Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Improving ECC scheme of proposals 802.16.1pc-00/14 and 802.16.1pc-00/13**
Date Submitted	**2000-03-06**
Source	Moshe Ran and Shahaf Wayer TelesciCOM, LTD 6 Hamachtesh st. Holon ISRAEL, 58810 Voice: +972-3-5589595/208 Fax: +972-3-5589091 mailto:Mran@telescicom.com
Re:	Call for Evaluations and Improvements, IEEE 802.16.1, Session #6, IEEE 802.16.1-00/07/r1.
Abstract	Iterative Soft-in/Soft-out decoding schemes are the best ECC known today for approaching the BWA channel capacity. This document specifically addresses an advanced coding and modulation mode for the 802.16 PHY layer by replacing the traditional Reed-Solomon (RS) codes by binary block product codes and use of Soft-in\Soft-out (SISO) iterative decoders (i.e. "Block Turbo Codes", BTC).
Purpose	The following ECC proposal is submitted as an improvement to the ECC coding scheme of the two PHY proposals **802.16.1pc-00/14 and 802.16.1pc-00/13.** Conclusions and recommendations are made to allow incorporation of the suggested improvements to proposed PHY.
Notice	
Release	
IEEE Patent Policy	

Improving ECC schemes of proposals 802.16.1pc-00/14 and 802.16.1pc-00/13

Moshe Ran and Israel S. Wayer

mran @telesciCOM.com

1.0 Scope

This document specifically addresses an advanced coding and modulation mode for the 802.16 PHY layer by replacing the traditional Reed-Solomon (RS) codes by binary block product codes and use of Soft-in\Soft-out (SISO) iterative decoders (i.e. "Block Turbo Codes", BTC). Compatibility with baseline downstream modes of the 802.16.1pc-00/13 [6] includes: transport multiplex adaptation [1], scrambling for energy dispersal. The bit mapping to Gray coded QPSK, 16QAM and optionally 64QAM constellation are compatible with 802.16.1pc-00/14 [7]. In the upstream channel, variable length\variable rate, block product codes to maintain maximum commonality with the variable length MAC messages, [7] and [6] is supported.

1.2 Motivation

Iterative Soft-in\Soft-out (SISO) decoders ("Turbo Codes") are considered in several evolving wireless communication standards including the 3-G mobile communications system and DVB-RCS.

Compared to traditional Error Control Coding (ECC) schemes such as Reed-Solomon codes, convolutional codes or concatenation of both (i.e. Reed Solomon Viterbi, RSV), the BTC techniques are more appealing in BWA applications. It can be demonstrated that these new ECC schemes outperform the legacy ECC in several aspects.

- **Performance**: more than 1 dB better coding-gain than the concatenated RSV scheme with comparable rate as specified in down link of [6] and 2.7 dB better than the (9,8) + RS (138,128) of [7]. Also better performance are achieved compared to variable length shortened RS employed in the uplink of both proposals. (see Appendix 3)
- **Flexibility**: variable code rates and variable block size can easily be implemented to support various protocols such as MPEG-2, ATM and short IP frames. Very short to very long single block code can be designed based on relatively short component codes (see Appendix 1).
- **Latency**: BTC has inherent block interleaver with much shorter delays than the convolutional interleaver used in EN300 421 [1, section 4.4.3]. This feature is particularly important for the up-stream channel where interleaver is not available.
- **CPE cost optimization:** In order to optimize the CPE cost, it is possible to initially reduce performance of the CPE, by employing non-iterative soft decoders. In a different assembly, with simple iterative decoders using current silicon technology, a higher performance at moderate cost is easily attainable. In the future, when much more complex processing at reasonable price will be available, a seamless upgrade to more powerful iterative decoders will be accomplished. This upgrade will take place without affecting the other technical specifications of the CPE.

- **Future implication:**. SISO iterative decoders have inherent ingredient that allow the highest performance when compared to other decoding strategies. Technological limitation of the two stage concatenated coding scheme is far from optimal performance.

We propose a binary product code based on shortened Hamming codes for the downstream PHY layer, which exactly match the 188-bytes information frame of the MPEG-2 as required in proposal [6]. Allowing the parameters of the product code to be configurable we get a family of product codes denoted Hamming Product Codes, HPC. This family of codes can be used to protect frame formats from 32bytes up to 512bytes and have code rates between 0.5 – 0.8 (see Appendix A). From this list an HPC code for the downstream of proposal [7] can be selected. In particular, a highly symmetric HPC code based on [39,32] x [39,32] shortened Hamming codes is available as an alternative to coding scheme proposed in [7]. For frame formats between 8bytes to 32bytes, we propose a high rate Parity Product Code (PPC) based on parity check codes. Both HPC and PPC have remarkable efficient SISO iterative decoders (i.e.,"Turbo decoders"). A reduced performance low cost CPE mode which performs simple (i.e.,non-iterative) soft decoding is added as a part of the improved PHY mode. Evidently, (see [5]) simple soft decision decoding adds roughly 3dB to conventional hard decision coding gain. SISO Iterative decoding gains roughly additional 3dB over soft decoding.

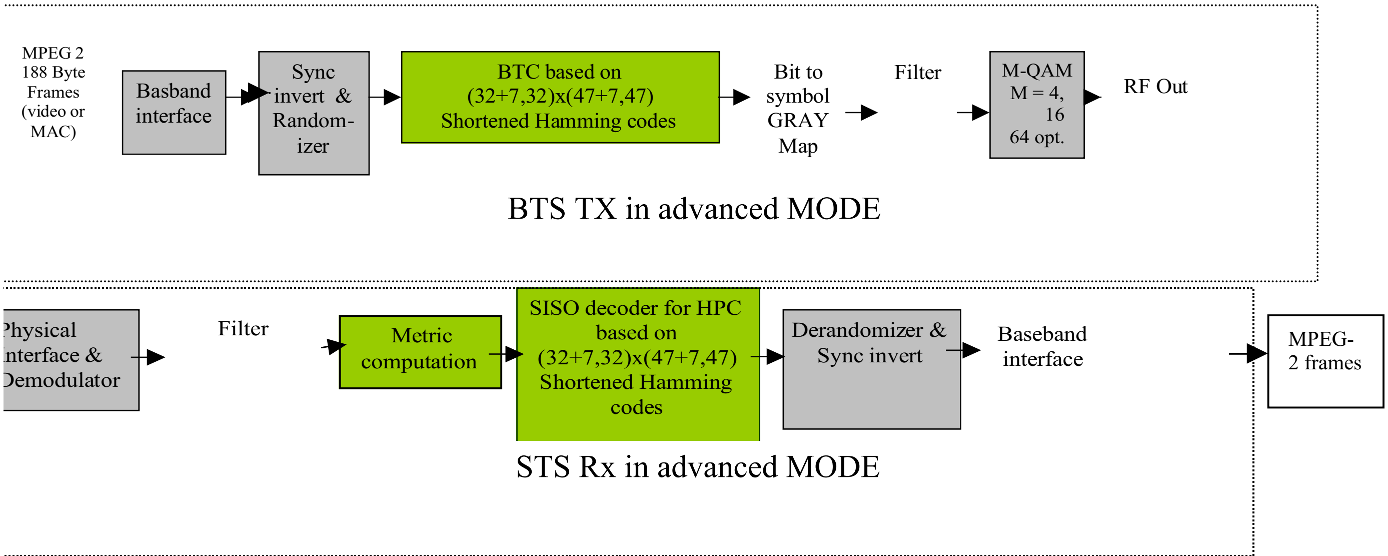
1.3 Coding, Interleaving, Scrambling & Modulation

In the downstream, following the encapsulation of the MAC packets into the MPEG frame, the data is randomized using the same mechanism as in the basic mode of [6]. The randomized bitstream is placed in a 32 x 47 array of information bits. Each row is encoded by (54,47) shortened Hamming code and each column by (39,32) shortened Hamming code. The resulting 54x39 coded array is block interleaved by writing consecutive information bits in columns and reading them out in rows. Coded bits are Gray mapped to a QPSK, 16QAM (optionally) or 64QAM (optionally) signal constellation. Finally, symbols are Nyquist filtered using a square-root raised cosine with roll-of that is programmable 0.15, 0.25 or 0.35. The downstream demodulator performs Soft-in\soft-out iterative decoding. A reduced performance low cost CPE mode which performs simple (i.e., non-iterative) soft decoding is added as a part of the improved PHY mode.

In the upstream, the parameters of the product code shall be configurable yielding a family of product codes denoted Hamming Product Codes, HPC. This family of codes shall be used in the upstream to protect frame formats from 18bytes up to 399bytes and have code rates between 0.44 – 0.8. For very short frame of length less than 32bytes, a very high rate Parity Product Code (PPC) based on parity check codes is proposed. Both HPC and PPC have remarkable efficient SISO iterative decoders (i.e.,"Turbo decoders"). The same modulation formats and filtering features as appeared in downstream will be supported in the upstream.

2.0 Downstream PMD sublayer

The encoding and decoding functions for the downstream physical layer in the advanced mode are summarized in the following block diagram.



Main features for proposal [6]:
Sync Invert and randomization: as in [1]
ECC: HPC(m=6,S1=25,S2=10) = (2106, 1504, 16)
interleaving: Block interleaver 54 bits or 39 bits (selectable through MAC message)
Modulation: QPSK, 16QAM and optionally 64QAM
Bit to symbol map*: **Gray-coded** for all modulation formats
Spectral shaping: $\alpha = 0.15$,0.25, 0.35 programmable through MAC messaging.
Symbol rates: configurable up to 40 Mbaud

Main features for proposal [7]:
Modulation, bit –to-symbol-mapping and spectral shaping as suggested for [7].

ECC: HPC(m=6, S1=25, S2=25) = (1521,1024)

2.1 Binary Hamming Product codes for MPEG-2 package format

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.

$(n_1, k_1, d_1) \times (n_2, k_2, d_2)$ is a general representation of a block code with length $n=n_1n_2$ dimension $k= k_1k_2$ and minimum distance $d=d_1d_2$. 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).

This idea can be generalized straightforward to 3-D array codes based on three components code (n_i, k_i) i=1,2,3.

B. Proposed product code for MPEG-2

The general product code based on shortened binary Hamming codes as component codes is given by

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

This code will be referred in the sequel as a Hamming Product Code **HPC(m, S1,S2).**

An MPEG package contains 188 bytes of 8 bits each. Thus, a product code which contains exactly these 188x8=1504 bits is realized with the following parameters:

m = 6, S1 = 25, S2 = 10.

This implies (39, 32) and (54, 47) shortened Hamming components codes which constitute the binary product code (39x54, 32x47, 4x4) = (2106, 1504, 16) with code rate 0.714.

The shortened Hamming code (64-S, 57 - S) with shortening parameter S shall be implemented by appending S bits, all set to zero, before the information bits at the input of (64,57) extended Hamming encoder. This encoder shall be implemented by appending a parity check column the generator matrix of the Hamming code (63,57) generated by the primitive polynomial of degree 6:

$$g(x) = X^6 + X^1 + 1$$

The block interleaving shall have two selections. Either 64 - S1 = 39 or 64 - 10 = 54 bits.

3rd. Error performance requirements for HPC(m=6,S1=25,S2=10)

The modem in advanced MODE shall meet the BER versus E_b/N_0 performance requirements given below:

PAM level	BER post HPC	Required E_b/N_0
M=2	10E-6	3.7dB
M=2	10E-9	4.5dB
M=4	10E-6	7.0dB
M=4	10E-9	7.8dB
M=8	10E-6	11dB
M=8	10E-9	12dB

Notes

1. The figures include a modem implementation margin of 1dB for 2-PAM (4QAM), and 2dB for 4-PAM (16QAM) and 8-PAM (64QAM).

2. The conversion to C/N should be taken according to the following formula:

$C/N = E_b/N_0 + 10\log 0.714 + 10\log[\log 2(M^2)] - 10\log(1 - \alpha/4)$ [dB]
where, α is roll-off factor (0.15 to 0.35). C/N describes the ratio of signal-to-noise in the transmission channel. The following applies:

$C/N = S/N + 10\log(1 - \alpha/4)$ [dB].

3. The performance of the proposed ECC scheme is evaluated for comparison purpose in AWGN channels. Empirical results in the area of SISO iterative decoding shows that on Rayleigh fading channels the BER versus E_b/N_0 curve has the same slope as in Gaussian channels. This can be explained by the fact that in the iterative decoding process, the soft output tends to a Gaussian distribution after

few iterations for any identically distributed input data. Thus the BER versus E_b/N_0 curves on the Rayleigh fading channels is as steep as for the Gaussian channel but shifted to the right.

4. Low cost reduced performance CPE equipment can be operated in non-iterative mode by performing only simple (i.e., non-iterative) soft decoding. This sub-mode will have the penalty of loosing roughly 3 dB of coding gain in AWGN channels.

3.0 PHY Layer -up stream in the advanced mode

In this mode, most of the transmission formats of baseline upstream such as: randomization, preamble prepend, Gray bit-to-symbol mapping and pulse shaping are compatible with [6]. Only slight modifications are required to be consistent with [7]. These modifications will be defined in later stage. However, in order to give better protection against errors, two schemes for variable length block product code are proposed. For information packets between 18 to 255, HPC(m, S1, S2) are proposed. (See Appendix 1 for detailed list of available configurations). For short packets between 8bytes and 32bytes and relatively very high rate, block product code based on $(k_1+1,k_1) \times (k_2+1, k_2)$ parity product codes, denoted PPC(k_1, k_2), are suggested. Unlike Reed-Solomon codes based on GF(256) which are limited to codewords of length up to 255, HPC codes can support up to 399 bytes.

Main features:

- Randomizer: XOR with configurable 16 taps LFSR
- ECC: variable information length between 18 - 399bytes, variable rate 0.44 to 0.8 based on HPC(m,S1,S2) m = 5 or 6, S1, S2 configurable between 0 to 32. [see note 1]

  Short burst mode: variable length between 8 to 32 bytes based on $(k_1+1,k_1) \times (k_2+1, k_2)$ parity product codes, denoted PPC(k_1, k_2).
- interleaving: Bit block-interleaver, configurable 2^m - S1 or 2^m - S2
- Modulation: Gray coded QPSK, 16QAM and optionally 64QAM
- Bit to symbol map*: **Gray-coded** for all modulation formats
- Spectral shaping: $\alpha = 0.15 - 0.35$ programmable through MAC messaging.
- Symbol rates: configurable up to 40 Mbaud

NOTES:

1. Since upstream transmitter in [6] should support at least 6 burst profiles, where each burst profile contains 0-1023 bits of preamble, the spectral efficiency in the up-link is expected to be low. Thus, it is highly desired to estimate the expected size of packets and tailor the code to the expected size. For example, the most common packet size of IP traffic is 40 bytes, which accounts for TCP ACKs, finish messages (FINs), and reset messages (RSTs). For this 40bytes packets HPC(m=5,S1=10, S2=10) specifically designed. Furthermore, overall average packet size of IP traffic varies from 175 to about 400. Thus, most of variable length IP traffic are efficiently protected with HPC code with m=5 or 6. In the baseline mode, when using Reed-Solomon encoding scheme, efficiency is improved by using "shortened last word mechanism". This mechanism can be supported also here. Alternatively, much better solution, which avoids "shortened last word" can be realized by selecting the proper block size from the list of available HPC between 18 to 399 bytes. (see Appendix 1)

2. In an IP traffic there are also 552bytes, 576bytes and 1500 bytes which are encoded in the baseline mode as sequence of several Reed-Solomon codewords. However, by allowing 3-D product codes where the 2-D code is HPC with m=5 or 6 and the third code is a parity check code we can extend the protection capability of the Turbo code to much larger packet size.

3.1 Protecting variable length packages using HPC(m,S1, S2) code

The generator polynomial for the shortened Hamming component codes shall be based on the following primitive binary polynomial.

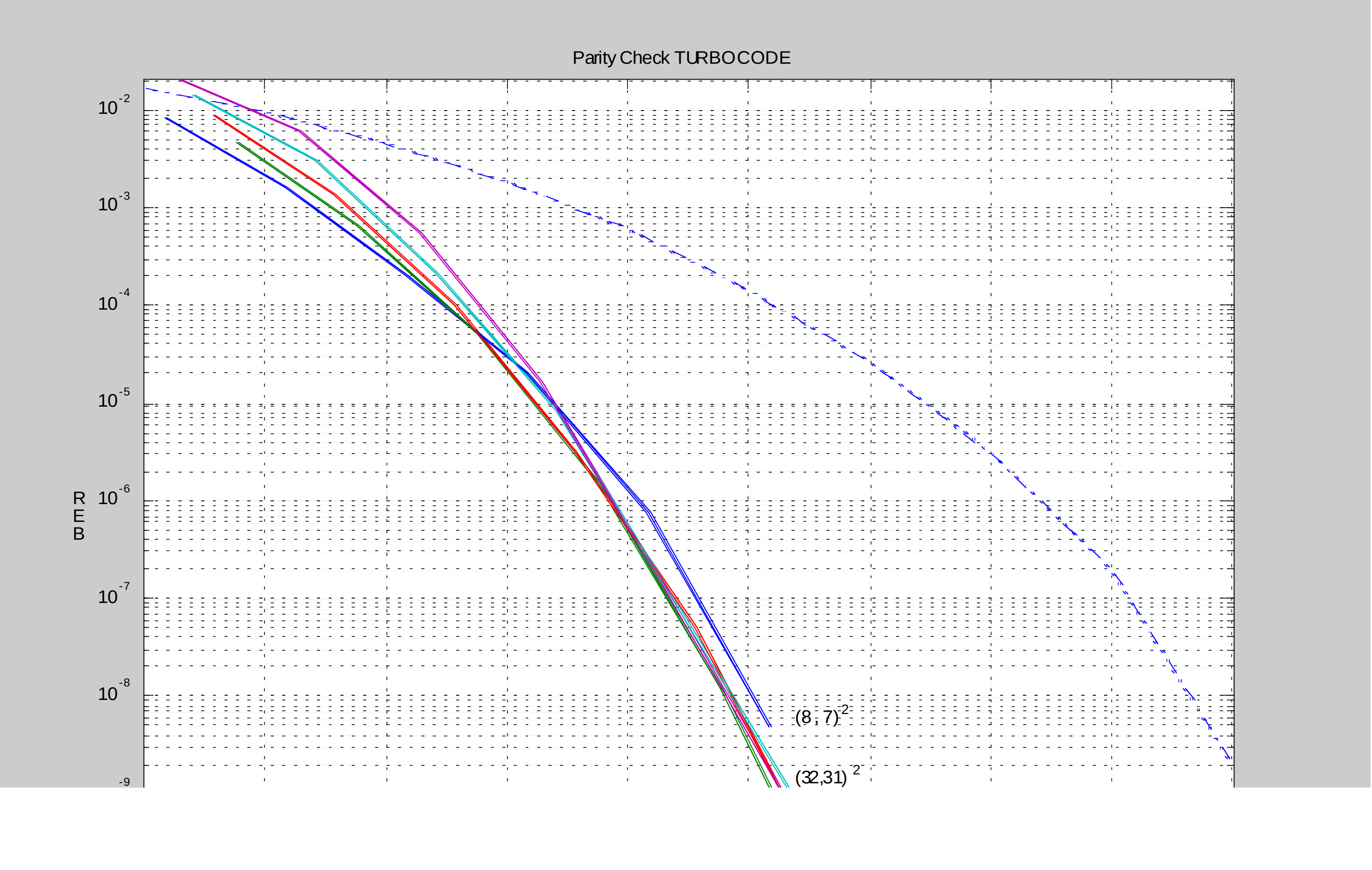
m=5 $g(x) = X^5 + X^2 + 1$

m=6 $g(x) = X^6 + X + 1$

The HPC encoder and interleaver consists of a rectangular array of 2^m x 2^m bits. The block encoder for HPC(m, S1, S2) accepts S1 bits of zeros followed by $k_1 = 2^m - m - 1 - S1$ bits of data. Those k_1 bits are written in columns of the array where last bit is regarded as the MSB. A sequence of m parity check bits are computed based on g(x) followed by an overall parity check bit. This procedure is repeated column by column until the first $k_2 = 2^m - m - 1 - S2$ columns of the encoder array. When this process, called column encoding, is finished a line encoding process starts by appending S2 bits of zero followed by a sequence of m parity check bits for the first row followed by overall parity check bit. This process is repeated until all n_1 lines are encoded. The coded bits are read from the array row-by-row and Gray mapped to symbols in the constellation map.

3.2 Protecting Short IP packages:

Applications of BTC to variable length and relatively short IP packages can be realized with the aid of parity check product codes (PPC). Typical performance for several choices of PPC are given below.



4.0 Summary and Recommendations

SISO iterative decoders based on Hamming Product codes (HPC) and Parity product codes (PPC) are suggested as an amendment to the PHY proposals of [6] and [7]. We demonstrated that the families of codes presented outperform the conventional two stage decoders based on RS-Interleaver-Convolutional encoder as suggested for the DL of [6]. The benefits to the PHY proposed in [7] is even more remarkable since it can improve the coding gain by 2.7dB compared to the (9,8) + RS(138,128) coding scheme (see Appendix 3 for simulation results).

Another advantage to proposal [7] is due to opportunity to have a block interleaver as an inherent part of the product code. For the uplink of both proposals [6] and [7] we suggest a coding scheme based on variable length HPC, or for very short information bursts, a coding scheme based on Parity Product codes, PPC.

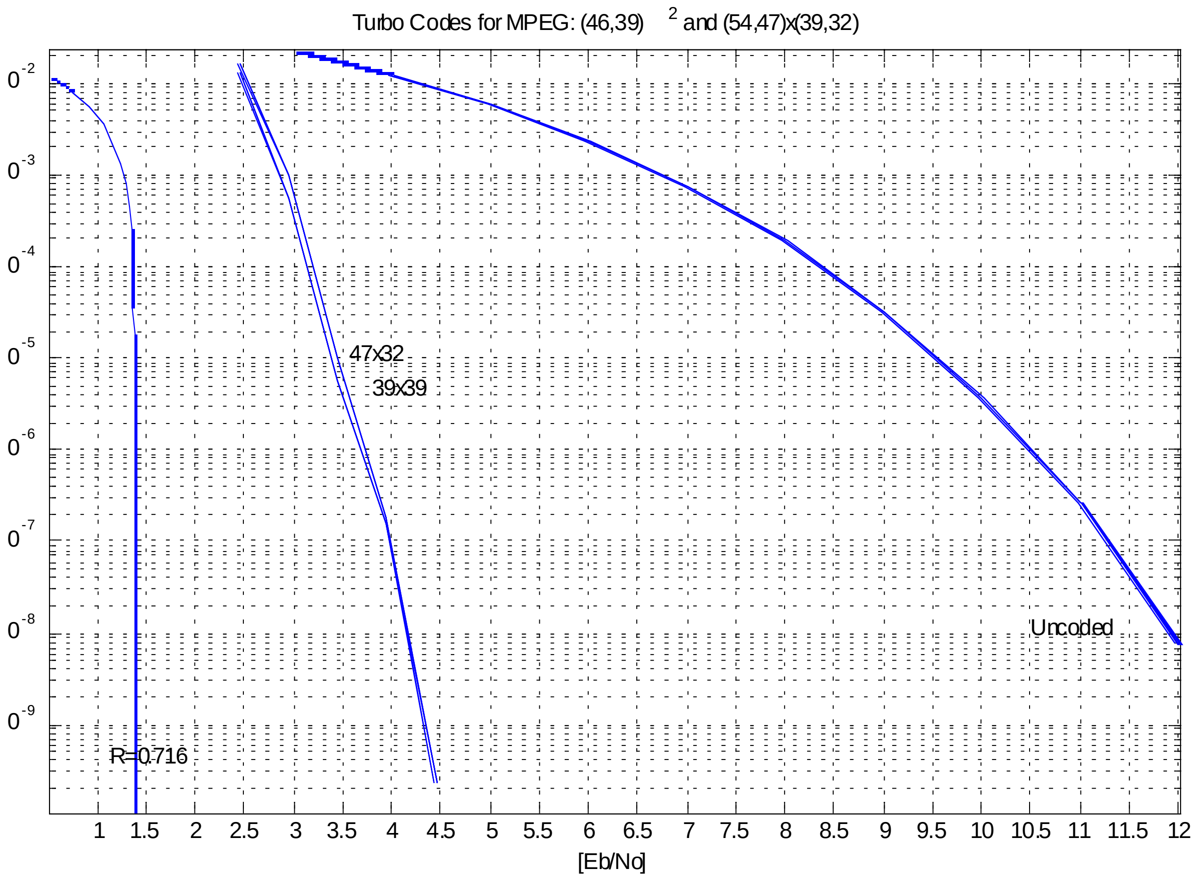
Appendix 1: A list of Available information frame formats between 18 and 255 bytes

144=12*12 (18 bytes) R=0.444
160=10*16 (20 bytes) R=0.485
168=12*14 (21 bytes) R=0.467
176=11*16 (22 bytes) R=0.500
192=12*16 (24 bytes) R=0.485
208=13*16 (26 bytes) R=0.498
216=12*18 (27 bytes) R=0.500
224=14*16 (28 bytes) R=0.509
240=15*16 (30 bytes) R=0.519
256=16*16 (32 bytes) R=0.529
272=16*17 (34 bytes) R=0.538
280=14*20 (35 bytes) R=0.538
288=16*18 (36 bytes) R=0.545
304=16*19 (38 bytes) R=0.553
320=16*20 (40 bytes) R=0.559
336=16*21 (42 bytes) R=0.566
352=16*22 (44 bytes) R=0.571
360=18*20 (45 bytes) R=0.577
368=16*23 (46 bytes) R=0.577
384=16*24 (48 bytes) R=0.582
400=20*20 (50 bytes) R=0.592
408=17*24 (51 bytes) R=0.591
432=18*24 (54 bytes) R=0.600
440=20*22 (55 bytes) R=0.604
456=19*24 (57 bytes) R=0.608
480=20*24 (60 bytes) R=0.615
504=21*24 (63 bytes) R=0.622
520=20*26 (65 bytes) R=0.625
528=22*24 (66 bytes) R=0.629
552=23*24 (69 bytes) R=0.634
560=20*28 (70 bytes) R=0.615
576=24*24 (72 bytes) R=0.640
600=24*25 (75 bytes) R=0.645
616=22*28 (77 bytes) R=0.629
624=24*26 (78 bytes) R=0.650
648=24*27 (81 bytes) R=0.635
672=24*28 (84 bytes) R=0.640
696=24*29 (87 bytes) R=0.644
704=22*32 (88 bytes) R=0.645
720=24*30 (90 bytes) R=0.649
728=26*28 (91 bytes) R=0.650
736=23*32 (92 bytes) R=0.651
744=24*31 (93 bytes) R=0.653
768=24*32 (96 bytes) R=0.656
784=28*28 (98 bytes) R=0.640
792=24*33 (99 bytes) R=0.660
800=25*32 (100 bytes) R=0.662
816=24*34 (102 bytes) R=0.663
832=26*32 (104 bytes) R=0.667
840=28*30 (105 bytes) R=0.649
864=27*32 (108 bytes) R=0.652
888=24*37 (111 bytes) R=0.673
896=28*32 (112 bytes) R=0.656
928=29*32 (116 bytes) R=0.661
936=26*36 (117 bytes) R=0.680
952=28*34 (119 bytes) R=0.663

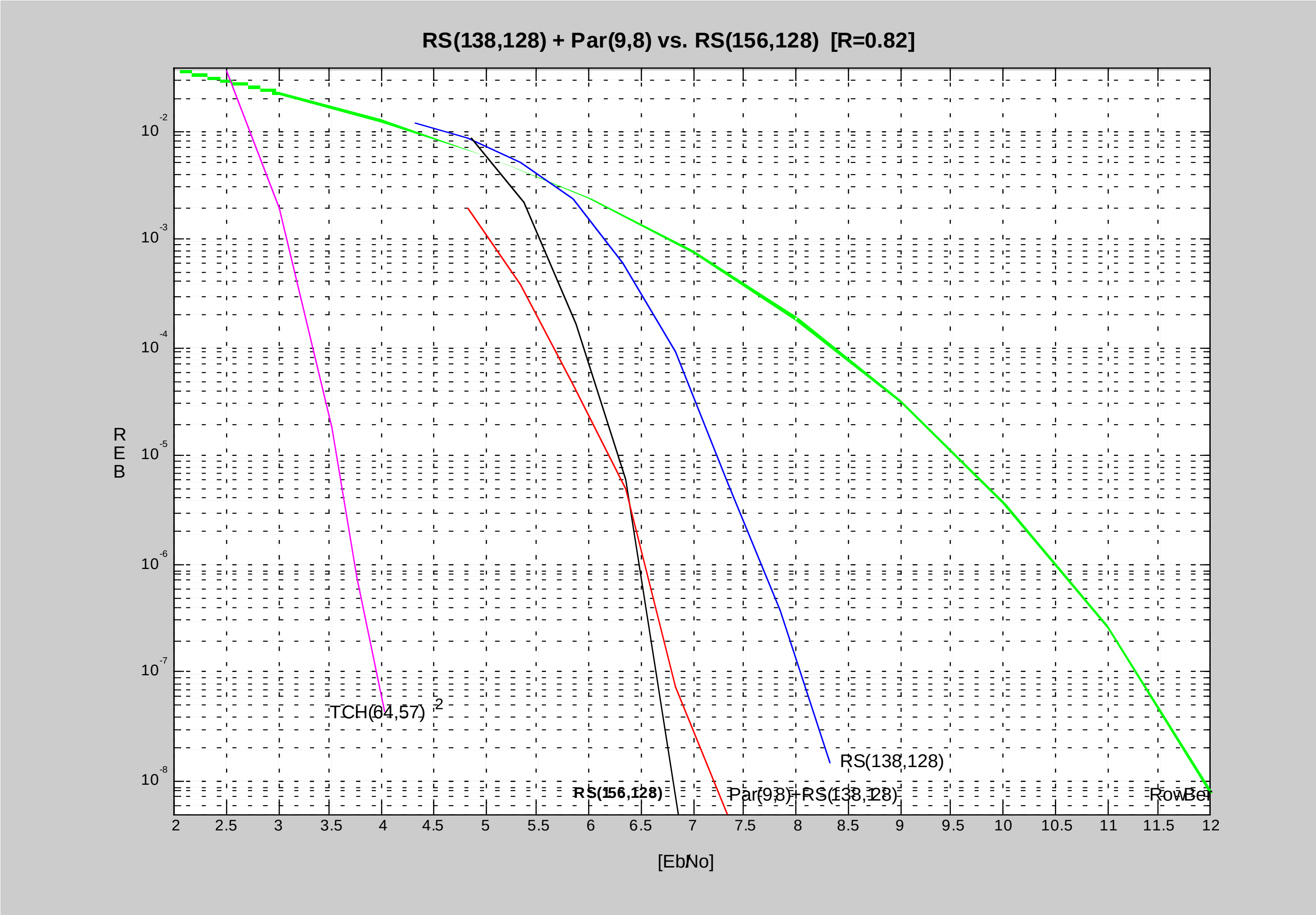
960=30*32 (120 bytes) R=0.665
992=31*32 (124 bytes) R=0.669
1008=28*36 (126 bytes) R=0.670
1024=32*32 (128 bytes) R=0.673
1040=26*40 (130 bytes) R=0.691
1056=32*33 (132 bytes) R=0.677
1064=28*38 (133 bytes) R=0.676
1080=30*36 (135 bytes) R=0.679
1088=32*34 (136 bytes) R=0.680
1120=32*35 (140 bytes) R=0.684
1152=32*36 (144 bytes) R=0.687
1160=29*40 (145 bytes) R=0.686
1176=28*42 (147 bytes) R=0.686
1184=32*37 (148 bytes) R=0.690
1200=30*40 (150 bytes) R=0.690
1216=32*38 (152 bytes) R=0.693
1224=34*36 (153 bytes) R=0.694
1240=31*40 (155 bytes) R=0.694
1248=32*39 (156 bytes) R=0.696
1280=32*40 (160 bytes) R=0.698
1296=36*36 (162 bytes) R=0.701
1312=32*41 (164 bytes) R=0.701
1320=33*40 (165 bytes) R=0.702
1344=32*42 (168 bytes) R=0.703
1360=34*40 (170 bytes) R=0.706
1368=36*38 (171 bytes) R=0.707
1376=32*43 (172 bytes) R=0.706
1400=35*40 (175 bytes) R=0.709
1408=32*44 (176 bytes) R=0.708
1440=36*40 (180 bytes) R=0.713
1472=32*46 (184 bytes) R=0.712
1480=37*40 (185 bytes) R=0.716
1496=34*44 (187 bytes) R=0.715
1504=32*47 (188 bytes) R=0.714
1512=36*42 (189 bytes) R=0.718
1520=38*40 (190 bytes) R=0.719
1536=32*48 (192 bytes) R=0.716
1560=39*40 (195 bytes) R=0.722
1568=32*49 (196 bytes) R=0.718
1584=36*44 (198 bytes) R=0.722
1600=40*40 (200 bytes) R=0.724
1632=34*48 (204 bytes) R=0.724
1640=40*41 (205 bytes) R=0.727
1656=36*46 (207 bytes) R=0.727
1672=38*44 (209 bytes) R=0.729
1680=40*42 (210 bytes) R=0.729
1720=40*43 (215 bytes) R=0.732
1728=36*48 (216 bytes) R=0.731
1760=40*44 (220 bytes) R=0.734
1768=34*52 (221 bytes) R=0.731
1776=37*48 (222 bytes) R=0.734
1800=40*45 (225 bytes) R=0.736
1824=38*48 (228 bytes) R=0.737
1840=40*46 (230 bytes) R=0.739
1848=42*44 (231 bytes) R=0.739
1872=39*48 (234 bytes) R=0.740
1880=40*47 (235 bytes) R=0.741
1920=40*48 (240 bytes) R=0.743

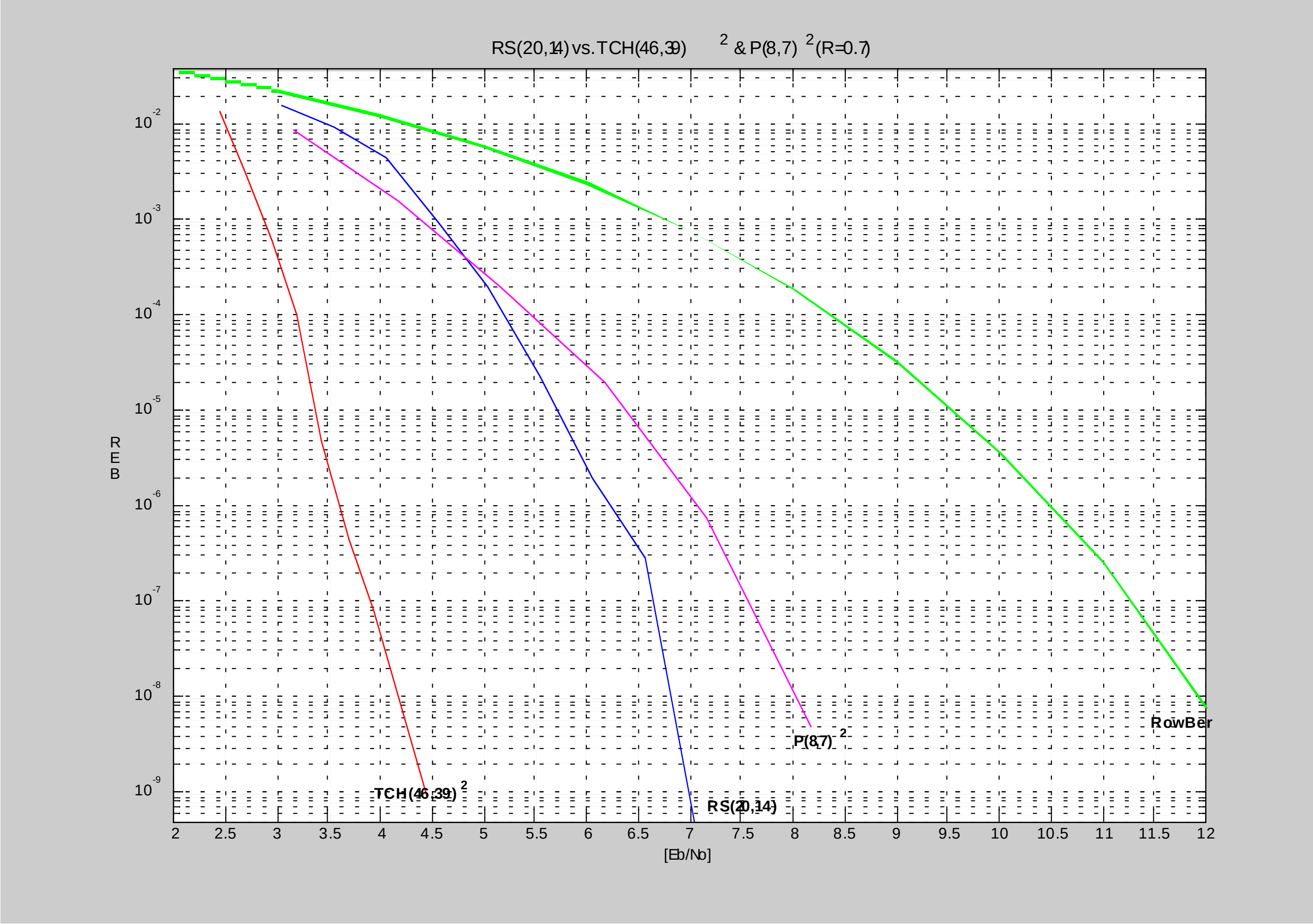
1936=44*44 (242 bytes) R=0.744
1944=36*54 (243 bytes) R=0.741
1960=40*49 (245 bytes) R=0.745
1968=41*48 (246 bytes) R=0.745
1976=38*52 (247 bytes) R=0.744
2000=40*50 (250 bytes) R=0.747
2016=42*48 (252 bytes) R=0.748
2024=44*46 (253 bytes) R=0.749
2040=40*51 (255 bytes) R=0.748

Appendix 2:

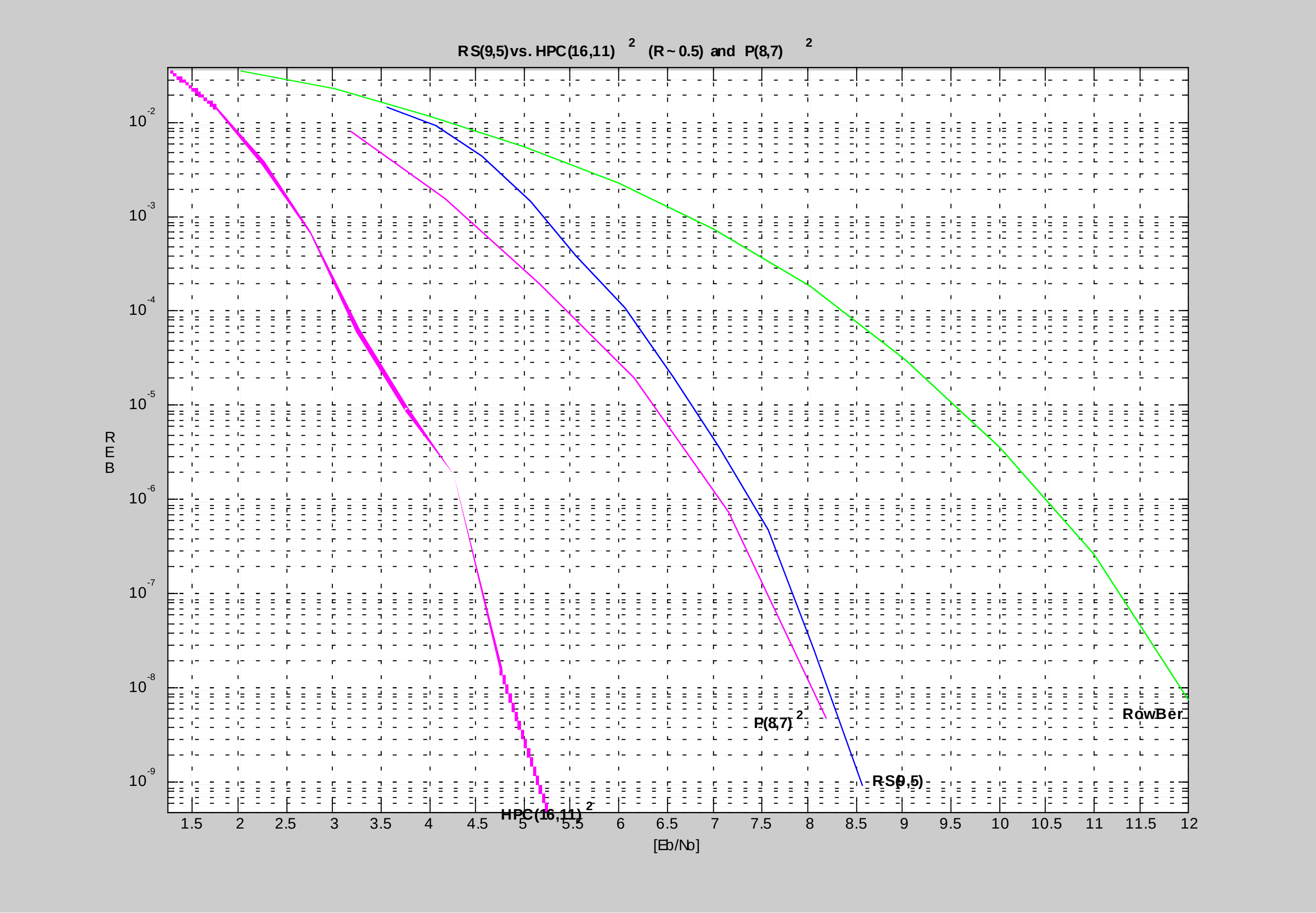

Turbo Codes for MPEG: (46,39) 2 and (54,47)x(39,32)
0-2
0-3
0-4
0-5
0-6
0-7
0-8
0-9
47x32
39x39
Uncoded
R=0.716
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5
8
8.5
9
9.5
10
10.5
11
11.5
12
[Eb/No]

Appendix 3:


RS(138,128) + Par(9,8) vs. RS(156,128) [R=0.82]
BER
[Eb/No]
TCH(64,57)^2
RS(138,128)
RS(156,128)
Par(9,8)+RS(138,128)
RowBer



RS(20,14) vs. TCH(46,39)^2 & P(8,7)^2 (R=0.7)
BER
[Eb/No]
TCH(46,39)^2
RS(20,14)
P(8,7)^2
Row Ber


RS(9,5) vs. HPC(16,11)^2 (R ~ 0.5) and P(8,7)^2
BER
[Eb/No]
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
10^-9
1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 6.5 7 7.5 8 8.5 9 9.5 10 10.5 11 11.5 12
P(8,7)^2
RowBer
RS(9,5)
HPC(16,11)^2

References

[1] 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."

[2] 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."

[3] ITU-T J.83 (04/97), Series J: Transmission of Television, Sound Program and Other Multimedia Signals: Digital transmission of television signals, "Digital multi-program systems for television, sound and data services for cable distribution."

[4] Data-Over-Cable Service Interface Specifications, "Radio Frequency Interface Specification," SP-RFIv1.1-I01-990311.

[5] Jhon B. Anderson, Digital Transmission Engineering, 1999 IEEE Press Ch.6.3

[6] 802.16.1pc-00/13

[7] 802.16.1pc-00/14