

## Referências Bibliográficas

- [1] Holma H., Toskala A., "WCDMA for UMTS," John Wiley & Sons,Ltd., England, 3rd. Edition, 2004.
- [2] IETF RFC 3261, "SIP: Session Initiation Protocol", Jun. 2002.
- [3] ITU-T Document Rec. H.248.1, "Gateway Control Protocol: Version 2", May 2002.
- [4] 3GPP Document, "Network Architecture", TS 23.002, v6.3.0, December 2003.
- [5] 3GPP Document, "Radio Interface Protocol Architecture", TS 25.301, v6.0.0, December 2003.
- [6] 3GPP Document, " UTRAN Iu interface: General aspects and principles", TS 25.410, v6.3.0, June 2005.
- [7] 3GPP Document, "RRC Protocol Specification", TS 25.331 v6.1.0, March 2004.
- [8] 3GPP Document, "Mobile Radio Interface Layer 3 Specification; Core Network Protocols;stage 3", TS 24.008, v6.4.0, March 2004.
- [9] 3GPP Document, "UTRAN Iu Interface RANAP Signalling", TS 25.413, v6.1.0, March 2004.
- [10] 3GPP Document, "Common Test Environments for User Equipment (UE); Conformance Testing", TS 34.108, v5.1.0, June 2004.
- [11] Daniel W., Vijay K.V., "Supporting Real-Time IP Multimedia Services in UMTS," IEEE Communications Magazine, Vol. 41, No. 11, pp. 148–155, November 2003.
- [12] Yi-Bing L., Ai-Chun P., Yieh-Ran H., "An All-IP Approach for UMTS Third-Generation Mobile Networks," IEEE Network, Vol. 16, No. 5, pp. 8–19, September/October 2002.
- [13] D. Plasse, "Call Control Scenarios in the *all-IP* UMTS Core Network", IEEE 11<sup>th</sup> International Symposium on Personal Indoor and Mobile Radio Communications –PIMRC 2000, Vol 1, pp 322-326, September 2000.

- [14] R. Koodli and M. Puuskari, "Supporting Packet-Data QoS in Next-Generation Cellular Networks", IEEE Communications Magazine, Vol. 39, No. 3, pp 180-188, February 2001.
- [15] Oliver H. and Thomas F.L., "Unified Mobility Manager: Enabling Efficient SIP/UMTS Mobile Network Control", IEEE Wireless Magazine, Vol. 10, No. 4, pp 66-75, August 2003.
- [16] IETF RFC 3311, "The Session Initiation Protocol (SIP) UPDATE Method", September 2002.
- [17] Vale E., Brandão J.C., Grivet M., "Voice and Multimedia Call Procedures in UMTS Networks", I2TS 2005 - 4th International Information and Telecommunication Technologies Symposium, UFSC, December 14-16, 2005.
- [18] 3GPP Document, "Radio Link Control (RLC) Protocol Specification", TS 25.322, v6.0.0, December 2003.
- [19] Bertsekas D.P., "Dynamic Programming and Optimal Control" , Vols I and II, Athena Scientific, 1995.
- [20] Proakis, J., "Digital Communications" , 4rd Edition, McGraw Hill, 2000.
- [21] Vale E., Brandão J.C., Grivet M., "Padding Optimization in UMTS", to be presented at PIMRC 2006 Conference, Helsinki, September 2006.
- [22] 3GPP Document, "Multiplexing and Channel Coding," TS 25.212 v6.2.0, June 2004.
- [23] Sohn I., Bang S.C., "Performance Studies of Rate Matching for WCDMA Mobile Receiver," VCT 2000 Conference, pp. 2661–2665, 2000.
- [24] Aftelak S, Bhatoolaul X., "Rate Matching Attribute Settings and Error Rate Performance Sensitivity for Selected UMTS FDD Services," IEEE Semiannual Vehicular Technoogy Conference, pp. 6-9, October 2003.
- [25] Baey S., Dumas M.C., "QoS Tuning and Resource Sharing for UMTS WCDMA Multiservice Mobile", IEEE Transactions on Mobile Computing, vol.1,no.3, pp.221-235, July-September 2002.
- [26] Haykin S., "Communication Systems", John Wiley & Sons, 2001.
- [27] Sumanasena M.A.K., Evans B.G., Tafazolli X., "IEE Seminar on Simulation and Modelling of Satellite Systems", London, UK, April 2004.
- [28] Adi W., Ben-Aoun M., Kassab N., "Variable Redundancy Coding for Mobile Channels", ISCC Conference, pp. 580-584, July 2001.
- [29] Bertsekas D. P., "Nonlinear Programming", Athenas Scientific, 1995.

- [30] Vale E., Brandão J.C., Grivet M., "UMTS Rate Matching Optimization", International Association of Science and Technology for Development - IASTED, CSA 2005, Banff, Canada, July 2005.
- [31] Vale E., Brandão J.C., Grivet M., "Performance of Voice Calls with Rate Matching Optimization in UMTS Networks", submetido para publicação na revista da SBRT.
- [32] 3GPP Document, "Services Provided by the Physical Layer", TS 25.302, v6.1.0, March 2004.
- [33] 3GPP Document, "Channel Coding and Multiplexing Examples", TR 25.944, v4.1.0, June 2001.
- [34] 3GPP Document, "Medium Access Control (MAC) Protocol Specification", TS 25.321, v6.1.0, March 2004.
- [35] IETF RFC 3267, "Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", June 2002.
- [36] IETF RFC 3095, "Robust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", July 2001.
- [37] A. Couvreur, Le Ny L.M., Minaburo A., Rubino G., Sericola B. and Toutain L. (2006), Performance Analysis of a Header Compression Protocol: The ROHC Unidirectional Mode, Telecommunication Systems, Vol 31, No 1, pp 85,98.
- [38] A. Giovanardi, G. Mazzini, M. Rossi and M. Zorzi (2000), Improved header compression for TCP/IP over wireless links. IEEE Electronics Letters, pp 1958–1960.
- [39] 3GPP Document, "Optimized RAB for Conversational IMS with Secondary Scrambling Code", TSG-RAN2, Meeting#41, Malaga, Spain, R2-040531, February 2004.
- [40] 3GPP Document, "Packet Data Convergence Protocol (PDCP) Specification", TS 25.323, v6.0.0, December 2003.
- [41] Hammer F., Reichl P., Nordstrom T., Kubin G., "Corrupted Speech Data Considered Useful", Acta Acustica, Vol.90, No. 6, pp 1052-1060, Nov/Dec 2004.
- [42] Vale E., Brandão J.C., Grivet M., "Reducing the VoIP Packet Overhead in UMTS Air Interface", submetido para publicação na revista Telecommunications Systems.

[43] Freiberg L. F., Muechkenheim J., "Universal Mobile Telephone System Network with Improved Rate Matching Method", United States Patent No. US 6,788,657 B1, Sep. 7,2004.

[44] 3GPP Document, "RAB Configuration for VoIMS", TSG RAN2#45bis, Sophia Antipolis, France, R2-050039, January 2005.

[45] 3GPP Document, "A Comparison Between GERAN Packet-Switched Call Setup Using SIP and GSM Circuit-Switched Call Setup Using RIL3-CC, RIL3-MM, RIL3-RR, and DTAP", TSG GERAN #1 Seattle, WA, August 28 - September 1, 2000.

[46] 3GPP Document, "SIP RAB with SRBs Multiplexing for IMS Voice Service Optimisation", TSG RAN WG2 Meeting #38, Sophia Antipolis, France, 6th-10th October 2003.

[47] IETF RFC 3320 , "Signaling Compression (SIGCOMP)" , January 2003.

[48] Nordberg M., Hannu H., Christoffersson J., Zaccomer L., "Improving SigComp Performance through Extended Operations", IEEE 58th Vehicular Technology Conference 2003.

[49] Vale E., Brandão J.C., Grivet M., "Techniques for Performance Improvement of IMS Voice Calls in UMTS Networks", submetido para publicação na revista da SBRT.

[50] 3GPP Document, "AMR Speech Codec - General Description", TS 26.071, v6.0.0, January 2005.

[51] 3GPP Document, "AMR Wideband Speech Codec - General Description", TS 26.171, v5.0.0, March 2001.

[52] 3GPP Document, "AMR Wideband Speech Codec; Frame Structure", TS 26.201, v5.0.0, March 2001.

## APÊNDICE 1

### Otimização do *Padding* para Grupos de Pacotes

Seja o caso em que está sendo considerado não apenas um mas sim  $Q$  grupos de pacotes sendo transmitidos, todos com o mesmo número  $N$  de elementos. Cada grupo pode ser representado por  $\Omega_q = \{x_1^{(q)}, x_2^{(q)}, \dots, x_N^{(q)}\}$ , todos associados ao mesmo conjunto de probabilidades. Para cada grupo é assumido que i)  $x_1^{(q)} > 0$  ii)  $x_{k+1}^{(q)} > x_k^{(q)}$  para todo valor de  $k$ .

Assim, deseja-se encontrar os valores de  $M$  ( $M < N$ ) comprimentos de quadros que podem acomodar os pacotes acima indicados, independentemente do grupo ao qual pertençam. Estes comprimentos de quadros são representados pelo vetor  $\underline{X} = \{X_1, X_2, \dots, X_M\}$  onde também assume-se que i)  $X_1 > 0$  ii)  $X_{m+1} > X_m$  para todo  $m$ .

A diferença agora, em relação ao caso de um único grupo, é que o objetivo torna-se o de minimização de uma função que sintetize todos os grupos. Um exemplo desta função pode ser aquela que representa o valor médio global de todas as perdas dos codecs associados. Em outros termos, esta nova função objetivo representa a "perda média", sendo dada por

$$D(\underline{X}) = \frac{1}{Q} \sum_{q=1}^Q D_q(\underline{X}) \quad (\text{A1.1})$$

em que

$$D_q(\underline{X}) = \sum_{m=1}^M X_m \cdot [F_q(X_m) - F_q(X_{m-1})] - \overline{x}_q \quad (\text{A1.2})$$

e  $F_q$  é a função distribuição de probabilidade discreta associada ao conjunto de probabilidades  $\{\alpha_k\}$  relativo a  $\Omega_q$ .

Diferentemente do caso tratado na Seção 4.1.1, o segundo somatório de (A1.2), representado por  $\overline{x}_q$ , não pode, via de regra, ser descartado, pois ele depende de  $q$ . Neste caso específico,  $D(\underline{X})$  possui um termo que é o valor médio tomado por todos os grupos, sendo que este termo pode agora ser ignorado, uma vez que ele não depende de  $\underline{X}$ . Conseqüentemente, a expressão pode ser reescrita como

$$D(\underline{X}) = \sum_{m=1}^M X_m \cdot [\bar{F}(X_m) - \bar{F}(X_{m-1})] \quad (\text{A1.3})$$

em que

$$\bar{F}(u) = \frac{1}{Q} \sum_{q=1}^Q F_q(u) \quad (\text{A1.4})$$

O procedimento para se conseguir a otimização é igual àquele desenvolvido na Seção 4.1.1 observando que, pela mesma razão anterior, cada componente  $X$  deve pertencer à união de todos os  $\Omega_q$ .

Outro caso de interesse é a consideração do valor máximo de todas as perdas, para todos os grupos. Nesta situação, tem-se

$$D(\underline{X}) = \max_{(q \in 1..Q)} D_q(\underline{X}) \quad (\text{A1.5})$$

Diferentemente do caso anterior, aqui não se dispõe de uma propriedade que torne este caso similar aos demais, no que concerne às técnicas de programação dinâmica. O fato da função objetivo expressa por (A1.5) ser demasiado descontínua, com a possibilidade de existência de diversos mínimos locais, inibe a utilização de técnicas clássicas de otimização, baseadas no cálculo de gradientes.

Contudo, existem outras técnicas de otimização que poderiam ser empregadas neste caso (*Tabu Search*, *Particle Swarm Optimization*, *Genetic Algorithms*, etc.). Uma vez que tais técnicas se utilizam de diferentes metáforas e são sub-ótimas, não haveria a garantia de que a solução encontrada estivesse próxima da solução ótima.

Por outro lado, torna-se interessante investigar quão próximas estão as soluções definidas pela técnica de otimização pelo valor do *padding* médio e aquela definida pela minimização de (A1.5). A esta última técnica será atribuído o nome de Minimax.

De forma a se testar estes algoritmos propostos, foram selecionados dois grupos constituídos pelos comprimentos de pacotes e taxas de ocorrência para as diferentes taxas de operação de dois tipos de codecs: o AMR e o AMR-WB. Considerou-se a utilização de ROHC e um período de 10 minutos de

conversação<sup>[21]</sup>. Foram então comparadas as soluções obtidas a partir das seguintes técnicas:

- Otimização pela perda média devido a *padding* (A1.1);
- Perda referente ao pior caso de codec na situação anterior;
- Otimização pela técnica de PSO (*Particle Swarm Optimization*);
- Otimização pela técnica minimax (método da "força bruta");

Estes resultados encontram-se apresentados nas Figuras 1.1 e 1.2.

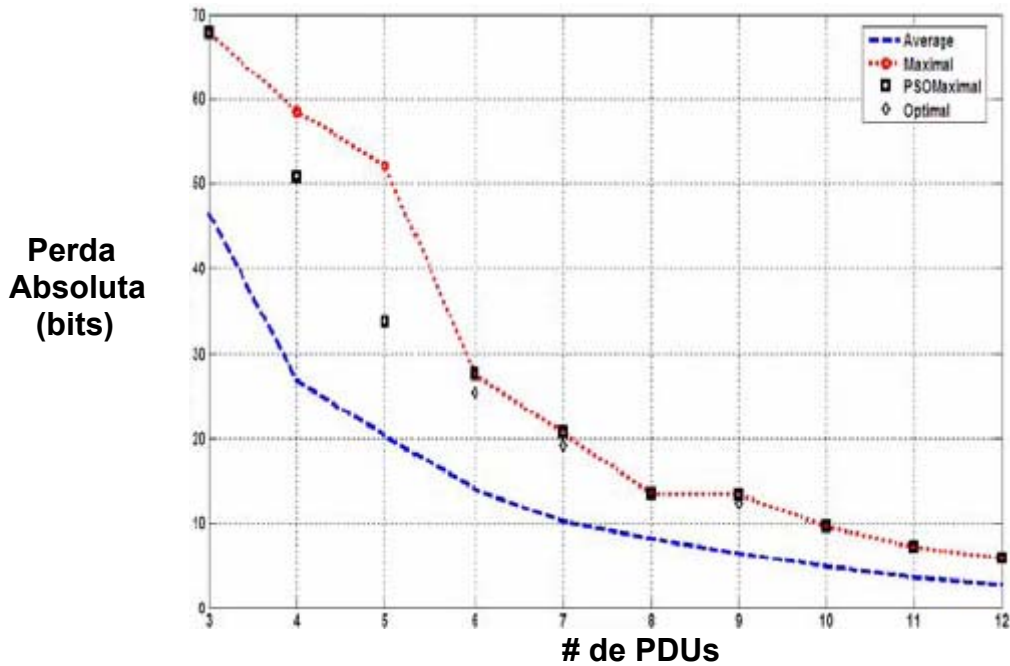


Figura 1.1 - Resultado para o Codec AMR.

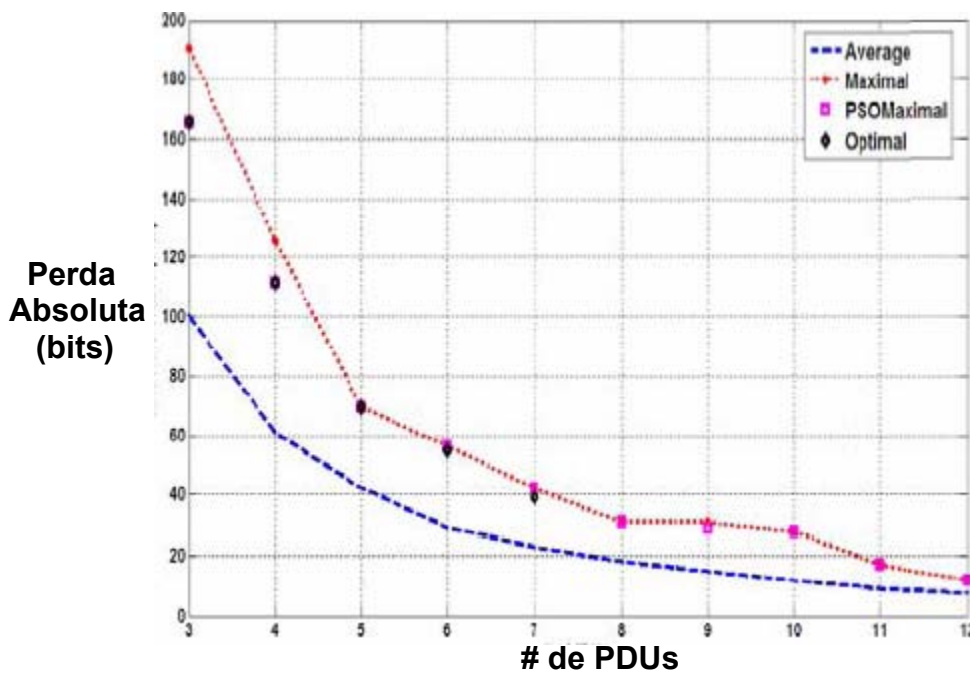


Figura 1.2 - Resultado para o Codec AMR-WB.

Um fato interessante que pode ser observado nestas curvas é que, embora a minimização em curso seja a perda média absoluta, o valor máximo correspondente situa-se, em geral, muito próximo dos resultados relativos à solução Minimax. Nas situações em que tal proximidade não ocorre, as soluções decorrentes do PSO ficam perto da solução Minimax, com um erro relativo não superior a 2%. Este fato pode ser observado para ambos os codecs.

Naturalmente não se pode afirmar que este comportamento vá ocorrer em todos os casos possíveis, porém isso é um indício de que o uso desta técnica pode prover soluções sub-ótimas de boa qualidade, para o caso Minimax.



## APÊNDICE 2

### Considerações sobre o Codec AMR e sua Utilização no UMTS

O codec AMR (Adaptive Multirate)<sup>[50]</sup> é muito utilizado em sistemas celulares. Como o próprio nome sugere, ele pode operar em diversas taxas, que variam desde 4,75 até 12,2 kbps. Uma versão mais recente deste codec, o AMR-WB (AMR-Wideband)<sup>[51,52]</sup>, permite trocas de taxas desde 6,6 até 23,85 kbps.

Para ambos os codecs, um quadro de voz tem a duração de 20ms. No caso do codec AMR, a frequência de amostragem é de 8000 amostras por segundo, o que conduz a 160 amostras de voz por quadro. Para o AMR-WB a frequência de amostragem é de 16000 amostras por segundo, o que acarreta 320 amostras de voz por quadro. Em qualquer caso, o codec somente pode alterar a sua taxa no intervalo de um quadro (20ms).

Os codecs dispõem os bits codificados nos quadros arrumados em graus de importância relativa. Assim, definem-se os bits de Classe A como os de maior importância. Segue-se o grupamento dos bits de Classe B e, por último, o grupamento dos bits de Classe C, considerados os de menor importância. No entanto, modernamente a maioria dos equipamentos decodificadores requisita o envio dos quadros de voz em qualquer caso, mesmo que tenham sido detectados, previamente, erros severos nos bits de Classe A. Parte-se do princípio de que é mais conveniente tentar-se recuperar a inteligibilidade da voz a partir de quadros com erro do que a partir de quadros que tenham sido suprimidos<sup>[41]</sup>.

A Figura 2.1 mostra o formato de um quadro do codec AMR<sup>[52]</sup>.

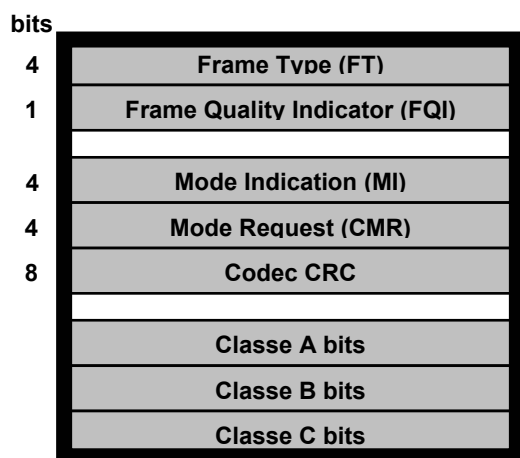


Figura 2.1 - Formato do Quadro Rádio do codec AMR.

O parâmetro *Frame Type* indica a taxa que está sendo utilizada na transmissão do quadro rádio. O parâmetro *Frame Quality Indicator* (FQI) assume o valor 1 quando o quadro contém os grupamentos de bits sem erro. FQI será igual a 0 em caso contrário. O parâmetro *Mode Request* (CMR) indica, para o receptor do quadro, a taxa que o lado emissor espera receber os quadros de retorno. O CRC tem por finalidade proteger os bits de Classe A. O Quadro 2.1 mostra a distribuição de bits, conforme as diversas taxas, para o codec AMR e o Quadro 2.2 para o codec AMR-WB. Note-se que, no caso do codec AMR-WB, só existem grupamentos de bits de Classe A e Classe B.

Quadro 2.1 - Parâmetros do Quadro AMR.

FT	MI	CMR	T(Kbps)	Número de bits			
				Tot.	A	B	C
0	0	0	4.75	95	42	53	0
1	1	1	5.15	103	49	54	0
2	2	2	5.9	118	55	63	0
3	3	3	6.7	134	58	76	0
4	4	4	7.4	148	61	87	0
5	5	5	7.95	159	75	84	0
6	6	6	10.2	204	65	99	40
7	7	7	12.2	244	81	103	60
8	-	-	SID	39	39	0	0

Quadro 2.2 - Parâmetros do Quadro AMR-WB.

FT	MI	CMR	T(Kbps)	Número de bits		
				Tot.	A	B
0	0	0	6.60	132	54	78
1	1	1	8.85	177	64	113
2	2	2	12.65	253	72	181
3	3	3	14.25	285	72	213
4	4	4	15.85	317	72	245
5	5	5	18.25	365	72	293
6	6	6	19.85	397	72	325
7	7	7	23.05	461	72	389
8	8	8	23.85	477	72	405
9	-	-	SID	40	40	0

Nos intervalos de voz é enviado um sinal SID (*Silent Descriptor*) contendo informação que possibilite ao receptor do quadro gerar um ruído de fundo (CN - *Confort Noise*) semelhante ao que está ocorrendo, naquele momento, no lado do emissor.

O Quadro 2.3 a seguir, mostra como são classificados os quadros de voz.

Quadro 2.3 - Classificação dos quadros de voz.

FQI	FT	TX_TYPE	COMENTÁRIOS
1	0-8	SPEECH_GOOD	Quadro de voz sem erro
0	0-8	SPEECH_BAD	Quadro de voz corrompido (erro detectado pelo CRC). A informação do quadro será utilizada pelo codec receptor para reduzir o efeito da falha
0	14	SPEECH_LOST	O FQI e o FT estão corretos, porém não há informação de voz útil. O quadro deve ser transmitido
1	9	SID_FIRST SID_UPDATE	Em SID_FIRST, os 35 bits de informação são "0" Em SID_UPDATE tem-se 35 bits de Comfort Noise
0	9	SID-BAD	Ocorre quando a informação de confort noise de um quadro SID_UPDATE apresenta-se corrompida
1	15	NO_DATA	O quadro (inclusive FQI e FT) não contém informação útil. Não deve ser transmitido

A Figura 2.2 ilustra o processo de envio de quadros de voz/SID. Define-se o *hangover* como sendo o período de 7 quadros de silêncio que deve ocorrer quando termina um surto de voz. Neste período, o parâmetro VAD (*Voice Active Detector*) do transmissor é feito igual a 0. Este período é preenchido com a transmissão de quadros de voz, porém sem informação útil de voz. Os parâmetros dos 7 quadros deverão ser: FQI=0 e FT=14 (SPEECH\_LOST, ver Quadro 1.3).

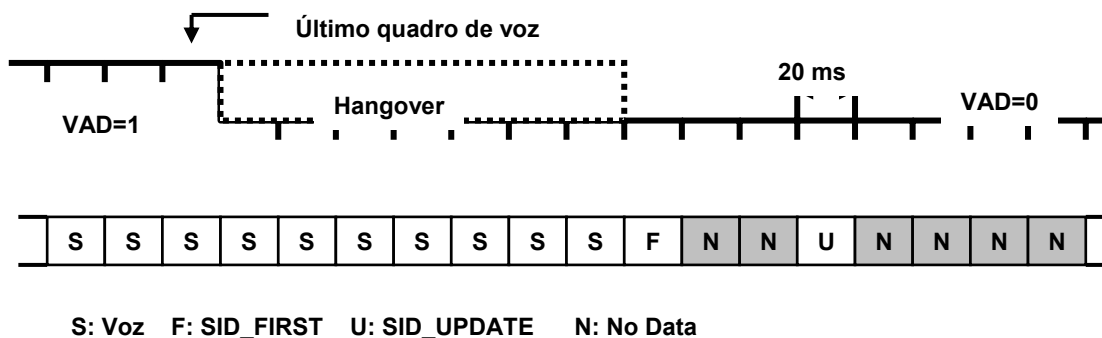


Figura 2.2 - Envio de Quadros de Voz/SID.

Convém observar que o primeiro quadro após o *hangover* é um SID\_FIRST. Após 3 quadros vem o SID\_UPDATE, que irá se repetir a cada 8 quadros.

## APÊNDICE 3

### Descrição do Simulador Desenvolvido para o Rate Matching

#### 3.1

#### Descrição dos Simuladores para Utilização com Codificadores Convolucionais

A Figura 3.1 mostra o diagrama em blocos do simulador para o mecanismo de repetição de bits com o uso de codificadores convolucionais, como descrito na Figura 15 da seção 4.4.2. Foi utilizado o Simulink do Matlab. Os codificadores convolucionais tiveram seus parâmetros adaptados para o cumprimento das recomendações pertinentes do 3GPP<sup>[22]</sup>, ou seja:

- Para o Codificador Convolutacional 1/2 (CC2):
  - *Constraint Length*: 9
  - *Vector Codes*: 561 e 753 (em octal)
- Para o Codificador Convolutacional 1/3 (CC3):
  - *Constraint Length*: 9

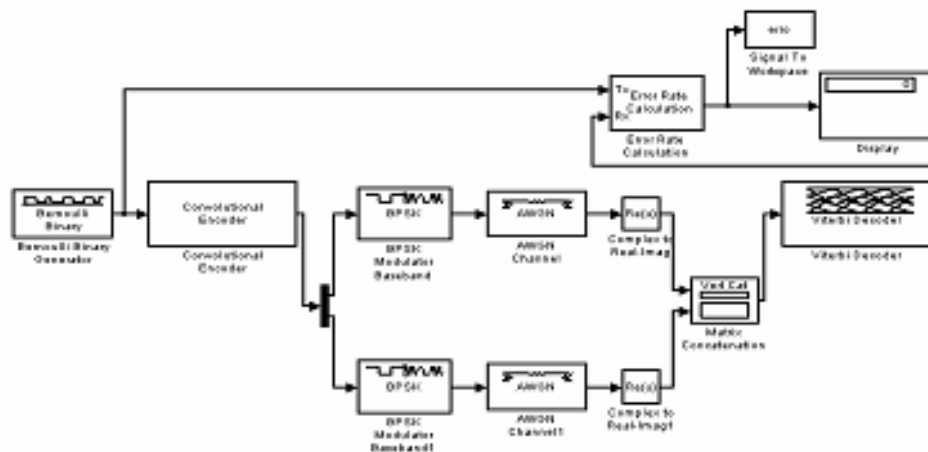


Figura 3.1 - Simulador para Repetição de Bits (CC2 e CC3).

A Figura 3.2 mostra o diagrama em blocos para o caso de utilização de supressão de bits (*puncturing*), ainda para a hipótese de utilização de codificadores convolucionais. Também foi utilizado o Simulink do Matlab.

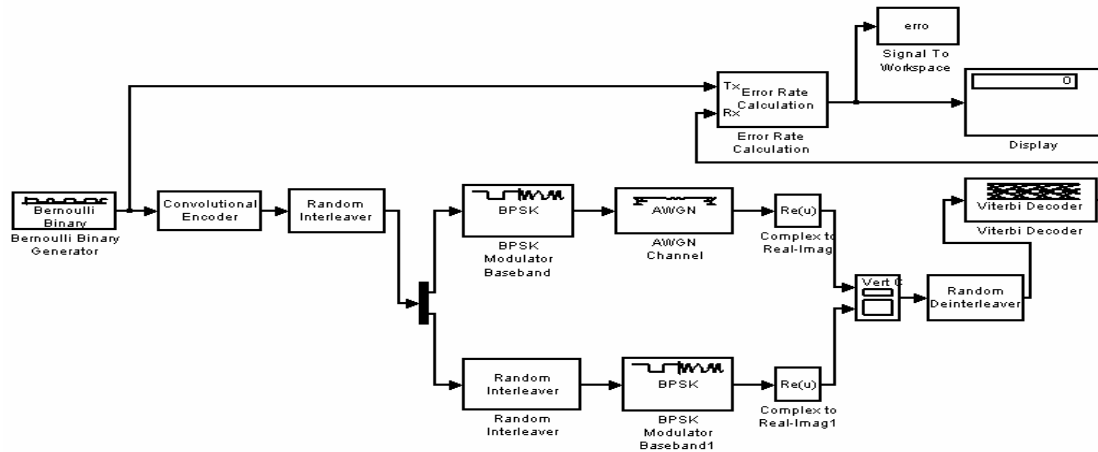


Figura 3.2 - Simulador para Supressão de Bits (CC2 e CC3).

### 3.2

#### Descrição do Codificador Turbo

O Codificador Turbo é recomendado pelo 3GPP para utilização com transmissão de dados. Existem dois tipos de Codificadores Turbo, para os propósitos do UMTS:

- PCCC (*Parallel Concatenated Convolutional Coder*)
- SCCC (*Serial Concatenated Convolutional Coder*)

O 3GPP definiu<sup>[22]</sup> que deveria ser utilizado apenas o PCCC, com os seguintes parâmetros:

- *Constraint Length*: 4
- *Feedforward Generator*: 15 (em octal)
- *Feedback Generator*: 13 (em octal)

As Figuras 3.3 e 3.4 mostram o diagrama esquemático do codificador e do decodificador para o PCCC. Nas figuras, I refere-se a *Interleaver* e APP é o bloco de *A Posteriori Probability*.

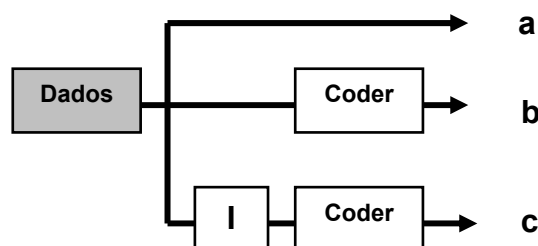


Figura 3.3 - Codificador do PCCC.

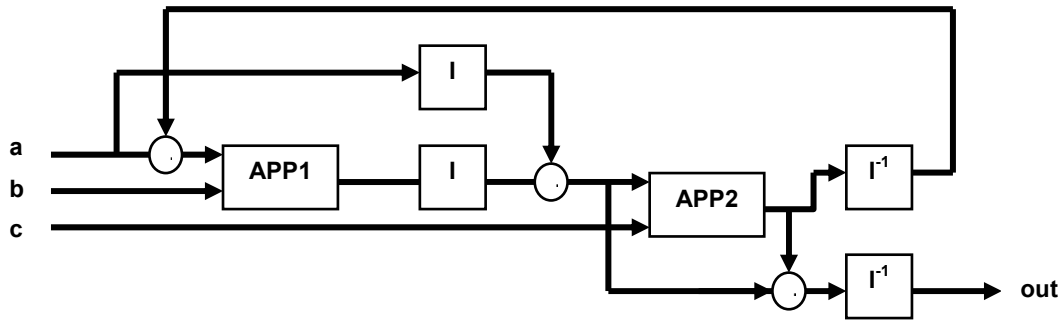


Figura 3.4 - Decodificador do PCCC.

### 3.3

#### Descrição dos Simuladores para Utilização com Codificadores Turbo

As Figuras 3.5 e 3.6 ilustram os diagramas em bloco dos simuladores construídos para os mecanismos de repetição e supressão de bits, respectivamente, utilizando o PCCC. Também aqui utilizou-se o Simulink do Matlab.

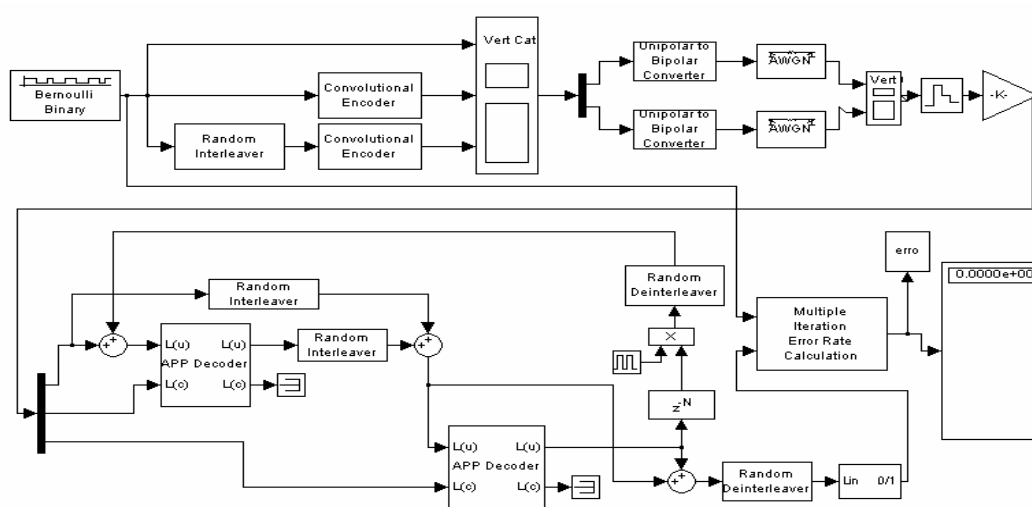


Figura 3.5 - Simulador para Repetição de Bits (TC3).

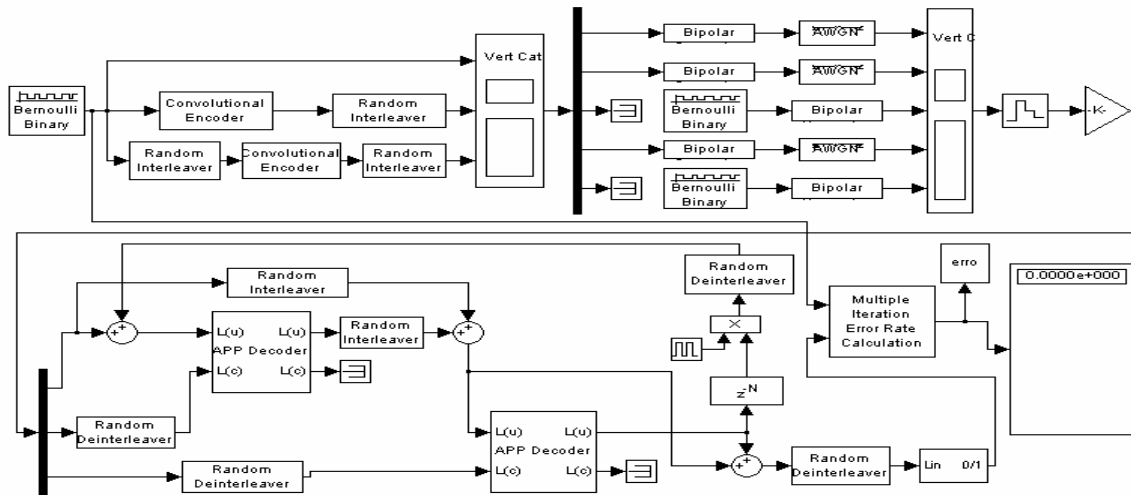


Figura 3.6 - Simulador para Supressão de Bits (TC3).

## APÊNDICE 4

### Um Modelo para a Função $P(N, \Delta N, E_b/N_0)$

A partir de uma inspeção visual nos dados obtidos através do processo de simulação referente aos codificadores convolucionais (CC2 e CC3) e turbo code (TC3), verifica-se que uma razoável aproximação pode ser obtida a partir da seguinte função

$$P(N, \Delta N, E_b / N_0) = 10^{\sum_{i=0}^{N_1} a_i(N, \Delta N) \cdot (E_b / N_0)^i} \quad (\text{A4.1})$$

com  $E_b / N_0$  expresso em dB.

Cada um dos coeficientes  $a$ 's pode ser estimado pelo método dos mínimos quadrados, e modelados por uma expressão da forma

$$a_i(N, \Delta N) = \sum_{j=0}^{N_2} b_{ij}(N) \cdot (\Delta N)^j \quad (\text{A4.2})$$

Repetindo o processo, cada um dos coeficientes  $b$ 's pode ser estimado pelo método dos mínimos quadrados, e modelados por uma expressão da forma

$$b_{ij}(N) = \sum_{k=0}^{N_3} c_{ijk} \cdot N^k \quad (\text{A4.3})$$

Dessa forma, a probabilidade em questão será modelada por

$$P(N, \Delta N, E_b / N_0) = 10^{f[N, \Delta N, E_b / N_0]} \quad (\text{A4.4})$$

em que

$$f[N, \Delta N, E_b / N_0] = \sum_{i=0}^{N_1} \sum_{j=0}^{N_2} \sum_{k=0}^{N_3} c_{ijk} \cdot (E_b / N_0)^i \cdot (\Delta N)^j \cdot N^k \quad (\text{A4.5})$$

Os valores de  $N_1$ ,  $N_2$  e  $N_3$  foram obtidos por tentativas, a partir da observação da melhor concordância dos pontos obtidos por simulação com as curvas decorrentes.



## APÊNDICE 5

### Descrição Simplificada do ROCH - *Robust Header Compression*

#### 5.1

##### Considerações Iniciais

O ROCH, desenvolvido pelo IETF<sup>[36]</sup> é o sistema mais moderno para compressão de cabeçalhos IP/UDP/RTP.

O objetivo é manter-se o que se denomina um *contexto*, que consiste basicamente do conjunto de informações constantes de um cabeçalho não comprimido, tanto no lado do compressor como no lado do descompressor. São então estabelecidos mecanismos para a instalação, verificação do estado de atualização e reparo do contexto, no lado do descompressor. O contexto irá conter informações de caráter fixo e dinâmico, ao longo do período de tempo da conexão. As informações de caráter fixo são aquelas que não variam nos cabeçalhos, em todos os pacotes que são enviados durante a conexão ( p.e. números IP de origem e destino, números de portas, conteúdo do campo de versão do IP, etc.). Já as informações de caráter dinâmico são aquelas que variam ao longo da conexão. São as seguintes:

- *Sequence Number* (SN)
- *Timestamp* (TS)
- *Marker* (M)

O ROCH admite operação em três modos distintos:

1. Modo Unidirecional (U): quando não existe um canal de retorno;
2. Modo Otimista (O): quando existe um canal de retorno, porém para cursar um pequeno tráfego (ACK);
3. Modo Confiável (R): quando existe um canal de retorno propiciando um tráfego mais completo (ACK, NACK). Este é o modo recomendado pelo 3GPP para utilização no UMTS;

O compressor poderá se encontrar em um dos seguintes estados, devendo sempre evoluir para um estado de ordem superior:

- *Initialization and Refresh* (IR): Aqui ele envia as informações de caráter fixo e dinâmico. Deve permanecer neste estado até ter

certeza de que as informações enviadas estão disponíveis no descompressor;

- *First Order* (FO): O compressor irá atualizar o contexto do descompressor com informações de ordem dinâmica, quando isto for necessário;
- *Second Order* (SO): Neste estado o compressor envia pacotes com o cabeçalho comprimido. É o estado onde a compressão é ótima;

O descompressor poderá estar em um dos estágios vistos a seguir, sendo que ele deverá começar do primeiro e evoluir para os seguintes:

- *No Context*: ocorre quando o descompressor ainda não conseguiu descomprimir um pacote;
- *Static Context*: quando o descompressor já possui as informações corretas de caráter fixo;
- *Full Context*: o descompressor é capaz de descomprimir corretamente os pacotes;

Observar que, uma vez que o descompressor receba as informações IR, ele poderá ir diretamente do estado *No Context* para *Full Context*. Caso ocorram falhas na descompressão dos cabeçalhos, o descompressor deverá cair para o estado *Static Context*, quando então receberá informações do compressor (que estará no estado FO).

## 5.2

### Princípio de Operação

A forma como o ROCH opera é por meio, inicialmente, do estabelecimento de funcionalidades entre o campo SN (*Sequence Number*) e os demais campos dos cabeçalhos. Então, a operação irá se resumir em manter o descompressor sempre atualizado com relação ao valor corrente de SN. Assim, por exemplo, o *Timestamp* (TS) poderá ser calculado a partir do valor de SN, uma vez que as comunicações de áudio e vídeo ocorrem em quadros de comprimento fixo e definido. Para o caso da transmissão de áudio com o codec AMR, em 20ms tem-se 160 amostras, como visto no Apêndice 1, de modo que o TS será acrescido de 160 unidades em SNs sucessivos.

### 5.3

#### Tipos de Pacotes

O ROCH opera com os seguintes tipos de pacotes:

- *Inicialização* (IR): Contém as informações fixas e dinâmicas, para o estabelecimento do contexto no descompressor. Poderá possuir um *payload* inicial agregado;
- *Inicialização Dinâmica* (IR-DYN): Contém somente as informações dinâmicas. É utilizado para atualização de contexto;
- *Feedback*: Para retorno de informação sobre o estado de descompressão;
- *Payload* com cabeçalho comprimido;

Em seguida serão analisados os diversos tipos de pacotes, com ênfase para aqueles do modo Confiável (R), que são de maior interesse para o UMTS. Os pacotes são agrupados em: ordem 0, ordem 1, ordem 2 e pacotes de *feedback*. As letras iniciais da classificação indicam os modos aos quais o pacote é aplicável.

- Pacotes de ordem 0: São aqueles que contém o cabeçalho no maior grau de compressão (o descompressor encontra-se no estado SO)
  - R-0: Não devem proporcionar atualização de contexto;
  - UO-0: Atualizam o valor corrente do contexto (apenas SN);
  - R-0-CRC: Podem ser utilizados para atualização de contexto, pois o CRC incluído é calculado a partir do cabeçalho original;

A Figura 5.1 ilustra estes pacotes.

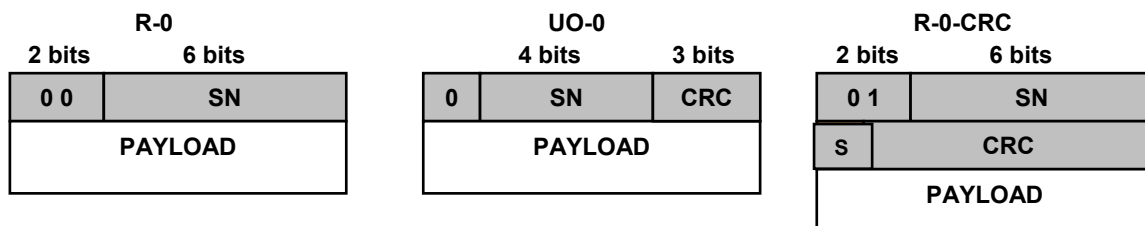


Figura 5.1 - Pacotes de Ordem 0.

Convém observar que o pacote carrega apenas os últimos (menos significativos) bits do SN. Este procedimento também vale para o TS, quando utilizado.

- Pacotes de ordem 1: São utilizados quando se necessita enviar mais bits de SN (ou o TS e o M)

A Figura 5.2 mostra o formato dos pacotes R-1 e UO-1. No caso do R-1, outros parâmetros podem ser enviados, através do uso de extensões (X=1, pacote com extensão, X=0 em caso contrário). Uma vez que R-1 não contém CRC, ele não poderá ser utilizado para atualização de contexto.

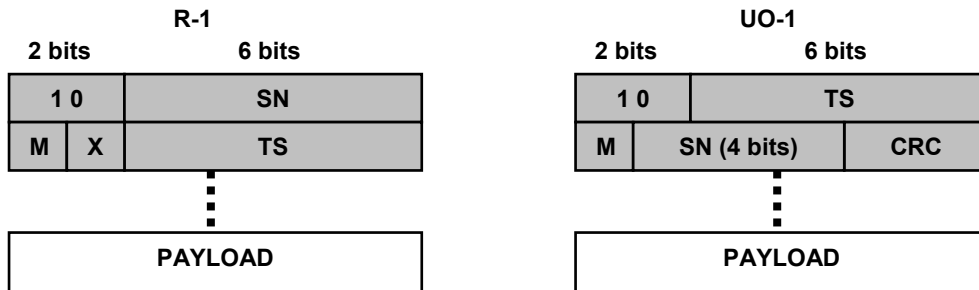


Figura 5.2 - Pacotes de Ordem 1.

- Pacotes de ordem 2: São utilizados quando se necessita enviar mais bits de SN (ou o TS e o M), sendo que, neste caso, outros parâmetros podem ser incluídos, através do uso de extensões. Podem atualizar contextos, uma vez que possuem o CRC.

A Figura 5.3 ilustra esta categoria de pacote.

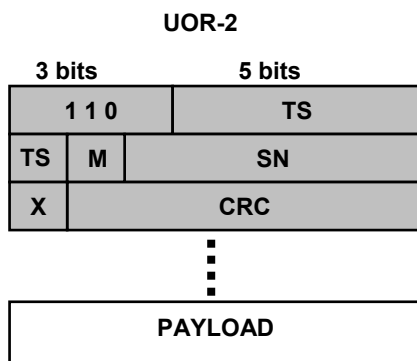


Figura 5.3 - Pacote de Ordem 2.

- Extensão de pacotes: O ROHC define 4 tipos de extensões, sendo que a de número 3 é a mais utilizada, para os propósitos deste trabalho.
  - Extensão 3: Serve para enviar sequências maiores (mais bits) do SN, o valor completo do TS, os parâmetros *Traffic-Class*, *Time-to-Live*, *Next Header* e outros. A Figura 5.4 ilustra a extensão para o caso em que se deseja enviar mais bits do SN.

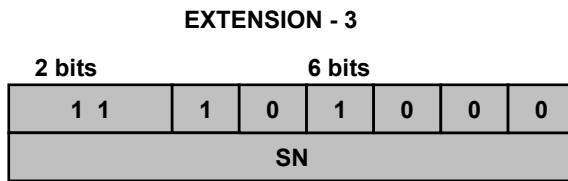


Figura 5.4 - Extensão 3.

- Pacotes de *Feedback*: Carregam a informação de retorno. Os mais utilizados são o ACK e o *Static NACK*. A Figura 5.5 ilustra estes pacotes.

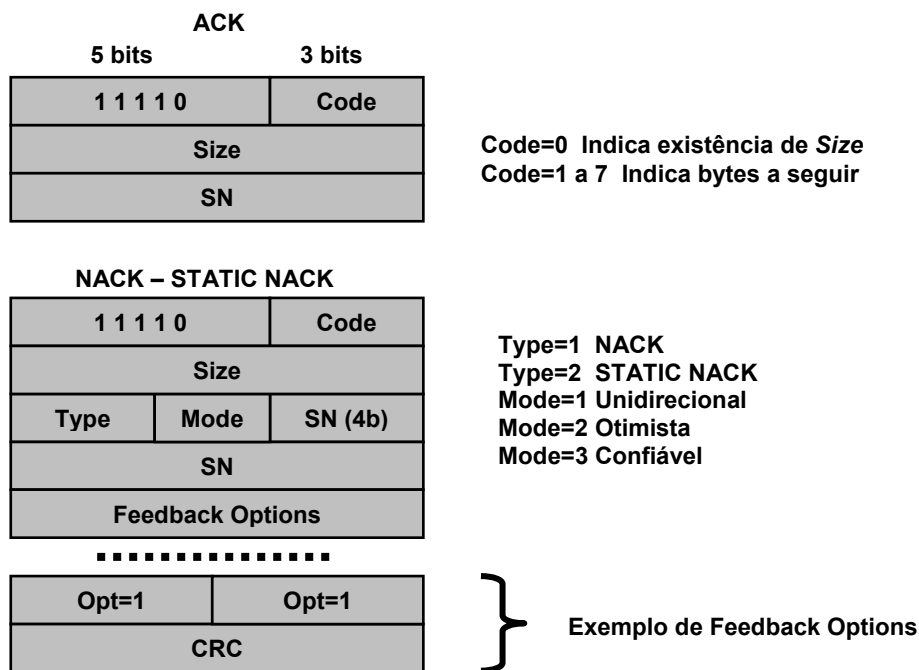


Figura 5.5 - Pacotes de *Feedback*.

## 5.4

### Operação do Modo Confiável

As observações seguintes sintetizam a operação do Modo Confiável:

- O Modo Confiável, como mencionado anteriormente, faz uso intensivo do canal de retorno;
- Para cada sequência de pacotes R-0-CRC descomprimidos corretamente, é enviada uma mensagem ACK para o compressor;
- Pacotes que chegam ao descompressor sem CRC (p.e. R-0) não geram ACKs, mesmo que corretamente descomprimidos;

- Caso seja recebido um NACK pelo compressor, ele irá enviar um pacote de atualização (p.e. R-1);
- Certos parâmetros da lógica do sistema ficam a cargo do implementador (p.e. frequência do envio de ACKs);

A seguir, é mostrada a sequência de eventos para um exemplo hipotético:

1. O compressor envia uma sequência de pacotes IR;
2. O compressor vai para o estado SO e envia pacotes R-0 até que julgue conveniente atualizar SN;
3. O compressor passa a enviar  $m$  pacotes R-0-CRC (o valor de  $m$  é da escolha do implementador);
4. Quando o compressor recebe ACK, ele torna a enviar pacotes R-0;
5. Caso o compressor julgue conveniente atualizar TS, ele envia o pacote R-2;
6. Sempre nos limites de uma sequência de voz/intervalo, o compressor atualiza M, com o envio de um pacote R-2;

## APÊNDICE 6

### Descrição do Simulador Utilizado para Análise do *Checksum* e CRC

Para a obtenção das curvas das Figuras 35, 36, 37 e 38 foram desenvolvidos simuladores utilizando os recursos proporcionados pelo Simulink do Matlab. A Figura 6.1 ilustra o diagrama em blocos do simulador utilizado para a obtenção das curvas das Figuras 35, 36 e 37. Para se obter os dados constantes da Figura 38, deve-se substituir o bloco denominado BSC (*Binary Symmetric Channel*) pelos blocos de "Modulador/Demodulador BPSK" e "Canal Rayleigh", do Simulink.

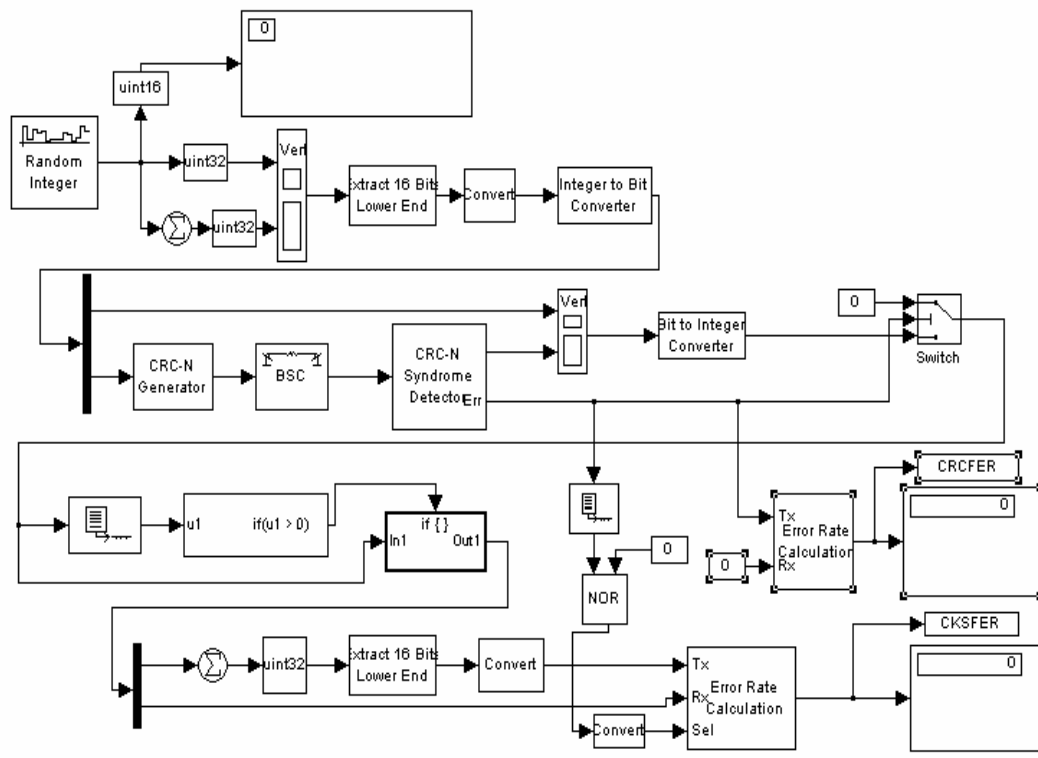


Figura 6.1 - Simulador para Obtenção das Curvas das Figuras 35, 36 e 37.

As curvas existentes na Figura 41, foram obtidas com o auxílio do simulador apresentado na Figura 6.2, também desenvolvido com recursos do Simulink do Matlab.

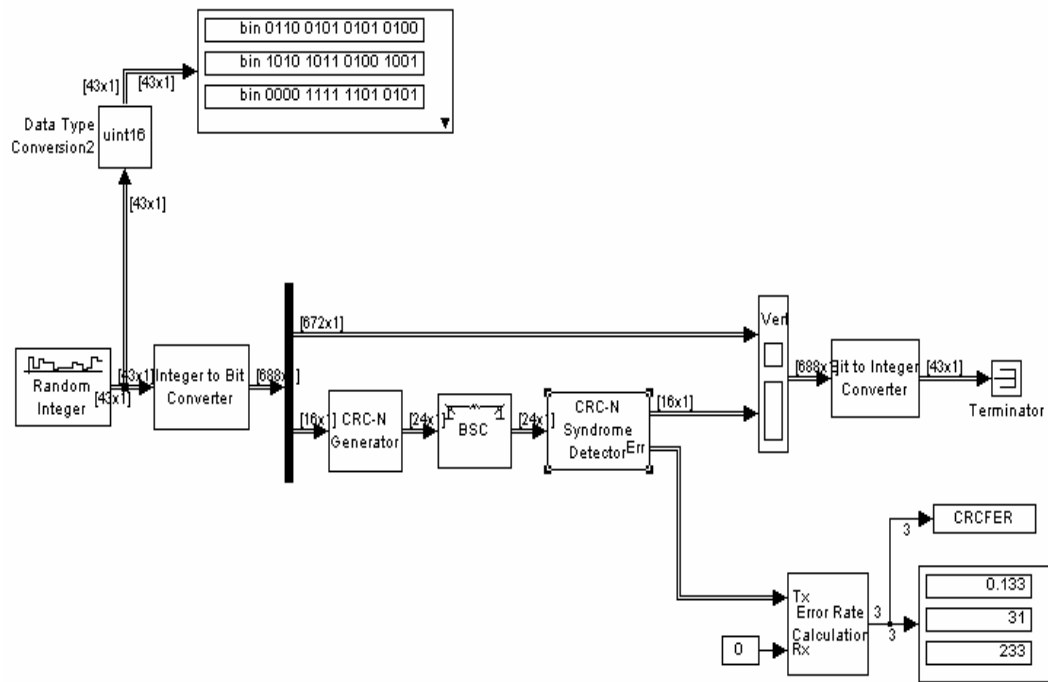


Figura 6.2 - Simulador para Obtenção das Curvas da Figura 42.