the patent holder that it will license applicants under reasonable terms and conditions for the purpose of implementing 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:r.b.marks@ieee.org> as early as possible, in written or electronic form, of any patents (granted or under application) that may cover technology that is under consideration by or has been approved by IEEE 802.16. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/letters>.

Evaluation of Block-Turbo Codes and Higher Order Modulations for IEEE802.16.1 air interface standard

Moshe Ran, TelesciCOM, mran@telescicom.com
Dave Williams, Advanced Hardware Architecture, davew@aha.com

1. Introduction

This contribution addresses several aspects of Block Turbo Codes (BTC) and higher order modulations for Broadband wireless Access (BWA) systems. Part of this work has been introduced (see [1]-[2]) within the discussions of the Physical (PHY) layer of the evolving IEEE802.16.1 air interface [7]. We consider several topics relevant to IEEE802.16.1

- some aspects regarding Upstream (U/L) and Downstream (D/L) code parameters design
- general description of BTC schemes for IEEE802.16.1 draft standard
- fundamental limits on BTC decoding
- BER performance for several packet size and code rate
- Hardware implementation aspects as latency, maximum bit rate, flexibility to parameters changing, gate count, parallel implementation etc.

2. Advanced Coding and Modulation scheme using Variable Length Binary Product Codes and Soft-in/Soft-out (SISO) iterative decoding

2.1 Background

This document specifically addresses an advanced coding and modulation scheme by replacing the traditional Reed-Solomon (RS) outer code combined with Forney-type depth 12-byte interleaver and 64-states convolutional inner code by binary block product codes and use of Soft-in\Soft-out (SISO) iterative decoders (i.e BTC). Compatibility with baseline D/L modes of the DVB [3],[4] includes: transport multiplex adaptation, scrambling for energy dispersal. However, bit-to-symbol mapping is Gray coded for all QPSK, 16QAM and optionally 64QAM constellations. In this aspect the current proposal agrees with MODE B of the PHY draft of IEEE802.16.1 [7].

To maximize commonalties between the two essentially distinct PHY modes – MODE A and MODE B - the same core codes are suggested for both MODE A and B. As demonstrated later, continuous mode can benefit from better code design based on larger BTC. Both modes rely on short RS for the U/L. This is replaced with short BTC based on Hamming Product Codes (HPC) or Hamming\Parity check codes

We note that MODE A is essentially continuous TDM transmissions while MODE B supports a forward channel, which is TDD\TDMA. Thus larger block codes can be supported in MODE A. Unlike MODE A, MODE B can support higher level modulation and adaptive modulation more efficiently. The cost paid for having this mode is having burst transmissions and more stringent requirements on the interleaver size. For that reasons several other schemes which bypass the long interleaver were introduced (see [7 ch. 5.2]). These schemes are based on RS outer coding with very simple soft decodable block codes as an inner code. Other approaches based on Convolutional Turbo Codes (CTC) are proposed for the BRAN/HA (see e.g.,[8]). Apparently, there are several main drawbacks for the implementation of CTC for BWA channels.

- **Interleaver design**: differences of more than 1.5 dB between two identical $(13,15)_{oct}$ CTC with code rate 0.75 and interleaver size of 424bits at BER of 10^{-5} were reported in [8]. The only difference between the two interleavers was that one of them was "optimized" in the sense that the interleaved CTC attained larger minimum free distance (7 compared to 3 in the other case). In general, since the CTC lacks algebraic structure the optimization process for best interleaver should be done ad-hoc for each block size and each coding rate.
- **Puncturing**; usually puncturing of rate _ convolutional component codes is used to obtain higher code rates (e.g., 2/3, 4/5, 6/7). By that it meant that several redundant bits are deleted in couples, in both component codes. The effect of this process produces a smaller free distance for the punctured code and imposes additional constraints on the packet size (N should be a multiple of 3 for 2/3 and 6/7). The codes obtained in this way are much less effective than the equivalent block codes with similar parameters.
- **Tail bits trellis termination**: to ensure burst-based transmission the convolutional code constituting the CTC has to be used as a block code. This can be done by closing the trellis by adding m-zeros, where m is the memory length. Alternatively, a cyclic trellis closing can be used: the memory of the convolutional encoder is initialized with the last m bits to be transmitted. This method, so-called "tail biting" trellis, imposes additional restrictions on implementations of CTC for BWA channels.

The main advantage of CTC is having a Soft Output Viterbi Algorithm (SOVA) based on trellis representation for decoding the convolutional component codes. However, it is well known [9] that all linear codes have trellis representations. Furthermore, minimal trellis representations (I.e., trellis representation with minimum number of states, see [10]) and "tail biting" trellises are also inherently apply for block codes. Thus, SOVA techniques apply also for soft decoding of block codes. The crucial point is that there exists in addition to SOVA, more efficient decoding for block codes based on their nice algebraic structure.

BTC provides a simple approach for the design and implementation of SISO iterative decoders for BWA channels. In particular, BTC do not suffer from the drawback of long interleaver. Thus BTC can be used with high rate codes both for burst-like transmissions and continuous transmissions. Block interleaving can be traded with larger block sizes with improving performance for the continuous mode. The flexibility of frame size and code rates can be easily accomplished by applying well-understood algebraic tools. Asymptotic behavior is dictated by the number of minimum Hamming distance that can be calculated in a closed form for many component codes. The codes can be employed to give error detection capabilities. The applications of BTC to higher level modulations can be done using approach similar to Pragmatic TCM.

2.2 Variable Length BTC based on symmetric Hamming Product Codes (HPC)

A. Convention and notations

(n,k,d) is a linear block code of length n dimension k and minimum Hamming distance d. The ratio k/n is the code rate. In many cases we shall drop the last parameter and we shall refer to (n, k) block code.

(n_1,k_1,d_1)x (n_2,k_2,d_2) is a general representation of a block code with length $n=n_1*n_2$ dimension $k=k_1*k_2$ and minimum distance $d=d_1*d_2$ .The code constructed in this way is called a "product code" (or 2-D array code), and (n_i, k_i) for i=1,2 are called the components codes. The codewords of the product code can be described by an n_1 times n_2 rectangular array, where the columns are a codewords of code (n_1,k_1) and the rows are codewords of (n_2,k_2). The general product code based on shortened binary Hamming codes as component codes is given by

$(2^m -S1, 2^m -m -1 -S1, 4) \times (2^m -S2, 2^m -m -1 -S2, 4)$.

This code is called a Hamming Product Code, **HPC(m,S1,S2).**

The case where S1=S2 leads to a highly symmetric case since the columns encoder is identical to the rows encoder. Symmetric HPC, denoted **HPC(m, S),** are highly desired since they result in an implementation reduction.

2nd. Enhancing BTC: shortening, puncturing and extending

1-D shortening HPC means stuffing fixed bits (usually zero's) instead of information bits in each of the components code. The following sequence of operations is performed:

1. S1 zero bits are added as a prefix to each column followed by k_1 – S1 information bits.
2. S2 zero bits are added as a prefix to each row followed by k_2 - S2 information bits.
3. Encoding the columns by Hamming code (n_1,k_1) and the rows by Hamming code (n_2,k_2).
4. Discarding the S1 rows and S2 columns with fixed bits before transmission.
5. The decoder properly inserts zero's padding to the received block and attached very high soft metric value to each restored fixed bit.

This procedure applies for any binary linear product codes and does not change the minimum Hamming distance of the code. Shortening reduces both length (i.e., n) and dimension (i.e., k) of the mother code. The resulted code is an (n_1 - S1, k_1- S1, d_1) * (n_2 – S2, k_2- S2, d_2) product code.

Shortening can be based on 2-D shortening to obtain a better enhancement of the product codes parameters. 2-D shortening means replacing up to k_1 (or k_2) information bits along the diagonal of the k_1 * k_2 array of information.

Puncturing BTC means replacing parity bits with fixed (zero's) bits. This procedure is not recommended as 1-D operation for HPC since it decreases the minimum distance of the product code.
The following steps describe 1-D puncturing of a product code:

1. Encode the k_1 bits of information in each column followed by $n_1 – k_1$ redundancy bits using (n_1,k_1) code.
2. Replace the last P1 redundancy bits out of $n_1 – k_1$ bits in each column with P1 zero's.
3. Encode the rows using (n_2,k_2) code. Replace the last P2 redundancy bits out of $n_2 – k_2$ bits in each column with P2 zero's.
4. Discarding the P1 rows and P2 columns with fixed bits before transmission.
5. The decoder properly inserts zero's padding to the received block and attached a very low soft metric value to each restored fixed bit.

The resulted 1-D punctured product code is (n_1 – P1, k_1, d'_1) * (n_2 – P2, k_2, d'_2) product code where usually $d'_1 < d_1$ and $d'_2 < d_2$.

2-D puncturing product codes is best performed by replacing some of the $n_1 – k_1$ parity check bits with zero's along the diagonals at the redundancy part of the product code.

Extending product codes means adding extra redundancy bits to the columns (or rows) of the product code. It is possible in this case to perform 2-D extending by applying the extra parity bits along the diagonals of 2-D shortened product codes.

C Error detection capability of BTC

There are applications, where additional CRC is suggested within ARQ method to improve the overall system performance. This is usually accomplished by adding extra bytes for the detection of frame errors. BTC has inherent frame error detection capability without using extra CRC mechanism. The following steps can describe this mechanism:

1) Apply additional half-iteration (i.e. if SISO iterative decoding begins with row decoding followed by column decoding, then perform additional row decoding. Similarly, if SISO iterative starts with column decoding, then).

2) Verify that all rows are valid codewords of the row code.
3) Check if all first k columns are valid codewords in the column code.
4) If NO in step 3 – declare FRAME -ERROR, Otherwise – no errored frame detected.

A scheme similar to that was discussed in [11].

3.0 BTC schemes for the IEEE802.16.1 air interface standard and their performance

The following ECC schemes are presented analyzed. The results are detailed in the Annex.

3.1 D/L codes:

All the codes analyzed for downstream are based on the same core of symmetric HPC based on extended Hamming (64,57) code. These frame format are suggested to reflect the current frame formats of MODE B (BTC1) and MODE B (BTC2). The codes are designed to have high rates between 0.673 to 0.79. Enhancing these codes as explained above can yield wide range of symmetric HPC.

- BTC1 is a Symmetric Hamming Product code HPC(6,25)=(39,32,4)^2 designed to deliver 128 bytes of information. This code has the following parameters:
  N=39^2 =1521, K = 32^2= 1024, R=(32/39)^2=0.673 and d=16.
- BTC2 is based on a mother code HPC(6,18)=(46,39,4)^2 to deliver 188 bytes of information. The mother code is 17 bits 2-D shortened along the diagonal which results in a code with
  N=46^2 – 17 = 2099, K=39^2 –17=1504, R=0.717, and d=16.
- BTC3 is the largest HPC based on HPC(m=6, S) which can be used to deliver integer numbers of bytes, namely 392 bytes.
  BTC3 = HPC(6,1) = (63,56)^2 = (3969, 3136,16), with R=0.79

  BTC3 is given to demonstrate the unique feature, well known from coding theory, that better codes are approached by increasing the block length. Much larger data formats can be delivered by constructing larger BTC. This feature for continuous transmissions as the TDM of MODE A can be used to improve performance as alternative to large block interleaver.

3.2 U/L codes

In order to reflect the current needs of short burst transmissions as required in both MODE A and MODE B of IEEE802.16.1 draft PHY, several coding schemes were investigated.

- P5= 5 bytes: BTC5 is a very simple (linear complexity) high code rate (R=0.727) Turbo Parity check code based on (11,10)*(5,4) . Parity Product codes (PPC) have relatively low coding gain due to minimum Hamming distance 4.
- P5= 5 bytes: BTC5’ is based on (11,10)* (8,4) components codes which implies (88, 40, 8) code with rate 0.455.
- P6= 14 bytes: BTC6 is based on (29,23)*(6,5) components codes, where additional 3 bits were 2-D shortened along the diagonal. The resulted code is (171, 112, 8) code with code rate R=0.655.
- P6’=14 bytes: BTC6’ is a high coding gain symmetric HPC(m=4) based on Hamming (16,11,4) components code. The code is 2-D 9 bits shortened to match with the 14 bytes data frame constraint.
  The code parameters are (247,112,16).
- P7=57 bytes: BTC7 is a product code based on non symmetric 1-D shortening the Hamming (32,26,4) code. This product code (30,24)*(25,19)=(750, 456,16) has code rate 0.608 and can be used to
  protect ATM frame formats over BWA channels.

4.0 Hardware implementation aspects of BTC for BWA channels

4.1 degradation due to small number of iterations

The coding gain loss due to small number of iterations, for rate 0.793 @BER=10^{-6}, compared to maximum 32 iterations is given in Table below.

No. iteration	Loss QPSK	Loss 16QAM	Loss 64QAM
1	2.6 dB		
2	0.9 dB		
4	0.4 dB		
6	0.2 dB		
32	0 dB		

In many cases the terminating condition can be used to speed up the performance of the algorithm. At low (small values of) BER, when a BTC decoder identifies a valid codeword, early termination of the process is performed.

4.2 Degradation due to q-bits quantization of soft information

The proposed BTC has -

2-PAM (QPSK): q=4 bits (one bit for sign and 3 for the reliability)
4-PAM (16-QAM) q=5 bits (one bit for sign and 4 for the reliability)
8-PAM (64-QAM) q=6 bits (one bit for sign and 5 for the reliability)

Regarding 2-PAM, the loss depicted from having finite length (3 bits for reliability), compared to floating-point, is less than 0.2dB at BER=10^{-6}. Reducing the quantization of reliability to 2 bits resulted in a degradation of about 0.3dB at BER=10^{-6}.

4.3 Simplicity of HPC, parallel implementations:

All components codes proposed for D/L are derived from basic core based on (extended) Hamming code (64,57,4). This code has well understood soft Maximum-Likelihood (ML) decoders and Bounded Distance (BD) soft decoders (Chase, modified-Chase, GMD etc.). The complexity of serial implementation of HPC is roughly 64^2 more than that of the component code. However, parallel implementation can reduce this number into an order of 100 times the complexity of the component decoder.

Another important issue is the symmetric architecture by which the same component code is used in columns and in rows. The symmetric code design enables the designer to exploit the basic shortening and puncturing operations in an optimal way. This is due to the fact that the same error correction capability in rows and in columns is maintained when stuffing zero-bits in a symmetric way (columns & rows or diagonals when doing shortening), or when deleting parity check bits along diagonals (i.e., puncturing).

References

[1] M. Ran, IEEE 802.16.1pc-00/18, Improving ECC schemes of proposals pc-00/14 and pc-00/13, TelesciCOM , March 2000 .

[2] M.Ran, IEEE802.16.1pc-00/22, An Amendment to Mode B of IEEE802.16.1, TelesciCOM, May 2000.

[3] ETSI EN 300 421 V1.1.2 (1997-08), "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/112 GHz satellite services."

[4] ETSI EN 301 210 V1.1.1 (1999-03), "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for Digital Satellite News Gathering (DSNG) and other contribution applications by satellite."

[5] DVB-RCS16r1, TPC replacement text for DVB-RCS001, Tandberg TV & Telenor

[6] C. Martin, BRAN18.5d38, Turbo Product Code: an Efficient Coding Scheme for Error Free Transmissions, Alcatel, May 2000.

[7] IEEE802.16.1pc-00/29r1, Physical Layer Proposal for the IEEE802.16.1 Air Interface Specifications, May 2000

[8] T. Journe and A. Gueguen, BRAN18.5d28.doc, Evaluation of Turbo-Codes for HA, France Telecom R&D and Mitsubishi Electric, May 2000.

[9] G. Solomon and H.G.A van Tilborg, A connection between block and convolutional codes, SIAM J. Appl. Math, vol. 37 pp.358-269, Oct. 1979.

[10] D.J. Muder, Minimal trellises for block codes, IEEE trans. Inform. Theory vol.34 pp.1049-1053, Sep. 1988.

[11] R. Pyndia, Iterative Decoding of Product Codes: Block Turbo Codes, , Int. Symp. on Turbo Codes, Sep. 3-5, 1997.

[12] H. Nickl, J. Hagenauer and F. Bukert, Approaching Shannon Capacity Limit by 0.27 dB Using Simple Hamming Codes, IEEE Comm. Letters, pp130-132, 1997.

[13] R.M. Pyndia, Near-Optimum Decoding of Product Codes: Block Turbo Codes, IEEE trans. Comm., pp.1003-1010, 1998.

(FEC_table_TSCAHA)

Annex: table of Criteria for Comparison for FEC schemes for IEEE802.16.1 air-interface draft standard

Assumptions for D/L:

1. QPSK, 16QAM, 64QAM modulations
2. 40 Mbaud modulation rate.
3. No ARQ assumed.
4. Interleaver may be employed in DL
5. TDM or TDMA
6. Binary input continuous Output (BICO) AWGN channel

Downlink table1: Parameters and hardware

FEC scheme	BC+RS	CC + RS	CTC	BTC
Rate 1 (R1)				0.673
Rate 2 (R2)				0.717
Rate 3 (R3)				0.790
Packet Size 1 128 Bytes (P1)	1024	1024	1024	1024
Packet Size 2 Mpeg 188 Bytes (P2)	1504	1504	1504	1504
P3=392 Bytes	NA	3136	3196	3196
P2+Interleaver I2=12 (P4=12*P2)	NA			
HW Complexity (for 64QAM, largest P, I)				Less than 400kg (based on Alcatel\ TelesciCOM estimates)
Max. Frequency Clock (64QAM)				160Mhz
Overall Latency Tx + Rx				Tx: < 0.1us Rx < 2 Blocks
Error detection capability				inherent without additional redundancy (see text)
Flexibility: Burst\continuous Variable P Variable Rate				YES

Assumptions for U/L:

1. QPSK, 16QAM, 64QAM modulations
2. 30 Mbaud modulation rate.
3. No ARQ assumed.
4. No interleaver.
5. UL is TDMA. Short packets [P5, P6 are shortest packets of 802.16.1 PHY mode B.]
6. Binary input continuous Output (BICO) AWGN channel

Uplink table 2: Parameters and hardware

FEC scheme	BC+RS	CC + RS	CTC	BTC
Rate				P5: 0.727, 0.455 P6: 0.655, 0.453 P7: 0.633
Rate 2 (R2)				
Rate 3 (R3)				
Packet Size 1 (P5) K=5 Bytes	40	40	40	40
Packet Size 2 (P6) K=14 Bytes	112	112	112	112
Packet Size 3 (P7) K=57 Bytes	456	456	456	456
Error detection capability				inherent without additional redundancy (see text)
Hardware Complexity (64QAM,P3)			50 kg ~ 300 kg	10kg to 300kg (based on TelesciCOM estimates)
Max. Frequency Clock (64QAM)			80 ~ 120 Mhz	160MHZ
Overall Processing delay (P7)				Less than 3 blocks
Flexibility				Yes

Table 3.1: Performance results table for Downlink at given modulation format –QPSK

Modulation QPSK	method	Eb/N0 dB BTC1= (39,32)^2 HPC(6,25) R1=0.673 P1=128Bytes	Eb/N0 dB BTC2= (46,39)^2 HPC(6,18) R2=0.717 P2=188Bytes	Eb/N0 dB BTC3= (63,56)^2 HPC(6,1) R3=0.790 P3=392Bytes	Eb/N0 dB UNCODED R0=1
BER1=1E-6	S/W sim1	3.5	3.6	3.5	
	S/W sim2	3.4	3.6	3.6	
	Union Bound	2.85	2.80	2.75	
	Shannon (BICO)	1.1	1.4	1.95	10.6
BER2=1E-9 (estimated)	S/W sim1	4.5	4.3	4.3	
	S/W sim2	4.3	4.2	4.2	
	Union Bound	4.1	3.9	3.85	
	Shannon (BICO)	1.1	1.4	1.95	12.6
*BER3=1E-11	S/W sim1				
	S/W sim2				
	Union Bound	4.75	4.60	4.45	
	Shannon (BICO)	1.1	1.4	1.95	13.6

Table 3.2: Performance results table for Downlink at given modulation format –16QAM

Modulation 16QAM	method	Eb/N0 dB BTC1= (39,32)^2 HPC(6,25) R1=0.673 P1=128Bytes	Eb/N0 dB BTC2= (46,39)^2 HPC(6,18) R2=0.717 P2=188Bytes	Eb/N0 dB BTC3= (63,56)^2 HPC(6,1) R3=0.790 P3=392Bytes	Eb/N0 dB UNCODED R0=1
BER1=1E-6	S/W sim1	6.5	6.6	6.9	
	S/W sim2	!!!	6.9	6.9	
	Union Bound	!!!???	!!!???	!!!???	
	Shannon (BICO)	3.7	4.0	5.2	14.4
BER2=1E-9, estimated	S/W sim1	7.5	7.8	7.5	
	S/W sim2	!!!	7.8	7.6	
	Union Bound	!!!???	!!!???	!!!???	
	Shannon (BICO)	3.7	4.0	5.2	16.4
*BER3=1E-11	S/W sim1				
	S/W sim2				
	Union Bound	!!!???	!!!???	!!!???	!!!???
	Shannon (BICO)	3.7	4.0	5.2	18.4

Table 3.3: Performance results table for Downlink at given modulation format –64QAM

Modulation 64QAM	method	Eb/N0 dB BTC1= (39,32)^2 HPC(6,25) R1=0.673 P1=128Bytes	Eb/N0 dB BTC2= (46,39)^2 HPC(6,18) R2=0.717 P2=188Bytes	Eb/N0 dB BTC3= (63,56)^2 HPC(6,1) R3=0.790 P3=392Bytes	Eb/N0 dB UNCODED R0=1
BER1=1E-6	S/W sim1	10.7	10.5	11.0	
	S/W sim2	!!!	10.9	10.8	
	Union Bound	!!!???	!!!???	!!!???	
	Shannon (BICO)	6.9	7.4	8.8	18.7
BER2=1E-9 (estimated)	S/W sim1	11.7	11.5	12.0	
	S/W sim2	!!!	11.5	11.7	
	Union Bound	!!!???	!!!???	!!!???	
	Shannon (BICO)	6.9	7.4	8.8	20.9
*BER3=1E-11	S/W sim1				
	S/W sim2				
	Union Bound				
	Shannon (BICO)	6.9	7.4	8.8	21.9

Table 4.1: Performance results table for Uplink at given modulation format -QPSK

Modulation QPSK	method	Eb/N0 dB P5=5bytes (11,10)*(5,4) R5=0.727	Eb/N0 dB P5=5bytes (11,10)*(8,4) R5’=0.455	Eb/N0 dB P6=14bytes (29,23)*(6,5) R6=0.655	Eb/N0 dB P6=14bytes (16,11)^2 - 9 R6’=0.453	Eb/N0 P7=57bytes (30,24) *(25,19) R7=0.608	Eb/N0 dB UNCODED R0=1
BER1 =1E-6	Sim1	7.2	6.5	5.4	4.0	3.4	
	Sim2	7.1			4.3	3.4	
	Union bound	7.1	6.2	5.2	3.8	3.2	
	Shannon	1.46	-0.02	0.99	-0.02	0.68	10.6
BER2 =1E-9 estimated	Sim1		8.7	7.3	5.8	4.7	
	Sim2	8.8			5.5	4.7	
	Union bound	8.7	7.8	6.6	5.2	4.4	
	Shannon	1.46	-0.02	0.99	-0.02	0.68	12.6
BER3 =1E-11 (optional)	Sim1						
	Sim2						
	Union bound			7.4	5.9	5.2	
	Shannon	1.46	-0.02	0.99	-0.02	0.68	13.6

Table 4.2: Performance results table for Uplink at given modulation format –16QAM

Modulation 16QAM	method	Eb/N0 dB P5=5bytes (11,10)*(5,4) R5=0.727	Eb/N0 dB P5=5bytes (11,10)*(8,4) R5'=0.455	Eb/N0 dB P6=14bytes (29,23)*(6,5) R6=0.655	Eb/N0 dB P6=14bytes (16,11)^2 - 9 R6'=0.453	Eb/N0dB P7=57bytes (30,24) *(25,19) R7=0.608	Eb/N0 dB UNCODED R0=1
BER1= 1E-6	Sim1				6.8		
	Sim2						
	Union bound						
	Shannon	4.2	1.7	3.5	1.7	3	14.4
BER2= 1E-9 (estimated)	Sim1				8.8		
	Sim2						
	Union bound						
	Shannon	4.2	1.7	3.5	1.7	3	16.4
BER3= 1E-11 (optional)	Sim1						
	Sim2						
	Union bound						
	Shannon	4.2	1.7	3.5	1.7	3	18.4

Table 4.3: Performance results table for Uplink at given modulation format –64QAM

Modulation 64QAM	method	Eb/N0 dB P5=5bytes (11,10)*(5,4) R5=0.727	Eb/N0 dB P5=5bytes (11,10)*(8,4) R5’=0.455	Eb/N0 dB P6=14bytes (29,23)*(6,5) R6=0.655	Eb/N0 dB P6=14bytes (16,11)^2 - 9 R6’=0.453	Eb/N0dB P7=57bytes (30,24) *(25,19) R7=0.608	Eb/N0 dB UNCODED R0=1
BER1=1E-6	Sim1			12.8	9.8	10.0	
	Sim2						
	Union bound						
	Shannon	7.5	3.8	6.7	3.8	5.9	18.7
BER2=1E-9 (estimated)	Sim1			15.5	11.8	11.5	
	Sim2						
	Union bound						
	Shannon	7.5	3.8	6.7	3.8	5.9	20.9
BER3=1E-11 (optional)	Sim1						
	Sim2						
	Union bound						
	Shannon	7.5	3.8	6.7	3.8	5.9	21.9