

UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA

DEPARTAMENTO DE COMUNICAÇÕES

UM ESTUDO COMPARATIVO ENTRE AS  
DIVERSAS ESTRATÉGIAS DE REQUISIÇÃO DE  
REPETIÇÃO AUTOMÁTICA

ARQ

Thales Coelho Borges Lima

Orientador

Prof. Dr. Renato Baldini Filho

Tese apresentada à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do Título de Mestre em Engenharia Elétrica.

Campinas, março de 1995

Este exemplar corresponde à redação final da tese  
defendida por Thales Coelho Borges Lima

e aprovada pela Comissão

Julgadora em 13 / 03 / 95

Renato Baldini Filho  
Orientador

# Resumo

Esta tese faz uma análise comparativa entre as estratégias de Requisição de Repetição Automática (ARQ) mais relevantes encontradas na literatura. As comparações são feitas em termos de throughput, que determina o desempenho dos diversos esquemas ARQ com relação às condições do canal de comunicação. Os parâmetros que determinam o melhor desempenho de um dado esquema ARQ sobre um outro são; os protocolos de transmissão/recepção, o tamanho do buffer do receptor e os códigos empregados nos processos de detecção e correção. A primeira parte deste trabalho, descreve o modelo do sistema de comunicação e os esquemas ARQ básicos: stop-and-wait, go-back- $N$  e selective-repeat. Em seguida, os esquemas ARQ selective-repeat mais significativos encontrados na literatura especializada, são descritos e comparados entre si. Esta classe de esquemas ARQ básicos apresentam um melhor desempenho, em termos de throughput, que o stop-and-wait e go-back- $N$ . A seguir, esquemas ARQ híbridos, particularmente o tipo-II, são descritos e estudados de maneira comparativa. Os esquemas híbridos apresentam desempenho superior, para altas taxas de erro, aos esquemas ARQ básicos, embora sejam um pouco mais complexos em termos de implementação e funcionamento. Os esquemas híbridos são capazes de simultaneamente detectar e corrigir erros, ao passo que, os esquemas ARQ básicos, utilizam apenas o procedimento de detecção de erro. Vários esquemas ARQ híbridos tipo-II adaptativos também são apresentados, uma vez que sua utilização se faz necessária em canais de comunicação onde as condições são variantes. Finalmente, são sugeridas algumas novas estratégias ARQ e adaptações de esquemas propostos para melhorar seus desempenhos.

# Agradecimentos

Agradeço a Deus, acima de tudo, por ter conseguido concluir mais uma etapa em minha vida.

Agradeço ao meu orientador e amigo, Professor Doutor Renato Baldini Filho, pela sua profunda colaboração para a realização deste trabalho; aconselhando e sugerindo-me idéias que foram muito importantes.

Agradeço também, a todos os professores com os quais tive a oportunidade de conviver dentro e fora da sala de aula, e em particular ao Professor Doutor Wilmar Bueno de Moraes, por valiosos conselhos e por ter sido um grande amigo.

Gostaria de agradecer a todos os meus grandes amigos que fiz durante este tempo que passei por esta universidade: Alcides Cintra, Alencar, Alexandre Surerus, Alexandre Amorim, Andrade, Carlos Pingarilho, Christian, Dinho, Elton, Fábio, Fançony, Flávia, Gerônimo, Ladislau, Leila, Leonardo Resende, Leonardo Guedes, Luiz Gonzaga, Marcelo Castro, Marcelo Segatto, Márcio, Marco Antonio, Mateus, Milaré, Moisés, Ricardo, Rodrigo, Rosângela, Sérgio, Tetsu... Amigos estes, que sempre poderei contar por toda a minha vida.

Finalmente, agradeço aos meus pais, que sempre incentivaram, apoiaram e ajudaram, em todos os momentos de minha vida, se sacrificando para sempre me proporcionar o que há de melhor.

# Glossário

$a_d^{(i)}$	Espectro de distância de um código convolucional.
$a(i), i = 1, 2, \dots$	Matrizes de puncionamento.
$a(X)$	Quociente de uma divisão.
$A_i$	Número de palavras de peso $i$ .
ACK	Sinal de recebimento correto.
ARQ	Sistema de Requisição de Repetição Automática.
ARQ-GH	Esquema ARQ híbrido tipo-II generalizado.
$b(X)$	Resto de uma divisão.
$B$	Número de blocos de informação de $K^2$ bits.
BCH	Códigos Bose Chaudhuri Hocquenghem
BSC	Canal Binário Simétrico.
$\mathbf{c}$	Palavra-código.
$c_i, i = 1, 2, \dots, m$	Segmentos de $n$ bits de $\mathbf{c}$ .
$C$	Código cíclico $(n, k)$ .
CPTC	Códigos convolucionais compatíveis em taxa.
$C_0$	Código de bloco de detecção de erro.

$C_1$	Código de bloco ou convolucional corretor de erro.
$C_3$	Código convolucional $(3, 2, m)$ .
$d$	Capacidade de detecção de erro e Distância de Hamming mínima de um código.
$d_i, i = 1, 2, \dots, m$	Distâncias de Hamming mínimas.
$d_{free}$	Distância livre do código.
$d_{min}$	É a menor distância de Hamming.
$D$	Mensagem de $k$ bits de informação.
$\tilde{D}$	Mensagem de $k$ bits de informação recebida pelo receptor.
$(D, Q)$	Palavra-código de informação.
$(\tilde{D}, \tilde{Q})$	Palavra recebida pelo receptor.
$E_{c,i}$	Evento no qual o receptor recupera a palavra corretamente.
$E_{d,i}$	Evento no qual o receptor detecta a presença de erros.
$E_{e,i}$	Evento onde o receptor não detecta a presença de erros.
$E_i$	Palavra estimada de uma palavra-código de informação $I_i$ .
$f_T$	Resto da divisão de $(ns - ns_0)$ por $3N$ .
$f_R$	Resto da divisão de $(ns - ns_0)$ por $3N$ .
FEC	Esquema de controle de erro.
FS	Estado de um flag.
FNS	Estado de um flag.
$g(X)$	Polinômio gerador de um código cíclico.

$G$	Matriz geradora de um código.
$G_i, i = 1, 2, \dots, m$	Submatrizes geradoras.
$G_i^{-1}$	Submatriz inversa.
$G_1(X)$	Polinômio gerador de um código convolucional.
$G_2(X)$	Polinômio gerador de um código convolucional.
$G_{31}(X)$	Polinômio gerador de um código convolucional de taxa 2/3.
$G_{32}(X)$	Polinômio gerador de um código convolucional de taxa 2/3.
$G_{1/2}$	Throughput do esquema ARQ híbrido tipo-II modificado.
GBN	Modo go-back- $N$ .
$h(X)$	Polinômio gerador da matriz de verificação de paridade.
$I$	Palavra-código de informação.
$\tilde{I}$	Palavra de informação relativa a $I$ no receptor.
$I_{n/4}$	Matriz identidade $4 \times n$ .
$I^*$	Palavra de informação decodificada.
$\tilde{I}^*(X)$	Sequência decodificada por um código convolucional.
$\tilde{I}(P)$	Palavra de $n$ bits.
$I(X)$	Mensagem de informação expressa em uma forma polinomial.
$I_1(\tilde{X})$	Quociente da divisão de $\tilde{P}_1(X)$ por $G_1(X)$ .
$I_2(\tilde{X})$	Quociente da divisão de $\tilde{P}_2(X)$ por $G_2(X)$ .
$(I, P(I))$	Palavra-código.
$k$	Comprimento da mensagem de informação.

$K$	Tamanho do bloco a ser codificado.
KM	Classe de códigos lineares.
$l_b$	Resto da divisão de $z_0 - ns$ por $3N$ .
$l_f$	Resto da divisão de $ns - ns_0$ por $3N$ .
$l$	Vetor de informação.
$m$	Número de bits de paridade, Profundidade de um código corretor de erro, Número de níveis de cascata e Memória de um codificador convolucional.
$M$	Matriz geradora.
$M_i$	Submatriz de $M$ .
$M_i^{-1}$	Submatriz inversa.
$n$	Comprimento da palavra-código.
$n$	Comprimento, em bits, de uma palavra-código.
$n_2$	Número de cópias de uma palavra-código.
$n_q$	Número de cópias de uma palavra-código.
$ns$	Número de sequência de uma palavra-código.
$N$	Número de palavras armazenadas no enlace.
NAK	Sinal de recebimento negativo.
$NP_d$	Número esperado de palavras com erro.
$p$	Probabilidade de transição de um canal BSC.

$p'$	Probabilidade de erro de bit.
$p''$	Probabilidade de erro de bit.
$p_c$	Probabilidade de decodificação correta.
$p_e$	Probabilidade de erro de decodificação.
$P$	Probabilidade de uma palavra ser aceita pelo receptor e Período de funcionamento.
$P_b$	Probabilidade de erro de bit após a decodificação.
$P_c$	Probabilidade de recebimento correto.
$P_C$	Probabilidade de decodificação correta.
$P_{c,i}$	Probabilidade de decodificação correta.
$P_{cs}$	Probabilidade do receptor recuperar a palavra de informação.
$P_d$	Probabilidade de que a palavra recebida contenha um padrão de erro detectável.
$P_{dec}$	Probabilidade de decodificação correta.
$P_E$	Probabilidade de decodificação incorreta.
$P_r(E_{c,i})$	Probabilidade do receptor obter a palavra de informação corretamente.
$P_r(E_{d,0}E_{c,1})$	Probabilidade conjunta.
$P_r(E_{d,0}E_{d,1}E_{c,2})$	Probabilidade conjunta.
$P_r(E_{c,1} E_{d,0})$	Probabilidade condicional.
$P_{r_d}$	Probabilidade de selecionar um caminho errado no algoritmo de Viterbi.
$P_t$	Probabilidade de recuperar uma palavra de informação.
$P(D)$	Palavra de $k$ bits de paridade.



$\tilde{P}(D)$	Palavra de $k$ bits de paridade que chega no receptor.
$(P(D), Q^{(1)})$	Palavra-código de paridade.
$(\tilde{P}(D), \tilde{Q}^{(1)})$	Palavra recebida pelo receptor correspondente à $(P(D), Q^{(1)})$ .
$P(D_e^{(i)})$	Probabilidade de que a sequência decodificada não contenha erro.
$P(D_d^{(i)})$	Probabilidade de que a sequência decodificada contenha um erro detectável.
$P(D_e^{(i)})$	Probabilidade de que a sequência decodificada contenha um erro não detectável.
$P(E)$	Probabilidade do receptor cometer um erro de decodificação.
$P(E^{(i)})$	Probabilidade do receptor cometer um erro de decodificação.
$P(E)_H$	Probabilidade do receptor cometer um erro de decodificação.
$P(E)_{HM}$	Probabilidade do receptor cometer um erro de decodificação.
$P(I)$	Palavra de $n$ bits de paridade.
$\tilde{P}(I)$	Palavra de paridade relativa a $P(I)$ .
$P_{NN}$	Probabilidade de receber dois blocos NAK's consecutivos referentes a uma mesma palavra.
$P(X)$	Sequência resultante do intercalamento de $P_1(X)$ , $P_2(X)$ e $P_3(X)$ .
$\tilde{P}(X)$	Sequência recebida pelo receptor relativa a $P(X)$ .
$P_1(X)$	Sequência de saída 1 codificada.
$\tilde{P}_1(X)$	Sequência que chega no receptor correspondente à $P_1(X)$ .
$\tilde{P}_1^{(1)}(X)$	Primeira sequência que chega no receptor correspondente à $P_1(X)$ .
$\tilde{P}_1^{(2)}(X)$	Segunda sequência que chega no receptor correspondente à $P_1(X)$ .
$P_2(X)$	Sequência de saída 2 codificada.

$\tilde{P}_2(X)$	Sequência que chega no receptor correspondente à $P_2(X)$ .
$\tilde{P}_2^{(1)}(X)$	Primeira sequência que chega no receptor correspondente à $P_2(X)$ .
$\tilde{P}_2^{(2)}(X)$	Segunda sequência que chega no receptor correspondente à $P_2(X)$ .
$P_3(X)$	Sequência resultante da soma de $P_{31}(X)$ e $P_{32}(X)$ .
$\tilde{P}_3(X)$	Sequência que chega no receptor correspondente à $P_3(X)$ .
$P_{31}(X)$	Sequência de saída 3 codificada, onde $I(X)$ é carregada na entrada 1 do codificador.
$P_{32}(X)$	Sequência de saída 3 codificada, onde $I(X)$ é carregada na entrada 2 do codificador.
$q$	Número de níveis de um esquema ARQ de Weldon.
$q_1$	Probabilidade condicional de decodificação correta.
$q_0$	Probabilidade de decodificação correta de uma palavra recebida pelo receptor.
$Q$	Número de bits de paridade.
$Q^{(1)}$	Número de bits de paridade.
$Q_1$	Probabilidade condicional.
$Q_2$	Probabilidade condicional.
$r$	Número de vezes que um bloco é retransmitido no esquema de Weldon.
$\tilde{R}_1(X)$	Resto da divisão.
$\tilde{R}_2(X)$	Resto da divisão.
$s_{prop}$	Número de bits que o enlace armazena.
SR	Modo selective-repeat.
ST	Modo sutter.

SR+GBN	Esquema ARQ misto selective-repeat + go-back- $N$ .
SR+ST	Esquema ARQ misto selective-repeat + stutter.
$\tilde{S}_1(X)$	Notação polinomial da síndrome.
$\tilde{S}_2(X)$	Notação polinomial da síndrome.
$t$	Capacidade de correção de erro.
$t_1$	Capacidade de correção de erro projetada.
$t_{emis}$	Tempo de atraso em emitir uma palavra.
$t_{prop}$	Atraso de propagação em segundos.
$T_{CC}$	Número médio de sequências transmitidas.
$T_{CP}$	Número médio de transmissões.
$T_{GBN}$	Número médio de transmissões.
$T_{HC}$	Número médio de transmissões.
$T_{SR}$	Número médio de transmissões.
$T_{SW}$	Número médio de transmissões.
$T(X, Y)$	Função geradora de um código convolucional.
$\mathbf{u}(X)$	Mensagem de informação.
$\mathbf{u}_1(X)$	Mensagem de informação.
$\mathbf{u}_2(X)$	Mensagem de informação.
$v$	Número de palavras transmitidas.
$\mathbf{v}$	Palavra-código de informação.
$\mathbf{v}_j$	Palavra não recebida corretamente.
$\mathbf{v}^*$	Palavra-código de paridade.

$\tilde{\mathbf{v}}^*$	Palavra-código correspondente a $\mathbf{v}^*$ .
$\mathbf{v}(X)$	Palavra-código.
$\mathbf{w}_1(X)$	Palavra-código.
$\mathbf{w}_2(X)$	Palavra-código.
$y_l$	Sequência de $d$ bits.
$z_j$	Sequência de $k$ bits.
$z_0$	Número de sequência.
$\tau$	Taxa de transmissão de dados em bits por segundo.
$\lambda$	Tempo inativo entre duas transmissões sucessivas.
$\phi_0$	Probabilidade de receber uma palavra-código com sucesso na segunda retransmissão.
$\phi_1$	Probabilidade de receber uma palavra-código com sucesso na terceira retransmissão.
$\phi_2$	Probabilidade de receber uma palavra-código com sucesso na quarta retransmissão.
$\sigma$	Probabilidade de que o número de erros em $(\tilde{P}(D), \tilde{D})$ , ultrapasse a capacidade projetada de detecção de erro do código $C_1$ .
$\eta_{CC}$	Throughput do esquema ARQ híbrido tipo-II com combinação de códigos.
$\eta_{CP}$	Throughput do esquema ARQ híbrido tipo-II com códigos puncionados.
$\eta_{GBN}$	Throughput do esquema ARQ go-back- $N$ .
$\eta_{GH_1}$	Throughput do esquema ARQ híbrido tipo-II generalizado que utiliza um código corretor de erro KM $(2l', l')$ .
$\eta_{GH_2}$	Throughput do esquema ARQ híbrido tipo-II generalizado que utiliza um código corretor de erro KM $(3l', l')$ .

$\eta_{HC}$	Throughput do esquema ARQ híbrido tipo-II com código de Hamming em cascata.
$\eta_{HM}$	Throughput do esquema ARQ híbrido tipo-II modificado considerando o buffer do receptor de tamanho infinito.
$\eta_{HMF_1}$	Throughput do esquema ARQ híbrido tipo-II modificado com buffer de tamanho finito $B$ , onde $B \leq N$ .
$\eta_{HMF_2}$	Throughput do esquema ARQ híbrido tipo-II modificado com buffer de tamanho finito $B$ , onde $B = s(N - 1) + 1 + b$ e $b$ e $s$ são dois números inteiros positivos assumindo que $b \leq N - 1$ .
$\eta_{II}$	Throughput do esquema ARQ híbrido tipo-II com buffer do receptor de tamanho $N$ .
$\eta_{SR}$	Throughput do esquema ARQ selective-repeat com buffer do receptor de tamanho infinito.
$\eta_{SRF}$	Throughput do esquema ARQ selective-repeat com buffer do receptor de tamanho finito e igual a $N$ .
$\eta_{SR+GBN}$	Throughput do esquema ARQ misto selective-repeat + go-back- $N$ .
$\eta_{SR+ST(1)}$	Throughput do esquema ARQ misto selective-repeat + stutter tipo 1.
$\eta_{SR+ST(2)}$	Throughput do esquema ARQ misto selective-repeat + stutter tipo 2.
$\eta_{SW}$	Throughput do esquema ARQ stop-and-wait.
$\eta_{1/2}$	Throughput do esquema ARQ híbrido tipo-II com código convolucional de taxa $1/2$ .

# Conteúdo

<b>1</b>	<b>ESTRATÉGIAS DE REQUISIÇÃO DE REPETIÇÃO AUTOMÁTICA - ARQ</b>	<b>1</b>
1.1	Introdução . . . . .	1
1.2	O Modelo do Sistema . . . . .	4
1.2.1	Enlace de Dados . . . . .	4
1.2.2	Transmissor . . . . .	5
1.2.3	Receptor . . . . .	5
1.3	Esquemas ARQ Básicos . . . . .	5
1.3.1	Esquema Stop-and-Wait . . . . .	6
1.3.2	Esquema Go-Back- <i>N</i> . . . . .	7
1.3.3	Esquema Selective-Repeat . . . . .	8
1.4	Desempenho dos Esquemas ARQ . . . . .	9
1.5	Códigos de Blocos Lineares para Detecção de Erro . . . . .	15
<b>2</b>	<b>ESQUEMAS ARQ SELECTIVE-REPEAT</b>	<b>17</b>
2.1	Introdução . . . . .	17
2.2	Esquemas ARQ com Modos Mistos de Retransmissão . . . . .	18
2.2.1	Esquemas Selective-Repeat mais Go-Back- <i>N</i> . . . . .	18
2.2.2	Esquema Selective-Repeat mais Stutter . . . . .	20
2.2.3	Esquema de Weldon . . . . .	25
2.3	Esquema de Detecção de Overflow no Buffer do Receptor . . . . .	30
2.3.1	Procedimentos de Transmissão e Retransmissão . . . . .	31
2.3.2	Operação do Receptor e Procedimento para Recuperação de Erros . . . . .	32
2.3.3	Operação Estado Normal . . . . .	34
2.3.4	Operação Estado Bloqueado . . . . .	36
2.3.5	Throughput do Sistema ARQ Selective-Repeat com Buffer Finito . . . . .	38
2.4	Comparações Entre os Diversos Esquemas Selective-Repeat . . . . .	40
2.5	Conclusão . . . . .	44

<b>3</b>	<b>ESQUEMAS ARQ HÍBRIDOS USANDO CÓDIGOS DE BLOCOS</b>	<b>45</b>
3.1	Introdução . . . . .	45
3.2	Esquema ARQ Híbrido Tipo-I . . . . .	46
3.3	Esquema ARQ Híbrido Tipo-II . . . . .	48
3.3.1	Códigos de Blocos Inversível de Taxa 1/2 . . . . .	52
3.4	Esquemas ARQ Híbridos Práticos . . . . .	52
3.4.1	Esquema ARQ Selective-Repeat Híbrido Tipo-II com Buffer Finito no Receptor . . . . .	52
3.4.2	Esquema ARQ Selective-Repeat Híbrido Tipo-II Modificado . . . . .	61
3.4.3	Esquema ARQ Híbrido Tipo-II Generalizado - ARQ-GH . . . . .	67
3.4.4	Esquema ARQ Híbrido Usando um Sistema de Codificação em Cascata . . . . .	72
3.5	Comparações Entre os Diversos Esquemas ARQ Híbridos Selective-Repeat que Utilizam Códigos de Blocos . . . . .	83
3.6	Conclusão . . . . .	87
<b>4</b>	<b>ESQUEMAS ARQ HÍBRIDOS USANDO CÓDIGOS CONVOLUCIONAIS</b>	<b>90</b>
4.1	Introdução . . . . .	90
4.2	Esquema ARQ híbrido tipo-II usando código convolucional de taxa 1/2 . . . . .	91
4.3	Esquema ARQ híbrido tipo-II usando código convolucional de taxa 2/3 . . . . .	94
4.4	Esquema ARQ Híbrido Tipo-II com Combinação de Códigos . . . . .	98
4.5	Esquema ARQ Híbrido tipo-II com Códigos Convolucionais Puncionados . . . . .	101
4.6	Comparações Entre os Diversos Esquemas ARQ Híbridos Tipo-II Selective-Repeat que Utilizam Códigos Convolucionais . . . . .	106
4.7	Conclusão . . . . .	109
<b>5</b>	<b>CONCLUSÕES FINAIS</b>	<b>111</b>
5.1	Comentários e Comparações entre os Esquemas ARQ . . . . .	111
5.2	Futuros Trabalhos . . . . .	118
5.3	Conclusão Final . . . . .	121
<b>A</b>	<b>Processo de Inversão</b>	<b>123</b>
<b>B</b>	<b>Descrição dos Códigos KM</b>	<b>127</b>
	<b>Referências</b>	<b>129</b>

# Lista de Figuras

1.1	Modelo do Sistema de Comunicação. . . . .	4
1.2	ARQ Stop-and-Wait. . . . .	6
1.3	ARQ Go-Back- $N$ com $N = 7$ . . . . .	7
1.4	ARQ Selective-Repeat. . . . .	8
1.5	Canal Binário Simétrico. . . . .	10
1.6	Throughput do ARQ selective-repeat com buffer do receptor infinito, go-back- $N$ e stop-and-wait com $n = 1000$ bits, $\tau = 1$ Mbit/s, e tempo de atraso de percurso igual a 700 ms. . . . .	13
1.7	Throughput do ARQ selective-repeat com buffer do receptor infinito, go-back- $N$ e stop-and-wait com $n = 1000$ bits, $\tau = 1$ Mbit/s, e tempo de atraso de percurso igual 30 ms. . . . .	14
1.8	Throughput do ARQ selective-repeat para vários comprimentos de códigos. . . . .	14
2.1	Esquema ARQ SR+GBN com $v = 1$ e $N = 5$ . . . . .	19
2.2	Esquema ARQ SR+ST 1 com $v = 1$ e $N = 5$ . . . . .	21
2.3	Esquema ARQ SR+ST 2 com $v = 1$ e $N = 5$ . . . . .	22
2.4	Throughput do esquema ARQ misto SR+GBN, para vários valores de $v$ , considerando $n = 1024$ bits e $N = 128$ palavras. . . . .	23
2.5	Throughput do esquema ARQ misto SR+ST 1, para vários valores de $v$ , considerando $n = 1024$ bits e $N = 128$ palavras. . . . .	24
2.6	Comparação do throughput do esquema ARQ misto SR+ST 2, com SR+ST 1 ( $v = 1$ ), selective-repeat ideal e go-back- $N$ , considerando $n = 1024$ bits e $N = 59$ palavras. . . . .	24
2.7	Comparação dos vários esquemas ARQ selective-repeat mistos, considerando $n = 1024$ palavras e $N = 1024$ palavras. . . . .	25
2.8	Comparação do throughput do esquema de Weldon com $q = 1$ , para diversos valores de $n_1$ , considerando o tempo de atraso de percurso $N = 128$ palavras. . . . .	28
2.9	Comparação do throughput do esquema de Weldon para $q = 1$ , para diversos valores de $n_1$ , considerando $N = 1024$ palavras. . . . .	28
2.10	Comparação do throughput do esquema de Weldon para vários valores de $q$ , considerando todos os $n_i = 1$ , onde $i = 1, 2, \dots, q$ . . . . .	29



2.11	Comparação do throughput do esquema de Weldon para vários valores de $n_i$ , considerando $N = 1024$ palavras: (1) esquema de Weldon com $q = 3$ e $n_1 = n_2 = n_3 = 1$ ; (2) esquema de Weldon com $q = 3$ , $n_1 = n_2 = 1$ e $n_3 = 3$ ; (3) esquema de Weldon com $q = 3$ , $n_1 = 2$ , $n_2 = 4$ e $n_3 = 6$ . . . . .	29
2.12	Procedimento de transmissão e retransmissão. . . . .	33
2.13	Esquema ARQ selective-repeat com buffer do receptor de tamanho $N = 7$ . . . . .	34
2.14	Operação no Estado Normal. . . . .	36
2.15	Operação Estado Bloqueado do receptor. . . . .	39
2.16	Operação estado bloqueado do receptor. . . . .	40
2.17	Throughput dos esquemas ARQ selective-repeat: (1) esquema SR+GBN para $v = 1$ ; (2) esquema de Weldon com $q = 1$ e $n_1 = 1$ ; (3) esquema SR+ST 1 para $v = 1$ ; (4) esquema ARQ selective-repeat com buffer finito de tamanho $N$ , capaz de detectar overflow no buffer do receptor; (5) selective-repeat ideal. . . . .	41
2.18	Throughput dos esquemas ARQ selective-repeat: (1), (2) e (3) esquema SR+GBN para $v = 1$ , $v = 2$ e $v = 3$ , respectivamente; (4) e (5) esquema ARQ selective-repeat com buffer finito de tamanho $N$ e $2N$ , respectivamente, capaz de detectar overflow no buffer; (6) selective-repeat ideal. . . . .	42
2.19	Comparação do throughput dos esquemas ARQ SR + ST1 com os esquemas de Weldon: (1) e (2) esquema de Weldon com $q = 1$ e $n_1 = 2$ e $n_1 = 3$ , respectivamente; (3) e (4) esquema ARQ misto SR+ST 1 com $v = 2$ e $v = 3$ , respectivamente; (5) esquema de Weldon com $q = 3$ , $n_1 = 1$ , $n_2 = 2$ e $n_3 = 3$ ; (6) esquema ARQ selective-repeat ideal. . . . .	43
2.20	Comparação do throughput dos esquemas ARQ SR + ST1 com os esquemas de Weldon: (1), (2) e (3) esquema de Weldon com $q = 1$ e $n_1 = 1$ , $n_1 = 2$ e $n_1 = 3$ , respectivamente; (4) e (5) esquema ARQ selective-repeat com buffer finito de tamanho $N$ e $2N$ , respectivamente, capaz de detectar overflow no buffer; (6) esquema de Weldon com $q = 3$ , $n_1 = 1$ , $n_2 = 2$ e $n_3 = 3$ ; (7) esquema ARQ selective-repeat ideal. . . . .	43
3.1	Throughput do esquema ARQ híbrido selective-repeat tipo-I para diversas capacidades de correção de erro : (1) selective-repeat ideal; (2) código BCH (1023,1013) com $t = 1$ ; (3) código BCH (1023,993) com $t = 3$ ; (4) código BCH (1023,953) com $t = 7$ ; (5) código BCH (1023,828) com $t = 20$ . . . . .	48
3.2	Esquema ARQ selective-repeat híbrido tipo-II com buffer do receptor de tamanho $N = 7$ . . . . .	53

3.3	Throughput de vários esquemas ARQ com $N = 128$ palavras e $n = 511$ bits: (1) ARQ selective-repeat ideal com buffer do receptor infinito; (2), (3) e (4) ARQ selective-repeat híbrido tipo-II com buffer do receptor de tamanho $N$ e parâmetro de correção de erro $t = 3, 5, e 10$ , respectivamente. . . . .	59
3.4	Throughput de vários esquemas ARQ com $N = 1023$ palavras e $n = 1023$ bits: (1) ARQ selective-repeat ideal com buffer do receptor infinito; (2), (3), (4) e (5) ARQ selective-repeat híbrido tipo-II com buffer do receptor de tamanho $N$ e parâmetro de correção de erro $t = 3, 5, 10 e 20$ , respectivamente. . . . .	60
3.5	Throughput de vários esquemas ARQ híbridos tipo-II modificado com buffer infinito e $t = 3$ , tendo como parâmetro o comprimento $n$ do código $C_1$ . . . . .	64
3.6	Throughput do esquema ARQ híbrido tipo-II modificado com buffer infinito e $n = 1023$ : (1) Selective-repeat ideal; (2), (3), (4) e (5) Esquemas híbridos tipo-II modificado com $t = 3, 5, 10 e 16$ , respectivamente. . . . .	65
3.7	Throughput do esquema ARQ selective-repeat tipo-II modificado com tamanho de bloco $n = 510$ bits e tempo de atraso de percurso $N = 128$ palavras: (1) ARQ selective-repeat ideal com buffer do receptor de tamanho infinito; (2) Esquema ARQ selective-repeat básico com buffer de tamanho $N$ ; (3), (4), (5) e (6) Esquemas ARQ selective-repeat híbrido tipo-II modificado com buffer de tamanho $N$ e capacidade de correção de erro projetada $t = 3, 5, 10 e 30$ , respectivamente. . . . .	67
3.8	Throughput do esquema ARQ selective-repeat tipo-II modificado com $n = 1023$ bits e tempo de atraso de percurso $N = 1024$ palavras: (1) ARQ selective-repeat ideal com buffer do receptor de tamanho infinito; (2), (3), (4) e (5) Esquemas ARQ híbrido tipo-II modificado com $t = 3, 5, 10 e 50$ , respectivamente. . . . .	68
3.9	Throughput dos esquemas ARQ-GH: (1) Selective-repeat ideal; (2) ARQ-GH utilizando um código KM (8,4,3); (3) ARQ-GH utilizando um código KM (12,4,5); (4) ARQ-GH utilizando um código KM (27,7,9). . . . .	73
3.10	Codificação para dois níveis de cascata . . . . .	74
3.11	Decodificação para dois níveis de cascata . . . . .	75
3.12	Taxa de erro de bit decodificada comparada à taxa de erro de bit do canal para dois códigos de Hamming . . . . .	76
3.13	Entrelaçamento para três níveis de cascata usando um código de Hamming (7,4) . . . . .	77

3.14	Throughput de três esquemas em cascata usando código de Hamming: (1) (7,4) e $n = 1344$ bits; (2) (15,11) e $n = 1331$ bits; (3) (31,26) e $n = 1352$ bits. . . . .	81
3.15	Comparação do throughput: (1) e (2) ARQ híbrido tipo-II modificado com $t = 5$ e $10$ , respectivamente; (3) e (4) ARQ híbrido tipo-I com $t = 5$ e $10$ , respectivamente. . . . .	83
3.16	Comparação do throughput: (1) ARQ híbrido tipo-I com $n = 1300$ bits e $t = 5$ ; (2) e (3) ARQ híbrido tipo-II em cascata, utilizando código de Hamming (15,11) e (31,26), respectivamente. . . . .	84
3.17	Comparação de Throughput: (1) ARQ selective-repeat ideal; (2), (3) e (4) ARQ híbrido modificado com $t = 5, 10$ e $20$ , respectivamente; (5) e (6) ARQ híbrido GH usando código KM (8,4,3) e (12,4,5), respectivamente. . . . .	85
3.18	Comparação de Throughput: (1), (2) e (3) ARQ Híbrido tipo-II modificado com $n = 1300$ e $t = 10, 20$ e $40$ , respectivamente; (4), (5) e (6) ARQ Híbrido tipo-II em cascata, com código de Hamming (7,4), (15,11) e (31,26), respectivamente. . . . .	86
3.19	Comparação do throughput: (1) ARQ selective-repeat ideal; (2) e (3) ARQ híbrido GH usando código KM (8,4,3) e (12,4,5), respectivamente, (4), (5) e (6) ARQ híbrido em cascata usando código de Hamming (7,4), (15,11) e (31,26), respectivamente. . . . .	87
4.1	Throughput do esquema ARQ híbrido tipo-II com código convolucional de taxa $1/2$ com buffer do receptor infinito: (1) Esquema Selective-repeat ideal; (2), (3), (4) e (5) Esquema híbrido tipo-II com código convolucional e memória do codificador $m$ igual a 4, 6, 8 e 10, respectivamente. . . . .	93
4.2	Throughput do esquema ARQ híbrido tipo-II com código convolucional de taxa $1/2$ com buffer do receptor de tamanho finito: (1) Esquema Selective-repeat ideal; (2), (3), (4) e (5) Esquema híbrido tipo-II com código convolucional e memória do codificador $m$ igual a 4, 6, 8 e 10, respectivamente. . . . .	94
4.3	Codificador para um código convolucional (3, 2, 2) . . . . .	96
4.4	Throughput do esquema ARQ híbrido tipo-II com taxa $2/3$ comparado com esquema ARQ híbrido tipo-II com taxa $1/2$ , considerando $m = 6$ . . . . .	97
4.5	Throughput do esquema ARQ híbrido tipo-II com combinação de códigos de repetição considerando o buffer do receptor infinito e utilizando um código convolucional (2, 1, 6), com polinômios geradores (133,171). . . . .	102
4.6	Exemplo de um codificador convolucional com duas tabelas de puncionamento compatíveis em taxa e período de puncionamento igual a 4. . . . .	103

4.7	Throughput do esquema ARQ híbrido tipo-II com códigos puncionados com diferentes taxas iniciais: (1) Selective-repeat ideal; (2) e (3) Esquema ARQ híbrido tipo-II com taxas iniciais iguais a $4/5$ e $7/8$ , respectivamente. . . . .	105
4.8	Throughput do esquema ARQ híbrido tipo-II com códigos convolucionais puncionados para diversos valores de $h$ . . . . .	106
4.9	Comparação do throughput: (1) Esquema ARQ selective-repeat ideal; (2) Esquema ARQ híbrido tipo-II com código convolucional de taxa $1/2$ e $m = 6$ e (3) Esquema ARQ híbrido tipo-II com combinação de código e $m = 6$ . . . . .	107
4.10	Comparação do throughput: (1) Esquema ARQ selective-repeat ideal; (2) Esquema ARQ híbrido tipo-II com código convolucional de taxa $1/2$ ; (3) Esquema ARQ híbrido tipo-II com combinação de código; (4) e (5) Esquemas ARQ híbridos tipo-II com códigos puncionados com $h = 1$ e taxas iniciais iguais a $4/5$ e $7/8$ , respectivamente. . . . .	107
4.11	Comparação do throughput: (1) Esquema ARQ híbrido tipo-II com código convolucional de taxa $1/2$ ; (2) Esquema ARQ híbrido tipo-II com combinação de código; (3), (4) e (5) Esquemas ARQ híbridos tipo-II com códigos puncionados de taxas iniciais iguais a $7/8$ e $h = 1, 2$ e $4$ , respectivamente. . . . .	108
4.12	Comparação do throughput: (1) Esquema ARQ selective-repeat ideal; (2) Esquema ARQ híbrido tipo-II com código convolucional de taxa $1/2$ ; (3) Esquema ARQ híbrido tipo-II com combinação de código; (4) Esquema ARQ híbrido tipo-II com código convolucional de taxa $2/3$ ; (5) e (6) Esquemas ARQ híbridos tipo-II com códigos puncionados com $h = 1$ e taxas iniciais iguais a $4/5$ e $7/8$ , respectivamente. . . . .	109
5.1	Throughput dos esquemas ARQ híbridos tipo-II com códigos em Cascata: (1) Esquema ARQ Selective-repeat ideal; (2) Código BCH (15,7) com $t = 2$ e $n = 1029$ ; (3) Código de Hamming (15,11) com $t = 1$ e $n = 1331$ . . . . .	115
5.2	Throughput dos esquemas ARQ híbridos tipo-II com códigos em cascata: (1) Esquema ARQ Selective-repeat ideal; (2) Código BCH (31,11) com $t = 5$ e $n = 1331$ ; (3) Código de Hamming (31,26) com $t = 1$ e $n = 1352$ . . . . .	116

5.3	Throughput de vários esquemas ARQ híbridos tipo-II: (1) esquema ARQ híbrido tipo-II com código convolucional de taxa $1/2$ , memória do codificador $m = 10$ e $n = 512$ bits; (2) Esquema ARQ híbrido tipo-II com combinação de códigos, memória $m = 10$ e $n = 512$ bits; (3) e (4) Esquemas ARQ híbridos tipo-II com código de Hamming (31,26) e $n = 1352$ e $676$ respectivamente; (5) Esquema ARQ híbrido tipo-II com códigos convolucionais puncionados com $h = 1$ , taxa inicial $7/8$ e $n = 512$ . . . . .	119
A.1	Circuito de Inversão . . . . .	126

# Lista de Tabelas

1.1	Cáculo de $P_c$ , $P_e$ e $P(E)$ para alguns códigos BCH. . . . .	11
4.1	Distâncias livres ( $d_{free}$ ) de duas famílias de códigos de repetição obtidos a partir de um código convolucional $(2, 1, 6)$ com polinômios geradores $(133, 171)$ . . . . .	100

# Capítulo 1

## ESTRATÉGIAS DE REQUISIÇÃO DE REPETIÇÃO AUTOMÁTICA - ARQ

### 1.1 Introdução

O objetivo deste trabalho é realizar comparações em termos de "throughput", entre os diversos esquemas ARQ encontrados na literatura especializada, analisando as estratégias de retransmissão e os códigos utilizados nos processos de detecção e correção de erro de um determinado esquema ARQ. O desempenho dos esquemas de requisição de repetição automática ARQ propostos na literatura especializada, são geralmente comparados com os esquemas ARQ básicos. Quando se deseja comparar os vários esquemas ARQ propostos, é necessário fixar alguns parâmetros para que estes esquemas funcionem nas mesmas condições de canal. A fixação destes parâmetros para cada esquema ARQ não é imediata para o cálculo do desempenho em termos de throughput.

A maior preocupação em comunicação de dados é como controlar os erros causados por ruídos do canal, para que dados sem erros possam ser liberados para o usuário. Uma solução para este problema é a aplicação de codificação, isto é, o uso de códigos corretores ou detectores

de erros. Há duas categorias básicas de esquemas de controle de erros em comunicação de dados: esquemas de requisição de repetição automática (ARQ - *automatic-repeat-request*) e esquemas corretores de erro (FEC - *forward-error-correction*).

Em qualquer sistema de controle de erro ARQ, é utilizado um código detector de erro de alta taxa ( $k/n$ ), ou seja, um código de bloco linear ( $n,k$ ) incorporado a um protocolo de retransmissão. Quando uma mensagem de  $k$  bits de informação estiver pronta para ser transmitida, são adicionados a ela  $n - k$  bits de verificação de paridade para formar uma palavra-código. Estes  $n - k$  bits são formados baseados no código utilizado pelo sistema. A palavra-código é então transmitida para o receptor. A palavra-código transmitida é contaminada por ruídos do canal, e como consequência, a palavra recebida correspondente pode conter erros de transmissão.

Utilizando o sistema de controle de erro ARQ, dados com erros serão passados para o usuário, somente se o receptor falhar no processo de detecção de erros. Usando um código detector de erros apropriado, a probabilidade de não detectar um erro é muito pequena. Esquemas ARQ são amplamente usados em sistemas de comunicação de dados para controle de erros, porque são simples e de alta confiabilidade. Entretanto, estes esquemas apresentam uma desvantagem: o throughput (razão entre o número total de bits de informação realmente entregues ao usuário e o número total de bits recebidos) não é constante e cai rapidamente com o aumento da taxa de erro do canal.

Em um sistema de controle de erro FEC, é utilizado um código corretor de erro (código de bloco ou convolucional) para corrigir os erros de transmissão. Novamente, bits de verificação de paridade são adicionados à cada mensagem de  $k$  bits de informação a ser transmitida, para formar uma palavra-código baseada no código utilizado pelo sistema. Quando o receptor detecta a presença de erros em uma palavra recebida, ele tenta localizar e corrigir estes erros. Após corrigi-los, a palavra decodificada é então liberada para o usuário. Um erro de decodificação é cometido se o receptor falhar na detecção de erros ou falhar em determinar as localizações exatas dos erros. Nestes dois casos, uma palavra com erro será passada para o usuário. Como não existe retransmissão em sistemas de controle de erro FEC, não há necessidade do canal de retorno. O throughput do sistema é constante, e é igual a taxa do código utilizado pelo sistema.

Sistemas de controle de erro FEC também possuem algumas desvantagens. Quando erros



são detectados na palavra recebida, ela deve ser decodificada e a palavra decodificada deve ser passada para o usuário sem considerar se a palavra está correta ou não. Desde que a probabilidade de um erro de decodificação é muito maior que a probabilidade de não detectar um erro, é difícil obter um sistema com uma alta confiabilidade usando esquemas FEC. Para conseguir um sistema com alta confiabilidade, um longo e poderoso código corretor de erro deve ser utilizado e uma grande coleção de padrões de erros devem ser corrigidos. Isto resulta em uma decodificação de difícil implementação e de alto custo. Por estas razões, esquemas ARQ são frequentemente preferidos sobre esquemas FEC para controle de erros em sistemas de comunicações de dados tais como: rede de comutação de pacotes e redes de comunicação entre computadores. Entretanto, em sistemas de comunicação (ou armazenamento de dados) onde canais de retorno não são disponíveis ou a retransmissão não é adequada por alguma razão, o esquema FEC é a única opção.

Este trabalho trata de esquemas ARQ. Eles representam uma solução para projeto de protocolos de retransmissão e particularmente, para o modo no qual o transmissor armazena, ordena e retransmite as palavras-código que tenham sido recebidas com erros. Diferentes esquemas ARQ surgem primeiramente, na tentativa de combater a diminuição do throughput de um sistema de controle de erro ARQ quando a taxa de erro do canal aumenta. Esta diminuição é causada pelo tempo adicional dispendido devido a um maior número de retransmissões de palavras recebidas com erros. Este problema torna-se particularmente drástico, se houver um tempo de atraso de percurso<sup>1</sup> significativo entre a transmissão de uma palavra-código e o recebimento da informação de status de erro pelo transmissor. Um longo atraso é inevitável quando se utiliza satélites ou longos canais terrestres.

Uma outra solução para o controle de erros, é utilizar esquemas ARQ híbridos, os quais incorporam os esquemas FEC e retransmissões (ARQ). Esquemas ARQ híbridos possuem um desempenho em termos de throughput superior aos sistemas que utilizam esquemas ARQ puros. Nos esquemas FEC, pode ser utilizado no processo de correção de erro, código de blocos ou código convolucional.

---

<sup>1</sup>Intervalo de tempo entre a transmissão de uma palavra código e o recebimento pelo transmissor do reconhecimento desta palavra.

## 1.2 O Modelo do Sistema

O sistema a ser considerado, consiste de um transmissor conectado a um receptor através de um meio de comunicação (enlace de dados), conforme mostra a figura 1.1. O transmissor envia palavras de  $n$  bits para o receptor. Devido ao ruído introduzido pelo canal, estas palavras ao serem recebidas, podem conter erros. O receptor também envia palavras de  $n$  bits para o transmissor e estas são assumidas não possuir erros. Os palavras de retorno conduzem ao transmissor a informação do estado do receptor. Os principais componentes do sistema são descritos abaixo.

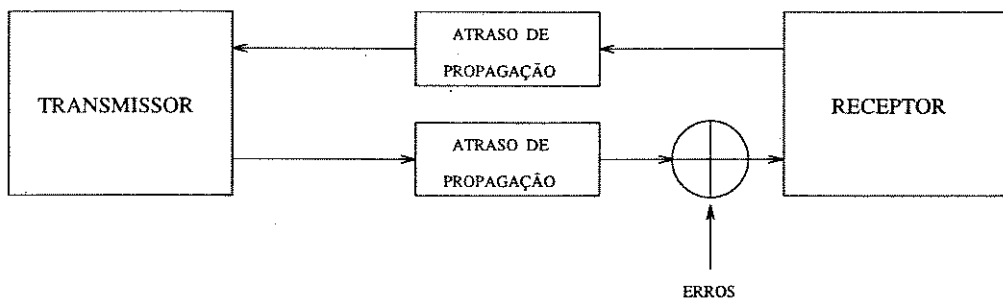


Figura 1.1: Modelo do Sistema de Comunicação.

### 1.2.1 Enlace de Dados

O enlace é capaz de transmissão bidirecional simultaneamente, a uma taxa de  $\tau$  bits/s em cada direção. Tem um atraso de propagação de  $t_{prop}$  segundos em cada direção. Portanto, o enlace armazena:

$$s_{prop} = 2\tau t_{prop} \quad (1.1)$$

bits de dados. O atraso na transmissão de uma palavra através de um enlace em ambas as direções é  $t_{emis} + t_{prop}$ , onde  $t_{emis}$  é o tempo de atraso de emitir uma palavra à taxa  $\tau$ . Isto é:  $t_{emis} = n/\tau$ . No receptor, há um atraso adicional entre o recebimento de uma palavra e o envio de uma resposta; do mesmo modo, no transmissor há um atraso entre o recebimento de uma resposta e a transmissão da próxima palavra. Ajustando o posicionamento relativo das duas

palavras, pode-se forçar um desses atrasos para zero, enquanto o outro, não será maior que o tempo de uma palavra. Deste modo, o número de palavras armazenadas no enlace é<sup>2</sup>:

$$N = \left\lceil \frac{\tau(3t_{emis} + 2t_{prop})}{n} \right\rceil \quad (1.2)$$

e o tempo de atraso de percurso é de  $nN/\tau$  segundos.

### 1.2.2 Transmissor

O transmissor forma palavras numeradas (ou seja, com um número de sequência) e passa-as para o meio de transmissão (enlace). Cada palavra contém uma sequência de bits de paridade e de controle. O transmissor e o receptor conversam por meio de um protocolo ARQ, o qual informa ao transmissor o estado de cada palavra recebida pelo receptor. O transmissor armazena uma cópia de cada palavra enviada, até receber uma confirmação de recebimento positiva do receptor referente a esta palavra. Como a capacidade de armazenamento do enlace é de  $N$  palavras, o transmissor também deve ser capaz de armazenar  $N$  palavras.

### 1.2.3 Receptor

No receptor, é calculada a síndrome de cada palavra recebida. Se a síndrome for diferente de zero, isto indica que foi detectado erro na palavra transmitida. O transmissor é então notificado que a palavra recebida contém erro por meio do canal de retorno. Uma cópia desta palavra que foi recebida com erro, é então transmitida para substituir a palavra recebida com erro.

## 1.3 Esquemas ARQ Básicos

Os sistemas ARQ podem ser divididos basicamente em três esquemas: stop-and-wait, go-back- $N$  e selective-repeat. Utiliza-se a nomenclatura inglesa para estes esquemas, pelo fato de serem universalmente aceitas.

Estes esquemas são descritos a seguir.

---

<sup>2</sup>a notação  $\lceil x \rceil$  refere-se ao menor inteiro maior ou igual a  $x$ .

### 1.3.1 Esquema Stop-and-Wait

Num sistema de controle de erro ARQ stop-and-wait, o transmissor envia uma palavra-código para o receptor e espera por uma confirmação de recebimento referente a esta palavra, conforme mostra a figura 1.2. Se a confirmação de recebimento for positiva (ACK), significa que a palavra transmitida foi recebida corretamente e o transmissor pode iniciar a transmissão da próxima palavra-código. Se a confirmação de recebimento for negativa (NAK), indica que foi detectado erro nesta palavra. O transmissor então inicia a retransmissão desta palavra e espera novamente por uma resposta do receptor. As retransmissões continuam até que o transmissor receba um ACK referente à palavra transmitida.

Este esquema é simples, mas ineficiente devido ao tempo inativo entre a confirmação do recebimento da palavra-código transmitida e a transmissão (ou retransmissão) de outra palavra-código. Uma solução para melhorar a eficiência deste esquema, seria aumentar o tamanho  $n$  da palavra. Mas o uso de códigos de grandes comprimentos resulta no aumento do número de retransmissões, pois quanto maior for o tamanho da palavra, maior será a probabilidade de ocorrer erro.

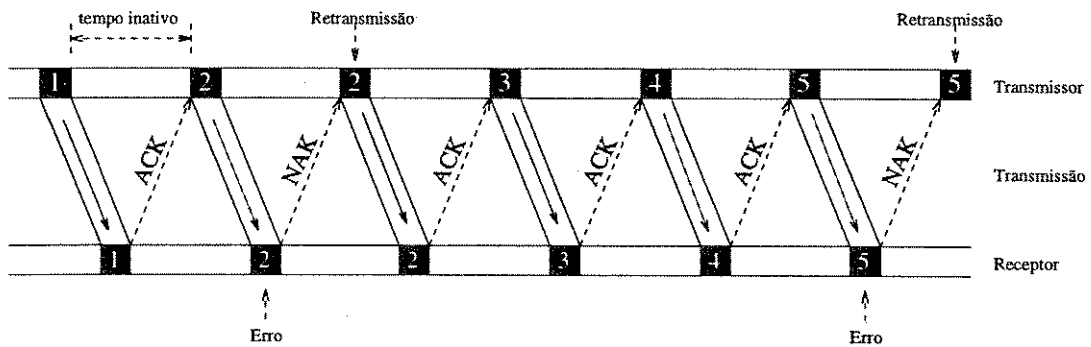


Figura 1.2: ARQ Stop-and-Wait.

Em sistemas de comunicação onde temos um tempo de atraso de percurso muito longo, o uso do esquema ARQ stop-and-wait se torna impraticável, pois o throughput é muito pequeno.



1000 bits. Em um tempo de atraso de percurso,  $N = 700$  palavras<sup>3</sup> são transmitidas. Portanto, quando uma palavra é recebida com erro, um total de 700 palavras-código são descartadas.

### 1.3.3 Esquema Selective-Repeat

Neste esquema, as palavras-código também são transmitidas continuamente. Entretanto, somente são retransmitidas as palavras cujas confirmações de recepção forem negativas, conforme mostra a figura 1.4.

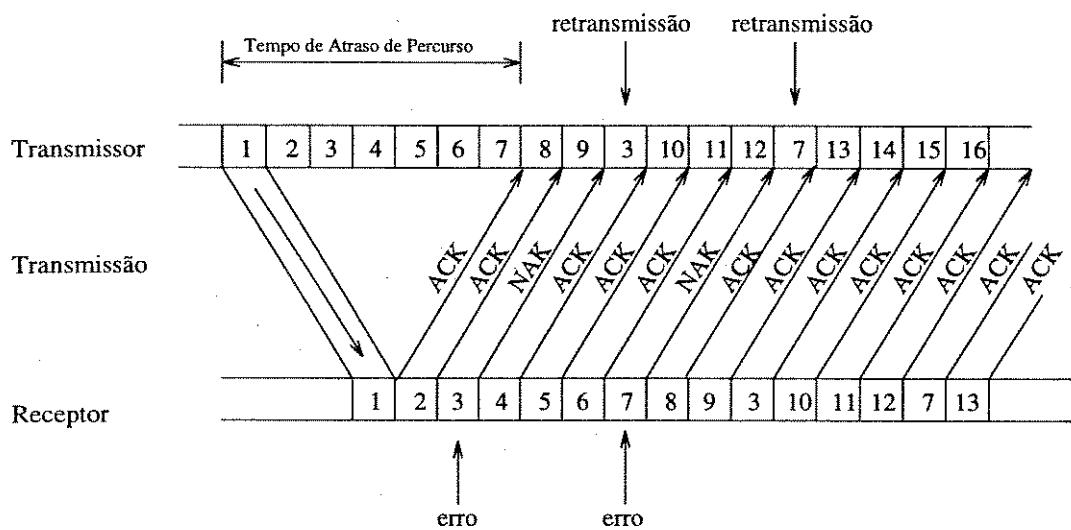


Figura 1.4: ARQ Selective-Repeat.

Como as palavras devem ser liberadas na ordem correta para o usuário, o receptor deve possuir um buffer (memória) suficientemente grande para armazenar as palavras livres de erros seguidas a uma palavra recebida com erro. Quando a palavra-código retransmitida for recebida com sucesso (sem erros), o receptor libera todas as palavras livres de erros armazenadas no buffer, até encontrar um espaço no buffer do receptor que foi reservado para uma outra palavra que foi recebida com erro, ou "esvaziar" o buffer se não houver nenhuma palavra subsequente no buffer que foi recebida com erro.

Como o enlace armazena  $N$  palavras, qualquer esquema ARQ selective-repeat deve possuir no mínimo uma memória (buffer) no receptor capaz de armazenar  $N$  palavras, caso contrário,

<sup>3</sup> $N = \frac{\text{tempo de atraso de percurso} \times \text{taxa}}{n}$

poderá ocorrer "overflow" (transbordamento) do buffer do receptor, e como consequência, palavras recebidas sem erros poderão ser descartadas por não ter espaço para armazená-las.

## 1.4 Desempenho dos Esquemas ARQ

A performance de sistemas de controle de erro ARQ, geralmente é medida por meio de dois parâmetros: **confiabilidade** e **throughput**.

O receptor comete um erro, quando ele aceita uma palavra-código cujo erro não foi detectado. Este evento é chamado de *evento erro*. Seja  $P(E)$  a probabilidade deste evento. Para um sistema ARQ ser confiável,  $P(E)$  deve ser muito pequena, de tal forma a minimizar a ocorrência de erro não detectável em uma palavra-código. Assim, deduz-se facilmente que os três esquemas ARQ básicos apresentam a mesma confiabilidade, considerando que eles estejam utilizando o mesmo código detector de erro.

Considere um código linear  $(n, k)$ , usado para detecção de erro. Então, pode-se definir as seguintes probabilidades:

- $P_c$  = probabilidade de que a palavra recebida não contenha erro;
- $P_d$  = probabilidade de que a palavra recebida contenha um padrão de erro detectável;
- $P_e$  = probabilidade de que a palavra recebida contenha um padrão de erro não detectável.

Tem-se  $P_c + P_d + P_e = 1$ .  $P_c$  depende das estatísticas de erro do canal e as probabilidades  $P_d$  e  $P_e$  dependem não somente das estatísticas do canal, mas do código detector de erro utilizado no sistema. Portanto, a probabilidade  $P(E)$  do receptor cometer um erro é dada por:

$$\begin{aligned}
 P(E) &= P_e + P_d P_e + P_d^2 P_e + \dots \\
 &= P_e (1 + P_d + P_d^2 + \dots) \\
 &= P_e \frac{1}{1 - P_d} \\
 &= \frac{P_e}{P_c + P_e}
 \end{aligned} \tag{1.3}$$

Por exemplo, para um canal BSC (canal binário simétrico) com probabilidade de transição  $p$ , conforme mostra a figura 1.5, tem-se :

$$P_c = (1 - p)^n \quad (1.4)$$

Se for utilizado um código linear  $(n, k)$  [Lin, 1983], então a probabilidade da palavra recebida conter um padrão de erro não detectável é limitada superiormente por:

$$P_e \leq 2^{-(n-k)}[1 - (1 - p)^n] \quad (1.5)$$

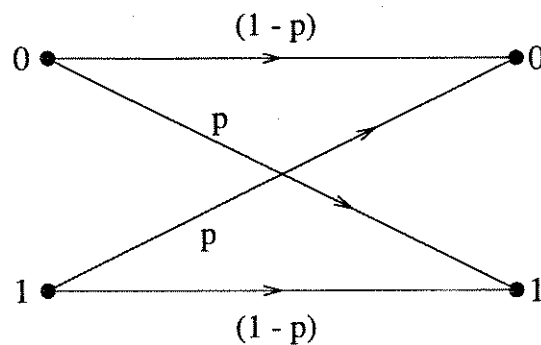


Figura 1.5: Canal Binário Simétrico.

A tabela 1.1 mostra o cálculo de  $P_c$ ,  $P_e$  e  $P(E)$  para alguns códigos BCH, considerando  $p = 10^{-3}$ .

O throughput de um sistema ARQ, é definido como a razão entre o número médio de bits de informação aceitos pelo receptor por unidade de tempo, pelo número de bits que poderiam ser transmitidos por unidade de tempo. Para o cálculo do throughput dos esquemas básicos, foi considerado que o canal de retorno é sem ruído. Portanto, este canal não introduz erro na informação transmitida no sentido inverso; do receptor para o transmissor.

Seja  $P$  a probabilidade de uma palavra recebida ser aceita pelo receptor, sendo dada por:

$$P = P_c + P_e \quad (1.6)$$

Considere primeiro, a análise do esquema stop-and-wait. Seja  $\lambda$  o tempo inativo entre duas transmissões sucessivas e  $\tau$ , a taxa de transmissão em bits/s. Embora não haja transmissão durante o tempo inativo, este deve ser considerado para o cálculo do throughput. Durante o



n	k	$P_e$	$P_c$	$P(E)$
127	120	$9.32 \times 10^{-04}$	$8.81 \times 10^{-1}$	$1.06 \times 10^{-03}$
	113	$7.28 \times 10^{-06}$	$8.81 \times 10^{-1}$	$8.27 \times 10^{-06}$
	106	$5.69 \times 10^{-08}$	$8.81 \times 10^{-1}$	$6.46 \times 10^{-08}$
	99	$4.45 \times 10^{-10}$	$8.81 \times 10^{-1}$	$5.05 \times 10^{-10}$
	92	$3.47 \times 10^{-12}$	$8.81 \times 10^{-1}$	$3.94 \times 10^{-12}$
	85	$2.71 \times 10^{-14}$	$8.81 \times 10^{-1}$	$3.08 \times 10^{-14}$
	78	$2.12 \times 10^{-16}$	$8.81 \times 10^{-1}$	$2.41 \times 10^{-16}$
	71	$1.66 \times 10^{-18}$	$8.81 \times 10^{-1}$	$1.88 \times 10^{-18}$
255	247	$8.80 \times 10^{-04}$	$7.75 \times 10^{-1}$	$1.13 \times 10^{-03}$
	239	$3.44 \times 10^{-06}$	$7.75 \times 10^{-1}$	$4.44 \times 10^{-06}$
	231	$1.34 \times 10^{-08}$	$7.75 \times 10^{-1}$	$1.73 \times 10^{-08}$
	223	$5.24 \times 10^{-11}$	$7.75 \times 10^{-1}$	$6.77 \times 10^{-11}$
	215	$2.05 \times 10^{-13}$	$7.75 \times 10^{-1}$	$2.64 \times 10^{-13}$
	207	$8.00 \times 10^{-16}$	$7.75 \times 10^{-1}$	$1.03 \times 10^{-15}$
	199	$3.13 \times 10^{-18}$	$7.75 \times 10^{-1}$	$4.03 \times 10^{-18}$
	191	$1.22 \times 10^{-20}$	$7.75 \times 10^{-1}$	$1.58 \times 10^{-20}$
511	502	$7.82 \times 10^{-04}$	$6.00 \times 10^{-1}$	$1.30 \times 10^{-03}$
	493	$1.53 \times 10^{-06}$	$6.00 \times 10^{-1}$	$2.55 \times 10^{-06}$
	484	$2.98 \times 10^{-09}$	$6.00 \times 10^{-1}$	$4.97 \times 10^{-09}$
	475	$5.83 \times 10^{-12}$	$6.00 \times 10^{-1}$	$9.71 \times 10^{-12}$
	466	$1.14 \times 10^{-14}$	$6.00 \times 10^{-1}$	$1.90 \times 10^{-14}$
	457	$2.22 \times 10^{-17}$	$6.00 \times 10^{-1}$	$3.71 \times 10^{-17}$
	448	$4.34 \times 10^{-20}$	$6.00 \times 10^{-1}$	$7.24 \times 10^{-20}$
	439	$8.48 \times 10^{-23}$	$6.00 \times 10^{-1}$	$1.41 \times 10^{-22}$
1023	430	$1.66 \times 10^{-25}$	$6.00 \times 10^{-1}$	$2.76 \times 10^{-25}$
	1013	$6.26 \times 10^{-04}$	$3.59 \times 10^{-1}$	$1.74 \times 10^{-03}$
	1003	$6.11 \times 10^{-07}$	$3.59 \times 10^{-1}$	$1.70 \times 10^{-06}$
	993	$5.97 \times 10^{-10}$	$3.59 \times 10^{-1}$	$1.66 \times 10^{-09}$
	983	$5.83 \times 10^{-13}$	$3.59 \times 10^{-1}$	$1.62 \times 10^{-12}$
	973	$5.69 \times 10^{-16}$	$3.59 \times 10^{-1}$	$1.58 \times 10^{-15}$
	963	$5.56 \times 10^{-19}$	$3.59 \times 10^{-1}$	$1.55 \times 10^{-18}$
	953	$5.43 \times 10^{-22}$	$3.59 \times 10^{-1}$	$1.51 \times 10^{-21}$
943	$5.30 \times 10^{-25}$	$3.59 \times 10^{-1}$	$1.48 \times 10^{-24}$	
933	$5.18 \times 10^{-28}$	$3.59 \times 10^{-1}$	$1.44 \times 10^{-27}$	

Tabela 1.1: Cálculo de  $P_c$ ,  $P_e$  e  $P(E)$  para alguns códigos BCH.

tempo de atraso de percurso, o transmissor poderia enviar  $n + \lambda\tau$  bits se não permanecesse inativo. Para uma palavra-código ser recebida corretamente, o número médio de bits que o transmissor poderia enviar é:

$$\begin{aligned}
 T_{sw} &= (n + \lambda\tau)P + 2(n + \lambda\tau)P(1 - P) + 3(n + \lambda\tau)P(1 - P)^2 + \dots \\
 &= (n + \lambda\tau)P[1 + 2(1 - P) + 3(1 - P)^2 + \dots] \\
 &= (n + \lambda\tau)P \cdot \frac{1}{P^2} \\
 &= \frac{n + \lambda\tau}{P}
 \end{aligned} \tag{1.7}$$

Portanto o throughput do sistema ARQ stop-and-wait é:

$$\begin{aligned}\eta_{sw} &= \frac{k}{T_{sw}} \\ &= \frac{P}{1 + \lambda\tau/n} \left(\frac{k}{n}\right)\end{aligned}\quad (1.8)$$

onde  $k/n$  é a taxa do código. Em sistemas de comunicação onde o tempo de atraso de percurso é grande,  $\lambda\tau/n$  torna-se grande. Portanto, para reduzir o valor de  $\lambda\tau/n$ , deve-se utilizar um código muito longo. Mas, por problemas de implementação e custo, utilizar um código muito longo, pode ser impraticável. Por exemplo, considere um sistema com uma taxa de 1Mb/s e um tempo de atraso de percurso de 700 ms, então tem-se  $\lambda\tau = 0.7 \times 10^6$ . Para fazer  $\lambda\tau/n$  pequeno,  $n$  deve ser na faixa de 1 milhão de bits. É impraticável usar um código tão longo. Por outro lado, se fizer  $n = 10.000$  bits,  $\lambda\tau/n = 70$ . Neste caso, o throughput se torna muito pequeno, o que é também inaceitável!

Em um sistema go-back- $N$ , o recebimento pelo transmissor de uma confirmação negativa referente a uma palavra transmitida, implica sempre na retransmissão de  $N$  palavras-código consecutivas. Para uma palavra-código ser recebida com sucesso, o número médio de transmissões é dado por:

$$\begin{aligned}T_{GBN} &= 1.P + (N+1)P(1-P) + (2N+1)P(1-P)^2 + \dots + (lN+1)P(1-P)^l + \dots \\ &= 1 + \frac{N(1-P)}{P}\end{aligned}\quad (1.9)$$

O throughput do go-back- $N$  é então dado por:

$$\begin{aligned}\eta_{GBN} &= \frac{1}{T_{GBN}} \left(\frac{k}{n}\right) \\ &= \frac{P}{P + (1-P)N} \left(\frac{k}{n}\right)\end{aligned}\quad (1.10)$$

O throughput do go-back- $N$ , depende da taxa de erro do canal e de  $N$  (número de blocos que podem ser transmitidos durante o tempo de atraso de percurso). Considere um sistema com taxa de 1Mb/s, atraso de percurso de 700 ms e  $n = 10.000$  bits. Portanto,  $N$  é igual a 70. Seja  $p = 10^{-5}$ , então,  $P_c = 0,9048$  e  $(1 - P_c)N \approx 6,66$ , que resulta em um throughput em torno de 11% da taxa do código.

Para o cálculo do throughput do sistema ARQ selective-repeat, assume-se que o buffer do receptor tem capacidade de armazenamento infinito (selective-repeat ideal). Portanto, o número médio de transmissões é:

$$\begin{aligned} T_{SR} &= 1.P + 2P(1 - P) + 3P(1 - P)^2 + \dots + lP(1 - P)^{l-1} + \dots \\ &= \frac{1}{P} \end{aligned} \quad (1.11)$$

sendo o throughput dado por:

$$\begin{aligned} \eta_{SR} &= \frac{1}{T_{SR}} \left( \frac{k}{n} \right) \\ &= P \left( \frac{k}{n} \right) \end{aligned} \quad (1.12)$$

Na prática, é impossível ter um buffer com capacidade de armazenamento infinito. Portanto, em sistemas ARQ selective-repeat práticos, pode ocorrer overflow do buffer do receptor, reduzindo o throughput deste sistema.

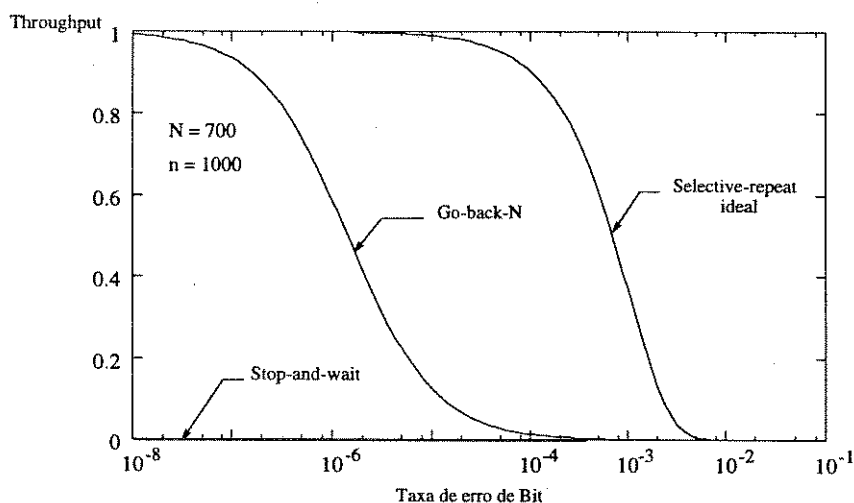


Figura 1.6: Throughput do ARQ selective-repeat com buffer do receptor infinito, go-back- $N$  e stop-and-wait com  $n = 1000$  bits,  $\tau = 1$ Mbit/s, e tempo de atraso de percurso igual a 700 ms.

As figuras 1.6 e 1.7 mostram uma comparação dos esquemas ARQ selective-repeat ideal, go-back- $N$  e stop-and-wait, para um sistema de comunicação. Observa-se, pelos gráficos, que o esquema stop-and-wait é impraticável para sistemas de comunicação com altas taxas de

dados e longo tempo de atraso de percurso, e que o esquema go-back- $N$  apresenta um melhor desempenho quando o tempo de atraso de percurso é pequeno, conforme mostra a figura 1.7.

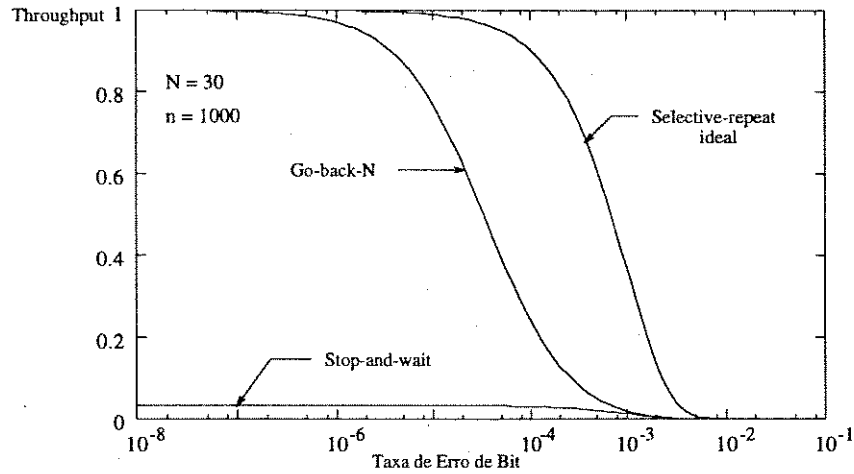


Figura 1.7: Throughput do ARQ selective-repeat com buffer do receptor infinito, go-back- $N$  e stop-and-wait com  $n = 1000$  bits,  $\tau = 1\text{Mbit/s}$ , e tempo de atraso de percurso igual 30 ms.

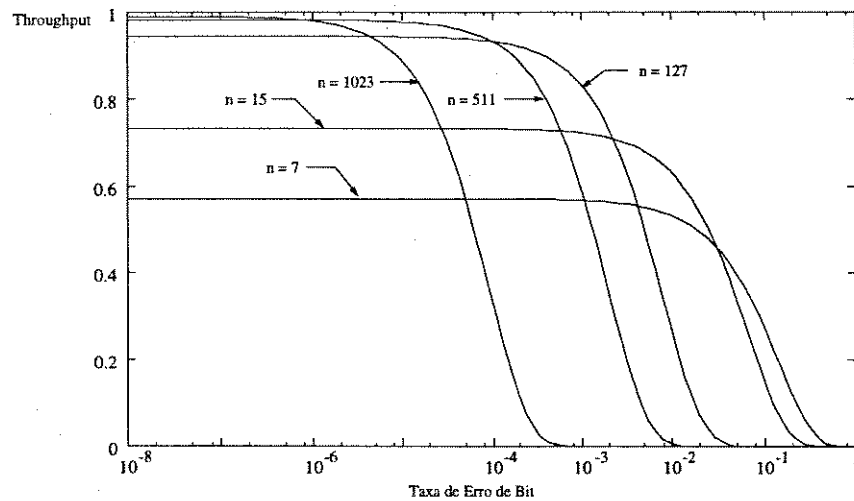


Figura 1.8: Throughput do ARQ selective-repeat para vários comprimentos de códigos.

A figura 1.8 mostra o throughput do ARQ selective-repeat para vários comprimentos de códigos. Analisando esta figura, observa-se que se a taxa de erro de bit do canal estiver muito baixa, o throughput é maior para sistemas que utilizam códigos com  $n$  grande. Mas se o canal estiver muito ruidoso, os códigos com  $n$  pequeno, apresentam um melhor desempenho, em

termos de throughput.

## 1.5 Códigos de Blocos Lineares para Detecção de Erro

A confiabilidade de um sistema ARQ, depende do código utilizado no processo de detecção de erro. Se o código for corretamente escolhido, a probabilidade do receptor cometer um erro de decodificação é muito pequena.

Considere a detecção de erro utilizando código de bloco linear em um canal de erro aleatório com taxa de erro de bit igual a  $p$ . [Korzihik, 1965] provou que existem códigos lineares com probabilidade de não detectar um erro  $P_e$ , que satisfazem o limitante:

$$P_e \leq 2^{-(n-k)}[1 - (1 - p)^n] \quad (1.13)$$

para todo  $n$ ,  $k$  e todo  $p$ ,  $0 \leq p \leq 1/2$ . A prova deste resultado é feita utilizando a média de  $P_e$ , sobre todos os códigos de blocos lineares  $(n, k)$  sistemáticos. É uma prova de existência e nenhum método foi encontrado para construção de códigos que satisfaçam o limitante dado pela equação 1.13. Entretanto, algumas pequenas classes de códigos conhecidas satisfazem o limitante abaixo:

$$P_e \leq 2^{-(n-k)} \quad (1.14)$$

para  $0 \leq p \leq 1/2$ . Estes são códigos perfeitos, tais como: os códigos de Hamming, o código de Golay (23,12), seus códigos duais, e os códigos BCH com distância igual a 5. Se um código que satisfaça o limitante dado pela equação 1.14 for usado para detecção de erro em um sistema ARQ, a  $P(E)$  pode ser muito pequena, usando um número moderado de bits de paridade ( $\sim 32$  bits).

Kasami [1982] considerou o conjunto de todos os códigos lineares  $(n, k)$  de peso par e provou que existem códigos com  $P_e$  satisfazendo o seguinte limitante:

$$P_e \leq 2^{-(n-k)}[1 + (1 - 2p)^n - 2(1 - p)^n] \quad (1.15)$$

para  $0 \leq p \leq 1/2$ . As seguintes classes de códigos satisfazem o limitante dado pela equação 1.15:

- Código de Golay estendido (24,12) com distância 8;
- Códigos de Hamming estendido com distância 4, de tamanhos  $2^m - 1$  e  $2^m$ , respectivamente;
- Códigos BCH primitivos com distância 6, de tamanho  $2^m - 1$  com  $m \geq 5$  ( $X + 1$  é um fator do polinômio gerador);
- Códigos BCH estendidos com distância 6, de tamanho  $2^m$ ;
- Códigos BCH primitivos com distância 8, de tamanho  $2^m - 1$  com  $m$  ímpar e  $m \geq 5$ ;
- Códigos BCH estendidos com distância 8, de tamanho  $2^m$  com  $m$  ímpar e  $m \geq 5$ .

## Capítulo 2

# ESQUEMAS ARQ

# SELECTIVE-REPEAT

### 2.1 Introdução

O alto desempenho em termos de throughput de um sistema ARQ selective-repeat ideal, é conseguido devido à capacidade de armazenamento do buffer do receptor (teoricamente infinito) e uma lógica mais complexa tanto no transmissor como no receptor. Se for usado um buffer finito, como é o caso de sistemas ARQ práticos, poderá ocorrer overflow no buffer do receptor, reduzindo o throughput do sistema. Entretanto, se for usado um buffer com grande capacidade de armazenamento e o overflow do buffer do receptor for apropriadamente tratado, mesmo reduzindo o throughput, o esquema ARQ selective-repeat apresenta um desempenho significativamente superior aos esquemas go-back- $N$  e stop-and-wait, particularmente em sistemas onde a taxa de transmissão de dados é alta e o tempo de atraso de percurso é grande.

Há dois métodos para tratar o problema de overflow em sistemas ARQ selective-repeat com buffer do receptor finito. O primeiro método, denominado de estratégia mista, consiste em projetar uma estratégia de retransmissão, de tal modo que o overflow do buffer do receptor possa ser evitado. O segundo, consiste em projetar um mecanismo através do qual, o transmissor possa detectar a ocorrência de overflow no buffer do receptor, e retransmitir as palavras-códigos

que forem perdidas. O primeiro método é de menor complexidade enquanto que o segundo, por ser mais complexo, apresenta um melhor desempenho em termos de throughput.

## 2.2 Esquemas ARQ com Modos Mistos de Retransmissão

Os esquemas ARQ selective-repeat capazes de evitar overflow no buffer [Miller, 1981], empregam estratégias de retransmissão mistas, ou seja, o transmissor não usa somente o modo selective-repeat para efetuar uma retransmissão. Há dois esquemas que empregam o modo misto: esquemas selective-repeat mais go-back- $N$  (SR+GBN) e selective-repeat mais stutter (SR+ST).

### 2.2.1 Esquemas Selective-Repeat mais Go-Back- $N$

Este esquema apresenta dois modos de retransmissão quando uma palavra-código for recebida com erro:

- Selective-repeat: o transmissor fica neste modo nas  $v$  primeiras retransmissões consecutivas.
- Go-back- $N$ : o transmissor é chaveado para este modo, quando  $v + 1$  retransmissões consecutivas de uma palavra-código não forem aceitas pelo receptor como uma palavra sem erro.

A figura 2.1 mostra o funcionamento do esquema SR+GBN considerando  $v = 1$ . Neste esquema, quando o transmissor receber o primeiro NAK referente a uma dada palavra-código, ele retransmite esta palavra e continua a transmitir novas palavras-código, no modo selective-repeat. Durante este modo, o receptor continua armazenando as palavras que foram recebidas corretamente subsequentes a uma palavra recebida com erro. Se o transmissor receber um outro NAK referente a esta mesma palavra anteriormente retransmitida, o transmissor é chaveado para o modo go-back- $N$  (como  $v$  igual a 1, o transmissor fica no modo selective-repeat somente



na primeira retransmissão); ele não transmite novas palavras e retransmite a palavra que não foi recebida corretamente e as  $N - 1$  palavras-códigos que foram transmitidas após a primeira retransmissão. O transmissor fica no modo go-back- $N$  até receber uma confirmação de recepção positiva referente a esta palavra-código. No receptor, quando for detectado erro na segunda tentativa de transmissão, as  $N - 1$  palavras recebidas subsequentes são descartadas, estando livres de erros ou não. No exemplo da figura 2.1, os blocos 11, 12, 13 e 14 são descartados pelo receptor, pois o bloco 7 foi recebido com erro na primeira retransmissão ( $v = 1$ ).

O SR+GBN fornece um throughput superior quando comparado com o esquema go-back- $N$ . Isto é conseguido devido aos benefícios obtidos usando o modo selective-repeat na primeira retransmissão. O uso do modo go-back- $N$  a partir da segunda retransmissão, evita a ocorrência de overflow no buffer do receptor, considerando o buffer com capacidade para armazenar  $N$  palavras-código. Pode-se obter um sistema com um melhor throughput se o transmissor for projetado para permitir mais de uma retransmissão no modo SR ( $v > 1$ ), para uma dada palavra-código que não foi recebida corretamente, antes de passar para o modo GBN. Se o sistema retransmite  $v > 1$  no modo SR, antes de chavear para o modo GBN, o buffer do receptor deve ser capaz de armazenar  $v(N - 1) + 1$  palavras-códigos para evitar overflow. O preço por utilizar  $v > 1$  é uma lógica mais complexa tanto no transmissor como no receptor, e um aumento na capacidade de armazenameto do buffer do receptor.

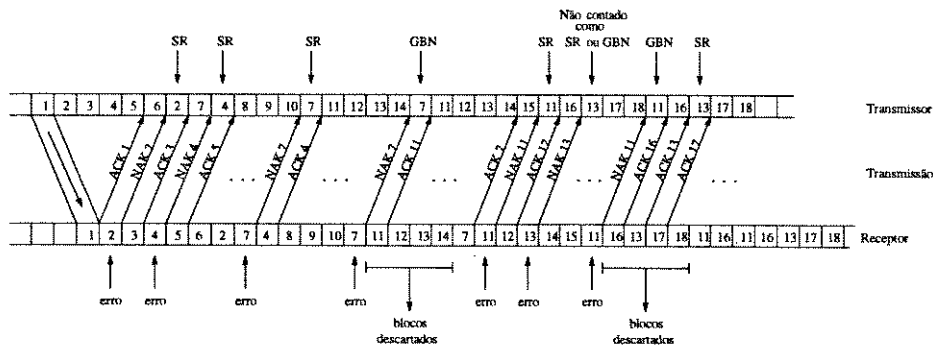


Figura 2.1: Esquema ARQ SR+GBN com  $v = 1$  e  $N = 5$ .

O throughput do esquema ARQ SR+GBN é dado por:

$$\eta_{SR+GBN} = \frac{P}{1 + (N - 1)(1 - P)^{v+1}} \left( \frac{k}{n} \right) \quad (2.1)$$

onde  $P$  é a probabilidade de que uma palavra seja recebida com sucesso. Se  $v = 0$ , o esquema SR+GBN torna-se o esquema ARQ go-back- $N$  básico:

$$\eta_{GBN} = \frac{P}{P + N(1 - P)} \binom{k}{n} \quad (2.2)$$

### 2.2.2 Esquema Selective-Repeat mais Stutter

Este esquema misto, é bem parecido com o esquema SR+GBN, mas ao invés de usar o modo go-back- $N$  após as  $v$  primeiras retransmissões sem sucesso de uma palavra, o transmissor utiliza um outro modo de transmissão denominado stutter (ST), no qual ele retransmite repetidamente a palavra-código recebida com erro, até receber um ACK referente a esta palavra. Existem dois esquemas diferentes que utilizam o modo stutter nas retransmissões.

#### 1. Esquema SR+ST 1

No esquema SR+ST 1, o modo selective-repeat é o mesmo do SR+GBN, ou seja, o transmissor pode retransmitir até  $v$  vezes uma palavra-código no modo SR. Considerando  $v = 1$ , após o recebimento do primeiro NAK referente a uma palavra  $i$ , o transmissor retransmite esta palavra e continua transmitindo novas palavras-códigos (modo selective-repeat). Se o transmissor receber novamente um NAK referente à primeira retransmissão da  $i$ -ésima palavra, o transmissor entra no modo ST onde ele retransmite repetidamente a palavra  $i$  até receber um ACK referente a esta palavra. O modo stutter evita a ocorrência de overflow. Se erros ocorrerem em quaisquer outras palavras  $i + 1, i + 2, \dots, i + N - 1$ , o transmissor inicia a retransmissão no modo SR das palavras que foram recebidas com erro. A figura 2.2 ilustra este procedimento. Neste caso, como  $v = 1$ , o tamanho máximo do buffer do receptor é igual a  $2(N - 1)$  palavras.

Se o transmissor receber um NAK referente a uma palavra  $j$ , e ele estiver no modo ST retransmitindo a palavra  $i$  ( $j > i$ ), o transmissor armazena a palavra  $j$ , até receber um ACK referente à palavra  $i$ , para então retransmitir a palavra  $j$  no modo SR. Portanto, a palavra  $j$  é retransmitida, seguida por novas palavras a partir do número de sequência que estava o transmissor quando entrou no modo ST.

Neste esquema, o transmissor tem que manter uma contagem do número de tentativas

de transmissões para cada palavra transmitida, no seu buffer de retransmissão. Quando o transmissor receber um NAK referente a qualquer palavra, o transmissor a retransmitirá no modo SR se ocorreu somente uma retransmissão, caso contrário, a retransmitirá no modo ST (lembrando novamente que  $v = 1$ ).

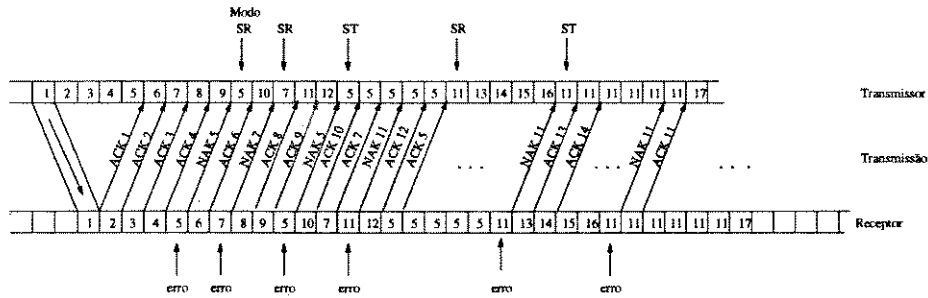


Figura 2.2: Esquema ARQ SR+ST 1 com  $v = 1$  e  $N = 5$ .

Este esquema pode ser estendido para permitir  $v > 1$  retransmissões no modo SR, antes de passar para o modo ST. Neste caso, o tamanho do buffer do receptor necessário para evitar overflow é de  $(N - 1)(v + 1)$  palavras-código.

O throughput do esquema SR+ST 1 é dado por:

$$\eta_{SR+ST(1)} = \frac{P}{1 + (N - 1)(1 - P)^{v+1}P} \left( \frac{k}{n} \right) \quad (2.3)$$

## 2. Esquema SR+ST 2

Este esquema, é similar ao esquema SR+ST 1 descrito anteriormente. O SR+ST 2 necessita somente de um contador no transmissor para verificar o número de transmissões da primeira palavra cuja confirmação de recepção foi negativa e que ainda esteja no buffer de retransmissão. A lógica tanto no transmissor, como no receptor é mais simples para este esquema, na qual todas as palavras que não foram recebidas corretamente, são tratadas no mesmo modo de retransmissão (SR ou ST), ao passo que, no esquema SR+ST 1, o modo de retransmissão depende da contagem de NAK de cada palavra.

No esquema SR+ST 2, numa operação livre de erros, o transmissor envia palavras continuamente. Mas quando o transmissor receber um NAK indicando que uma dada palavra  $i$  foi

recebida com erro, o modo de retransmissão é escolhido de acordo com o estado de um "flag" no transmissor; ativado (FS) ou não ativado (FNS).

- FS - considerando inicialmente o "flag" do transmissor no estado FNS, o "flag" passa para o estado FS, imediatamente após o recebimento de dois NAK's consecutivos referentes a uma dada palavra  $i$ , isto é, se duas transmissões consecutivas de qualquer palavra  $i$  forem detectadas com erro, o transmissor entra no estado FS.
- FNS - o "flag" do transmissor fica ativado (estado FS) até que o transmissor receba os ACK's correpondentes a todos os blocos que não foram recebidos corretamente e estão armazenados no buffer de retransmissão. Então, o "flag" do transmissor passa para o estado FNS.

No estado FS, uma palavra que não foi recebida corretamente no buffer de retransmissão, será retransmitida no modo ST, até que o transmissor receba um ACK referente a esta palavra. Este processo se repetirá para todas as palavras que não foram recebidas corretamente no buffer de retransmissão, até que sejam recebidas com sucesso.

No estado FNS, o transmissor envia novas palavras-código. Se receber um NAK, a palavra cuja confirmação de recebimento foi negativa é retransmitida, seguida por novas palavras-código, ou seja, quando o "flag" estiver no estado FNS, a retransmissão é no modo SR. O procedimento é ilustrado na figura 2.3. O tamanho máximo do buffer do receptor necessário para evitar overflow é de  $2(N - 1)$  palavras, considerando  $v = 1$ .

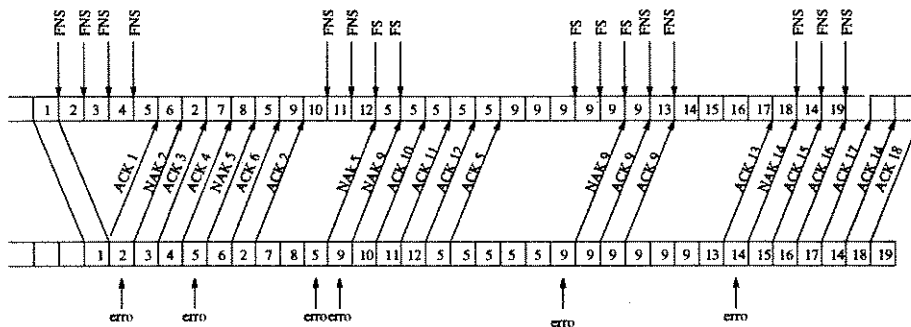


Figura 2.3: Esquema ARQ SR+ST 2 com  $v = 1$  e  $N = 5$ .

O throughput do esquema SR+ST 2 também é analisado por Miller [1981] e é dado por:

$$\eta_{SR+ST(2)} = \frac{P}{1 + (N - 1)(1 - P)^2 P + (N - 1)(1 - P)P_{NN}P^2} \binom{k}{n} \quad (2.4)$$

onde  $P_{NN}$  é a probabilidade de receber dois blocos NAK's consecutivos referentes a uma mesma palavra, ou seja, separadas por um intervalo de tempo igual a  $N$ .  $P_{NN}$  é dada por:

$$P_{NN} = \sum_{j=0}^{N-2} \sum_{k=2j}^{2(N-1)-2} \frac{(-1)^j (N - 1)!}{(N - j - 2)!(j + 1)!} \cdot \frac{(2(N - 1) - 2 - 2j)!}{(k - 2j)!(2(N - 1) - 2 - k)!} (1 - P)^{2+k} P^{2(N-1)-2-k} \quad (2.5)$$

Se o termo  $P_{NN}$  for igual a zero, a equação 2.5 reduz-se à equação do throughput do esquema SR+ST 1 com  $v = 1$ . O termo  $P_{NN}$  representa uma redução da performance do esquema SR+ST 2 sobre o esquema SR+ST 1.

As figuras 2.4 - 2.7 mostram o throughput dos esquemas ARQ mistos.

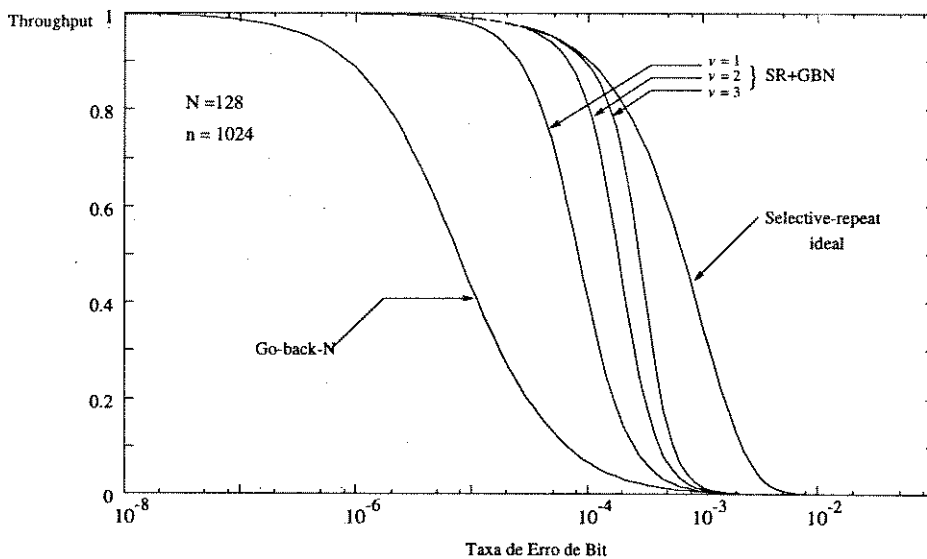


Figura 2.4: Throughput do esquema ARQ misto SR+GBN, para vários valores de  $v$ , considerando  $n = 1024$  bits e  $N = 128$  palavras.

Analisando as figuras 2.4 e 2.5 observa-se que à medida que o número de retransmissões no modo selective-repeat aumenta, há um incremento do throughput dos esquemas SR+GBN e SR+ST. Como as  $v$  primeiras retransmissões são no modo SR, estes esquemas mistos, tem um desempenho superior ao go-back- $N$ , mas inferior ao esquema ARQ selective-repeat ideal,

devido às limitações com relação ao tamanho do buffer do receptor.

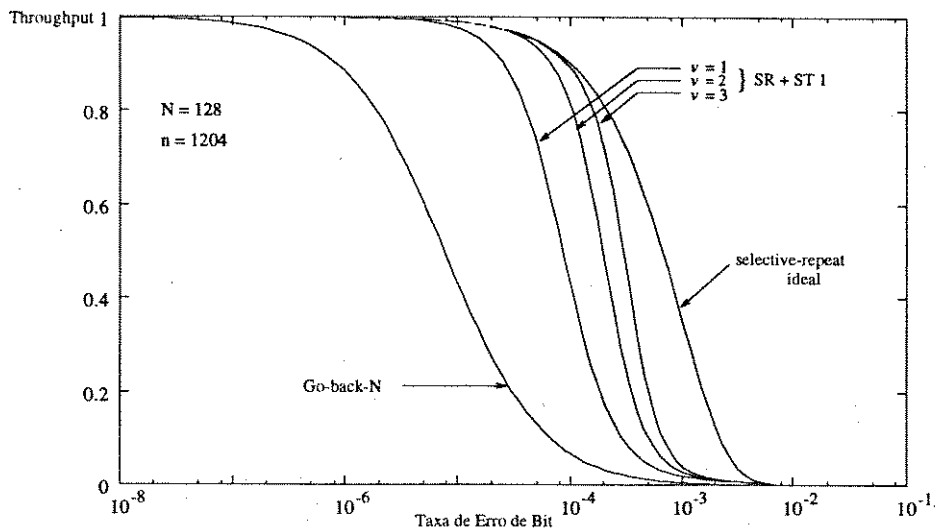


Figura 2.5: Throughput do esquema ARQ misto SR+ST 1, para vários valores de  $v$ , considerando  $n = 1024$  bits e  $N = 128$  palavras.

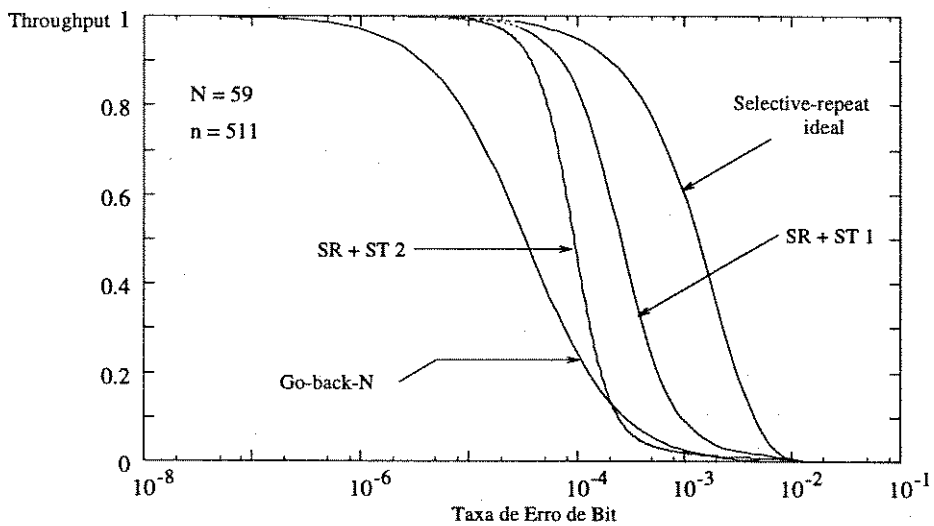


Figura 2.6: Comparação do throughput do esquema ARQ misto SR+ST 2, com SR+ST 1 ( $v = 1$ ), selective-repeat ideal e go-back-N, considerando  $n = 1024$  bits e  $N = 59$  palavras.

O throughput do esquema SR+ST 2 é mostrado na figura 2.6. Observa-se que o desempenho em termos de throughput do esquema SR+ST 2 é inferior ao SR+ST 1, devido ao termo  $P_{NN}$  na equação 2.4, mas ainda é superior ao esquema go-back-N.

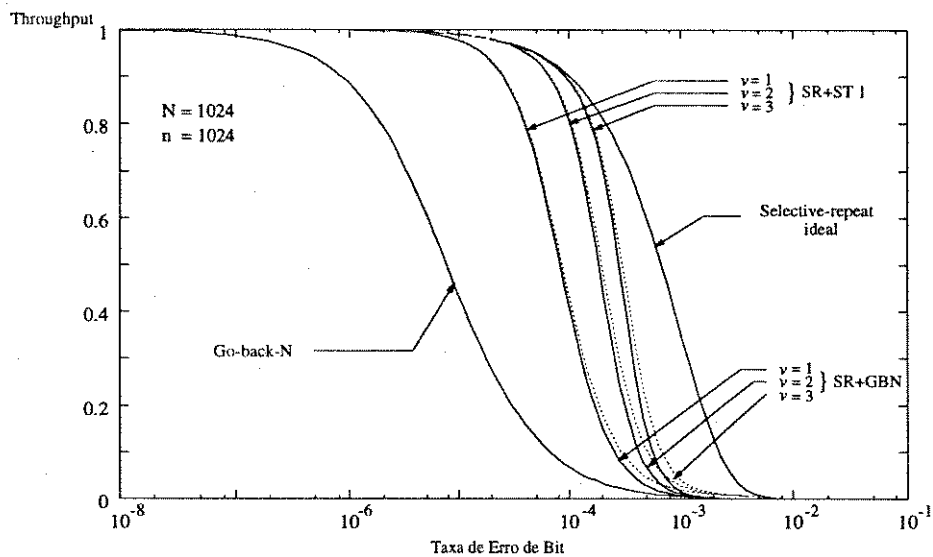


Figura 2.7: Comparação dos vários esquemas ARQ selective-repeat mistos, considerando  $n = 1024$  palavras e  $N = 1024$  palavras.

A figura 2.7 compara os esquemas ARQ mistos SR+ST 1 e SR+GBN. Observa-se que seus desempenhos em termos de throughput são praticamente idênticos para um mesmo  $v$ , onde o throughput do esquema SR+ST 1 é ligeiramente superior ao esquema SR+GBN.

### 2.2.3 Esquema de Weldon

O esquema de Weldon [Weldon, 1982] é uma variação dos esquemas ARQ mistos. Ao receber um NAK referente a uma palavra recebida com erro, o transmissor retransmite esta palavra  $r$  vezes consecutivas, onde  $r$  é o número de retransmissões seguidas de uma palavra recebida com erro e  $r$  aumenta à medida que o buffer do receptor aproxima-se do overflow. Por exemplo, se uma palavra  $i$  for recebida com erro, o transmissor envia  $n_1$  cópias desta palavra. Se todas estas  $n_1$  cópias forem novamente recebidas com erro, o transmissor envia agora  $n_2$  cópias desta palavra (onde  $n_2 > n_1$ ). A lógica de operação desta estratégia é:

- Nível 0: Se o transmissor receber um ACK, então envia uma nova palavra.
- Nível 1: Se o transmissor receber um NAK, a palavra correspondente a este NAK é retransmitida  $n_1$  vezes.

- Nível 2: Se todas as  $n_1$  cópias de uma palavra forem recebidas com erro (correspondendo a  $n_1$  NAK's), a palavra será retransmitida  $n_2$  vezes consecutivas.

⋮

- Nível  $q$ : Se todas  $n_{q-1}$  cópias de uma palavra forem recebidas com erro, ocorrerá overflow no buffer. A palavra será retransmitida  $n_q$  vezes consecutivas.
- Nível  $q + 1$ : Se todas  $n_q$  cópias de uma palavra forem recebidas com erro, continuará ocorrendo overflow no buffer. A palavra é repetida  $n_q$  vezes. A estratégia permanecerá neste nível até que o transmissor receba um ACK referente a esta palavra.

Em cada nível, o transmissor envia novas palavras toda vez que não estiver retransmitindo uma palavra que não foi corretamente recebida. Para  $q = 1$  e  $n_1 = 1$ , tem-se a estratégia selective-repeat mais go-back- $N$  (SR+GBN) com  $v = 1$ . Se múltiplos erros ocorrerem, o transmissor pode ser requisitado a enviar duas palavras ao mesmo tempo. Neste caso, a palavra com menor número de sequência, será transmitida primeiramente.

Após falhar as  $q + 1$  retransmissões de uma palavra e o nível  $q + 1$  for alcançado, o buffer do receptor é considerado cheio. As próximas  $N - 1$  palavras recebidas serão descartadas pelo receptor. O buffer do receptor pode não estar completamente cheio (como um resultado de espaços reservados no buffer para palavras com erros). Também, deve ser considerado que descartando palavras que foram recebidas corretamente, implica em uma redução do throughput. Ainda assim, o número de retransmissões necessárias para comunicar<sup>1</sup> a palavra  $i$  é independente na palavra  $j$  e suas repetições para todo  $j \neq i$ . Com isso, o número esperado de retransmissões é a mesma para todas as palavras.

O throughput do esquema de Weldon é dado por:

$$\eta_{WD} = 1 / \left[ \sum_{i=0}^q \left( \sum_{j=0}^i n_j \right) P_d^{\left( \sum_{j=0}^{i-1} n_j \right)} (1 - P_d^{n_i}) + \sum_{j=0}^q n_j P_d^{\sum_{j=0}^q n_j} + \frac{(n_q + N - 1) P_d^{\sum_{j=0}^q n_j}}{1 - P_d^{n_q}} \right] \quad (2.6)$$

<sup>1</sup>Uma palavra é dita ser comunicada, se for transmitida e recebida corretamente.



Na prática  $q$  é igual a unidade, o que reduz a equação 2.6 a:

$$\eta_{WD} = 1 / \left[ 1 + n_1 P_d + \frac{(n_1 + N - 1) P_d^{(1+n_1)}}{1 - P_d^{n_1}} \right] \quad (2.7)$$

onde  $P_d = 1 - P \approx 1 - P_c$ .

Para um dado canal definido por  $P_d$  e  $N$ , o número de repetições  $n_1$  deve ser escolhido de tal forma a maximizar o throughput. As regras para a escolha dos valores de  $n_1$  são dadas por [Chang, 1984]:

$$\begin{aligned} n_1 &= 1, & NP_d &\leq 1 \\ n_1 &= 2, & 1 \leq NP_d \text{ e } NP_d^2 &\leq 1 + P_d - P_d^2 \\ n_1 &= i, & (1 + P_d + \dots + P_d^{i-2}) - (i-2)P_d^{i-1} &\leq NP_d^{i-1} \text{ e} \\ & & NP_d^i &\leq (1 + P_d + \dots + P_d^{i-1}) - (i-1)P_d^i \end{aligned} \quad (2.8)$$

A quantidade  $NP_d$  é o número esperado de palavras com erro em um enlace de dados. Na maioria dos enlaces,  $NP_d$  é menor que um, e as palavras com confirmação de recebimento negativa devem ser repetidas somente um vez. Para enlaces de dados longos, com uma alta taxa de dados,  $NP_d$  pode exceder a unidade. Neste caso repete-se cada palavra que não foi recebida corretamente duas vezes. Somente para canais com  $N$  extremamente grande ou com uma probabilidade de erro de bit muito alta, é que as palavras devem ser repetidas mais de duas vezes.

As figuras 2.8 e 2.9 mostram o throughput do esquema de Weldon para vários valores de  $n_1$ , considerando  $q = 1$ . Analisando as curvas de throughput do esquema de Weldon, conclui-se que para maximizar o throughput, deve-se observar o valor do produto  $NP_d$  para então escolher o valor de  $n_1$ . Se a taxa de erro de bit do canal for alta, deve-se escolher um valor de  $n_1$  grande para obter um melhor desempenho do esquema de Weldon em termos de throughput.

Na estratégia de Weldon, há uma melhoria em termos de throughput a medida que o número de níveis  $q$  aumenta. Mantendo todos os  $n_i$  iguais a um (onde  $i = 1, 2, \dots, q$ ) a equação 2.6 reduz-se a:

$$\eta_{WD} = 1 / \left[ \sum_{i=0}^q (i+1) P_d^i (1 - P_d) + \frac{P_d^{q+1} (N + (q+1)(1 - P_d))}{1 - P_d} \right] \quad (2.9)$$

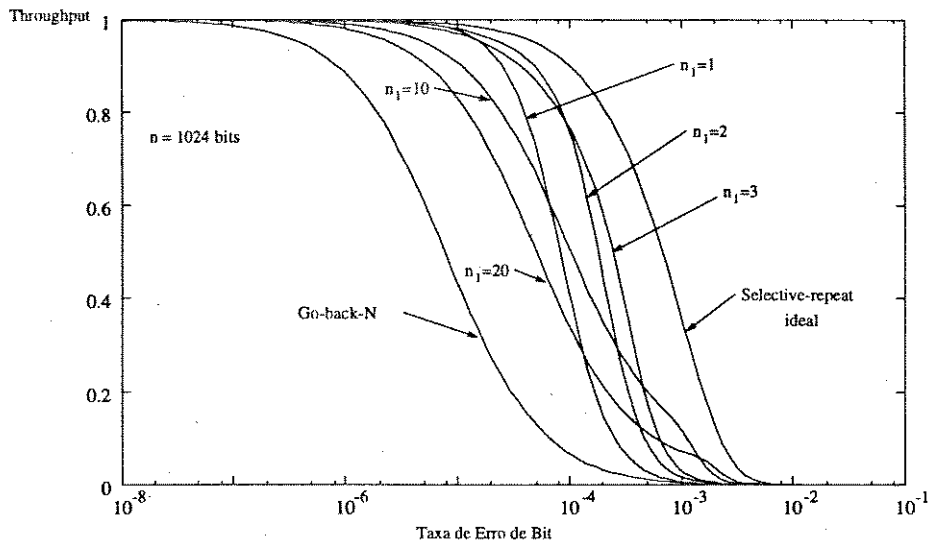


Figura 2.8: Comparação do throughput do esquema de Weldon com  $q = 1$ , para diversos valores de  $n_1$ , considerando o tempo de atraso de percurso  $N = 128$  palavras.

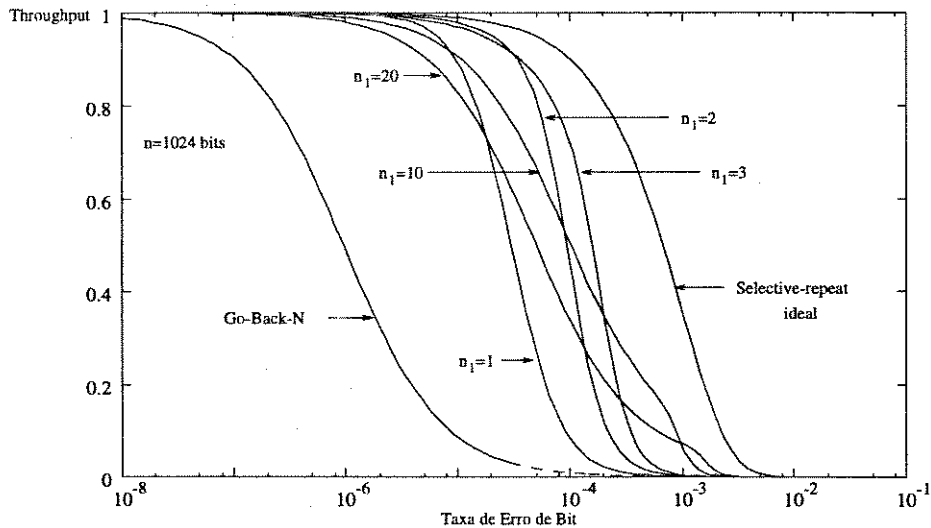


Figura 2.9: Comparação do throughput do esquema de Weldon para  $q = 1$ , para diversos valores de  $n_1$ , considerando  $N = 1024$  palavras.

A estratégia de Weldon é idêntica ao esquema ARQ misto SR + GBN, quando todos os  $n_i$  são iguais a 1 e  $q = v$  (lembrando que  $v$  é o número de retransmissões no modo selective-repeat). A figura 2.10 mostra o throughput do esquema de Weldon para vários valores de  $q$ . Analisando as curvas de throughput mostradas nesta figura, tem-se que para um valor de  $q$

grande, o throughput do esquema de Weldon se aproxima do selective-repeat ideal.

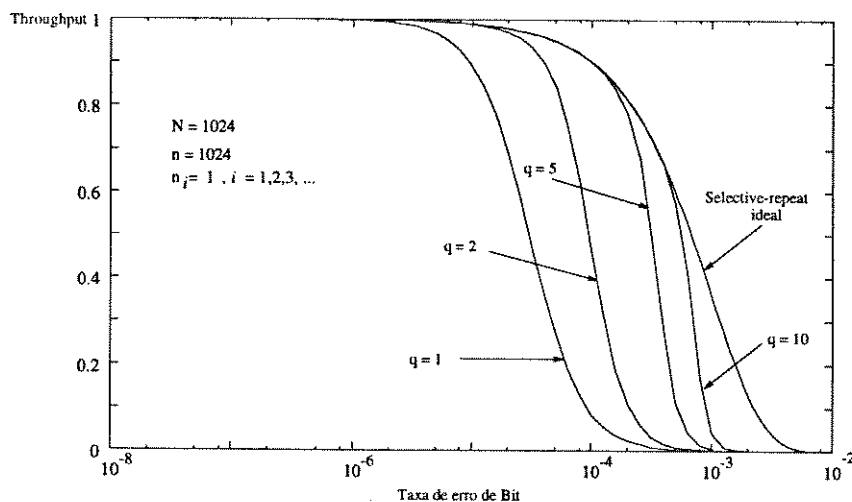


Figura 2.10: Comparação do throughput do esquema de Weldon para vários valores de  $q$ , considerando todos os  $n_i = 1$ , onde  $i = 1, 2, \dots, q$ .

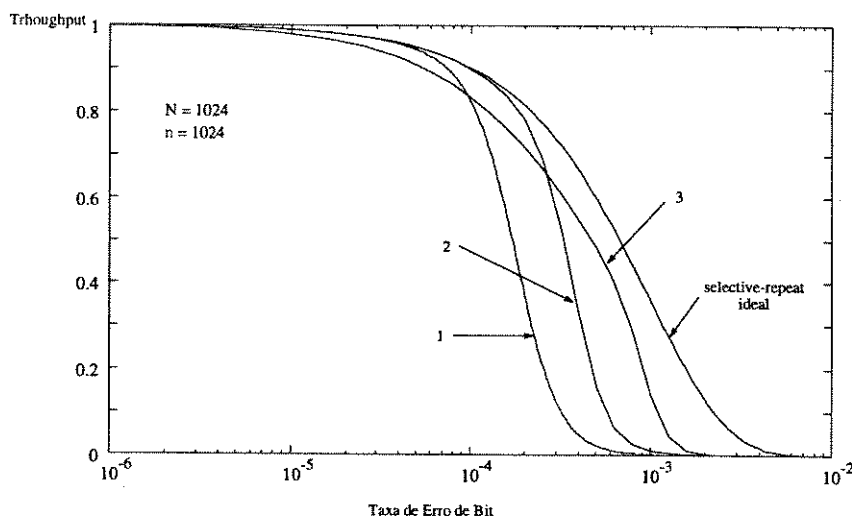


Figura 2.11: Comparação do throughput do esquema de Weldon para vários valores de  $n_i$ , considerando  $N = 1024$  palavras: (1) esquema de Weldon com  $q = 3$  e  $n_1 = n_2 = n_3 = 1$ ; (2) esquema de Weldon com  $q = 3$ ,  $n_1 = n_2 = 1$  e  $n_3 = 3$ ; (3) esquema de Weldon com  $q = 3$ ,  $n_1 = 2$ ,  $n_2 = 4$  e  $n_3 = 6$ .

A figura 2.11 mostra as curvas de throughput com valores de  $n_i$  variáveis. Observa-se nesta figura que se o valor de  $n_i$  aumenta à medida que é necessário uma nova retransmissão, há uma

melhoria em termos de throughput do esquema de Weldon, para altas taxas de erro de bit.

## 2.3 Esquema de Detecção de Overflow no Buffer do Receptor

Nesta seção, será descrito um esquema [Yu, 1981] selective-repeat com capacidade de armazenamento finito na recepção e capaz de detectar overflow no buffer do receptor.

Seja  $N$  o número de palavras-código que podem ser transmitidas durante o tempo de atraso de percurso. O receptor deve portanto, ser capaz de armazenar pelo menos  $N$  palavras. Geralmente, o receptor tem um buffer de tamanho  $iN$ ,  $i = 1, 2, 3, \dots$ . Isto implica que ocorrerá overflow no buffer se falhar a  $i$ -ésima retransmissão. Considera-se para efeito de análise de overflow do buffer do receptor, um buffer de tamanho  $N$ .

Em cada palavra-código é associado um número de sequência. As palavras são numeradas sequencialmente de 1 a  $3N$ . Estes números, são usados ciclicamente, isto é, a palavra  $3N + 1$  tem o número de sequência igual a 1. Para um buffer de tamanho  $N$ , a faixa de 1 a  $3N$  é suficiente para o receptor distinguir se uma palavra recebida com número de sequência  $ns$ , é uma nova palavra-código, ou uma palavra que já foi aceita e liberada para o usuário. Em cada palavra-código são adicionados  $\log_2 3N$  bits para indicar o número de sequência.

Suponha que um código linear  $(n, k)$  seja usado para detecção de erro. Quando uma palavra-código é recebida, sua síndrome é computada; se não houver erros, a informação contida na palavra é enviada para o usuário ou a palavra é armazenada no buffer, até que chegue sua vez de ser liberada para o usuário. Se houver erro, a palavra é descartada e um espaço no buffer do receptor é reservado para armazenar sua cópia a ser retransmitida.

Quando uma palavra-código estiver pronta para ser transmitida, ela é numerada e colocada na *fila de entrada*<sup>2</sup> do transmissor. Após a transmissão, uma cópia é armazenada no buffer de retransmissão até receber uma confirmação de recebimento positiva (ACK). A resposta chega após um tempo de atraso de percurso. Durante este intervalo,  $N - 1$  palavras-código são

---

<sup>2</sup>É uma fila no transmissor onde são colocadas as palavras, em ordem com relação ao número de sequência, para serem transmitidas.

transmitidas. Após o término deste tempo de atraso de percurso, o transmissor deve receber uma confirmação de recebimento da palavra-código transmitida. Se a resposta for um ACK, a palavra é liberada do buffer de retransmissão. Entretanto, se a resposta for um NAK, inicia-se a retransmissão desta palavra-código. Existe uma terceira possibilidade; a resposta confirmando o recebimento é perdida no canal de retorno, e não chega ao transmissor. A palavra no buffer de retransmissão, torna-se uma palavra *time-out*<sup>3</sup>. Neste caso, considera-se que a palavra-código não foi recebida com sucesso ou perdeu-se durante a transmissão. Esta palavra é então retransmitida.

**Definição:** Seja  $ns_0$  o número de sequência da primeira palavra que não foi recebida corretamente (NAK) ou que não tenha recebido nenhuma confirmação de recepção, armazenada no buffer de retransmissão. O índice "forward", denotado por  $f_T$ , de uma palavra-código com número de sequência  $ns$  no buffer de retransmissão ou *fila de entrada*, é definido como o resto da divisão de  $ns - ns_0$  por  $3N$ , matematicamente  $f_T$  é expresso como:

$$f_T \equiv (ns - ns_0) \text{ mod } 3N \quad (2.10)$$

### 2.3.1 Procedimentos de Transmissão e Retransmissão

Na operação de transmissão, também é computado o índice "forward"  $f_T$  da palavra-código no buffer de retransmissão que está para se tornar uma palavra *time-out* (neste caso, esta palavra é denominada palavra *time-out corrente*), para decidir se transmite a próxima palavra da fila de entrada ou inicia-se uma retransmissão. As regras de decisão são:

1. Se o transmissor receber uma confirmação de recepção positiva com relação à palavra *time-out corrente*, e seu índice  $f_T$  for menor que  $N$  (isso implica que não há palavra *time-out* no buffer de retransmissão), a primeira palavra da fila de entrada será transmitida. Se não houver palavras na fila de entrada, o transmissor fica inativo, até que cheguem novas palavras-código na fila.

---

<sup>3</sup>Quando uma palavra-código foi transmitida a um tempo de atraso de percurso e o tempo de chegada de uma resposta referente a esta palavra expira, sem que um ACK ou NAK seja recebido pelo transmissor.

2. Se o transmissor receber um NAK ou não receber resposta referente a palavra time-out corrente e o índice forward for menor que  $N$ , ele retransmite esta palavra. Se a palavra time-out corrente for a palavra mais antiga no buffer de retransmissão que recebeu um NAK ou uma palavra que não recebeu resposta (isto indica que o transmissor recebeu um segundo NAK referente a esta palavra time-out corrente, ou seja,  $f_T = 0$ ), todas as palavras no buffer de retransmissão com índice forward maior ou igual a  $N$  são colocadas novamente na fila de entrada para serem retransmitidas. Estas palavras foram transmitidas, mas quando chegaram no receptor, o buffer do receptor estava cheio (overflow) e não havia espaço para armazená-las.
3. Se o índice forward da palavra time-out corrente for maior ou igual a  $N$ , a primeira palavra-código na fila de entrada será transmitida (esta palavra pode ser uma palavra que foi movida do buffer de retransmissão para a fila de entrada devido ao overflow do buffer do receptor).

As operações de transmissão e retransmissão, são ilustradas nas figuras 2.12 e 2.13

### 2.3.2 Operação do Receptor e Procedimento para Recuperação de Erros

O transmissor envia palavras-código continuamente para o receptor. Na recepção, a síndrome de cada palavra é computada e para cada palavra recebida corretamente, o receptor envia um ACK. Quando não há interferência de ruídos no canal, a transmissão procede-se continuamente e dados são liberados para o usuário na ordem correta e o buffer do receptor fica vazio. O receptor está no *estado normal* quando o buffer estiver vazio e não houver espaço reservado para palavras recebidas com erros.

Quando uma palavra for recebida com erro ou com número de sequência fora de ordem, e o receptor estiver no estado normal, este envia um NAK para o transmissor e entra no *estado bloqueado*. No estado bloqueado, o receptor continua verificando a síndrome das palavras que chegam. Ele armazena as palavras cujas síndromes são iguais a zero no buffer do receptor

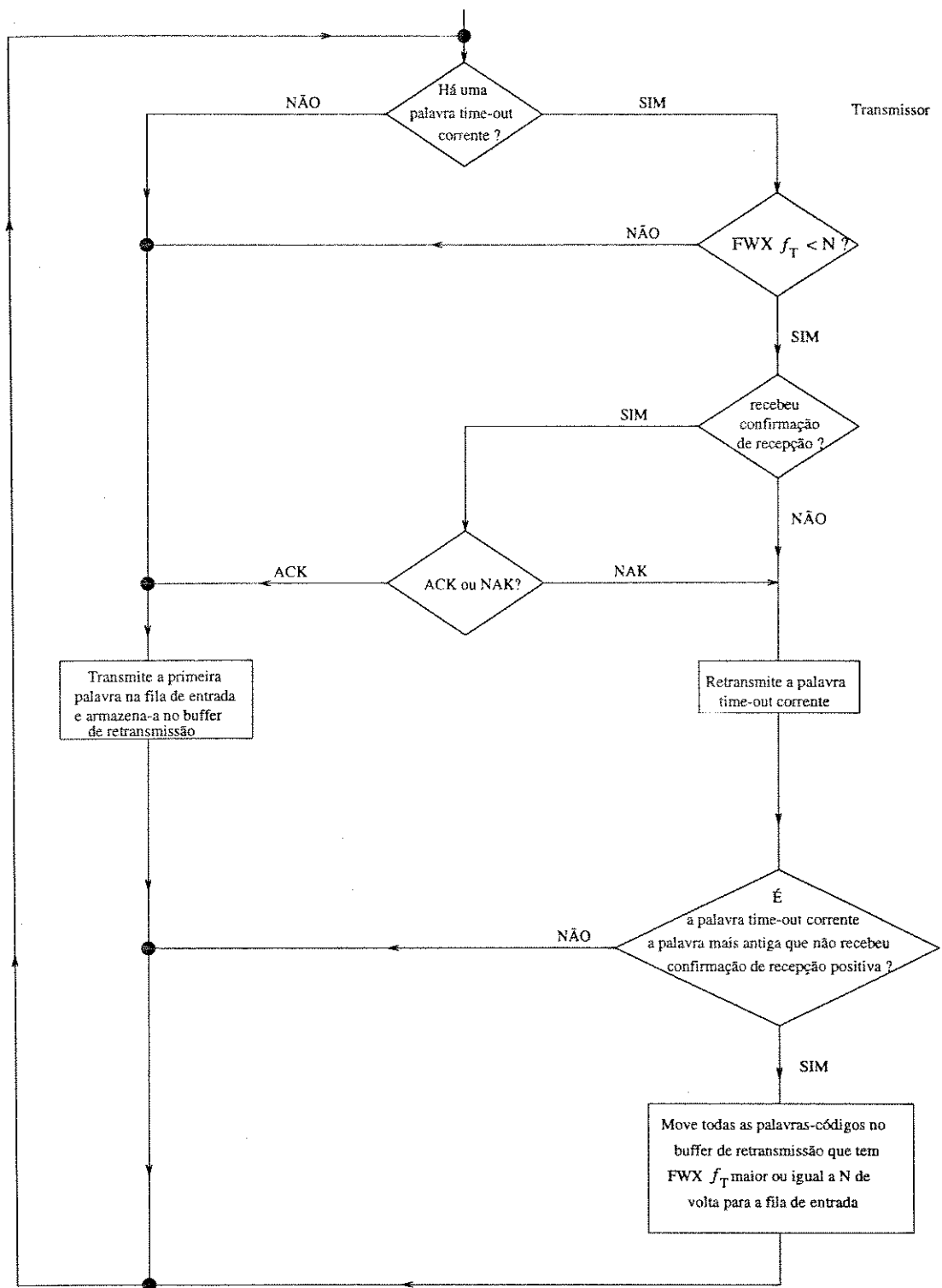


Figura 2.12: Procedimento de transmissão e retransmissão.

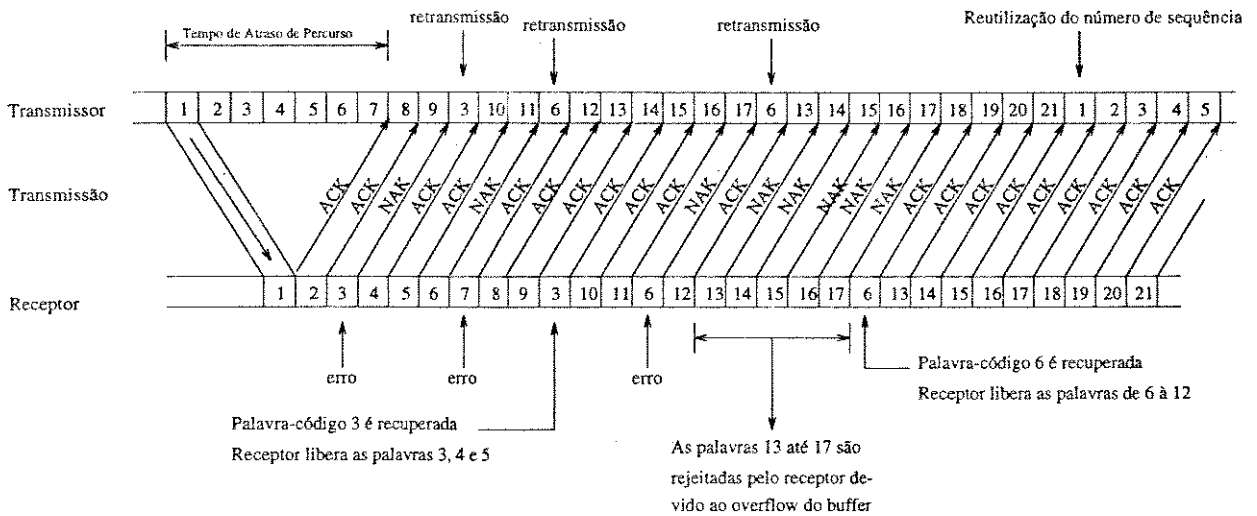


Figura 2.13: Esquema ARQ selective-repeat com buffer do receptor de tamanho  $N = 7$ .

sequencialmente, reservando espaços apropriados para as palavras cujas síndromes são diferentes de zero, até preencher todo o buffer. Nenhuma palavra é liberada para o usuário. Quando as palavras retransmitidas chegam após um tempo de atraso de percurso, suas síndromes são novamente computadas. As palavras que foram recebidas com sucesso (síndrome zero) são armazenadas no buffer do receptor nos locais previamente reservados. Quando a retransmissão de uma palavra que não havia sido recebida corretamente for recebida com sucesso, o receptor libera esta e todas as palavras subsequentes com síndrome zero até encontrar o próximo espaço reservado a outra palavra que não foi recebida corretamente. Se todas as palavras armazenadas no receptor forem liberadas e não houver mais espaço reservado no buffer, o receptor retorna ao estado normal. Caso contrário, o receptor continua no estado bloqueado.

### 2.3.3 Operação Estado Normal

Quando uma palavra-código com número de sequência  $ns$  for recebida, o receptor computa sua síndrome e seu índice forward denotado por  $f_R$ . O índice "forward"  $f_R$  da palavra recebida corrente é definido como o resto resultante da divisão de  $ns - ns_0$  por  $3N$ , isto é:

$$f_R \equiv (ns - ns_0) \text{ mod } 3N \quad (2.11)$$

onde  $ns_0$  é o número de sequência da última palavra aceita e liberada para o usuário.



Se não for detectado erros na palavra recebida corrente e  $f_R$  for igual a 1 ( $f_R = 1$  indica que a palavra recebida corrente tem o número de sequência correto), a palavra recebida será aceita e liberada para o usuário. O receptor procede na verificação de erros da próxima palavra que está para ser recebida.

Entretanto, se for detectado erro na palavra recebida corrente ou  $f_R$  for diferente de 1, mas menor que  $N + 1$  ( $1 < f_R \leq N$  indica que o número de sequência da palavra corrente está fora de ordem), o receptor entra no estado bloqueado.

Há três casos de combinações entre síndrome e o valor de  $f_R$  a serem considerados:

1. Síndrome diferente de zero e  $f_R$  igual a 1. O receptor descarta a palavra recebida com erro e reserva o primeiro espaço no buffer do receptor para esta palavra.
2. Síndrome diferente de zero e  $f_R$  maior que 1. Indica que  $f_R - 1$  palavras entre a última palavra liberada ao usuário e a palavra recebida corrente foram perdidas. O receptor descarta esta palavra recebida com erro e reserva os primeiros  $f_R$  espaços no buffer do receptor para as palavras perdidas e para esta palavra descartada.
3. Síndrome igual a zero e  $f_R$  maior que 1. Indica que  $f_R - 1$  palavras entre a última palavra aceita e liberada para o usuário e a palavra recebida corrente foram perdidas. O receptor reserva os primeiros  $f_R - 1$  espaços do buffer do receptor para estas palavras perdidas e armazena a palavra recebida corrente no  $f_R$ -ésimo espaço do buffer do receptor.

Se o índice forward  $f_R$  da palavra recebida corrente for maior que  $N$ , esta é considerada como uma palavra que foi previamente aceita e liberada para o usuário, sendo portanto ignorada e um ACK é enviado para o transmissor. O receptor permanece no estado normal e inicia o procedimento para processar a próxima palavra que está chegando. O fluxograma da figura 2.14 detalha a operação do receptor no estado normal, onde  $v_j$  é a  $j$ -ésima palavra que chega no receptor.

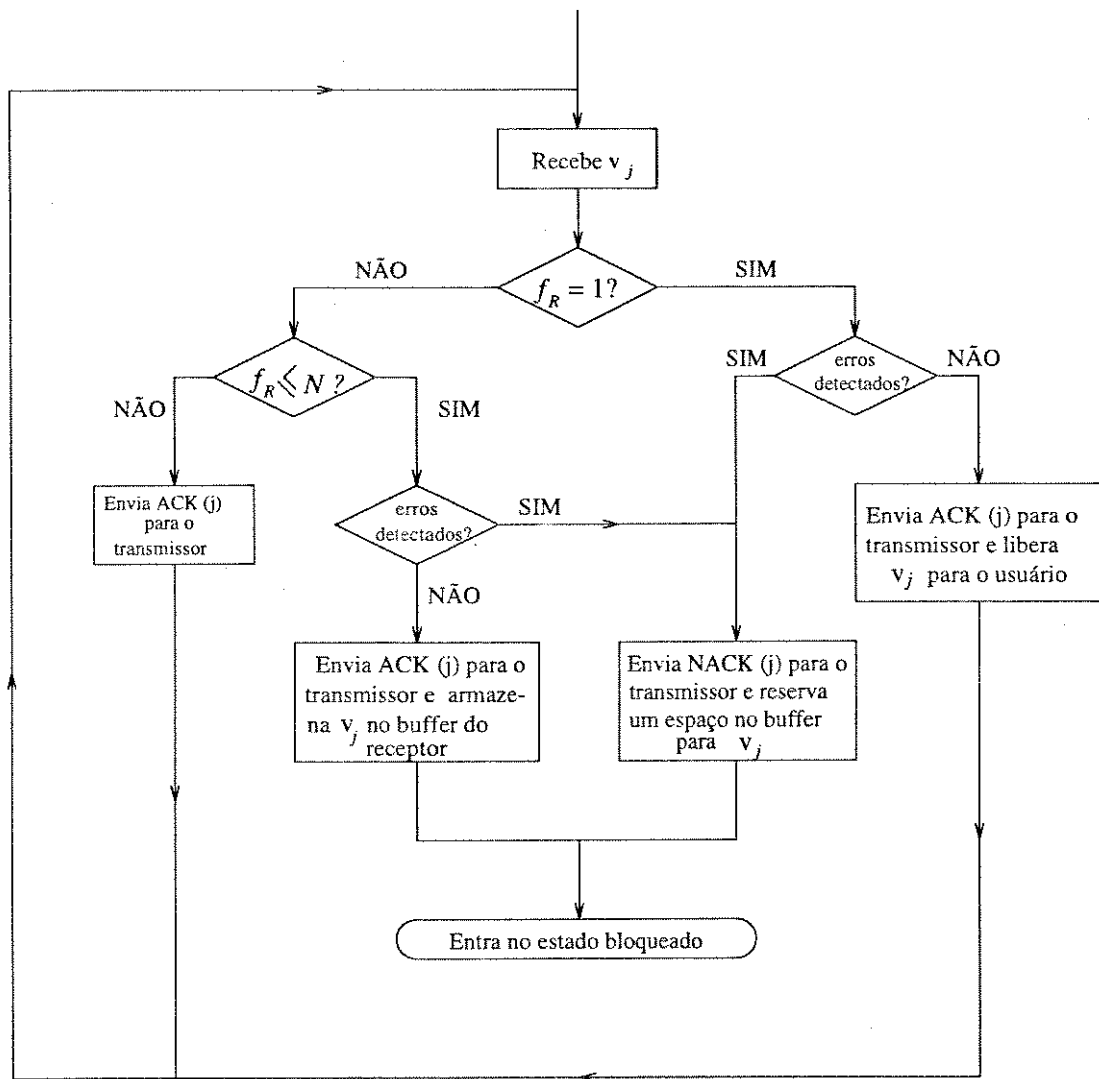


Figura 2.14: Operação no Estado Normal.

### 2.3.4 Operação Estado Bloqueado

O receptor entra no estado bloqueado quando for detectado erro na palavra recebida ou a palavra recebida estiver com número de sequência fora de ordem. Quando o receptor passa para o estado bloqueado, cada palavra recebida subsequente com síndrome igual a zero é armazenada no buffer do receptor em locais apropriados e um espaço no buffer é reservado para cada palavra que for recebida com erro ou perda. Quando o buffer estiver cheio, todas as novas palavras recebidas subsequentes, serão descartadas ocorrendo overflow no buffer do receptor.

Seja  $v_j$  a primeira palavra que não foi recebida com sucesso. Quando uma retransmissão

de  $\mathbf{v}_j$  for recebida com sucesso, o receptor libera  $\mathbf{v}_j$  e todas as palavras com síndromes iguais a zero subsequentes, que estão armazenadas no buffer do receptor (ver figura 2.13). Suponha que o receptor possa liberar  $L + 1$  palavras consecutivas, incluindo  $\mathbf{v}_j$  ( $\mathbf{v}_j, \mathbf{v}_{j+1}, \dots, \mathbf{v}_{j+L}$ , onde  $0 \leq L < N$ ). Como a retransmissão é seletiva e desde que  $\mathbf{v}_{j+1}$  até  $\mathbf{v}_{j+L}$  foram palavras recebidas com sucesso antes de  $\mathbf{v}_j$ , as palavras transmitidas após a retransmissão de  $\mathbf{v}_j$  são novas palavras  $\mathbf{v}_{j+N}, \mathbf{v}_{j+N+1}, \dots, \mathbf{v}_{j+N+L-1}$ . Quando estas  $L$  novas palavras chegam no receptor, suas síndromes são computadas. As palavras que tem síndromes iguais a zero são armazenadas temporariamente no buffer do receptor, até que estejam prontas para serem liberadas, e locais apropriados são reservados no buffer do receptor para as palavras cujas síndromes são diferentes de zero.

Se a palavra retransmitida  $\mathbf{v}_j$  não for recebida com sucesso, as palavras com síndrome zero armazenadas no buffer, não poderão ser liberadas para o usuário. Entretanto, não há espaço no buffer do receptor para armazenar as novas palavras  $\mathbf{v}_{j+N}, \mathbf{v}_{j+N+1}, \dots, \mathbf{v}_{j+N+L-1}$  (sendo  $N$  o tamanho do buffer). Neste caso, ocorrerá overflow no buffer e as novas palavras,  $\mathbf{v}_{j+N}, \mathbf{v}_{j+N+1}, \dots, \mathbf{v}_{j+N+L-1}$ , serão rejeitadas, estando livres de erros ou não.

Seja  $ns_0$  o número de sequência da primeira palavra que não foi recebida com sucesso. No estado bloqueado, quando uma palavra com número de sequência  $ns$  for recebida, seu índice "forward" com relação à palavra mais antiga que não foi recebida com sucesso, denotado por  $l_f$  será calculado. Este índice  $l_f$  é definido como o resto da divisão de  $ns - ns_0$  por  $3N$ , isto é:

$$l_f \equiv (ns - ns_0) \text{ mod } 3N \quad (2.12)$$

Se a síndrome da palavra recebida corrente for igual a zero e  $l_f < N$ , esta palavra é armazenada no buffer do receptor e envia um ACK para o transmissor. Se for detectado erro na palavra recebida corrente (síndrome diferente de zero) e  $l_f < N$ , a palavra será descartada e um espaço no buffer é reservado para esta palavra e é enviado um NAK para o transmissor. Se  $l_f \geq N$ , o receptor computa o índice "backward"  $l_b$  da palavra recebida, com relação à última palavra recebida que foi armazenada no buffer do receptor, ou para a qual foi reservada um espaço no buffer do receptor. O índice backward  $l_b$  é definido como o resto da divisão de  $z_0 - ns$

por  $3N$ , isto é:

$$l_b \equiv (z_0 - ns) \pmod{3N} \quad (2.13)$$

onde  $z_0$  é o número de sequência da última palavra recebida ou para a qual foi reservada um espaço no buffer do receptor e  $ns$  é o número de sequência da palavra recebida corrente. Se  $l_b < 2N$ , a palavra recebida corrente é considerada como uma palavra que foi previamente aceita e liberada para o usuário. Esta palavra é ignorada e um ACK é enviado para o transmissor. Se  $l_b \geq 2N$ , a palavra recebida corrente é uma nova palavra. Entretanto, como o buffer do receptor está cheio, ocorrerá overflow no buffer. Neste caso, a palavra corrente é rejeitada e um NAK é enviado para o transmissor. A operação do estado bloqueado é detalhada no fluxograma da figura 2.15.

### 2.3.5 Throughput do Sistema ARQ Selective-Repeat com Buffer Finito

Assumindo que os erros do canal são aleatoriamente distribuídos, isto é, um canal binário simétrico (BSC) e não há interferência de ruídos no canal de retorno, pode-se estabelecer um limitante inferior do sistema ARQ com buffer finito. Suponha que um código  $(n, k)$  apropriado seja usado para detecção de erro. Seja  $p$  a probabilidade de transição do BSC. Então, a probabilidade  $P$  do vetor ser aceito pelo receptor é:

$$P \approx P_c = (1 - p)^n \quad (2.14)$$

Seja

$\phi_0 = 1 - (1 - P)^2$ ,  $\phi_1 = 1 - (1 - P)^3$ ,  $\phi_2 = 1 - (1 - P)^4$  as probabilidades de sucesso da segunda, terceira e quarta retransmissões de uma palavra-código respectivamente. Definindo:

$$\lambda_0 = \frac{\phi_1}{1 - \phi_2} (1 - \phi_1 \phi_2^{N-1}) \quad (2.15)$$

$$\lambda_1 = P^2 \{ \phi_0^{N-2} + (1 - P) \phi_1^{N-2} + (1 - P)^2 \phi_2^{N-2} \} \quad (2.16)$$

$$\lambda_2 = 3 - P^2 \phi_0^{N-2} - \phi_0^2 \phi_1^{N-2} - \phi_1^2 \phi_2^{N-2} \quad (2.17)$$

Então, o throughput do esquema ARQ selective-repeat com o buffer do receptor de tamanho

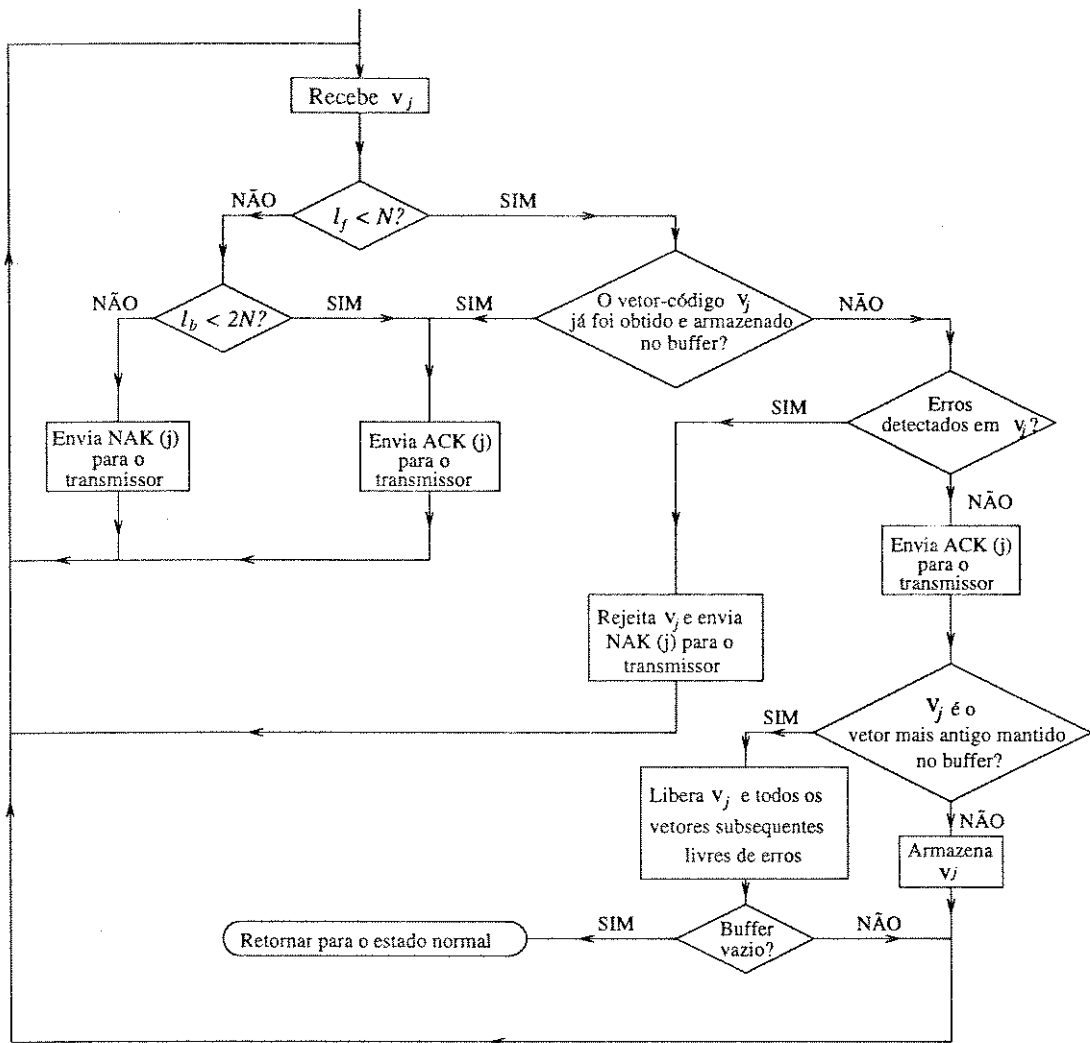


Figura 2.15: Operação Estado Bloqueado do receptor.

$N$  tem o seguinte limitante inferior:

$$\eta_{SRF} \geq \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2 N} \left( \frac{k}{n} \right) \quad (2.18)$$

A figura 2.16 mostra uma comparação do throughput dado pela equação 2.18 com o go-back- $N$  e selective-repeat com o buffer do receptor com capacidade de armazenamento infinito. O selective-repeat com buffer de tamanho  $N$  é menos eficiente do que o ARQ selective-repeat com buffer infinito, devido à redução do tamanho do buffer do receptor. Entretanto, seu desempenho é significativamente superior ao ARQ go-back- $N$ , particularmente em sistemas de comunicação onde o tempo de atraso de percurso é grande e a taxa de dados é alta. Como

exemplo, seja o tempo de atraso de percurso igual a 700 ms. Para  $n = 524$  bits e  $N = 128$  palavras, a taxa de transmissão de dados é de 100 kbps, e para  $n = 1024$  bits e  $N = 1024$  palavras é de 1,54 Mbps.

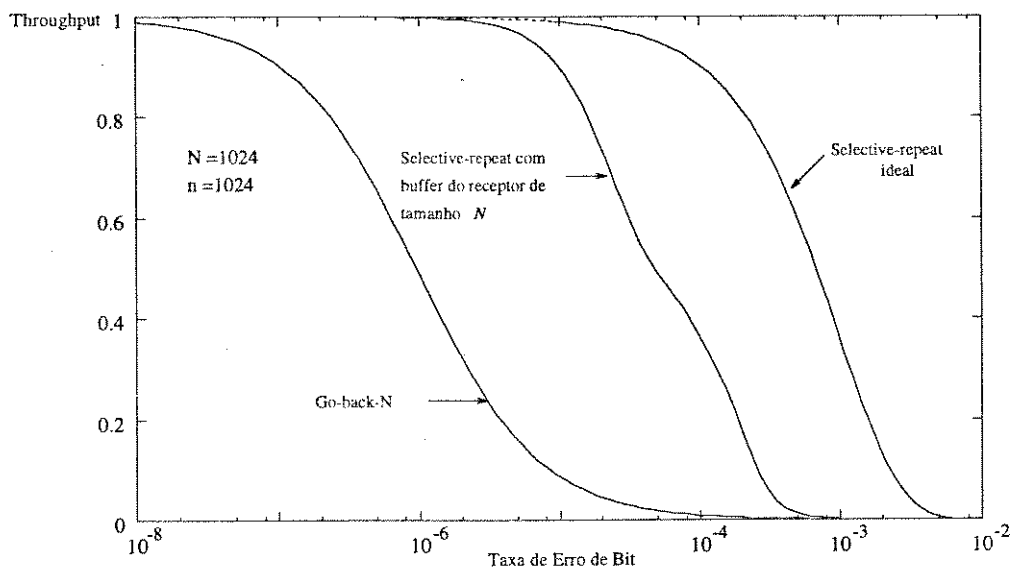


Figura 2.16: Operação estado bloqueado do receptor.

O esquema ARQ selective-repeat descrito nesta seção pode ser estendido para qualquer tamanho do buffer do receptor múltiplo de  $N$ , ou seja,  $2N, 3N, \dots$ . À medida que o tamanho do buffer do receptor aumenta, o throughput também aumenta, aproximando-se do selective-repeat ideal.

## 2.4 Comparações Entre os Diversos Esquemas Selective-Repeat

Nesta seção, os sistemas ARQ selective-repeat descritos nas seções anteriores, são analisados conjuntamente, comparando-os em termos de throughput, conforme mostram as figuras 2.17 - 2.20.

A figura 2.17 compara o limitante inferior do throughput de todos os esquemas ARQ selective-repeat estudados. Observa-se que o desempenho do esquema selective-repeat capaz de

detectar overflow no buffer é superior aos outros esquemas para uma alta taxa de erro de bits, e que o throughput do esquema ARQ misto SR+ST 1 é ligeiramente superior aos esquemas ARQ SR+GBN e Weldon, para os parâmetros considerados.

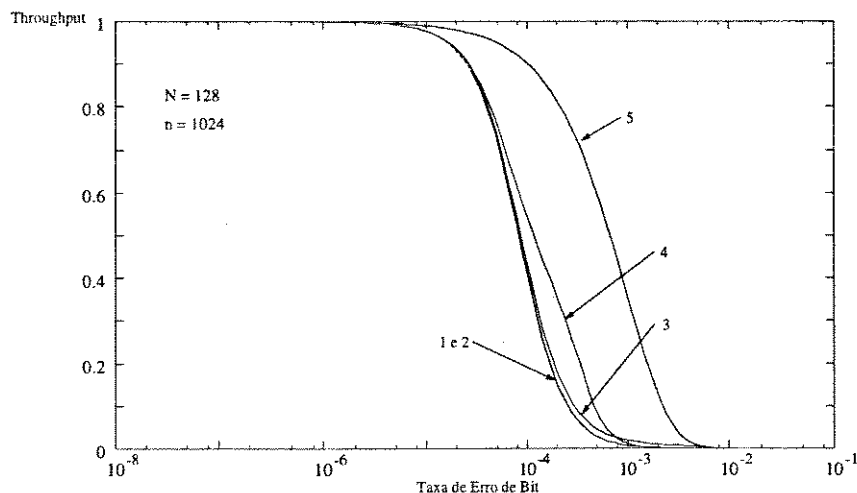


Figura 2.17: Throughput dos esquemas ARQ selective-repeat: (1) esquema SR+GBN para  $v = 1$ ; (2) esquema de Weldon com  $q = 1$  e  $n_1 = 1$ ; (3) esquema SR+ST 1 para  $v = 1$ ; (4) esquema ARQ selective-repeat com buffer finito de tamanho  $N$ , capaz de detectar overflow no buffer do receptor; (5) selective-repeat ideal.

A figura 2.18 faz uma comparação entre os esquemas ARQ misto SR+GBN com o selective-repeat capaz de detectar overflow. Para uma taxa de erro de bit entre  $10^{-6}$  e  $10^{-5}$ , os esquemas ARQ SR+GBN com  $v = 2$  e  $3$  e o selective-repeat com buffer do receptor de tamanho  $2N$  possuem um alto valor de throughput, ao passo que, o desempenho dos esquemas ARQ SR+GBN com  $v = 1$  e selective-repeat com buffer de tamanho  $N$ , começam a diminuir. Para uma taxa em torno de  $10^{-4}$ , o esquema que apresenta o melhor desempenho em termos de throughput, é o esquema ARQ SR+GBN com  $v = 3$ . Para uma taxa de erro de bit acima de  $10^{-4}$ , o throughput do esquema ARQ SR+GBN com  $v = 3$  cai rapidamente, enquanto que o esquema ARQ selective-repeat capaz de detectar overflow com buffer de tamanho  $2N$ , ainda mantém um valor de throughput em torno de 0,4. Os mesmos comentários valem para o ARQ misto SR+ST 1, já que os valores de throughput são semelhantes com o SR+GBN.

- [Yamamoto, 1980] Hirosuke Yamamoto and Kohji Itoh, "Viterbi Decoding Algorithm for Convolutional Codes with Repeat Request," *IEEE Transactions on Information Theory*, VOL. 26, n<sup>o</sup>. 5, September 1980.
- [Yu, 1981] P. S. Yu and Shu Lin, "An Efficient Selective-Repeat ARQ for Satellite Channels and Its Throughput Analysis," *IEEE Transactions on Communications*, VOL. COM-29, pp. 353-363, March 1981.



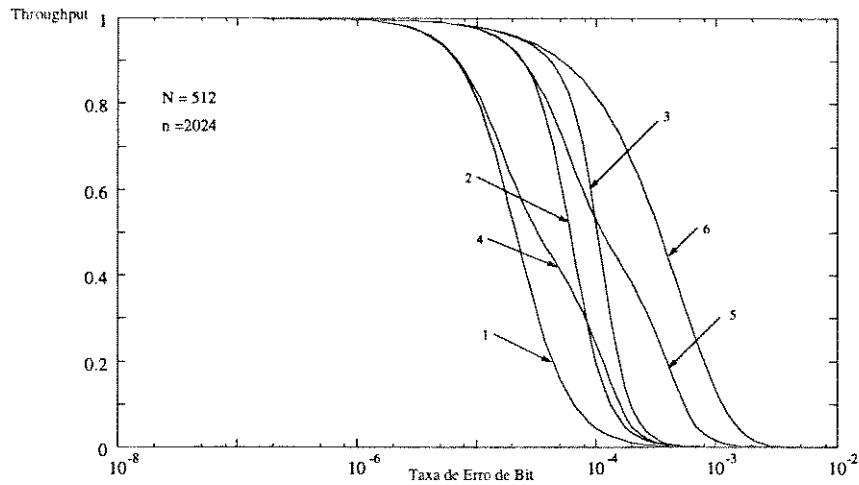


Figura 2.18: Throughput dos esquemas ARQ selective-repeat: (1), (2) e (3) esquema SR+GBN para  $v = 1$ ,  $v = 2$  e  $v = 3$ , respectivamente; (4) e (5) esquema ARQ selective-repeat com buffer finito de tamanho  $N$  e  $2N$ , respectivamente, capaz de detectar overflow no buffer; (6) selective-repeat ideal.

A figura 2.19 compara o esquema SR + ST 1 com  $v = 2$  e 3 com o esquema de Weldon. Observa-se que o throughput do esquema SR+ST 1 com  $v = 2$  é ligeiramente superior ao esquema de Weldon com  $q = 1$  e  $n_1 = 2$ . Para uma taxa de erro de bit igual a  $10^{-4}$ , o esquema ARQ SR+ST 1 com  $v = 3$ , tem um desempenho igual ao selective-repeat ideal, ao passo que, os throughputs dos esquemas de Weldon com  $q = 1$  e  $n_1 = 2$  e 3 e SR+ST 1 com  $v = 2$ , começam a cair. Mas se o valor de  $q$  do esquema de Weldon for maior que um e o valor de  $n_i$  for aumentando à medida que é necessário uma nova retransmissão de uma dada palavra (conforme mostra a curva 5 da figura 2.19), o throughput do esquema de Weldon supera o throughput do esquema SR+ST 1 para  $v = 3$ . Para verificar a superioridade do esquema de Weldon sobre o SR+ST 1, deve-se comparar as curvas (4) e (5), que possuem basicamente valores iguais com relação aos parâmetros usados.

Da figura 2.20 conclui-se que: se aumentar a capacidade de armazenamento do buffer do receptor no esquema selective-repeat que detecta overflow, o seu desempenho em termos de throughput supera os esquemas mistos e o esquema de Weldon. Mas se for utilizado um valor grande de  $v$  nos esquemas mistos, ou um valor grande de  $q$  e  $n_i$  variável (incrementando à

medida que  $i$  aumenta) no esquema de Weldon, pode-se obter um throughput equivalente ao esquema selective-repeat com buffer do receptor de tamanho  $2N$ .

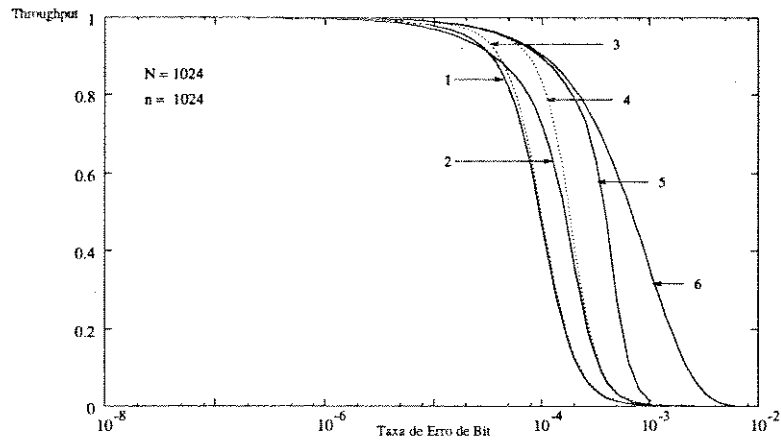


Figura 2.19: Comparação do throughput dos esquemas ARQ SR + ST1 com os esquemas de Weldon: (1) e (2) esquema de Weldon com  $q = 1$  e  $n_1 = 2$  e  $n_1 = 3$ , respectivamente; (3) e (4) esquema ARQ misto SR+ST 1 com  $v = 2$  e  $v = 3$ , respectivamente; (5) esquema de Weldon com  $q = 3$ ,  $n_1 = 1$ ,  $n_2 = 2$  e  $n_3 = 3$ ; (6) esquema ARQ selective-repeat ideal.

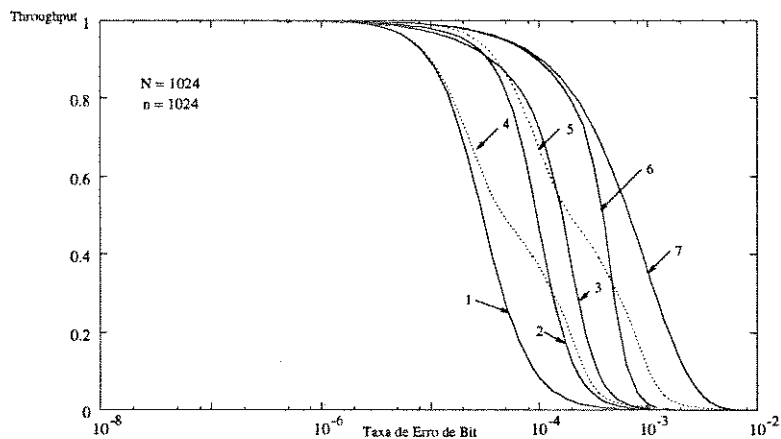


Figura 2.20: Comparação do throughput dos esquemas ARQ SR + ST1 com os esquemas de Weldon: (1), (2) e (3) esquema de Weldon com  $q = 1$  e  $n_1 = 1$ ,  $n_1 = 2$  e  $n_1 = 3$ , respectivamente; (4) e (5) esquema ARQ selective-repeat com buffer finito de tamanho  $N$  e  $2N$ , respectivamente, capaz de detectar overflow no buffer; (6) esquema de Weldon com  $q = 3$ ,  $n_1 = 1$ ,  $n_2 = 2$  e  $n_3 = 3$ ; (7) esquema ARQ selective-repeat ideal.

## 2.5 Conclusão

Todos os esquemas ARQ selective-repeat analisados, possuem um throughput superior ao esquema go-back- $N$ . Os dois esquemas ARQ com transmissões mistas; SR+GBN e SR+ST, basicamente possuem o mesmo throughput. A eficiência destes esquemas é dada pelo número de retransmissões no modo selective-repeat ( $v$ ), ou seja, à medida que este número aumenta, mais eficiente se torna estes esquemas mistos. Se  $v \rightarrow \infty$ , o throughput tende a ser o mesmo do esquema ARQ selective-repeat ideal. Mas um valor grande de  $v$  implica em um buffer do receptor com grande capacidade de armazenamento, já que o tamanho do buffer depende de  $v$ .

O esquema de Weldon é uma variação dos esquemas ARQ mistos. De acordo com os valores dos parâmetros  $q$  e  $n_i, i = 1, 2, \dots, q$ , obtém-se diferentes valores de throughput. Se todos os  $n_i$  forem iguais a 1 e  $q$  for igual a  $v$ , o throughput do esquema de Weldon é o mesmo do esquema misto SR+GBN. Se  $q = 3$ , e  $n_i$  for incrementado à medida que é necessário uma nova retransmissão, a estratégia de Weldon fornece bons resultados em termos de throughput, superando os esquemas SR+ST e SR+GBN, mas apresenta uma complexidade maior em termos de implementação se comparado com estes esquemas ARQ mistos.

Os esquemas ARQ selective-repeat capazes de detectar a ocorrência de overflow no buffer do receptor são mais difíceis de implementar, mas são mais eficientes que os esquemas mistos e Weldon para altas taxas de erro de bits, sendo esta eficiência dada pela capacidade de armazenamento do buffer do receptor.

## Capítulo 3

# ESQUEMAS ARQ HÍBRIDOS USANDO CÓDIGOS DE BLOCOS

### 3.1 Introdução

Comparando os dois esquemas de controle de erros (ARQ e FEC), tem-se que os esquemas ARQ são simples e proporcionam um sistema de alta confiabilidade. Entretanto, se a taxa de erro do canal aumentar, o throughput cai rapidamente. Os sistemas FEC mantêm o throughput constante e igual a taxa do código ( $k/n$ ), sem considerar a taxa de erro do canal. Mas os sistemas FEC têm duas desvantagens. Primeiro, se for detectado erro na palavra recebida, esta deve ser decodificada e liberada para o usuário sem considerar se está correta ou não. Desde que a probabilidade de erro de decodificação é muito maior que a probabilidade de não detectar um erro, é difícil conseguir um sistema com alta confiabilidade usando esquemas FEC. Segundo, para obter um sistema de alta confiabilidade, deve-se usar um longo e poderoso código, tornando a decodificação difícil de implementar e de alto custo.

As deficiências dos esquemas ARQ e FEC podem ser resolvidas se os dois esquemas forem apropriadamente combinados. A combinação dos dois esquemas básicos de controle de erro é conhecida como ARQ híbrido. Um sistema ARQ híbrido consiste de um subsistema FEC contido em um sistema ARQ. A função da parte FEC é reduzir a frequência de retransmissões,

corrigindo alguns padrões de erro que ocorrem com mais frequência, resultando no aumento do throughput. Quando um padrão de erro menos frequente ocorrer e for detectado, o receptor requisita a retransmissão da palavra com erro, para não liberar para o usuário uma mensagem não confiável, aumentando assim, a confiabilidade. Como resultado, uma combinação apropriada de FEC e ARQ, fornece um sistema mais confiável se for comparado com FEC isolado e um throughput maior que os sistemas ARQ puros. Além disso, a implementação do decodificador é simples, pois é projetado para corrigir uma pequena coleção de padrões de erro. Os esquemas ARQ híbridos podem ser classificados em duas categorias:

1. Esquemas ARQ híbridos Tipo-I
2. Esquemas ARQ híbridos Tipo-II

### **3.2 Esquema ARQ Híbrido Tipo-I**

Um esquema ARQ híbrido tipo-I utiliza um código projetado para detectar e corrigir erros simultaneamente. Quando for detectado erro na palavra recebida, o receptor primeiramente tenta corrigi-los. Se o número de erros estiver dentro da capacidade de correção de erro projetada do código, os erros serão corrigidos e a mensagem decodificada será liberada para o usuário ou armazenada no buffer do receptor até que esteja pronta para ser liberada. Se for detectado um padrão de erro não corrigível, o receptor descarta a palavra recebida e solicita uma retransmissão desta palavra. Quando a palavra retransmitida for recebida o receptor novamente verifica se houve erros. Se houver erros não corrigíveis, o receptor solicita uma nova retransmissão e descarta a palavra retransmitida. Este processo continuará até que a palavra seja recebida ou decodificada corretamente.

O processo de correção de erros pode ser incluído em qualquer esquema ARQ. Como o código em um sistema ARQ híbrido é usado para corrigir e detectar erros, é necessário um número maior de bits de paridade do que em um código usado somente para detecção de erro em um sistema ARQ puro. Com consequência, há um aumento da redundância para a transmissão de cada palavra-código. Quando a taxa de erro do canal é baixa, o sistema ARQ híbrido tipo-I tem

um throughput menor do que o seu correspondente sistema ARQ. Entretanto, quando a taxa de erro do canal é alta, o sistema ARQ híbrido tipo-I apresenta um desempenho em termos de throughput maior do que seu correspondente sistema ARQ, pois a capacidade de correção de erro do código, reduz a frequência de retransmissões, como mostra a figura 3.1.

Os esquemas ARQ híbridos tipo-I são mais adequados para sistemas de comunicações nos quais um nível constante de ruídos e interferências é previsto no canal. Neste caso, pode-se projetar um código corretor de erro que corrige a maioria das palavras recebidas, reduzindo o número de retransmissões e aumentando a performance do sistema. Mas para um canal não-estacionário, onde a taxa de erro de bit varia, o esquema ARQ híbrido tipo-I apresenta algumas deficiências. Quando a taxa de erro de bit é baixa (por exemplo, um canal de satélite com tempo bom), a transmissão procede-se normalmente e basicamente não há necessidade de correção de erro. Como resultado, os bits extras de paridade incluídos para a correção de erros em cada transmissão representam um desperdício. Por outro lado, quando o canal estiver muito ruidoso, a capacidade de correção de erro projetada pode se tornar inadequada, aumentando o número de retransmissões, o que reduz o throughput.

Observa-se pela figura 3.1 que; quanto maior for a capacidade de correção do código, mais bits de paridade são necessários, reduzindo por consequência, a taxa de transmissão de informação. Como resultado, para pequenas taxas de erros de bits, o throughput de um código com uma capacidade de correção alta é inferior ao esquema selective-repeat ideal. Entretanto, quando a taxa de erro do canal aumenta, o throughput do esquema ARQ selective-repeat cai rapidamente, enquanto que o ARQ híbrido mantém um valor de throughput significativo.

Seja  $P_{dec}$  a probabilidade de decodificação correta de uma palavra recebida. Esta probabilidade depende da capacidade de correção de erro projetada do código que está sendo utilizado. Para determinar o throughput dos esquemas ARQ híbridos tipo-I, basta substituir  $P$  por  $P_{dec}$  nas equações 1.8, 1.10, 1.12 e 2.18, para obter o throughput dos esquemas: ARQ stop-and-wait híbrido tipo-I, ARQ go-back- $N$  híbrido tipo-I, ARQ selective-repeat híbrido tipo-I com buffer do receptor de tamanho infinito e ARQ selective-repeat híbrido tipo-I com buffer do receptor de tamanho  $N$ , respectivamente.

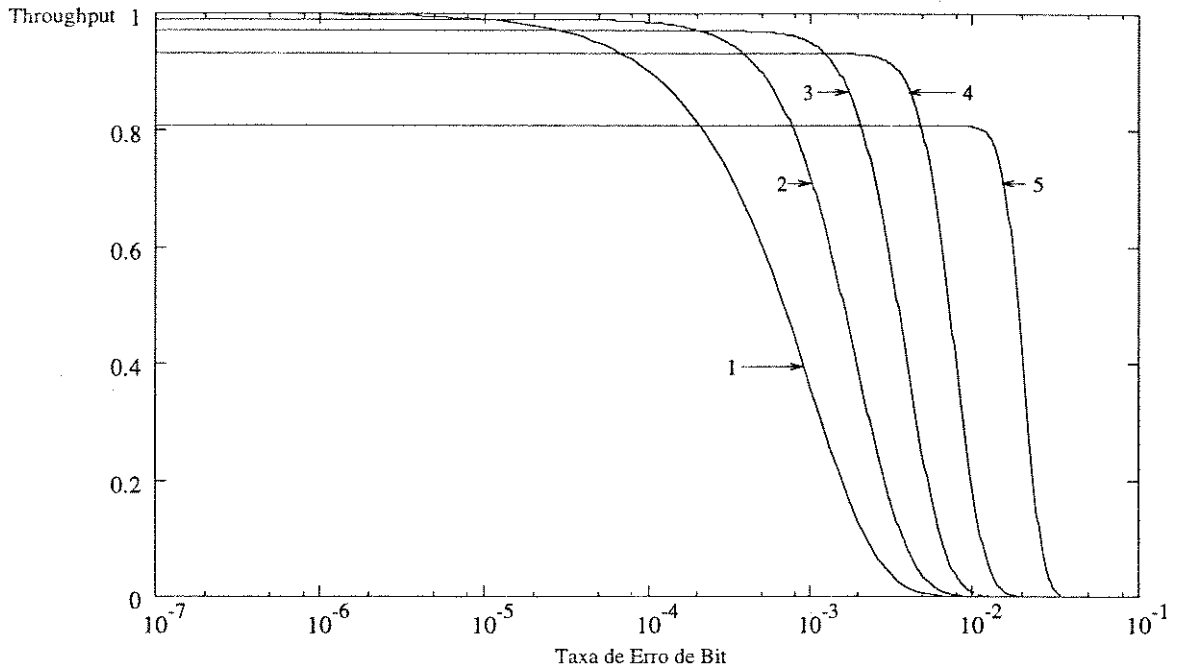


Figura 3.1: Throughput do esquema ARQ híbrido selective-repeat tipo-I para diversas capacidades de correção de erro : (1) selective-repeat ideal; (2) código BCH (1023,1013) com  $t = 1$ ; (3) código BCH (1023,993) com  $t = 3$ ; (4) código BCH (1023,953) com  $t = 7$ ; (5) código BCH (1023,828) com  $t = 20$ .

### 3.3 Esquema ARQ Híbrido Tipo-II

Para um canal com uma taxa de erro de bit não-estacionária, é desejado que o sistema ARQ híbrido seja adaptativo pois, quando a taxa de erro de bits do canal estiver muito pequena, o sistema híbrido se comportará exatamente como um esquema ARQ puro, que utiliza somente bits de paridade para a detecção de erro em cada transmissão. Neste caso, o throughput do esquema ARQ híbrido é o mesmo do esquema ARQ básico correspondente. Quando o canal estiver muito ruidoso, será necessário adicionar bits extras de paridade. Este conceito forma a base dos esquemas ARQ híbridos tipo-II.

Este esquema é baseado no conceito de que os bits de paridade para correção de erros são enviados para o receptor somente quando forem necessários. Normalmente, justapõe-se à mensagem alguns bits de paridade usados somente para a detecção de erro. Quando o receptor

detecta a presença de erros na palavra de informação<sup>1</sup> recebida, ele armazena esta palavra e ao mesmo tempo, solicita a transmissão de uma outra palavra que contém somente bits de verificação de paridade (palavra de paridade), que é formada baseada num código corretor de erros e na palavra de informação. Quando a palavra de paridade for recebida pelo receptor, ela será utilizada para corrigir os erros da palavra de informação que foi recebida com erro e que está armazenada no buffer do receptor. Se os erros forem corrigidos, a mensagem correta é liberada para o usuário. Se os erros não estiverem dentro da capacidade de correção do código, o receptor requisita uma segunda retransmissão. Esta segunda retransmissão pode ser a palavra de informação ou a palavra de paridade, dependendo da estratégia de retransmissão utilizada. Este processo se repete até que a informação correta seja obtida. Se o código corretor de erro utilizado e a estratégia de retransmissão forem apropriadamente escolhidos, o sistema ARQ híbrido tipo-II apresenta desempenho superior ao sistema ARQ híbrido tipo-I.

No esquema ARQ híbrido tipo-II são usados dois códigos: o primeiro é um código  $C_0(n, k)$  de alta taxa, projetado somente para a detecção de erro, e o segundo é um código  $C_1(2k, k)$  inversível de taxa meio, projetado simultaneamente para correção e detecção de erro; este código corrige no máximo  $t$  erros e detecta no máximo  $d$  erros, onde ( $d \geq t$ ). Um código  $(2k, k)$  é dito inversível se, tendo somente os  $k$  bits de paridade de uma palavra-código, é possível obter os  $k$  bits de informação por um processo de inversão (a ser visto em 3.3.1).

Quando uma mensagem  $D$  de  $k$  bits de informação estiver pronta para ser transmitida, ela é codificada em uma palavra-código  $(D, Q)$  de  $n$  bits baseada no código detector de erro  $C_0$ , onde  $Q$  corresponde aos  $n - k$  bits de paridade. A palavra-código  $(D, Q)$  é então transmitida. Ao mesmo tempo, o transmissor computa os  $k$  bits de paridade, denotados por  $P(D)$ , da mensagem  $D$  utilizando o código corretor de erro  $C_1$ . O bloco de paridade de  $k$  bits  $P(D)$ , não é transmitido, mas armazenado no buffer de retransmissão do transmissor (juntamente com a mensagem  $D$ ) para ser usado na retransmissão, se necessário.

Seja  $(\tilde{D}, \tilde{Q})$  a palavra recebida correspondente à  $(D, Q)$ . Quando  $(\tilde{D}, \tilde{Q})$  é recebida, o receptor computa sua síndrome baseada em  $C_0$ . Se a síndrome for zero, então assume-se que a mensagem  $\tilde{D}$  não contém erros, sendo aceita pelo receptor. Se for diferente de zero, indica que foi detectado

---

<sup>1</sup>Palavra de informação, é a palavra que contém os bits que constituem a mensagem.



erros na palavra recebida. A mensagem  $\tilde{D}$  é então armazenada no buffer do receptor e um NAK é enviado para o transmissor. Após o recebimento deste NAK, o transmissor codifica o bloco de  $k$  bits de paridade  $P(D)$  em uma palavra-código  $(P(D), Q^{(1)})$  de  $n$  bits baseada em  $C_0$ , onde  $Q^{(1)}$  corresponde aos  $n - k$  dígitos de paridade. A primeira retransmissão é a palavra de paridade  $(P(D), Q^{(1)})$ .

Quando a palavra recebida  $(\tilde{P}(D), \tilde{Q}^{(1)})$  correspondente à  $(P(D), Q^{(1)})$  for recebida, sua síndrome é computada baseada em  $C_0$ . Se for zero, é considerado que  $\tilde{P}(D)$  não contém erro e a mensagem  $D$  é obtida de  $\tilde{P}(D)$  pelo processo de inversão. Se a síndrome não for zero, então o bloco de paridade  $\tilde{P}(D)$  e a mensagem  $\tilde{D}$  (armazenada no buffer do receptor) são usados para uma possível correção de erros da mensagem baseada no código  $C_1$  de taxa  $1/2$ , pois  $(D, P(D))$  é uma palavra-código em  $C_1$ . Se os erros em  $(\tilde{D}, \tilde{P}(D))$  formarem um padrão de erro corrigível, então estes erros serão corrigidos. A mensagem decodificada  $D$  será aceita pelo receptor e um ACK será enviado para o transmissor. Se os erros em  $(\tilde{D}, \tilde{P}(D))$  formarem um padrão de erro não corrigível, mas detectável, descarta-se  $\tilde{D}$  e armazena-se em sua posição no buffer do receptor, o bloco de paridade  $\tilde{P}(D)$ . Neste caso, é enviado um NAK para o transmissor. Quando o transmissor receber o segundo NAK referente à mensagem  $D$ , ele retransmite a palavra-código  $(D, Q)$ . No receptor, ao receber a palavra  $(\tilde{D}, \tilde{Q})$ , sua síndrome novamente é computada baseada no código  $C_0$ . Se for zero,  $\tilde{D}$  é assumido estar livre de erros, sendo aceita pelo receptor. O bloco  $\tilde{P}(D)$  armazenado no buffer do receptor é então descartado. Se a síndrome não for zero,  $\tilde{D}$  e  $\tilde{P}(D)$  são usados para correção de erros baseados no código  $C_1$ . Se os erros formam um padrão de erro corrigível, a mensagem  $D$  é decodificada e aceita pelo receptor e é enviado um ACK para o transmissor. Entretanto, se os erros formarem um padrão de erro detectável, mas não corrigível, o bloco de paridade  $\tilde{P}(D)$  que estava no buffer do receptor é descartado, sendo armazenado em seu lugar a mensagem  $\tilde{D}$ . Neste caso, envia-se um NAK para o transmissor. A próxima retransmissão será a palavra de paridade  $(P(D), Q^{(1)})$ . Portanto, as retransmissões são repetições alternadas da palavra-código de informação  $(D, Q)$  e palavra-código de paridade  $(P(D), Q^{(1)})$ . O receptor armazena a mensagem recebida  $\tilde{D}$  e o bloco de paridade recebido  $\tilde{P}(D)$  alternadamente. As retransmissões continuam até receber a mensagem  $\tilde{D}$  corretamente ou até a mensagem ser recuperada pelo processo de inversão ou

decodificação.

A característica mais importante do esquema ARQ híbrido tipo-II é a retransmissão dos bits de paridade para a correção de erros baseado no código  $C_1$  inversível de taxa  $1/2$ . Esta estratégia de retransmissão da paridade pode ser incorporada a qualquer esquema ARQ básico; stop-and-wait, go-back- $N$  e selective-repeat. Se for usado em conjunto com o esquema ARQ selective-repeat, esta estratégia se torna mais eficiente. Devido à propriedade de inversão, a mensagem  $D$  pode ser reconstruída unicamente do bloco de paridade  $P(D)$ . Então, o bloco  $P(D)$  e a mensagem  $D$ , contém o mesma quantidade de informação. Como resultado, a redundância na transmissão ou retransmissão é apenas o número  $n - k$  de bits de paridade necessários para a detecção de erros baseados no código  $C_0$ , e que é necessário em qualquer esquema ARQ. Se a taxa de erro do canal estiver baixa, o esquema ARQ híbrido tipo-II tem o mesmo throughput do seu correspondente esquema ARQ básico. Quando a taxa de erro do canal estiver alta, o throughput do ARQ híbrido tipo-II mantém-se alto, por causa da redução no número de retransmissões devido à capacidade de correção do código  $C_1$  e pela retransmissão do bloco de paridade.

Comparando os dois esquemas ARQ híbridos, têm-se que a decodificação do ARQ híbrido tipo-II é um pouco mais complexa do que o ARQ híbrido tipo-I, considerando que os dois têm a mesma capacidade projetada de correção de erros. Os circuitos adicionais necessários no esquema ARQ híbrido tipo-II são: um circuito de inversão baseado no código  $C_1$  e um circuito para detecção de erro baseado no código  $C_0$ . A desvantagem do esquema ARQ híbrido tipo-I é a redundância usada para correção de erros que é incluída em cada transmissão, sem considerar a taxa de erro do canal. Se a taxa de erro do canal estiver baixa, o throughput é menor do que seu correspondente ARQ básico devido aos bits extras de paridade. O esquema ARQ híbrido tipo-II elimina esta desvantagem, pois é um esquema adaptativo. Este esquema é atrativo em sistemas de comunicações de altas taxa de dados onde o tempo de atraso de percurso é grande e a taxa de erro do canal é não-estacionária, como por exemplo, os sistemas de comunicação via satélite.

### 3.3.1 Códigos de Blocos Inversível de Taxa 1/2

A mensagem  $D$  pode ser obtida a partir do bloco de paridade  $P(D)$  pelo processo de inversão, evitando assim, alguma operação de decodificação complicada. A propriedade de inversão também reduz a frequência de retransmissão. Por exemplo, se a mensagem recebida  $\tilde{D}$  contém mais que  $t$  erros (onde  $t$  é a capacidade projetada de correção de erro do código  $C_1$ ) e o bloco de paridade  $\tilde{P}(D)$  não contém erro, o processo de decodificação baseado em  $(\tilde{D}, \tilde{P}(D))$  não será capaz de obter a mensagem original  $D$ , sendo necessária uma outra retransmissão. Entretanto, a partir do bloco de paridade  $\tilde{P}(D)$  recebido sem erro, é possível obter a mensagem  $D$  pelo processo de inversão, evitando desta forma, uma nova retransmissão. O processo de inversão é mostrado em detalhes no apêndice A.

## 3.4 Esquemas ARQ Híbridos Práticos

### 3.4.1 Esquema ARQ Selective-Repeat Híbrido Tipo-II com Buffer Finito no Receptor

Este esquema [Lin, 1982] é uma combinação da retransmissão da paridade e o ARQ selective-repeat com buffer finito no receptor. Seja  $N$  o número de palavras-código que podem ser transmitidas durante o tempo de atraso de percurso. É assumido que o buffer do receptor tem espaço para armazenar  $N$  palavras e a faixa dos números de sequência é igual a  $3N$ .

Seja  $D$  uma mensagem de  $k$  bits a ser transmitida. Há três palavras-código associadas com esta mensagem:

1. A palavra-código de informação  $(D, Q)$  formada pelo código de detecção de erro  $C_0$ .
2. A palavra-código  $(D, P(D))$  que é formada baseada no código de correção de erro inversível de taxa 1/2  $C_1$ .
3. A palavra-código de paridade  $(P(D), Q^{(1)})$  formada pelo bloco de paridade  $P(D)$  e codificada por  $C_0$ .

Para simplificar, denomina-se  $\mathbf{v} = (D, Q)$  a palavra-código de informação de  $D$  e  $\mathbf{v}^* = (P(D), Q^{(1)})$  a palavra-código de paridade da mensagem  $D$ . Usa-se  $\tilde{\mathbf{v}}$  e  $\tilde{\mathbf{v}}^*$  para denotar as palavras recebidas correspondentes à  $\mathbf{v}$  e  $\mathbf{v}^*$ , respectivamente. Na transmissão ou retransmissão a palavra-código de informação  $\mathbf{v}$  e a palavra-código de paridade  $\mathbf{v}^*$  têm o mesmo número de sequência.

Quando a palavra-código de informação  $\mathbf{v}$  estiver pronta para a transmissão, ela é numerada e armazenada na fila de entrada do transmissor. Após ser transmitida, esta palavra e sua correspondente palavra-código de paridade  $\mathbf{v}^*$  são armazenadas no buffer de retransmissão do transmissor, até que  $\mathbf{v}$  seja recebida corretamente. Se um ACK for recebido após o tempo de atraso de percurso, ambas  $\mathbf{v}$  e  $\mathbf{v}^*$  são liberadas do buffer de retransmissão. Se o transmissor receber um NAK após o tempo de atraso de percurso,  $\mathbf{v}^*$  será transmitida para o receptor para correção de erro (se necessário). Após outro tempo de atraso de percurso, se o transmissor receber um ACK, ambas  $\mathbf{v}$  e  $\mathbf{v}^*$  são liberadas do buffer de retransmissão; caso contrário, isto é, se receber um NAK, apenas  $\mathbf{v}$  será retransmitida. O transmissor retransmite  $\mathbf{v}$  e  $\mathbf{v}^*$  alternadamente, até obter a mensagem corretamente, conforme mostra a figura 3.2.

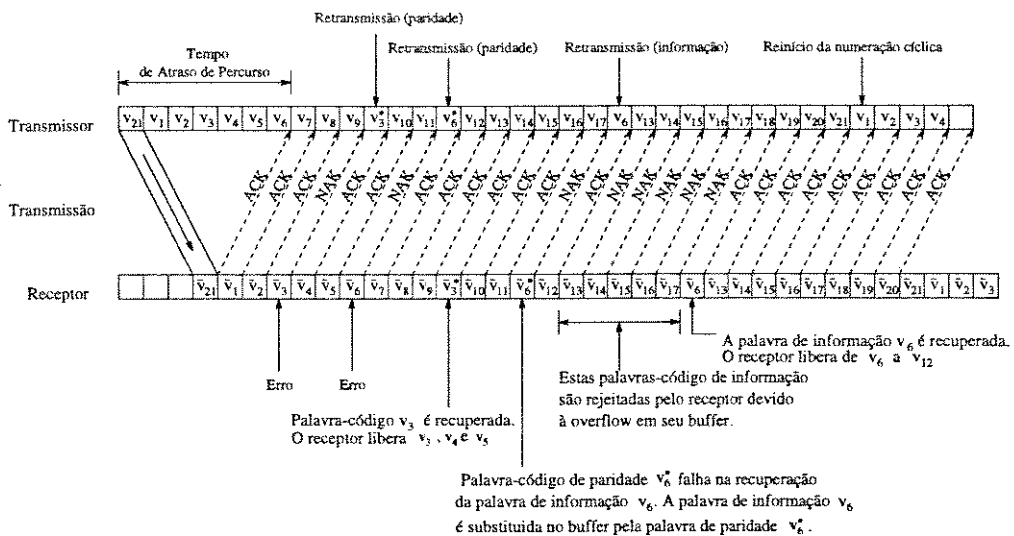


Figura 3.2: Esquema ARQ selectivo-repeat híbrido tipo-II com buffer do receptor de tamanho  $N = 7$ .

## Procedimentos de Transmissão e Retransmissão

Quando o transmissor envia uma palavra-código, seja de informação ou de paridade, é computado o índice *forward*<sup>2</sup>  $f_T$  da palavra-código no buffer de retransmissão, que está para se tornar uma palavra "time-out" (esta palavra é referenciada como a palavra "time-out corrente"). Baseado em  $f_T$ , o transmissor decide se transmite a próxima palavra de informação que está na fila de entrada, ou se inicia um processo de retransmissão. As regras de decisão são:

1. Se o transmissor receber um ACK referente à palavra transmitida e  $f_T < N$  (não há palavra "time-out corrente"), então, a primeira palavra de informação que estiver na fila de entrada será transmitida.
2. Se o transmissor receber um NAK referente à palavra "time-out corrente"  $\mathbf{v}_j$  ou se não receber nenhuma resposta, e  $f_T < N$ , inicia-se a retransmissão da palavra-código de paridade  $\mathbf{v}_j^*$ . Caso  $\mathbf{v}_j^*$  tenha sido a última palavra retransmitida, envia-se novamente  $\mathbf{v}_j$ . Se a palavra "time-out" corrente for uma palavra que havia sido recebida com erro pelo receptor ou cuja resposta referente a seu estado não foi recebida pelo transmissor, receber novamente uma resposta negativa (NAK) ou não receber nenhuma resposta (neste caso,  $f_T = 0$ ), todas as palavras-código de informação que estão no buffer de retransmissão com  $f_T \geq N$  serão novamente colocadas na fila de entrada para serem retransmitidas (estas palavras foram rejeitadas pelo receptor devido a overflow em seu buffer).
3. Se  $f_T \geq N$ , a primeira palavra-código na fila de entrada é a próxima palavra a ser transmitida.

## Operação do Receptor no Estado Normal

No estado normal, o receptor recebe palavras-código de informação e não há palavra armazenada em seu buffer. Ao receber uma palavra-código de informação, o receptor computa sua síndrome baseada no código  $C_0$  e no seu índice forward  $f_R$  com relação a última palavra-código de

---

<sup>2</sup>A definição do índice forward  $f_T$ , é a mesma dada no esquema ARQ selective-repeat com buffer finito, no capítulo 2.

informação aceita e liberada para o usuário. Se não detectou erros na palavra recebida (síndrome igual a zero) e  $f_R = 1$ , a palavra de informação recebida é aceita e liberada para o usuário. O receptor passa então para o processamento da próxima palavra.

Se for detectado erro na palavra de informação recebida ou  $1 < f_R \leq N$  (indica que o número de sequência da palavra recebida está fora de ordem), o receptor passa para o estado bloqueado. No primeiro caso, o receptor armazena em seu buffer, a palavra recebida com erro e as  $N - 1$  subsequentes palavras recebidas. No segundo caso,  $f_R - 1$  palavras de informação entre a última palavra aceita e liberada para o usuário e a palavra recebida corrente foram perdidas. O receptor reserva  $f_R - 1$  posições em seu buffer para estas palavras perdidas, e armazena a palavra recebida corrente na  $f_R$ -ésima posição no buffer.

Se o índice forward  $f_R$  da palavra recebida corrente for maior do que  $N$ , esta palavra será considerada como uma palavra de informação que foi previamente aceita e liberada, sendo então ignorada e envia um ACK para o transmissor. O receptor continua no estado normal e passa a processar a próxima palavra recebida.

### **Operação do Receptor no Estado Bloqueado**

O receptor entra no estado bloqueado quando for detectado a presença de erros na palavra de informação recebida  $\tilde{v}$  ou quando receber uma palavra de informação com seu número de sequência fora de ordem.

Para o primeiro caso, quando detecta-se a presença de erro em  $\tilde{v}$  e  $f_R = 1$ , a mensagem com erro  $\tilde{D}$  é armazenada na primeira posição do buffer do receptor e um NAK será enviado para o transmissor. O receptor continua o processo de detecção de erros das palavras de informação que estão chegando e armazena-as (somente a parte da palavra que contém a mensagem de informação) em locais apropriados de seu buffer de acordo com os números de sequência, até preencher todo o buffer. Para cada palavra recebida, é enviado uma mensagem informando ao transmissor, se a palavra foi recebida corretamente ou não.

No segundo caso, quando  $f_R > 1$ , são reservados locais apropriados para as palavras de informação que foram perdidas. A palavra de informação corrente é armazenada na  $f_R$ -ésima posição (novamente, somente os bits de informação). As palavras de informação subsequentes

à palavra corrente, são armazenadas em posições vazias que restam no buffer do receptor.

Quando a primeira palavra-código de paridade  $\tilde{\mathbf{v}}_j^* = (\tilde{P}(D_j), \tilde{Q}_j^{(1)})$  for recebida, ela será usada para obter a mensagem  $D_j$  que foi anteriormente recebida com erro (ou que foi perdida). Se a síndrome de  $\tilde{\mathbf{v}}_j^*$  for igual a zero, obtém-se  $D_j$  utilizando o processo de inversão em  $\tilde{P}(D_j)$ . Se a síndrome de  $\tilde{\mathbf{v}}_j^*$  não for zero, então  $\tilde{P}(D_j)$  e  $\tilde{D}_j$  (armazenada no buffer) juntamente são usadas para corrigir os erros, baseados no código inversível de taxa  $1/2 C_1$ . Se os erros forem corrigíveis, a mensagem  $D_j$  é recuperada. Após a mensagem  $D_j$  ter sido recuperada, o receptor libera  $D_j$  para o usuário e também todas as mensagens subsequentes livres de erros armazenadas no buffer do receptor consecutivamente;  $D_j, D_{j+1}, \dots, D_{j+L}$ , onde  $0 \leq L < N$ . Desde que a retransmissão é seletiva e desde que  $D_{j+1}$  até  $D_{j+L}$  foram obtidas em ordem, as palavras recebidas após  $\tilde{\mathbf{v}}_j^*$  são novas palavras de informação;  $\tilde{\mathbf{v}}_{j+N}, \tilde{\mathbf{v}}_{j+N+1}, \dots, \tilde{\mathbf{v}}_{j+N+L-1}$ . Quando estas palavras de informação chegam no receptor, suas síndromes são computadas e temporariamente armazenadas no buffer, até que estejam prontas para serem liberadas para o usuário ou para serem corrigidas.

Se  $\tilde{P}(D_j)$  falhar na recuperação da mensagem  $D_j$ , não será possível liberar as mensagens que foram recebidas sem erros, e que estão armazenadas no buffer do receptor. Também não há espaço no buffer do receptor para armazenar as novas palavras de informação que estão chegando:  $\tilde{\mathbf{v}}_{j+N}, \tilde{\mathbf{v}}_{j+N+1}, \dots, \tilde{\mathbf{v}}_{j+N+L-1}$ . Neste caso, ocorrerá overflow no buffer e estas novas palavras serão rejeitadas pelo receptor.

Se o transmissor receber o segundo NAK referente à palavra-código  $\mathbf{v}_j$ , esta palavra é novamente enviada para o receptor. Ao receber  $\tilde{\mathbf{v}}_j$ , o receptor tenta recuperar a mensagem  $D_j$ . Se a síndrome de  $\tilde{\mathbf{v}}_j$  for zero, então assume-se que  $\tilde{D}_j$  não contém erros, liberando-a para o usuário. Se a síndrome não for zero,  $\tilde{D}_j$  e  $\tilde{P}(D_j)$  são usadas para corrigir os erros baseados no código  $C_1$ . Se o receptor novamente falhar na obtenção da mensagem  $D_j$ ,  $\tilde{P}(D_j)$  (que estava armazenada no buffer do receptor) é descartada e  $\tilde{D}_j$  será armazenada em seu lugar. As retransmissões alternadas para a palavra-código  $\mathbf{v}_j$  continuarão até que a mensagem  $D_j$  seja recebida ou decodificada corretamente. As retransmissões são sempre repetições alternadas da palavra-código de informação  $\mathbf{v}_j$  e da palavra-código de paridade  $\mathbf{v}_j^*$ . Portanto, no receptor  $\tilde{D}_j$  e  $\tilde{P}(D_j)$  são armazenadas alternadamente.

Seja  $ns_0$  o número de sequência da primeira palavra de informação recebida com erro e armazenada no buffer do receptor. No estado bloqueado, ao receber uma palavra (de informação ou paridade) com número de sequência  $ns$ , o receptor computa o índice "forward"  $l_f$  da palavra recebida:

$$l_f = ns - ns_0 \pmod{3N} \quad (3.1)$$

Se  $l_f < N$ , a palavra recebida (que pode ser uma palavra de informação ou uma palavra de paridade) será usada para recuperar a mensagem correspondente. Se  $l_f \geq N$ , o receptor computa o índice "backward" (denotado por  $l_b$ ) desta palavra recebida, com relação à última palavra que foi armazenada no buffer do receptor. O índice backward é definido como:

$$l_b = z_0 - ns \pmod{3N} \quad (3.2)$$

onde  $z_0$  é o número de sequência da última palavra que foi armazenada no buffer do receptor. Se  $l_b < 2N$ , a palavra recebida corrente, é considerada como uma palavra que foi anteriormente aceita e liberada para o usuário. Esta palavra é descartada e o receptor envia um ACK para o transmissor. Se  $l_b \geq 2N$ , a palavra recebida é uma nova palavra de informação; entretanto, o buffer está cheio e ocorrerá overflow. Neste caso, a palavra recebida corrente é descartada e o receptor envia um NAK para o transmissor.

Quando uma palavra retransmitida é recebida pelo receptor, este verifica se esta palavra e a palavra que está armazenada no buffer formam o par  $\tilde{D}$  e  $\tilde{P}(D)$ . Se formarem este par, o receptor tenta obter a mensagem original  $D$ . Se não formarem este par, o receptor novamente verifica se a palavra recebida e sua correspondente palavra armazenada no buffer, são ambas palavras de informação ou palavras de paridade. Se forem ambas palavras de informação ou de paridade, o receptor armazena a que estiver sem erro; se ambas estiverem com erro, o receptor substituirá a palavra que estava armazenada pela palavra recebida.

Para a análise do throughput do esquema ARQ selective-repeat tipo-II com buffer de tamanho  $N$ , são feitas algumas considerações com relação ao canal; é um BSC (canal binário simétrico) e que não há interferência do ruído no canal de retorno. Com isso, é possível obter um limitante inferior do throughput. Novamente, seja  $p$  a probabilidade de transição do canal e  $P$ , a probabilidade de que a palavra-código de informação ou de paridade seja recebida com



sucesso (isto é, a síndrome baseada em  $C_0$  é igual a zero).

Suponha que  $C_1$  seja capaz de corrigir qualquer combinação de  $t$  erros ou menos e simultaneamente detectar qualquer combinação de  $d$  ( $d > t$ ) erros ou menos. Seja  $Q_1$  a probabilidade condicional de uma mensagem  $D_j$  ser obtida a partir da primeira palavra de paridade recebida  $\tilde{P}(D_j)$  ou por inversão ou por decodificação baseada em  $C_1$ , dado que são detectados erros na palavra de informação recebida  $\tilde{v}_j$ . Seja  $Q_2$  a probabilidade condicional de que a mensagem  $D_j$  seja recuperada na segunda retransmissão de  $v_j$ , dado que erros são detectados na primeira palavra-código de paridade recebida  $\tilde{v}_j^*$  e na primeira palavra-código de informação  $\tilde{v}_j$ , e que o primeiro bloco de paridade  $\tilde{P}(D_j)$  falha na obtenção da mensagem  $D_j$ , mas detecta a presença de erros em  $(\tilde{P}(D_j), \tilde{D}_j)$ . Então, é possível mostrar que [Lin, 1982]:

$$Q_1 = P + \frac{1}{1-P} \left\{ \sum_{i=0}^t \binom{2k}{i} p^i (1-p)^{2k-i} + (1-p)^{2n} - 2(1-p)^n \left[ (1-p)^k + \sum_{i=1}^t \binom{k}{i} p^i (1-p)^{k-i} \right] \right\} \quad (3.3)$$

e que

$$Q_2 \geq P + \frac{1}{(1-P)(1-Q_1)} \left\{ \sum_{i=1}^{t-1} \Delta_i S_{t-i} (1 - \Delta_0 - S_{t-i}) \right\} \quad (3.4)$$

onde

$$\Delta_i = \binom{k}{i} p^i (1-p)^{k-i},$$

$$S_j = \sum_{l=1}^j \Delta_l.$$

Definindo

$$\theta_0 = 1 - (1-P)(1-Q_1),$$

$$\theta_1 = 1 - (1-P)(1-Q_1)(1-Q_2),$$

$$\theta_2 = Q_1 + (1-Q_1)Q_2.$$

Assim obtém-se o limitante inferior do throughput do esquema ARQ selective-repeat híbrido

tipo-II com buffer do receptor de tamanho  $N$ :

$$\eta_{II} \geq \frac{\delta_0}{\delta_0 + \delta_1 + \delta_2 N} \quad (3.5)$$

onde

$$\begin{aligned} \delta_0 &= \frac{\theta_2}{1 - \theta_1} (1 - \theta_2 \theta_1^{N-1}), \\ \delta_1 &= PQ_1 \theta_0^{N-2} + P(1 - Q_1) Q_2 \theta_1^{N-2}, \\ \delta_2 &= 2 - PQ_1 \theta_0^{N-2} - \theta_0 \theta_2 \theta_1^{N-2}. \end{aligned} \quad (3.6)$$

As figuras 3.3 e 3.4 mostram o limitante inferior do throughput versus a taxa de erro de bit para vários valores de  $n, k, t$ , e  $N$ . Note que o desempenho em termos de throughput do esquema ARQ selective-repeat híbrido tipo-II com  $t = 5$ , é equivalente ao esquema selective-repeat ideal. E se a capacidade de correção de erro projetada ( $t$ ) for maior que 5, obtém-se um desempenho superior ao selective-repeat ideal. O throughput melhora à medida que o valor de  $t$  aumenta.

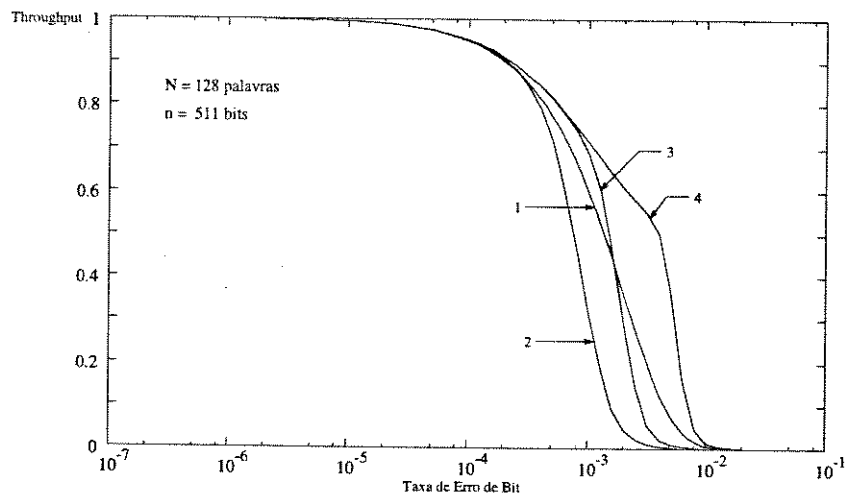


Figura 3.3: Throughput de vários esquemas ARQ com  $N = 128$  palavras e  $n = 511$  bits: (1) ARQ selective-repeat ideal com buffer do receptor infinito; (2), (3) e (4) ARQ selective-repeat híbrido tipo-II com buffer do receptor de tamanho  $N$  e parâmetro de correção de erro  $t = 3, 5$ , e  $10$ , respectivamente.

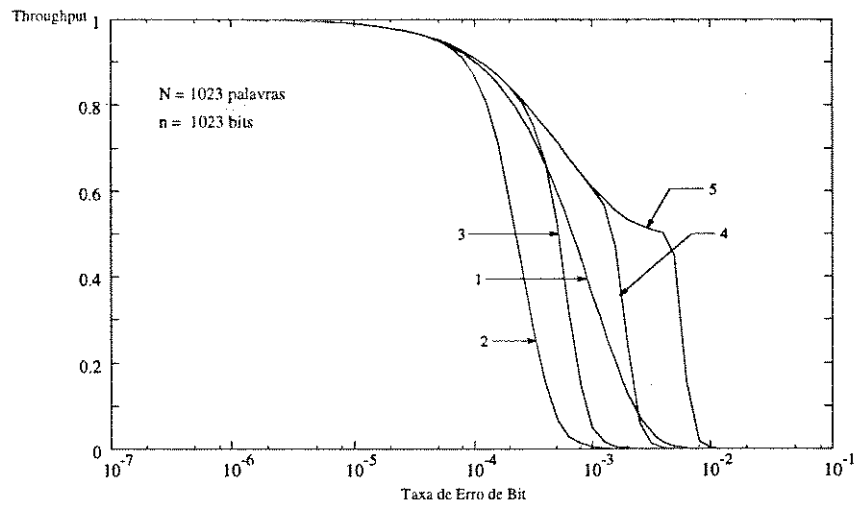


Figura 3.4: Throughput de vários esquemas ARQ com  $N = 1023$  palavras e  $n = 1023$  bits: (1) ARQ selective-repeat ideal com buffer do receptor infinito; (2), (3), (4) e (5) ARQ selective-repeat híbrido tipo-II com buffer do receptor de tamanho  $N$  e parâmetro de correção de erro  $t = 3, 5, 10$  e  $20$ , respectivamente.

Se a capacidade de detecção de erro do código  $C_1$  for suficientemente grande, o esquema ARQ híbrido tipo-II possui a mesma confiabilidade do esquema ARQ selective-repeat básico.

Seja  $P_e$  a probabilidade de  $C_0$  falhar na detecção de erros. Considere também:

$$\sigma = \sum_{i>d} \binom{2k}{i} p^i (1-p)^{2k-i} \tag{3.7}$$

a probabilidade de que o número de erros em  $(\hat{P}(D), \tilde{D})$  ultrapasse a capacidade projetada de detecção de erros  $d$  do código  $C_1$ . Portanto, a probabilidade do receptor cometer um erro de decodificação  $P(E)_H$  é limitada por [Lin, 1982]:

$$\frac{P_e}{P_e + P_c} \leq P(E)_H \leq \frac{P_e}{P_e + P_c} + \frac{\sigma}{P_e + P_c} \tag{3.8}$$

onde  $P_c = (1-p)^n$ . Se for utilizado um valor de  $d$  suficientemente grande, pode-se obter um valor de  $\sigma$  menor ou igual a  $P_e$ . Como resultado, tem-se  $P(E)_H$  da mesma ordem que a probabilidade de erro de um esquema ARQ puro.

Como exemplo, seja  $C_0$  um código linear (524,500) e  $C_1$  um código BCH (1000,500) encurtado com  $d_{min} \geq 111$ . Suponha que  $C_1$  seja usado para corrigir 10 erros ou menos ( $t \leq 10$ ), e

simultaneamente detectar 100 erros ou menos ( $d \leq 100$ ). Seja  $p = 10^{-2}$ . Então, obtém-se:

$$P_c = (1 - 10^{-2})^{524} = 5,16 \times 10^{-3}$$

$$\sigma \leq 1,8 \times 10^{-63}$$

Como  $n - k = 24$ , segue da equação 1.5 que

$$P_e \leq 6 \times 10^{-8}, \quad (3.9)$$

Observa-se que  $\sigma \ll P_e$ . Portanto, tem-se:

$$P(E)_H = \frac{P_e}{P_e + P_c} \leq 1,17 \times 10^{-5} \quad (3.10)$$

que é a mesma probabilidade de um esquema ARQ puro (ver equação 1.3).

### 3.4.2 Esquema ARQ Selective-Repeat Híbrido Tipo-II Modificado

Este esquema difere do descrito anteriormente [Wang, 1983], nos seguintes aspectos: o código  $C_1$  é usado somente para a correção de erros e o processo de detecção de erros na transmissão e retransmissão é baseado somente no código  $C_0$ . A capacidade de correção de erro projetada de  $C_1$  pode ser aumentada para  $t = \lfloor (d_{min} - 1)/2 \rfloor$ . Este aumento na capacidade de correção resulta em vantagens do esquema tipo-II modificado sobre o esquema ARQ híbrido tipo-II original, particularmente quando o tamanho  $n$  da palavra-código é pequeno.

O esquema ARQ híbrido modificado emprega dois códigos  $C_0$  e  $C_1$  e uma estratégia de retransmissão alternada das palavras-código de informação e de paridade.  $C_0$  é um código  $(n, k)$  de alta taxa usado na detecção de erro e  $C_1$  é um código inversível  $(2n, n)$  de taxa meio, projetado apenas para a correção de erros.

Seja  $I = (D, Q)$  a palavra-código de informação de  $n$  bits, baseada na mensagem de informação  $D$  e no código detector de erro  $C_0$ . A palavra de paridade de  $n$  bits  $P(I)$  correspondente a  $I$ , é formada baseada em  $I$  e no código corretor de erro  $C_1$ . Então,  $(I, P(I))$  é uma palavra-código em  $C_1$ . Como  $C_1$  é um código inversível, a palavra de informação  $I$  pode ser obtida invertendo  $P(I)$ . Suponha que  $\tilde{I}$  seja a palavra de informação recebida relativa a  $I$ . No receptor, quando  $\tilde{I}$  é recebida, sua síndrome é computada baseada em  $C_0$ . Se a síndrome for zero,  $\tilde{I}$  é

considerada sem erros sendo então aceita pelo receptor. Ao mesmo tempo, é enviado um ACK para o transmissor. Se a síndrome não for zero, indica que foi detectado erros em  $\tilde{I}$ . Então,  $\tilde{I}$  é armazenada no buffer do receptor e um NAK é enviado para o transmissor. Após receber o NAK, o transmissor envia a palavra de paridade  $P(I)$  para o receptor.

Seja  $\tilde{P}(I)$  a palavra de paridade recebida. Ao receber  $\tilde{P}(I)$ , o receptor primeiramente obtém o inverso de  $\tilde{P}(I)$ , denotado por  $\tilde{I}(P)$ , baseado em  $C_1$ . Se não houver erros em  $\tilde{P}(I)$ , então  $\tilde{I}(P)$  deve ser a palavra original  $I$  que é uma palavra-código de  $C_0$ . Após a obtenção de  $\tilde{I}(P)$ , o receptor computa a sua síndrome baseado em  $C_0$ . Se a síndrome de  $\tilde{I}(P)$  for zero,  $\tilde{I}(P)$  é considerada estar livre de erros, sendo aceita pelo receptor. Se a síndrome de  $\tilde{I}(P)$  não for zero, então a palavra de paridade  $\tilde{P}(I)$  e a palavra de informação com erro  $\tilde{I}$  (armazenada anteriormente), são usadas para a correção de erros baseado no código  $C_1$ . Seja  $I^*$ , a palavra de informação decodificada. Após o processo de decodificação, a síndrome de  $I^*$  é verificada baseada em  $C_0$ . Se for zero,  $I^*$  é assumida estar livre de erros sendo aceita pelo receptor. Se a síndrome de  $I^*$  não for zero, indica que existe erros. Neste caso, o receptor descarta a palavra de informação com erro  $\tilde{I}$ , armazena em seu lugar a palavra de paridade com erro  $\tilde{P}(I)$  e solicita uma nova retransmissão. A próxima retransmissão será a palavra de informação  $I$ . As retransmissões continuarão até que a palavra de informação seja obtida corretamente. As retransmissões são repetições alternadas de  $I$  e  $P(I)$ .

No esquema ARQ híbrido tipo-II modificado, o número de bits de paridade disponíveis para a correção de erro é maior do que no esquema ARQ híbrido tipo-II original. Consequentemente, o esquema modificado tem vantagens sobre o esquema ARQ híbrido tipo-II original. Como exemplo, considere  $n = 100$  e  $k = 80$ . No esquema ARQ híbrido tipo-II original, a distância de Hamming mínima de  $C_1$  é limitada superiormente em 23 [Mac, 1977] e portanto, a capacidade de correção de erro  $t$  de  $C_1$  é limitada superiormente em 11. Desde que, parte da distância mínima deve ser usada para a detecção de erro após a decodificação, a capacidade efetiva de correção de erro  $t_1$  deve ser menor que  $t$ . Por outro lado, no esquema modificado,  $C_1$  pode ser um código BCH (200,100) encurtado que é capaz de corrigir pelo menos 13 erros, isto é  $t \geq 13$ . Como  $C_1$  é usado somente para a correção de erros, pode-se utilizar toda esta capacidade no processo de correção de erro.

O throughput do esquema ARQ híbrido tipo-II modificado considerando o buffer do receptor de tamanho infinito é dado por [Wang, 1983]:

$$\eta_{HM} = \frac{1 + (1 - P)P_t/P}{2 - P} \cdot \left\{ P \binom{k}{n} \right\} \quad (3.11)$$

onde a probabilidade de recuperar a palavra de informação após o recebimento da palavra de paridade é:

$$P_t = P + (1 - P)q_1 \quad (3.12)$$

A probabilidade condicional de uma decodificação correta baseada nas palavras de informação e paridade  $(\tilde{I}, \tilde{P})$ , dado que foram detectados erros em ambas  $\tilde{I}$  e  $\tilde{P}$  é dada por:

$$q_1 = \frac{1}{1 - y}(q_0 - y) \quad (3.13)$$

e a probabilidade de decodificação correta de  $(\tilde{I}, \tilde{P})$  é<sup>3</sup>:

$$q_0 = \sum_{j=0}^{t_1} \binom{2n}{j} p^j (1 - p)^{2n-j} \quad (3.14)$$

Finalmente, o valor de  $y$  é obtido por:

$$y = (1 - p)^n \left\{ 2 \sum_{l=0}^{t_1} \binom{n}{l} p^l (1 - p)^{n-l} - (1 - p)^n \right\} \quad (3.15)$$

Pela equação do throughput, tem-se que  $\frac{1+(1-P)P_t/P}{2-P}$  é o ganho do esquema ARQ híbrido tipo-II modificado sobre o esquema selective-repeat ideal. Para  $t_1$  maior do que zero, tem-se  $q_1 \geq 0$  e  $P_t \geq P$ . Portanto, o ganho é sempre maior ou igual a 1. Consequentemente,  $\eta_{HM} \geq \eta_{SR} = P(k/n)$ . Quando a taxa de erro do canal  $p$  é baixa, então  $q_1 \approx P \approx 1$ . Neste caso, o ganho é aproximadamente igual a 1 e  $\eta_{HM} = \eta_{SR}$ . Quando a taxa de erro do canal é alta,  $q_1 \geq P$  para  $t > 0$ . Neste caso, o ganho é muito maior que 1 e  $\eta_{HM} \gg \eta_{SR}$ . A capacidade de correção de erro projetada  $t_1$  do código  $C_1$ , determina o ganho do throughput do esquema híbrido modificado sobre o esquema ARQ selective-repeat ideal.

Observe que o esquema ARQ selective-repeat ideal é um caso particular do esquema ARQ selective-repeat híbrido tipo-II modificado quando a palavra de paridade  $P(I)$  é simplesmente

<sup>3</sup> $t_1$  é a capacidade de correção projetada do código  $C_1$ .

idêntica à sua correspondente palavra de informação  $I$ , não há processo de correção de erro e cada decisão é feita baseada na palavra de informação recebida. Neste caso,  $q_0 = q_1 = 0$  e  $\eta_{HM} = \eta_{SR} = P(k/n)$ . As figuras 3.5 e 3.6 mostram as curvas de throughput versus a taxa de erro de bit do esquema modificado com buffer do receptor de tamanho infinito.

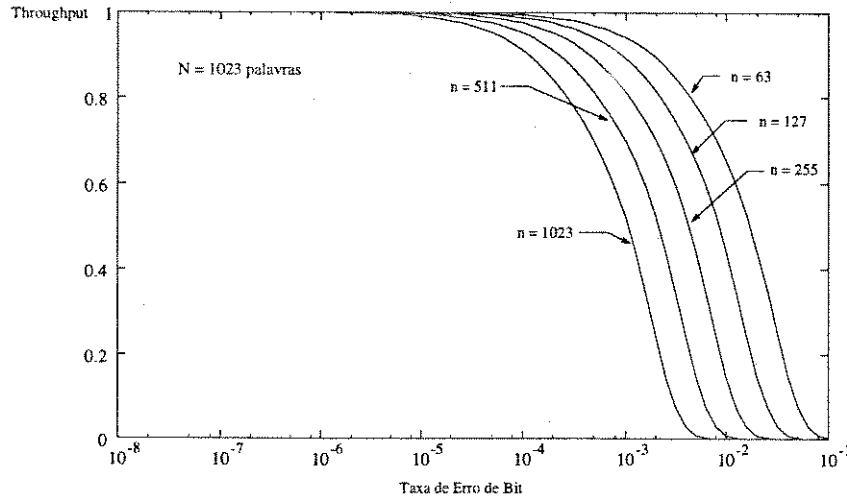


Figura 3.5: Throughput de vários esquemas ARQ híbridos tipo-II modificado com buffer infinito e  $t = 3$ , tendo como parâmetro o comprimento  $n$  do código  $C_1$ .

A figura 3.5 mostra que, utilizando códigos com a mesma capacidade de correção ( $t = 3$ ), o desempenho em termos de throughput, será melhor para os sistemas que utilizam códigos de comprimentos menores.

A figura 3.6 mostra que se for utilizado um código com capacidade de correção maior ou igual a três, o throughput do esquema ARQ híbrido modificado com buffer infinito, é superior ao selective-repeat ideal.

Para o cálculo do throughput com buffer do receptor de tamanho  $B$  finito, existem dois casos a serem considerados:

1) Se  $B \leq N$ , onde  $N$  é o número de blocos que o transmissor pode enviar durante o tempo de atraso de percurso. Então:

$$\eta_{HMF_1} \geq \frac{1 + P_t \phi_0(B, P, P_t)/P}{1 + (2 - P - \sigma)N/\sigma} \cdot \left\{ P \left( \frac{k}{n} \right) \right\} \quad (3.16)$$

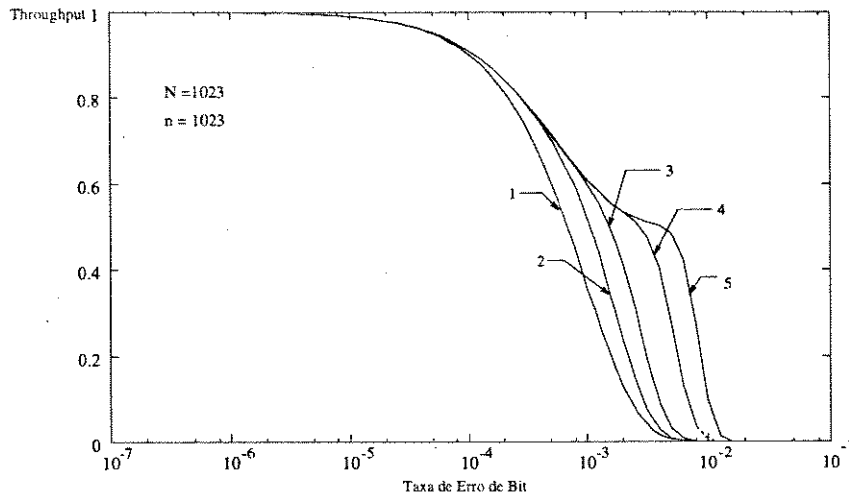


Figura 3.6: Throughput do esquema ARQ híbrido tipo-II modificado com buffer infinito e  $n = 1023$ : (1) Selective-repeat ideal; (2), (3), (4) e (5) Esquemas híbridos tipo-II modificado com  $t = 3, 5, 10$  e  $16$ , respectivamente.

onde  $\phi_0(B, P, P_t)$  é dado por:

$$\phi_0(B, P, P_t) = \sum_{j=1}^B \frac{(-1)^{j+1} \binom{B}{j} (1-P)^j [1 + P(1-P_t)^j / P_t]}{1 - (1-\sigma)^j} \quad (3.17)$$

e

$$\sigma = 1 - (1-P)(1-P_t) \quad (3.18)$$

2) Se  $B = s(N-1) + 1 + b$ , onde  $s$  e  $b$  são dois inteiros positivos e  $b \leq N-1$ , então:

$$\eta_{HMF_2} \geq \frac{[P_{cs} + P_t \phi_s(N, P, P_t)] / P}{Z + F} \cdot \left\{ P \binom{k}{n} \right\} \quad (3.19)$$

onde a probabilidade do receptor recuperar a palavra de informação nas primeiras  $s+1$  transmissões e retransmissões é dada por:

$$P_{cs} = \begin{cases} 1 - (1-P)(1-\sigma)^{s/2}, & \text{para } s \text{ par;} \\ 1 - (1-\sigma)^{(s+1)/2}, & \text{para } s \text{ ímpar;} \end{cases} \quad (3.20)$$

e  $Z, F$  e  $\phi_s(N, P, P_t)$  são definidos como:



$$Z = \begin{cases} \frac{(2-P)-[(2-P)+(s-sP-P)\sigma](1-\sigma)^{s/2}}{\sigma}, & \text{para } s \text{ par} \\ \frac{(2-P)-[(2-P)+(s+1)\sigma](1-\sigma)^{(s+1)/2}}{\sigma}, & \text{para } s \text{ ímpar} \end{cases} \quad (3.21)$$

$$F = \begin{cases} \left[ \frac{(2-P-\sigma)N+(1-P)\sigma}{\sigma} \right] (1-\sigma)^{s/2}, & \text{para } s \text{ par;} \\ \left[ \frac{(2-P)N+\sigma}{\sigma} \right] (1-\sigma)^{(s+1)/2}, & \text{para } s \text{ ímpar} \end{cases} \quad (3.22)$$

$$\phi_s(N, P, P_t) = \begin{cases} \frac{\sum_{j=1}^N (-1)^{j+1} \binom{N}{j} (1-\sigma)^{sj/2} [(1-P)^j + (1-\sigma)^j P/P_t]}{1-(1-\sigma)^j}, & \text{para } s \text{ par;} \\ \frac{\sum_{j=1}^N (-1)^{j+1} \binom{N}{j} (1-\sigma)^{(s+1)j/2} [(1-P)^j + P/P_t]}{1-(1-\sigma)^j}, & \text{para } s \text{ ímpar} \end{cases} \quad (3.23)$$

Note que para  $B \rightarrow \infty$  (ou  $s \rightarrow \infty$ ), a equação 3.19 se reduz à equação do throughput do esquema modificado com buffer do receptor infinito. A figura 3.7 mostra o throughput do esquema ARQ híbrido tipo-II modificado, considerando o tamanho do buffer do receptor igual a  $N + 1$ . O código  $C_1$  utilizado para a correção de erros, é um código BCH encurtado  $(2n, n)$ .

A figura 3.7 compara o esquema ARQ híbrido tipo-II modificado com buffer do receptor de tamanho finito com o esquema ARQ selective-repeat com buffer finito e infinito. O desempenho em termos de throughput do esquema modificado utilizando um código com capacidade de correção de erro  $t \geq 5$ , já é superior ao esquema selective-repeat ideal considerando  $n = 512$  bits e  $N = 128$  palavras. Se for utilizada a capacidade máxima de correção de erro do código, mesmo para taxas de erros maiores que  $10^{-2}$ , o throughput se mantém alto (em torno de 0.5). A figura 3.8 faz a mesma comparação em termos de throughput mostrada na figura 3.7, com  $n = 1023$  bits e  $N = 1024$  palavras.

Se a capacidade de correção de erro projetada for escolhida tal que  $p < t/4n$  e  $P_t \cong 1$  (ou  $\sigma \cong 1$ ), isto indica que a palavra de informação será recebida corretamente ou recuperada corretamente a partir da palavra de paridade retransmitida. Consequentemente, tem-se:

$$\eta_{HM} \cong \frac{1}{2-P} \left( \frac{k}{n} \right) \quad (3.24)$$

Da equação 3.24, observa-se que o throughput do esquema ARQ selective-repeat híbrido

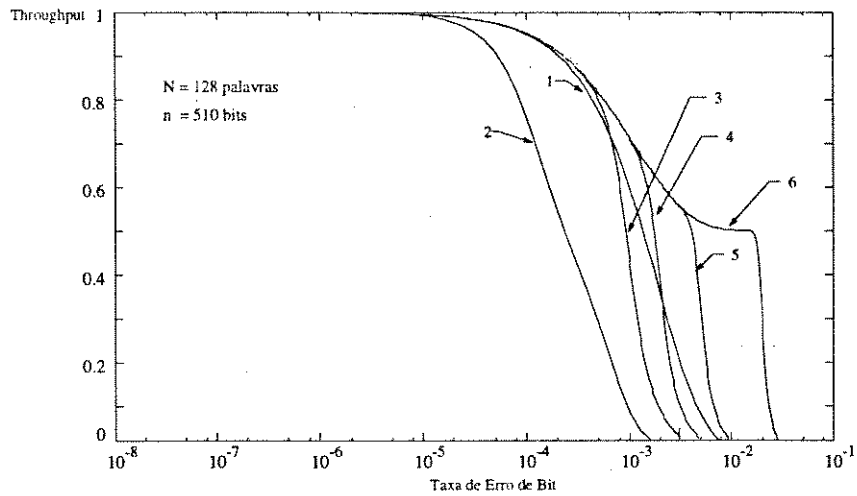


Figura 3.7: Throughput do esquema ARQ selective-repeat tipo-II modificado com tamanho de bloco  $n = 510$  bits e tempo de atraso de percurso  $N = 128$  palavras: (1) ARQ selective-repeat ideal com buffer do receptor de tamanho infinito; (2) Esquema ARQ selective-repeat básico com buffer de tamanho  $N$ ; (3), (4), (5) e (6) Esquemas ARQ selective-repeat híbrido tipo-II modificado com buffer de tamanho  $N$  e capacidade de correção de erro projetada  $t = 3, 5, 10$  e  $30$ , respectivamente.

tipo-II modificado, permanece com um valor alto (em torno de 50%), mesmo para altas taxas de erro de bit.

A confiabilidade do sistema ARQ selective-repeat híbrido tipo-II modificado, é igual a do esquema ARQ selective-repeat básico utilizando o mesmo código  $C_0$  para a detecção de erro, e é dada por [Wang, 1983]:

$$P(E)_{HM} = \frac{P_e}{P_e + P_c} \tag{3.25}$$

### 3.4.3 Esquema ARQ Híbrido Tipo-II Generalizado - ARQ-GH

Este esquema [Krishna, 1987] é uma generalização do esquema ARQ híbrido tipo-II modificado. A primeira retransmissão é feita utilizando a palavra de paridade formada por um código corretor de erro  $(2n, n)$  e pela palavra de informação. Na segunda retransmissão, não é utilizada a palavra de informação, sendo usada uma outra palavra de paridade formada por um código corretor de erro  $(3n, n)$ . Este esquema, pode ser generalizado para qualquer número de retrans-

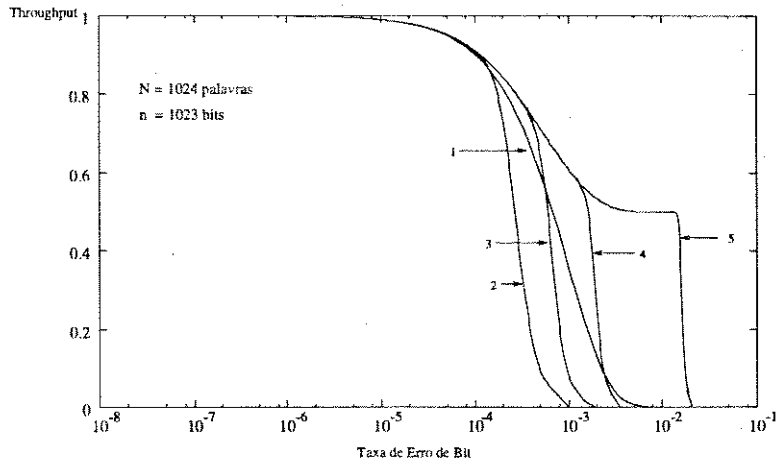


Figura 3.8: Throughput do esquema ARQ selective-repeat tipo-II modificado com  $n = 1023$  bits e tempo de atraso de percurso  $N = 1024$  palavras: (1) ARQ selective-repeat ideal com buffer do receptor de tamanho infinito; (2), (3), (4) e (5) Esquemas ARQ híbrido tipo-II modificado com  $t = 3, 5, 10$  e  $50$ , respectivamente.

missões de palavras de paridade, antes de retransmitir novamente a palavra de informação.

O esquema generalizado, também utiliza dois códigos: o primeiro é um código  $(n, k)$  de alta taxa  $C_0$  projetado somente para a detecção de erro, e o segundo é um código  $C_1$  usado adaptativamente para a correção de erro. O código  $(mn, n)$  corretor de erro  $C_1$ , com distância de Hamming mínima  $d$ , é selecionado de forma que sua matriz geradora possa ser particionada em  $m$  submatrizes, de dimensão  $(n \times n)$ . O inteiro  $m$  é definido como a *profundidade* do código. Seja  $G$  a matriz geradora do código  $C_1$  e  $G_1, G_2, \dots, G_m$  as submatrizes que compõem  $G$ . Então,  $G$  pode ser escrita como:

$$G = [G_1|G_2|\dots|G_m] \tag{3.26}$$

O subcódigo  $C_1^{(i)}$  com matriz geradora  $G^i = [G_1|G_2|\dots|G_i]$ , tem distância mínima  $d_i$ , tal que  $d_i < d_j$  para todo  $1 \leq i < j \leq m$ . A profundidade do subcódigo  $C_1^{(i)}$  é  $i$  por definição. Note que o código  $C_1^{(m)}$  é igual a  $C_1$ .

Este código pode ser incorporado em um esquema ARQ-GH da seguinte maneira. Seja  $I$  a palavra de informação formada a partir da mensagem  $D$  e do código  $C_0$ . A palavra-código de comprimento de  $mn$  bits é formada usando  $I$  e o código  $C_1 (mn, n)$ . Esta palavra-código é

representada por  $\mathbf{c} = (c_1|c_2|\dots|c_m)$ , onde cada segmento  $c_i$  da palavra  $\mathbf{c}$  corresponde a uma matriz geradora  $G_i$ ,  $i = 1, 2, \dots, m$ . Cada  $c_i$  tem tamanho igual a  $n$ . A palavra de informação  $I$  pode ser unicamente obtida a partir de  $c_i$ , se e somente se, a matriz  $G_i$  ( $n \times n$ ) correspondente for inversível. A matriz  $G_1$  deve ser inversível para que a palavra de informação possa ser obtida a partir de  $c_1$ . É desejável (mas não necessário) que as matrizes  $G_i$  ( $n \times n$ ),  $i = 2, \dots, m$  sejam inversíveis. Esta condição, é particularmente importante, quando erros em surtos destroem uma das transmissões, deixando as outras transmissões livres de erros.

Na transmissão de uma dada palavra de informação  $I$ , a sequência de palavras que são enviadas para o receptor, até que haja sucesso na recepção, é a sequência de palavras-código:  $c_1, c_2, \dots, c_m, c_1, c_2, \dots, c_m, c_1, \dots$ . Ao receber uma palavra  $c_i$ , o receptor adota uma das duas estratégias:

1. Inverte  $c_i$  (se a matriz associada  $G_i$  for inversível); verifica sua síndrome baseada em  $C_0$ . Se for necessário, decodifica usando  $c_i$  e as outras palavras recebidas anteriormente; executa a detecção de erro novamente (baseada em  $C_0$ ), ou
2. Decodifica usando  $c_i$  e as palavras recebidas anteriormente; executa a detecção de erro baseada em  $C_0$ .

Quando não for detectada a presença de erros em uma palavra, é enviado um ACK correspondente a esta palavra para o transmissor. Para a sequência de palavras transmitida acima, o receptor executa a correção de erro, baseada nos códigos  $C_1^{(2)}, C_1^{(3)}, \dots, C_1^m, C_1^m, \dots$ , tendo as distâncias  $d_2, d_3, \dots, d, d, \dots$ . Portanto, após cada retransmissão, um código com distância maior, isto é, com capacidade de correção de erro maior, é usado para a correção de erro, até chegar no código  $C_1^m$  (que é igual à  $C_1$ ). Note que o esquema ARQ híbrido tipo-II modificado é um caso especial do ARQ-GH para  $m = 2$ .

Para a transmissão de uma palavra-código de informação  $I$ , o transmissor computa uma palavra-código de paridade de tamanho  $mn$ , baseada em  $C_1$ . Então, tanto no transmissor como no receptor, é necessário um buffer de tamanho  $mn$  bits para cada palavra a ser transmitida. O transmissor necessita somente de dois codificadores; um baseado em  $C_1$  e outro em  $C_0$ . Já

o receptor requer um decodificador para cada um dos códigos  $C_1^{(2)}, \dots, C_1^{(m-1)}, C_1$ . Também é necessário um circuito inversor para cada  $c_i, i = 1, 2, \dots, m$ .

A complexidade do sistema ARQ-GH pode parecer proibitiva. Especialmente, na configuração do receptor, que necessita de  $m$  decodificadores. Se o procedimento de decodificação não for o mesmo para todos os códigos:  $C_1^{(2)}, C_1^{(3)}, \dots, C_1^{(m-1)}, C_1$ , o custo de ter  $(m - 1)$  circuitos decodificadores diferentes no receptor, pode reduzir o desempenho do sistema. Mas se o procedimento de decodificação for o mesmo para todos os códigos, o esquema ARQ-GH pode ter vantagens significativas sobre sistemas ARQ híbridos tipo-II modificado e sobre o original. Existe uma classe de códigos lineares denominados de códigos KM, que pode ser usada no esquema ARQ-GH, reduzindo a complexidade do processo de decodificação (vide apêndice B).

### Um Esquema ARQ-GH Baseado em Códigos KM

Será descrito nesta seção, um esquema ARQ-GH que utiliza códigos KM. Para uma melhor compreensão, este esquema será analisado por meio de um código KM específico.

Seja o código KM binário (12,4,5), então  $N = k+d-1 = 8$ . Supõe-se  $P(u) = P_1(u)P_2(u)P_3(u) = u^2(u^2 + 1)(u^2 + u + 1)$  e  $s = 2$  ( $s$  é um inteiro tal que  $s$  adicionado ao grau do polinômio  $P(u)$  resulta no mesmo valor de  $N$ ). A matriz geradora deste código é dada por

$$\begin{aligned}
 M &= \left[ \begin{array}{cccc|cccc|cccc}
 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1
 \end{array} \right] \\
 &= [M_1|M_2|M_3] \tag{3.27}
 \end{aligned}$$

A distância mínima deste código é 6 e a distância mínima projetada é 5. As matrizes  $M_1$ ,  $M_2$  e  $M_3$  são inversíveis. Usando um código  $C_0$  para detecção de erro e definindo três matrizes de dimensão  $(n \times n)$  como:

$$G_i = M_i \otimes I_{n/4}, \quad i = 1, 2, 3. \tag{3.28}$$

onde  $I_{n/4}$  é a matriz identidade  $n/4$  e o símbolo  $\otimes$  representa o produto Kronecker de duas matrizes [Therrian, 1984]. Se a matriz  $M_i, i = 1, 2, 3$  for inversível, a matriz  $G_i, i = 1, 2, 3$  também será com  $G_i^{-1}$  dada por:

$$G_i^{-1} = M_i^{-1} \otimes I_{n/4}, \quad i = 1, 2, 3. \quad (3.29)$$

Seja  $l$  um vetor correspondente à palavra de informação  $I$  a ser transmitida. O transmissor computa três palavras  $c_i^T = l^T G_i, i = 1, 2, 3$ . Como as matrizes  $G_i, i = 1, 2, 3$  são inversíveis, a palavra-código  $I$ , pode ser unicamente obtida a partir de uma das palavras-código  $c_i, i = 1, 2, 3$ . Sejam  $I_1, I_2$  e  $I_3$  as palavras associadas com  $c_1, c_2$  e  $c_3$ , respectivamente. Seja  $\tilde{I}_i$  a palavra recebida correspondente à  $I_i$  e  $E_i$  a palavra estimada de  $I_i$  baseada em  $\tilde{I}_i$  e em  $G_i, i = 1, 2, 3$ . Se não houver erros durante a transmissão,  $\tilde{I}_i = I_i$  e, conseqüentemente  $E_i = I_i, i = 1, 2, 3$ .

A análise do throughput para o esquema ARQ-GH, é feita considerando a capacidade de armazenamento do buffer do receptor infinita. O código  $C_1$  tem profundidade  $m = 3$ . A matriz geradora deste código, tem a forma  $G = [G_1|G_2|G_3]$ . O código  $C_1$  é derivado de um código KM  $(3l', l')$ , com capacidade de correção  $t_1$ . O código  $C_1^{(2)}$  é obtido do código  $(2l', l')$  com matriz geradora  $G = [G_1|G_2]$ , com capacidade de corrigir  $t_2$  erros. Observe que  $t_1 > t_2$ .

Considere um sistema que utiliza o código  $C_1^{(2)}$  para a correção de erros. O throughput deste sistema é limitado inferiormente por:

$$\eta_{GH_1} \geq \frac{P_c + P_t - P_c P_t}{2 - P_c} \left( \frac{k}{n} \right) \quad (3.30)$$

onde  $P_c$  é dado pela equação 3.12, sendo  $q_1$  dado pela equação 3.13 e  $q_0$  e  $y$  dados por:

$$q_0 = \left[ \sum_{j=0}^{t_2} \binom{2l'}{j} p^j (1-p)^{2l'-j} \right]^{n/l'} \quad (3.31)$$

$$y = (1-p)^n \left\{ 2 \left[ \sum_{j=0}^{t_2} \binom{l'}{j} p^j (1-p)^{l'-j} \right]^{n/l'} - (1-p)^n \right\} \quad (3.32)$$

Como exemplo, se  $C_1^{(2)}$  for um código  $(8,4,3)$ , tem-se que  $l' = 4$  e  $t_2 = 1$ .

Considere agora, um outro sistema que utiliza o código  $C_1$ . O throughput deste sistema é limitado inferiormente por:

$$\eta_{GH_2} \geq \frac{(1 - a_1)^2}{[P_c(1 + 2a_1) + a_2(2 + a_1) + 3a_3]} \left(\frac{k}{n}\right) \quad (3.33)$$

onde  $a_1 = (1 - P_c)^2$ ,  $a_2 = (1 - P_c)P_c$ , e  $a_3 = (1 - P_c)^2P_c$

As probabilidades  $q_0$  e  $y$  são dadas por:

$$q_0 = \left[ \sum_{j=0}^{t_1} \binom{3l'}{j} p^j (1-p)^{3l'-j} \right]^{n/l'} \quad (3.34)$$

$$y = (1-p)^n \left\{ 3 \left[ \sum_{j=0}^{t_1} \binom{2l'}{j} p^j (1-p)^{2l'-j} \right]^{n/l'} - 3(1-p)^n \left[ \sum_{j=0}^{t_1} \binom{l'}{j} p^j (1-p)^{l'-j} \right]^{n/l'} - 5(1-p)^{2n} \right\} \quad (3.35)$$

A expressão de  $q_1$  é a mesma da equação 3.13. Se  $C_1$  for um código (12,4,5), tem-se que  $l' = 4$  e  $t_1 = 2$ . A figura 3.9 mostra o throughput destes dois sistemas.

A figura 3.9, mostra que os esquemas ARQ-GH utilizando códigos KM apresentam desempenhos superiores ao esquema ARQ selective-repeat ideal. E à medida que a capacidade de correção do código aumenta, obtém-se valores significativos do throughput, para canais com altas taxas de erro de bits.

Para um sistema ARQ-GH possuir a mesma confiabilidade de um esquema ARQ básico, é necessário que a probabilidade de não detectar erro especificada pela matriz de verificação de paridade  $\tilde{H}_i^T = G_i^{-1} H^T$  satisfaça a equação 1.5 do capítulo 1.

### 3.4.4 Esquema ARQ Híbrido Usando um Sistema de Codificação em Cascata

Os sistemas de controle adaptativos de erro são uma boa solução, quando se tem canais variantes com o tempo. Estes sistemas variam a capacidade corretora do código corretor de erro de

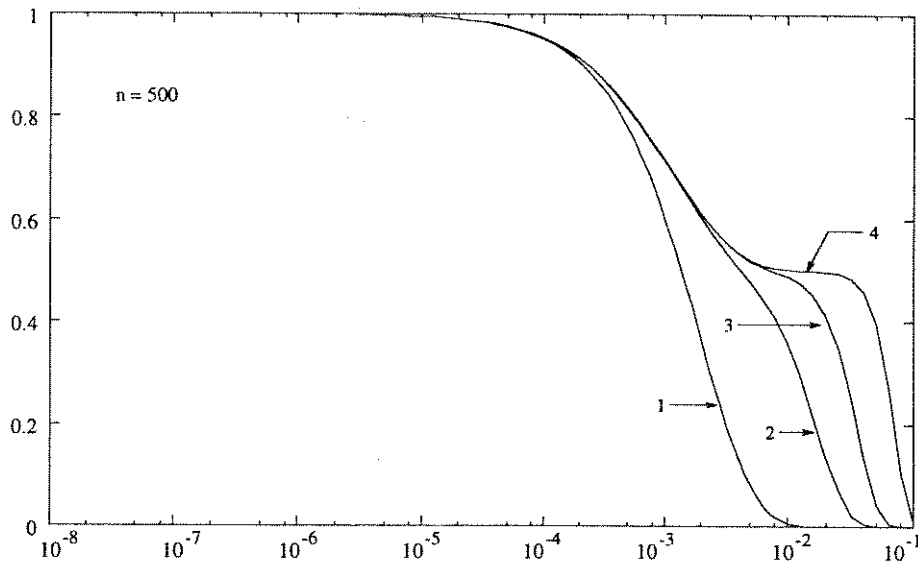


Figura 3.9: Throughput dos esquemas ARQ-GH: (1) Selective-repeat ideal; (2) ARQ-GH utilizando um código KM (8,4,3); (3) ARQ-GH utilizando um código KM (12,4,5); (4) ARQ-GH utilizando um código KM (27,7,9).

acordo com as condições do canal. O sistema ARQ híbrido adaptativo descrito nesta subseção [Kousa, 1991], utiliza códigos de Hamming em cascata. A principal característica deste sistema, é que o receptor utiliza o mesmo decodificador para processar os blocos transmitidos e como resultado, obtém um ótimo desempenho com um mínimo de complexidade.

### Códigos de Hamming em Cascata

Considere uma sequência de  $n$  bits não codificada cujos  $n$  bits de informação, estão no nível zero de cascadeamento.

- Codificador

Um código de Hamming ( $N, K$ ) é utilizado para a correção de erro. Os  $n$  bits de informação são divididos em grupos de  $K$  bits, onde cada grupo de  $K$  bits são mapeados em  $N$  bits pelo código de Hamming ( $N, K$ ). O número total de bits neste estágio é  $(n/K)N$ . Este estágio é referido como o primeiro nível do cascadeamento. O segundo nível do cascadeamento é obtido, dividindo os  $(n/K)N$  bits em grupos de  $K$  bits e mapeando em  $N$  bits novamente pelo mesmo



codificador. Ao término deste estágio, haverá um total de  $n(N/K)^2$  bits. Continuando este processo, no  $i$ -ésimo nível, haverá um total de  $n(N/K)^i$  bits. A figura 3.10 ilustra este processo para dois níveis de cascadeamento. Para obter um número inteiro de subblocos após cada divisão por  $K$  em qualquer estágio, a sequência de bits inicial  $n$  deve ter tamanho  $K^m$  ou qualquer inteiro múltiplo de  $K^m$ , onde  $m$  é o nível máximo de cascadeamento desejado.

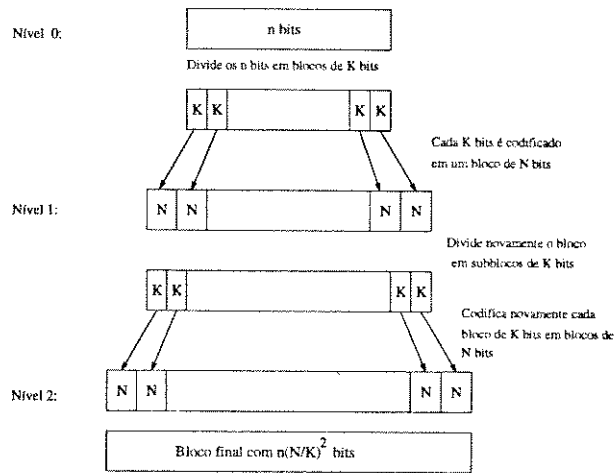


Figura 3.10: Codificação para dois níveis de cascata

- Decodificador

Suponha que o receptor receba uma sequência de bits, para a qual se conhece o nível  $m$  de cascadeamento. Seja  $p$  a taxa de erro de bit do canal. O receptor considera a sequência de bits recebida como um bloco de  $N$  bits e computa a síndrome de cada bloco para executar a correção de erro. Após completar a correção de primeira ordem no nível  $m$ , o receptor descarta os  $N - K$  bits de paridade, conservando apenas os  $K$  bits de informação, mas agora com uma probabilidade de erro de bit diferente, ou seja  $p'$ . A seguir o decodificador passa para o  $(m - 1)$  nível de cascadeamento. O receptor novamente divide estes bits em blocos de  $N$  bits cada, verifica os erros em cada bloco e descarta os bits de paridade. Desta forma, um outro nível de cascata é decodificado, com probabilidade de erro de bit igual a  $p''$ . Este processo continua até o receptor decodificar a mensagem de informação  $n$ . A figura 3.11 mostra o processo de decodificação para dois níveis de cascadeamento.

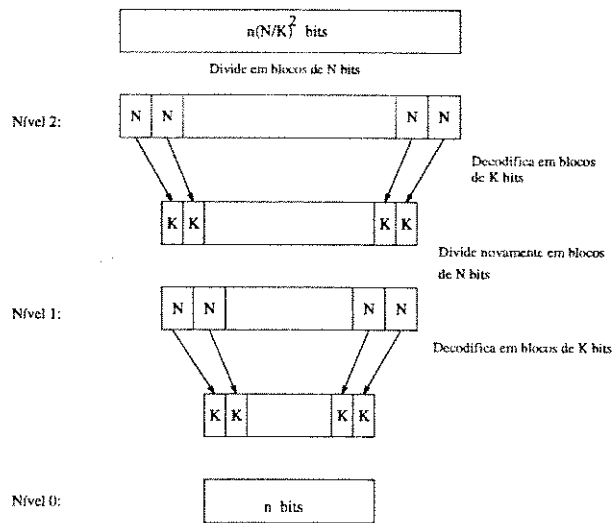


Figura 3.11: Decodificação para dois níveis de cascata

A probabilidade de erro de bit após a decodificação ( $p'$ ) para um código de Hamming ( $N, K$ ), onde a taxa de erro de bits do canal é  $p$ , é dada por [Kousa, 1991]:

$$p' = \frac{1}{N} \sum_{i=0}^N \left[ (i+1) \binom{N}{i} - A_i - 2A_{i-1}(N-i+1) \right] \cdot p^i (1-p)^{N-i} \quad (3.36)$$

onde  $A_i$  é o número de palavras-código de peso  $i$ . A figura 3.12 mostra  $p'$  versus  $p$  para os códigos de Hamming (7,4) e (15,11), mostrando uma redução considerável da taxa de erro de bit devido à codificação.

Se a escolha dos  $K$  bits para um subbloco for cuidadosamente elaborada, então, a cada codificação em cascata, resulta em um sistema com uma capacidade de correção de erro maior. Uma escolha mais simples para o segundo nível do cascadeamento, seria tomar  $K$  bits adjacentes por vez como um bloco. Entretanto, isto não pode ser feito pois a decodificação de segunda ordem pode nunca resultar em um decisão correta a menos que, a sequência esteja livre de erros ou contenha exatamente um erro por bloco. Por outro lado, se houver erros após a decodificação de primeira ordem (quando o canal introduz mais de um erro por bloco), a maioria destes erros deixados, ficarão confinados à seção de informação e a decodificação de segunda ordem pode jamais corrigi-los. Uma solução a este problema, seria o entrelaçamento dos bits de um bloco. No transmissor, para executar a codificação no segundo nível, cada  $K$

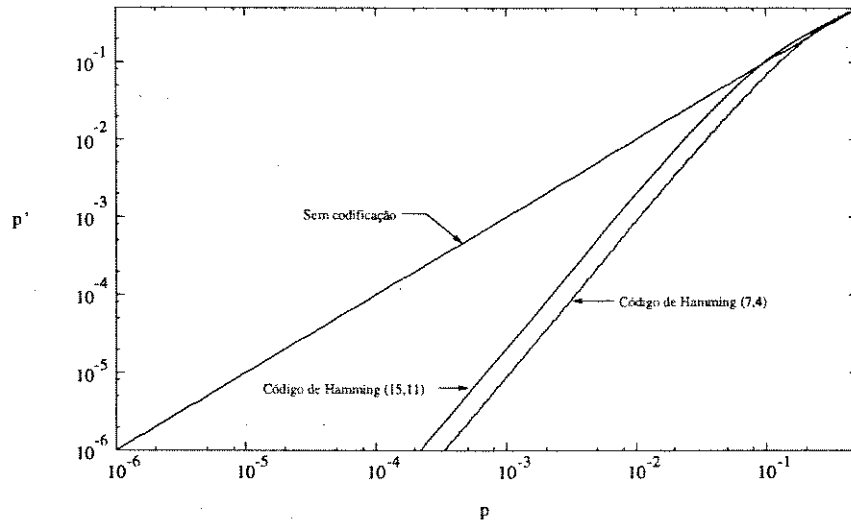


Figura 3.12: Taxa de erro de bit decodificada comparada à taxa de erro de bit do canal para dois códigos de Hamming

bits seriam escolhidos de  $K$  diferentes blocos do primeiro nível de codificação. A utilização do processo de entrelaçamento na codificação, garante que os erros de bit em cada bloco após o desentrelaçamento, sejam estatisticamente independentes, pois cada bit decodificado vem de um bloco diferente. Portanto, a probabilidade de erro de bit com dois níveis de decodificação é dada por:

$$p'' = \frac{1}{N} \sum_{i=1}^N \left[ (i+1) \binom{N}{i} - A_i - 2A_{i-1}(N-i+1) \right] \cdot p'^i (1-p')^{N-i} \quad (3.37)$$

onde  $p'$  é dado pela equação 3.36. A figura 3.13 mostra o entrelaçamento de um código de Hamming, para três níveis de cascata. No receptor, deve ser feita uma operação de desentrelaçamento análoga.

Pode-se agora, obter a probabilidade de decodificação correta de um sistema em cascata. Se  $K$  bits forem enviados através de um canal BSC tendo uma taxa de erro de bit  $p$ , então a probabilidade de recebê-los corretamente será igual a  $(1-p)^K$ .

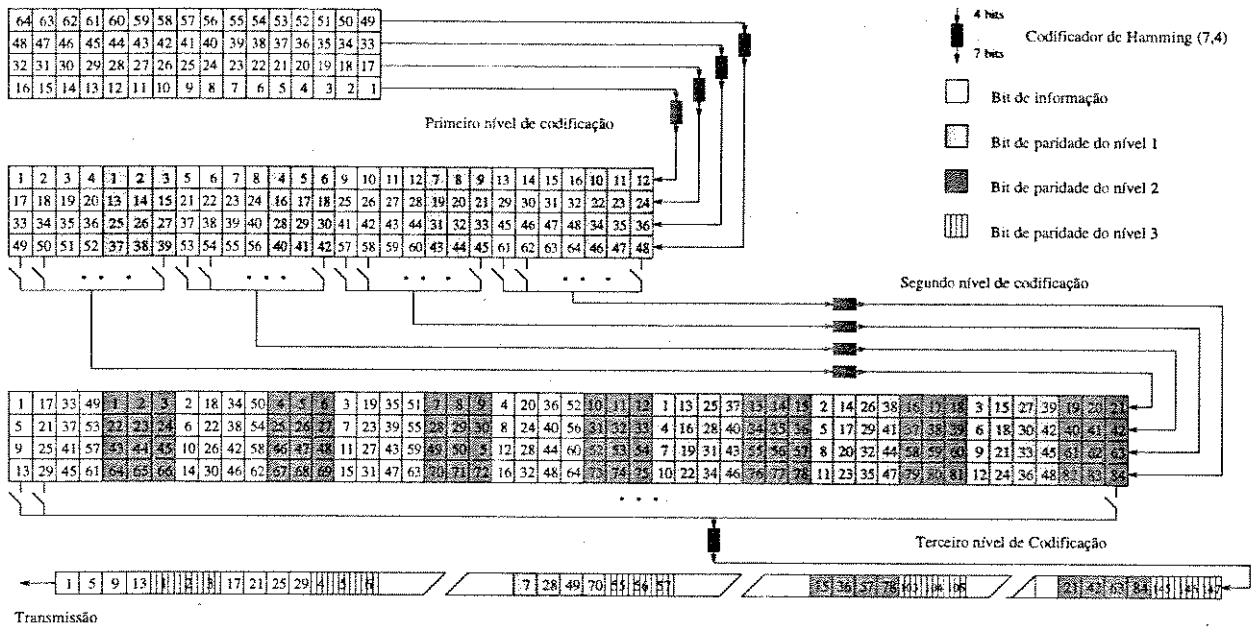


Figura 3.13: Entrelaçamento para três níveis de cascata usando um código de Hamming (7,4)

Se for empregado um código de Hamming, a probabilidade de recebê-los corretamente <sup>4</sup> é dada por  $(1 - p)^N + Np(1 - p)^{N-1}$ .

Agora, suponha que seja usado dois níveis de cascadeamento. No receptor, após o primeiro passo de decodificação, a probabilidade de erro de bit se reduzirá a  $p'$ . Devido ao entrelaçamento, os  $N$  bits são independentes. A probabilidade de decodificação correta de um bloco após o segundo passo de decodificação é dada por  $(1 - p')^N + Np'(1 - p')^{N-1}$ .

Se três níveis de cascadeamento forem usados, então a probabilidade de decodificação correta de um bloco de  $N$  bits é dada por  $(1 - p'')^N + Np''(1 - p'')^{N-1}$ , onde  $p''$  é obtido da equação 3.37. Esta técnica se aplica a qualquer nível de cascadeamento sob as condições de um entrelaçamento perfeito. Note que o tamanho do bloco é de  $N$  bits. Em qualquer nível de cascadeamento, deve haver  $n/K$  blocos de  $N$  bits a ser decodificado. Seja  $P_{c,i}$  a probabilidade de decodificação correta do superbloco (o bloco de  $n/K$  subbloco), quando os bits são codificados no nível  $i$ .

$${}^4P_{dec} = \sum_{i=0}^t \binom{N}{i} p^i (1 - p)^{N-i}$$

Portanto, para  $m = 1$ , (número de níveis de cascadeamento igual a 1) tem-se:

$$P_{c,1} = \left[ (1-p)^N + Np(1-p)^{N-1} \right]^{n/K} \quad (3.38)$$

Isto segue do fato que todos os  $n/K$  subblocos são independentes. Mas para  $m > 1$ , embora os  $N$  bits em um bloco sejam independentes, os bits são dependentes em diferentes blocos devido a etapas de decodificações anteriores. A probabilidade de decodificação correta do superbloco, não pode ser obtida simplesmente multiplicando as probabilidades individuais dos subblocos. Uma solução para este problema, seria utilizar o limitante de união (union bound).

Seja

$p_c \equiv$  probabilidade de decodificação correta de um subbloco

$P_C \equiv$  probabilidade de decodificação correta de um superbloco

$p_e \equiv$  probabilidade de decodificação incorreta de um subbloco

$P_E \equiv$  probabilidade de decisão incorreta de um superbloco, onde

$$P_E \leq p_e + p_e + \dots + p_e = (n/K)p_e \quad (3.39)$$

e portanto  $P_C \geq 1 - P_E$ .

Aplicando a limitante de união para o caso de  $i = 2$ , tem-se:

$$P_{c,2} \geq 1 - \frac{n}{K} \left\{ 1 - \left[ (1-p')^N + Np'(1-p')^{N-1} \right] \right\} \quad (3.40)$$

A equação 3.40 depende de  $p_e$  e do número de subblocos  $n/K$ .

### Esquema ARQ Híbrido Adaptativo Baseado em Códigos de Hamming em Cascata

Na sequência inicial de  $k$  bits deve-se adicionar alguns bits de redundância para detecção de erro baseado no código  $C_0(n, k)$  na primeira transmissão. Quando o receptor detecta a presença de erros na palavra recebida, ele armazena esta palavra e solicita uma retransmissão. A retransmissão consiste de um superbloco composto de subblocos de bits de paridade, que é baseado na mensagem original e no código de Hamming no primeiro nível de codificação. Este superbloco é utilizado para corrigir os erros da palavra recebida com erro armazenada no buffer do receptor. Após a decodificação, o bloco decodificado é novamente verificado pelo código

$C_0$ . Se o bloco decodificado não estiver correto, o transmissor armazena os bits de paridade do primeiro nível e solicita uma segunda retransmissão. A segunda retransmissão é um outro superbloco composto de subblocos de bits de paridade formado pelo mesmo código corretor de erro, baseado na mensagem original e nos bits de paridade da primeira retransmissão. É feito um entrelaçamento apropriado antes da codificação e o desentrelaçamento após a decodificação. Após o recebimento deste superbloco de segundo nível, o receptor tenta novamente corrigir a palavra armazenada no buffer. Este processo é repetido se necessário, até transmitir o  $m$ -ésimo nível de superblocos de paridade. Se após a  $m$ -ésima retransmissão, a palavra decodificada ainda estiver com erro, o transmissor envia novamente o bloco de informação. O receptor substitui o bloco de informação antigo por este novo bloco recebido e tenta decodificar a mensagem com ajuda de todos os bits de paridade disponíveis. Se a decodificação estiver incorreta, o transmissor envia novamente o superbloco de bits de paridade formado a partir do primeiro nível de cascata. Se houver necessidade de uma outra retransmissão, a próxima sequência será o superbloco de segundo nível e assim sucessivamente. Com isso, a sequência de retransmissões para sucessivos NAK's será  $I, P_1, P_2, \dots, P_m, I, P_1, \dots$ , onde  $I$  é a mensagem original de  $n$  bits e  $P_i$  é o  $i$ -ésimo superbloco de paridade. Deve ser notado que o sistema transmite blocos de diferentes tamanhos, podendo ocasionar um aumento na complexidade de transmissão.

A análise do throughput é feita para um sistema com dois níveis de cascata ( $m = 2$ ), sendo definido por:

$$\eta_{HC} = \frac{k}{T_{HC}} \quad (3.41)$$

onde  $k$  é o número de bits de informação e  $T_{HC}$  é o número médio de transmissões dado por:

$$\begin{aligned} T_{HC} = n \left\{ & P_{c,0} + \left[ 1 + \frac{N-K}{K} \right] P_r(E_{d,0}E_{c,1}) \right. \\ & + \left[ 1 + \frac{N-K}{K} + \frac{N-K}{K} \left( \frac{N}{K} \right) \right] P_r(E_{d,0}E_{d,1}E_{c,2}) \\ & + \left[ 1 + \frac{x}{K} + \frac{N-K}{K} \left( \frac{N}{K} \right) \right] P_r(E_{d,0}E_{d,1}E_{d,2}) \\ & \left. + \left( \frac{N}{K} \right)^2 P_r(E_{d,0}E_{d,1}E_{d,2}) \left( \frac{1}{P_{c,2}} \right) \right\} \quad (3.42) \end{aligned}$$

$E_{c,i}$  é o evento no qual o receptor recupera a palavra corretamente;  $E_{d,i}$  é o evento no qual o receptor detecta a presença de erros e solicita uma próxima retransmissão; e  $E_{e,i}$  é o evento

onde o receptor não detecta a presença de erros. Analisando as probabilidades de um sistema com dois níveis em cascata ( $m = 2$ ), tem-se que:

1.  $P_{c,0} = P_c = (1 - p)^n$ .
2.  $P_r(E_{c,3}) = P_r(E_{c,4}) = \dots = P_{c,2}$  e conseqüentemente:  $P_r(E_{d,3}) = P_r(E_{d,4}) = \dots = 1 - P_{c,2} = P_{d,2}$
3.  $P_r(E_{d,0}E_{c,1}) = P_r(E_{c,1}|E_{d,0})P_r(E_{d,0})$
4.  $P_r(E_{c,1}|E_{d,0})$  é a probabilidade de decodificação correta no primeiro nível, dado que foi detectado erro na palavra de informação, sendo dado por:

$$P_r(E_{c,1}|E_{d,0}) = \left\{ \frac{[(1-p)^N + Np(1-p)^{N-1}]^K - [(1-p)^N + (N-K)p(1-p)^{N-1}]^K}{1 - (1-p)^{K^2}} \right\}^B \quad (3.43)$$

Onde  $B$  é o número de blocos de informação de  $K^2$  bits que serão transmitidos.

5. A probabilidade conjunta  $P_r(E_{d,0}E_{d,1}E_{c,2})$  é definida como:

$$P_r(E_{d,0}E_{d,1}E_{c,2}) = P_r(E_{d,0})P_r(E_{d,1}|E_{d,0})P_r(E_{c,2}|E_{d,0}E_{d,1}). \quad (3.44)$$

onde o primeiro termo  $E_{d,0}$  é igual à  $(1 - P_{c,0})$ . O segundo termo  $P_r(E_{d,1}|E_{d,0})$  é dado por  $(1 - P_r(E_{c,1}|E_{d,0}))$ . E o terceiro termo:

$$P_r(E_{c,2}|E_{d,0}E_{d,1}) > \left\{ \frac{P_{c,2} - P_{c,1}}{1 - P_{c,1}} \right\}^B \quad (3.45)$$

6. A outra probabilidade conjunta  $P_r(E_{d,0}E_{d,1}E_{d,2})$  é definida como:

$$P_r(E_{d,0}E_{d,1}E_{d,2}) = P_r(E_{d,0})P_r(E_{d,1}|E_{d,0})P_r(E_{d,2}|E_{d,0}E_{d,1}) \quad (3.46)$$

onde:

$$\begin{aligned} P_r(E_{d,0}E_{d,1}E_{c,2}) &= 1 - P_r(E_{c,2}|E_{d,0}E_{d,1}) \\ &< 1 - \left\{ \frac{P_{c,2} - P_{c,1}}{1 - P_{c,1}} \right\}^B \end{aligned} \quad (3.47)$$

A figura 3.14 mostra o throughput de vários esquemas ARQ híbridos usando diferentes códigos de Hamming, com dois níveis de cascata ( $m = 2$ ). Observa-se que à medida que o valor de  $N$  do código de Hamming aumenta, obtém-se um melhor desempenho em termos de throughput.

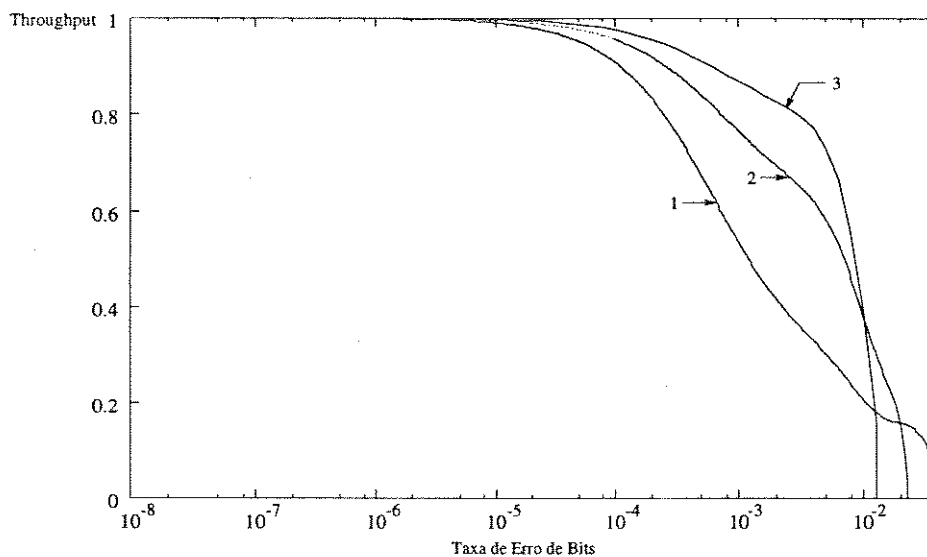


Figura 3.14: Throughput de três esquemas em cascata usando código de Hamming: (1) (7,4) e  $n = 1344$  bits; (2) (15,11) e  $n = 1331$  bits; (3) (31,26) e  $n = 1352$  bits.

Para se ter um sistema ARQ com alta confiabilidade, é necessário que o código de detecção de erro  $C_0$  seja escolhido de forma a minimizar a probabilidade  $P(E)$  (probabilidade do receptor cometer um erro) dada pela equação 1.3.

Para um sistema ARQ em cascata, com uma sequência inicial de  $K^m$  bits ou qualquer múltiplo deste número, existe algumas restrições com relação ao tamanho do código de detecção de erro usado neste sistema. Para satisfazer essas restrições, o código de detecção de erro, deve ser da forma de  $(aK^m, k)$ , onde  $k$  é o número de bits de informação e  $a$  é um número inteiro. Esta condição, geralmente é satisfeita pois, para a maioria dos canais práticos, apenas dois níveis de cascadeamento já são suficientes, e outros cascadeamentos adicionais não resultariam em nenhuma melhoria significativa. Se o tamanho da palavra-código do código de Hamming  $(N, K)$  não for muito grande,  $K^m$  poderá ser pequeno se comparado com o tamanho de um código de detecção de erro eficiente. Para satisfazer a condição acima, deve-se construir um



código detector de erro  $(aK^m, k)$  a partir de um inteiro múltiplo de  $K^m$ . Esta condição, pode às vezes ser satisfeita, mas não para a maioria dos casos. Para estes casos onde a condição não é satisfeita, usa-se o conceito de códigos estendidos ou encurtados. [Berlekamp, 1968]. Para a construção do código de detecção, procede-se da seguinte forma: escolha  $a$ , tal que  $aK^m$  esteja próximo de  $n$ , que é o tamanho do código de detecção de erro  $C_0$ . Surge com isso, dois casos:

1.  $aK^m < n$ : nesta situação, encurta-se o código  $C_0$ , omitindo  $(n - aK^m)$  bits de mensagem. Desta forma, o código  $(n, k)$  é modificado para o código  $(aK^m, k - n + aK^m)$ .
2.  $aK^m > n$ : para este caso, estende-se o código  $C_0$  adicionando  $(aK^m - n)$  bits de paridade. e o código modificado será um código  $(aK^m, k)$ .

Para o primeiro caso, se o número de bits apagados for pequeno, a probabilidade  $P_e$  (probabilidade de uma palavra recebida conter um erro não detectável) do código encurtado pode ser praticamente da mesma ordem da probabilidade do código original. No segundo caso, a capacidade de detecção de erro é alterada. O código estendido é pelo menos tão eficiente quanto o código original. O custo de se aumentar o código é a diminuição da taxa do código. Mas se  $n$  for suficientemente grande, que geralmente é o caso, e se  $aK^m$  for aproximadamente igual à  $n$ , a redução da taxa é praticamente insignificante.

Agora, são mostrados algumas possíveis escolhas do código de detecção de erro para dois níveis de cascata ( $m = 2$ ):

1. Código de Hamming (7,4) com  $a = 128$ . Todos os códigos BCH primitivos estendidos de tamanho  $2^l$ ,  $l \geq 4$ , são convenientes.
2. Código de Hamming (15,11) com  $a = 17$ . Um tamanho ideal para o código é de 2057.
3. Código de Hamming (31,26) com  $a = 6$ .  $n$  deve ser da ordem de 4056.

Portanto, a confiabilidade deste sistema depende das probabilidades de não se detectar erro em todos os níveis de cascata e do código de detecção  $C_0$ , e a escolha apropriada do código  $C_0$  é essencial para o desempenho do esquema ARQ híbrido em cascata.

### 3.5 Comparações Entre os Diversos Esquemas ARQ Híbridos Selective-Repeat que Utilizam Códigos de Blocos

Nesta seção, são feitas várias comparações em termos de throughput, entre os diversos esquemas ARQ híbridos descritos anteriormente.

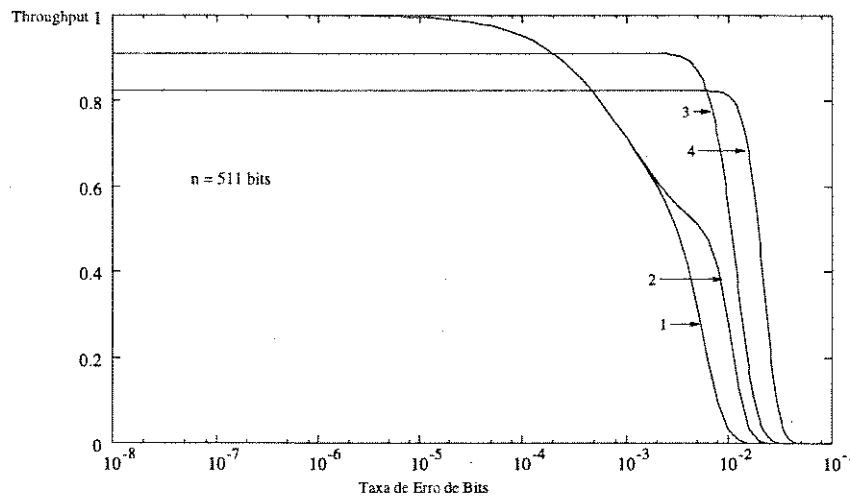


Figura 3.15: Comparação do throughput: (1) e (2) ARQ híbrido tipo-II modificado com  $t = 5$  e  $10$ , respectivamente; (3) e (4) ARQ híbrido tipo-I com  $t = 5$  e  $10$ , respectivamente.

A figura 3.15 compara os esquemas ARQ híbridos tipo-I e tipo-II modificado. Observa-se que; se a taxa de erro de bit do canal for menor que  $10^{-4}$ , obtém-se desempenhos melhores em termos de throughput, usando esquemas tipo-II, pois a taxa do código ( $k/n$ ) corretor de erro é muito menor no tipo-I (devido a um número maior de bits de paridade usado em cada transmissão). Mas, para uma taxa de erro acima de  $10^{-3}$ , o throughput do esquema tipo-II modificado reduz-se rapidamente, enquanto que o tipo-I ainda mantém um valor significativo. Para uma taxa de erro de bit em torno de  $10^{-2}$  o valor do throughput ainda se mantém praticamente igual a taxa do código, para  $t \geq 5$ , ao passo que, o desempenho do esquema ARQ híbrido tipo-II modificado, é muito baixo, se comparado com o tipo-I. Para obter um desempenho significativo, em termos de throughput, nos esquemas híbridos tipo-II modificado

para altas taxas de erro de bit, deve-se usar um código capaz de corrigir muitos erros, o que aumenta a complexidade do sistema.

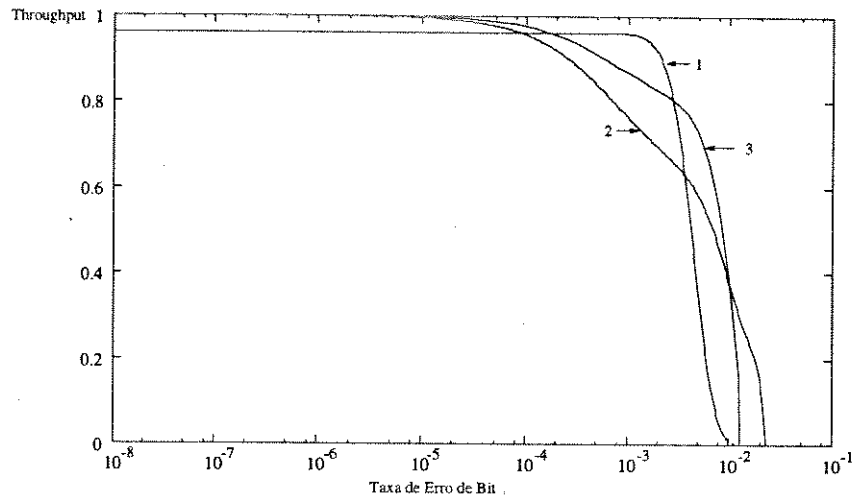


Figura 3.16: Comparação do throughput: (1) ARQ híbrido tipo-I com  $n = 1300$  bits e  $t = 5$ ; (2) e (3) ARQ híbrido tipo-II em cascata, utilizando código de Hamming (15,11) e (31,26), respectivamente.

A figura 3.16 faz uma comparação entre o esquema híbrido tipo-I utilizando uma capacidade de correção de erro igual a 5 bits e o esquema híbrido tipo-II que utiliza códigos de Hamming em cascata. Quando a taxa de erro de bit do canal é pequena ( $< 10^{-4}$ ), o esquema tipo-II em cascata possui um desempenho superior ao esquema híbrido tipo-I. Para uma taxa de erro de bit entre  $10^{-4}$  e  $\sim 2 \times 10^{-3}$ , o throughput do esquema tipo-I ainda se mantém, ao passo que o throughput do esquema tipo-II em cascata, começa a cair. Mas para uma taxa de erro de bit menor ou igual à  $10^{-3}$ , os esquemas em cascata usando códigos de Hamming (15,11) e (31,26), ainda possuem um desempenho significativo (throughput  $> 0.7$ ), embora sejam inferiores ao throughput do esquema tipo-I. Para uma taxa de erro de bit em torno de  $10^{-2}$  e  $n = 1300$  bits, o throughput do esquema tipo-I é praticamente nulo, ao passo que o esquema híbrido tipo-II em cascata ainda mantém um valor significativo, para os dois códigos de Hamming usados.

A figura 3.17 compara dois esquemas híbridos tipo-II: o esquema modificado para várias capacidades de correção de erro e o esquema ARQ-GH usando códigos KM. Analisando a figura, observa-se que o throughput mantém-se o mesmo para todos os esquemas quando a taxa de

erro de bits é menor ou igual a  $10^{-3}$ . As variações dos throughputs ocorrem apenas para taxas de erros maiores que  $10^{-3}$ . É necessário usar no esquema modificado, um código com grande capacidade corretora de erros ( $t > 10$ ), para se obter o mesmo desempenho do esquema ARQ-GH. Para uma taxa de erro de bit igual a  $10^{-2}$ , o throughput do esquema híbrido tipo-II modificado é nulo se a capacidade do código corretor de erro usado for menor que 10. Se o sistema modificado usar um código capaz de corrigir mais que 10 erros, consegue-se obter um valor de throughput acima de 0.4 para esta taxa de erro de bit. O esquema ARQ-GH ainda mantém um valor de throughput razoável para esta taxa, especialmente para o esquema ARQ-GH que utiliza o código KM (12, 4, 5).

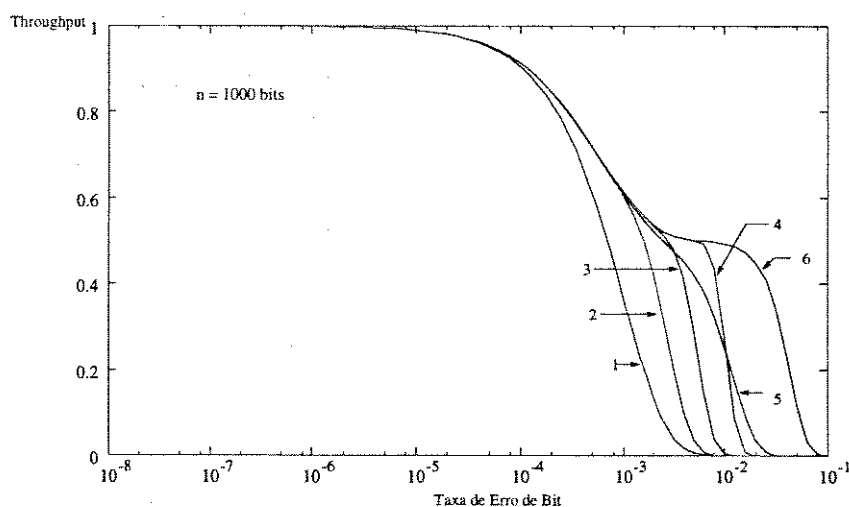


Figura 3.17: Comparação de Throughput: (1) ARQ selective-repeat ideal; (2), (3) e (4) ARQ híbrido modificado com  $t = 5, 10$  e  $20$ , respectivamente; (5) e (6) ARQ híbrido GH usando código KM (8,4,3) e (12,4,5), respectivamente.

A comparação entre o esquema híbrido tipo-II modificado e o esquema híbrido em cascata, é mostrado na figura 3.18. Os esquemas em cascata que utilizam códigos de Hamming (15,11) e (31,26), possuem um desempenho em termos de throughput, muito acima do esquema modificado, para uma taxa de erro de bit em torno de  $10^{-3}$ . Para uma taxa de erro na faixa de  $10^{-3}$  à  $10^{-2}$ , os sistemas em cascata com códigos de Hamming (15,11) e (31,26) ainda possuem desempenho superior aos esquemas modificado, mesmo quando se usa nestes esquemas, códigos capazes de corrigir vários erros ( $t \approx 40$ ). O esquema em cascata com código de Hamming (7,4),

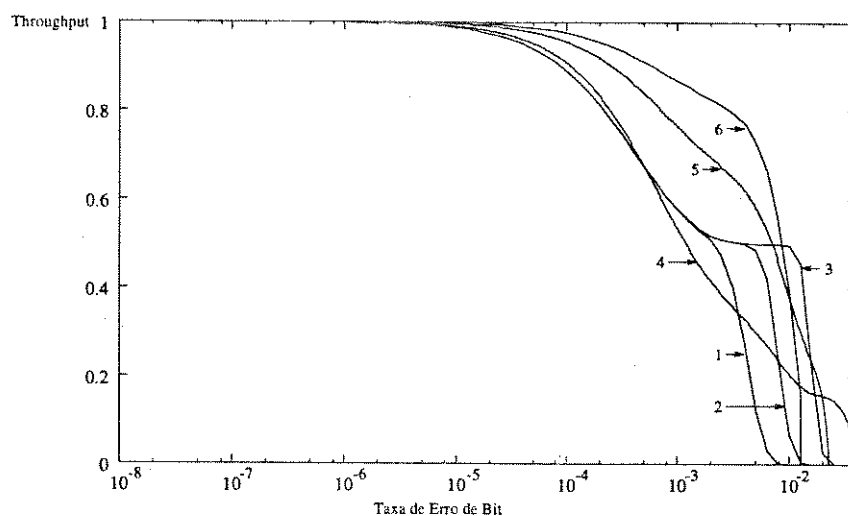


Figura 3.18: Comparação de Throughput: (1), (2) e (3) ARQ Híbrido tipo-II modificado com  $n = 1300$  e  $t = 10, 20$  e  $40$ , respectivamente; (4), (5) e (6) ARQ Híbrido tipo-II em cascata, com código de Hamming (7,4), (15,11) e (31,26), respectivamente.

possui um desempenho aproximadamente igual ao esquema modificado, para uma taxa de erro de bit menor que  $10^{-3}$ . Acima deste valor de taxa de erro de bit, o desempenho em termos de throughput deste esquema é inferior ao esquema modificado para  $t > 10$ .

A figura 3.19 compara os dois esquemas ARQ híbridos tipo-II, que apresentam os melhores desempenhos: o esquema em cascata e o GH. Para uma taxa de erro de bit menor que  $10^{-4}$ , estes esquemas apresentam o mesmo desempenho. Acima de  $10^{-4}$ , os throughputs dos esquemas em cascata com códigos de Hamming (7,4) e ARQ-GH, começam a reduzir, ao passo que, o throughput do esquema em cascata com códigos de Hamming (15,11) e (31,26) praticamente não se altera. Para uma taxa de erro de bit igual a  $10^{-3}$ , o valor do throughput do esquema em cascata com códigos de Hamming (15,11) e (31,26) é superior ao do ARQ-GH. Para uma taxa acima de  $10^{-2}$ , o esquema ARQ-GH com código KM (12,4,5) ainda mantém um valor de throughput em torno de 0.5. Para esta taxa de erro de bit, o throughput do esquema em cascata para os códigos de Hamming analisados, é praticamente nulo. O esquema híbrido em cascata com códigos de Hamming (7,4) possui um desempenho similar ao esquema ARQ-GH com código KM (8,4,3).

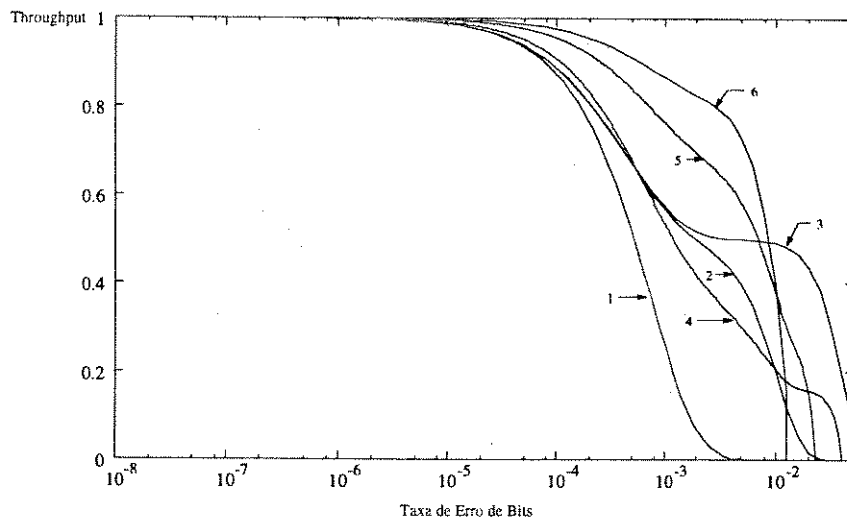


Figura 3.19: Comparação do throughput: (1) ARQ selective-repeat ideal; (2) e (3) ARQ híbrido GH usando código KM (8,4,3) e (12,4,5), respectivamente, (4), (5) e (6) ARQ híbrido em cascata usando código de Hamming (7,4), (15,11) e (31,26), respectivamente.

### 3.6 Conclusão

Os sistemas ARQ híbridos, são uma combinação dos sistemas ARQ e FEC. Como estes dois esquemas individualmente aplicados apresentam algumas deficiências, os sistemas ARQ híbridos surgiram como uma solução para contornar estas deficiências. Os esquemas ARQ híbridos são divididos em duas categorias denominadas: tipo-I e tipo-II. O ARQ híbrido tipo-I usa apenas um código para correção e detecção de erro. Como consequência, há um aumento do número de bits de paridade em cada palavra a ser transmitida, reduzindo portanto, o throughput, especialmente quando a taxa de erro de bit é muito baixa. Para contornar este problema, o esquema ARQ híbrido tipo-II trabalha como um esquema ARQ puro no processo de transmissão, enviando somente os bits de paridade necessários para a detecção de erro. Os bits de paridade que serão utilizados no processo de correção de erro, somente são enviados, quando o receptor solicitar uma retransmissão de uma palavra que tenha sido recebida com erro. Como resultado, se a taxa de erro de bit do canal estiver baixa, obtém-se um alto desempenho em termos de throughput. Quando o canal estiver ruidoso, os dois esquemas híbridos apresentam um desempenho significativo em termos de throughput, ao passo que, o throughput dos esquemas

ARQ básicos, deteriora-se rapidamente.

Os esquemas híbridos analisados neste capítulo, se concentraram nos esquemas tipo-II, por apresentarem um melhor desempenho que o tipo-I. Estes esquemas utilizam dois códigos; um para o processo de correção de erro e outro para a detecção de erro. Os esquemas híbridos tipo-II com buffer do receptor de tamanho finito (que segue o conceito básico do esquema híbrido tipo-II) e o esquema modificado, utilizam duas palavras-código no processo de transmissão/retransmissão: a palavra de paridade e a palavra de informação. A diferença entre estes dois esquemas está no código corretor de erro  $C_1$ . No esquema ARQ híbrido tipo-II,  $C_1$  é um código  $(2k, k)$  inversível de taxa meio (onde  $k$  é o número de bits de informação a ser codificado). Os  $k$  bits de paridade gerados pelo código  $C_1$ , são codificados em uma palavra de  $n$  bits baseada no código detector de erro  $C_0$ , formando a palavra de paridade a ser transmitida. No esquema modificado, o código  $C_1$  é um código  $(2n, n)$  inversível de taxa meio ( $n$  é o tamanho da palavra-código a ser transmitida). Os  $n$  bits gerados a partir de  $C_1$ , constitui a própria palavra de paridade a ser enviada. O código  $C_1$  no esquema modificado usa toda a sua capacidade de correção de erro. Já no esquema híbrido tipo-II básico, o código  $C_1$  é usado tanto para a correção como para a detecção de erro. Como consequência, a capacidade de correção do código, não é totalmente utilizada no processo de correção de erro. Pelo fato de utilizar toda a capacidade de correção de erro do código, o esquema ARQ híbrido tipo-II modificado, apresenta um desempenho superior ao esquema híbrido tipo-II original.

O esquema ARQ-GH, é uma generalização do esquema ARQ híbrido tipo-II modificado. Com a utilização dos códigos KM no esquema GH, a complexidade dos procedimentos de codificação/decodificação diminuem consideravelmente. O tamanho e a distância mínima dos códigos KM podem ser modificados sem alterar a configuração do conjunto codificador/decodificador e portanto, pode-se projetar um sistema ARQ-GH com código KM adaptativo para o controle de erros. A complexidade de implementação de um sistema ARQ-GH com códigos KM, pode ser menor que a de um sistema híbrido tipo-II original ou modificado que utiliza um código longo BCH capaz de corrigir vários padrões de erro.

O esquema híbrido tipo-II adaptativo que utiliza códigos de Hamming em cascata, possui desempenho superior ao esquema ARQ-GH pois, no sistema GH, toda vez que uma retransmis-

são é solicitada, o transmissor envia uma palavra de bits de paridade com o mesmo tamanho da palavra de informação. Como resultado, na primeira retransmissão, a taxa do código cai para  $1/2$ ; na segunda retransmissão, a taxa reduz-se a  $1/3$ , e assim sucessivamente. Isto é, uma redução significativa da taxa de transmissão da informação. Por outro lado, no esquema híbrido com códigos de Hamming em cascata, após a primeira retransmissão a taxa cai para  $(K/N)$ ; a taxa do código de Hamming que está sendo utilizado. A segunda retransmissão reduz a taxa para  $(K/N)^2$ , e assim sucessivamente. Se for utilizado um código de Hamming (31, 26), somente na quarta retransmissão é que a taxa será menor que  $1/2$ . Esta análise é feita considerando o esquema ARQ selective-repeat com buffer do receptor de tamanho infinito. Se for utilizado outro esquema ARQ básico ou utilizar buffer finito, esta análise pode não ser válida.

Há mais uma vantagem do esquema ARQ híbrido em cascata. Como existem códigos de Hamming com diferentes taxas, pode-se escolher um código de Hamming que se adapte melhor a um determinado canal. Somado a esta vantagem, tem-se que decodificadores para códigos de Hamming são muito simples de implementar.

Como desvantagem do esquema ARQ híbrido em cascata, há a necessidade de dois circuitos adicionais: um *entrelaçador* (interleaving) no transmissor e um *desentrelaçador* (de-interleaving) no receptor, com a finalidade de evitar surtos de erros. Estes dois circuitos não são necessários nos outros esquemas híbridos. Uma outra desvantagem, está associada ao tamanho variável dos blocos na retransmissão, ao passo que, nos outros sistemas híbridos, os blocos são do mesmo tamanho. A transmissão de blocos com mesmo tamanho são mais fáceis de tratar e requerem um número menor de bits de controle. Finalizando, a escolha do código de detecção de erro  $C_0$ , é menos flexível que nos outros esquemas híbridos, pois o tamanho  $n$  do código  $C_0$  deve ser igual a  $aK^m$ , onde  $a$  é um número inteiro,  $K$  é o número de bits a ser codificado pelo código de Hamming e  $m$  é o número de níveis de cascata.



## Capítulo 4

# ESQUEMAS ARQ HÍBRIDOS USANDO CÓDIGOS CONVOLUCIONAIS

### 4.1 Introdução

A maioria dos trabalhos na área de ARQ foram feitos utilizando códigos de blocos. Entretanto, os códigos convolucionais apresentam algumas características interessantes que possibilitam o seu uso em sistemas de requisição de repetição automática:

1. Os códigos convolucionais possuem ótima capacidade de correção e detecção de erros.
2. A decodificação do código convolucional é um processo sequencial e por isso, uma retransmissão pode ser solicitada, antes de terminar a decodificação de uma palavra-código. O que não é possível para códigos de blocos.
3. No processo de decodificação, se o número de erros ultrapassar a capacidade de correção do código e a decodificação da mensagem ainda não foi completada, o processo de decodificação é interrompido, e o receptor solicita somente a retransmissão da parte da mensagem que estava sendo decodificada até o momento, aumentando portanto, o throughput do sistema.

Este capítulo mostra a utilização de códigos convolucionais em esquemas ARQ híbridos no processo de correção de erros. Como os esquemas híbridos requerem códigos de baixas taxas ( $k/n$ ), os códigos convolucionais também podem ser utilizados. Estes códigos podem ser usados tanto nos esquemas ARQ híbridos tipo-I quanto no tipo-II. Nos esquemas híbridos tipo-II, eles são usados como códigos corretores de erros (código  $C_1$ ).

## 4.2 Esquema ARQ híbrido tipo-II usando código convolucional de taxa 1/2

Neste esquema [Wang, 1983], usa-se um código de bloco  $C_0$  ( $n, k$ ) somente para a detecção de erro. O código  $C_1$  usado no processo de correção de erro, é um código convolucional  $(2, 1, m)$  de taxa 1/2 e memória  $m$ . A decodificação é feita pelo algoritmo de Viterbi.

Seja  $G_1(X)$  e  $G_2(X)$  os dois polinômios geradores do código convolucional  $C_1$ . Quando a mensagem de informação de  $k$  bits estiver pronta para ser transmitida, primeiramente ela é codificada em uma palavra-código  $I(X)$  baseada no código detector de erro  $C_0$ . A palavra-código  $I(X)$  é então transformada em duas sequências:  $P_1(X) = I(X)G_1(X)$  e  $P_2(X) = I(X)G_2(X)$ , onde cada sequência tem comprimento de  $n + m$  bits. Observa-se que a sequência de  $2(n + m)$  bits, obtida pela intercalação de  $P_1(X)$  e  $P_2(X)$  é uma palavra-código formada a partir de  $I(X)$ , baseada no código convolucional de taxa 1/2. O processo de transmissão consiste em primeiramente enviar a sequência  $P_1(X)$ . A sequência  $P_2(X)$  é armazenada no buffer do transmissor para ser transmitida futuramente, se necessário.

Seja  $\tilde{P}_1(X)$  a sequência recebida correspondente à  $P_1(X)$ . No receptor,  $\tilde{P}_1(X)$  é dividida por  $G_1(X)$ . Considere  $\tilde{I}_1(X)$  e  $\tilde{R}_1(X)$  o quociente e o resto desta divisão, respectivamente. Se  $\tilde{R}_1(X)$  for igual a zero, verifica-se a síndrome de  $\tilde{I}_1(X)$  utilizando o código detector de erro  $C_0$ . Se a síndrome  $\tilde{S}_1(X)$  for zero, assume-se que  $\tilde{I}_1(X)$  não contém erros sendo idêntica à  $I(X)$ . O receptor aceita  $\tilde{I}_1(X)$  e solicita a transmissão da próxima sequência de informação. Se  $\tilde{R}_1(X)$  ou  $\tilde{S}_1(X)$  forem diferentes de zero, isto indica que a sequência  $\tilde{P}_1(X)$  contém erros.  $\tilde{P}_1(X)$  é então armazenada no buffer do receptor para ser corrigida futuramente (se necessário). Ao

mesmo tempo, o receptor envia um NAK para o transmissor indicando que a sequência recebida contém algum erro. Após o recebimento deste NAK, o transmissor envia a sequência  $P_2(X)$  para o receptor<sup>1</sup>.

A sequência recebida  $\tilde{P}_2(X)$  corresponde à sequência  $P_2(X)$  transmitida. No receptor,  $\tilde{P}_2(X)$  é dividida por  $G_2(X)$ , obtendo-se  $\tilde{I}_2(X)$  e  $\tilde{R}_2(X)$  que são o quociente e resto da divisão, respectivamente. Se  $\tilde{R}_2(X)$  for zero, verifica-se a síndrome de  $\tilde{I}_2(X)$  baseado em  $C_0$ . Se a síndrome  $\tilde{S}_2(X)$  for igual a zero, então assume-se que não há erros em  $\tilde{I}_2(X)$ , sendo esta sequência idêntica à  $I(X)$ . Neste caso, o receptor aceita  $\tilde{I}_2(X)$  e descarta  $\tilde{P}_1(X)$  armazenada anteriormente e solicita a transmissão da próxima sequência de informação. Por outro lado, se  $\tilde{R}_2(X)$  ou  $\tilde{S}_2(X)$  forem diferentes de zero,  $\tilde{P}_2(X)$  e  $\tilde{P}_1(X)$  são decodificados baseado no código convolucional  $C_1$  de taxa 1/2, usando um decodificador de Viterbi. Seja  $\tilde{I}^*(X)$  a sequência decodificada. Após a decodificação baseada em  $C_1$ , calcula-se a síndrome de  $\tilde{I}^*(X)$  baseada no código  $C_0$ . Se a síndrome for zero, o receptor aceita  $\tilde{I}^*(X)$ . Senão,  $\tilde{P}_1(X)$  é descartado, e  $\tilde{P}_2(X)$  é armazenado em seu lugar no buffer do receptor. Neste caso, o receptor envia um outro NAK para o transmissor. A próxima retransmissão será novamente  $P_1(X)$ . As retransmissões alternadas de  $P_1(X)$  e  $P_2(X)$  continuarão até que  $I(X)$  seja obtido corretamente.

Seja  $P_b$  a probabilidade de erro de bit após a decodificação baseada no código convolucional  $C_1$  (2, 1,  $m$ ) e no decodificador de Viterbi. Seja  $T(X, Y)$  a função geradora do código  $C_1$  [Lin, 1983], então  $P_b$  é dada por:

$$P_b \leq \frac{1}{k} \frac{\partial}{\partial Y} T(X, Y) \Big|_{X=2\sqrt{p(1-p)}, Y=1} \quad (4.1)$$

onde  $p$  é a probabilidade de transição de um canal BSC. Seja

$$q = (1 - P_b)^{n+m} \quad (4.2)$$

$$P = (1 - p)^{n+m} \quad (4.3)$$

$$P_t = P + (1 - P)q \quad (4.4)$$

---

<sup>1</sup>A transmissão de  $\tilde{P}_2(X)$  consiste no primeiro processo de retransmissão de  $I(X)$ .

Portanto, o throughput deste esquema usando um código convolucional  $(2, 1, m)$  e um decodificador de Viterbi, é dado por:

$$\eta_{1/2} = G_{1/2} \cdot \frac{n}{n + m}. \quad (4.5)$$

onde  $G_{1/2}$  é o throughput dado pelas equações 3.11, 3.16 e 3.19, para um tamanho do buffer do receptor infinito, finito e menor ou igual a  $B$  e finito e maior que  $B$ , respectivamente. As figuras 4.1 e 4.2 mostram as curvas de throughput de esquemas híbridos utilizando códigos convolucionais de taxa  $1/2$ .

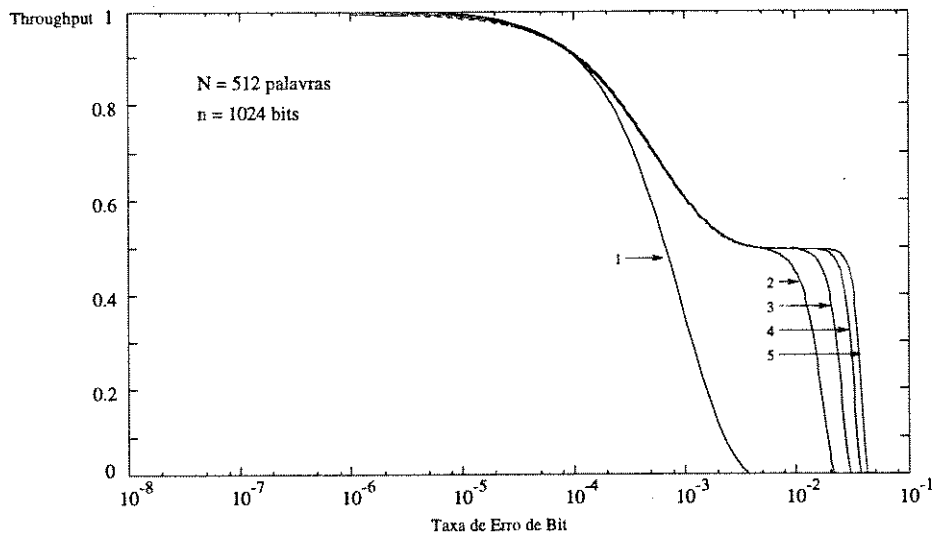


Figura 4.1: Throughput do esquema ARQ híbrido tipo-II com código convolucional de taxa  $1/2$  com buffer do receptor infinito: (1) Esquema Selective-repeat ideal; (2), (3), (4) e (5) Esquema híbrido tipo-II com código convolucional e memória do codificador  $m$  igual a 4, 6, 8 e 10, respectivamente.

As figuras 4.1 e 4.2 mostram o throughput de vários esquemas ARQ híbridos que utilizam códigos convolucionais de taxas  $1/2$  no processo de correção de erro, variando a memória do codificador. A medida que o valor da memória do codificador aumenta, resulta em uma ligeira melhoria em termos de throughput. Mas deve ser considerado que o aumento da memória do codificador implica em uma maior complexidade de implementação. O esquema ARQ híbrido com código convolucional de taxa  $1/2$ , apresenta desempenho superior ao esquema selective-repeat ideal.

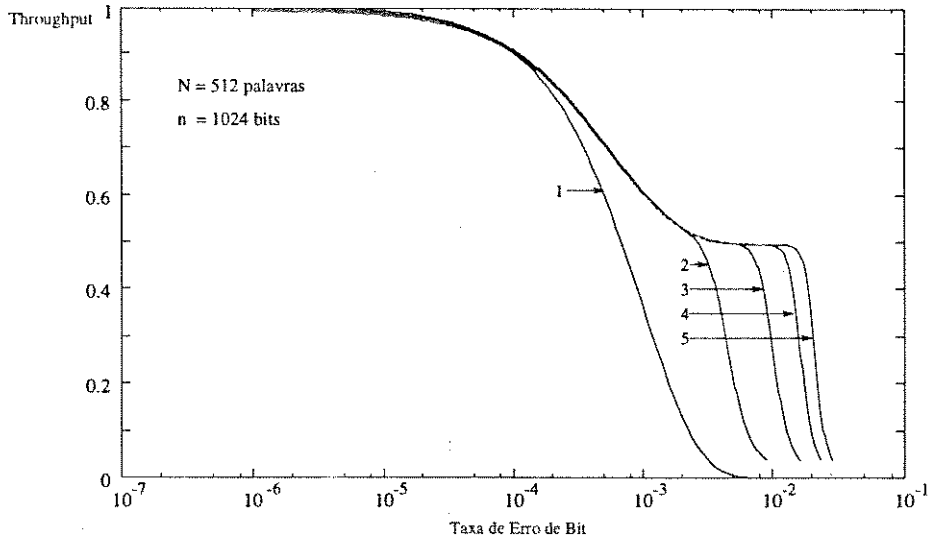


Figura 4.2: Throughput do esquema ARQ híbrido tipo-II com código convolucionacional de taxa 1/2 com buffer do receptor de tamanho finito: (1) Esquema Selective-repeat ideal; (2), (3), (4) e (5) Esquema híbrido tipo-II com código convolucionacional e memória do codificador  $m$  igual a 4, 6, 8 e 10, respectivamente.

### 4.3 Esquema ARQ híbrido tipo-II usando código convolucionacional de taxa 2/3

Este esquema [Lin, 1984] ainda utiliza um código de bloco  $C_0(n, k)$  para detecção de erro. Entretanto, o código usado para a correção de erro, é um código convolucionacional  $(3, 2, m)$  denotado por  $C_3$  e matriz geradora:

$$G(X) = \begin{bmatrix} G_1(X) & 0 & G_{31}(X) \\ 0 & G_2(X) & G_{32}(X) \end{bmatrix} \quad (4.6)$$

A sequência de  $k$  bits de informação, é codificada pelo código detector de erro  $C_0$  formando a palavra-código  $I(X)$ . Esta palavra é transformada em duas sequências  $P_1(X) = I(X)G_1(X)$  e  $P_{31}(X) = I(X)G_{31}(X)$ . A sequência resultante do intercalamento de  $P_1(X)$  e  $P_{31}(X)$ , pode ser vista como uma sequência codificada de  $I(X)$  baseada no código convolucionacional  $C_1$  de taxa 1/2, onde os polinômios geradores são  $G_1(X)$  e  $G_{31}(X)$ . A sequência  $P_1(X)$  é transmitida para

o receptor enquanto que  $P_{31}(X)$  é armazenada no buffer do transmissor para ser utilizada, se necessário, numa retransmissão. Quando  $\tilde{P}_1(X)$  é recebida, utiliza-se o mesmo procedimento usado para com o código de taxa 1/2. Se erros forem detectados na sequência  $\tilde{P}_1(X)$ , ela é armazenada no buffer do receptor, e este envia um NAK para o transmissor. Após o recebimento do NAK, o transmissor gera duas novas sequências a partir de  $I(X)$ ;  $P_2(X) = I(X)G_2(X)$  e  $P_{32}(X) = I(X)G_{32}(X)$ . As sequências  $P_2(X)$  e  $P_{32}(X)$  quando intercaladas, formam uma sequência codificada que pode ser vista como uma palavra do código convolucional de taxa 1/2 ( $C_2$ ), com polinômios geradores  $G_2(X)$  e  $G_{32}(X)$ .

A sequência  $P_2(X)$  é transmitida, enquanto que  $P_{32}(X)$  é armazenada no buffer do transmissor para ser usada no processo de retransmissão (observe que o codificador usando um código de taxa  $(3, 2, m)$  se comporta como dois codificadores  $(2, 1, m)$  independentes). Ao receber  $\tilde{P}_2(X)$ , o receptor executa o mesmo procedimento de verificação de erro; divide  $\tilde{P}_2(X)$  por  $G_2(X)$  obtendo  $\tilde{I}_2(X)$ ; se o resto desta divisão for igual a zero, verifica a síndrome de  $\tilde{I}_2(X)$  baseada no código detector de erro  $C_0$ ; se for zero, libera a mensagem de informação para o usuário. Se forem detectados erros em  $\tilde{P}_2(X)$ , um NAK é enviado para o transmissor, e o receptor armazena  $\tilde{P}_2(X)$ . Após o recebimento do segundo NAK, o transmissor envia a sequência  $P_3(X) = P_{31}(X) + P_{32}(X) = I(X)G_{31}(X) + I(X)G_{32}(X)$  para o transmissor. Observe que a sequência obtida pelo intercalamento de  $P_1(X)$ ,  $P_2(X)$  e  $P_3(X)$ , é uma sequência codificada baseada no código convolucional  $C_3$  de taxa 2/3. Após o recebimento de  $\tilde{P}_3(X)$ , a sequência resultante do intercalamento de  $\tilde{P}_1(X)$ ,  $\tilde{P}_2(X)$  e  $\tilde{P}_3(X)$  será decodificada baseada no código  $C_3$ , usando um decodificador de Viterbi. Para uma melhor compreensão, o processo de codificação é exemplificado abaixo.

- Exemplo: A matriz geradora do código convolucional  $(3, 2, 2)$  mostrado na figura 4.3 é dada por:

$$G(X) = \begin{bmatrix} 1 + X^2 & 0 & 1 + X + X^2 \\ 0 & 1 + X^2 & 1 + X^2 \end{bmatrix}$$

Seja  $I = (101101)$  a sequência de bits a ser codificada. Na forma polinomial ela é expressa

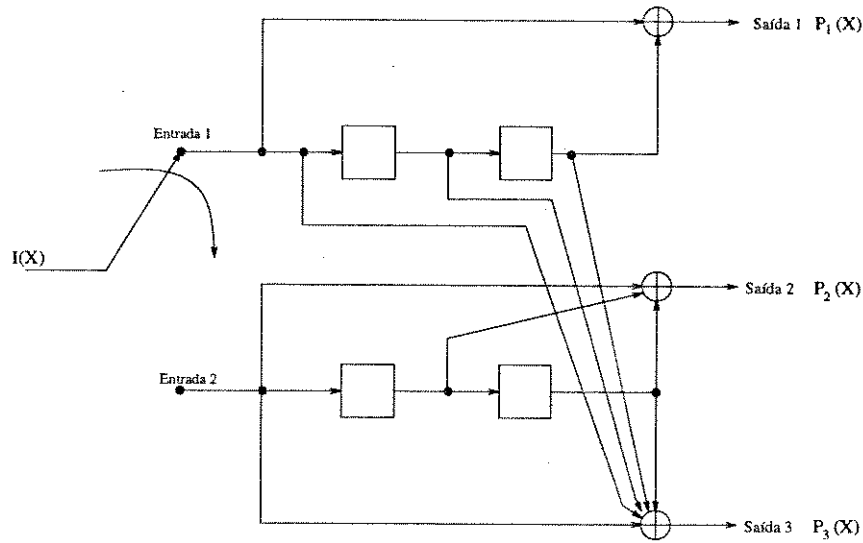


Figura 4.3: Codificador para um código convolucional (3, 2, 2)

por:  $I(X) = 1 + X^2 + X^3 + X^5$ . Portanto, as seqüências codificadas são:

- Quando a seqüência de entrada  $I(X)$  é carregada à entrada 1 do codificador:

$$P_1(X) = I(X)G_1(X) = 1 + X^3 + X^4 + X^7$$

$$P_{31}(X) = I(X)G_{31}(X) = 1 + X + X^6 + X^7$$

- Quando a seqüência de entrada  $I(X)$  é carregada à entrada 2 do codificador:

$$P_2(X) = I(X)G_2(X) = 1 + X + X^6 + X^7$$

$$P_{32}(X) = I(X)G_{32}(X) = 1 + X^3 + X^4 + X^7$$

A seqüência  $P_3(X) = P_{31}(X) + P_{32}(X)$  é dada por:

$$P_3(X) = X + X^3 + X^4 + X^6$$

Portanto, a seqüência  $P(X)$  resultante do intercalamento de  $P_1(X)$ ,  $P_2(X)$  e  $P_3(X)$ , é dada por [Lin, 1983]:

$$\begin{aligned} P(X) &= [P_1(X), P_2(X), P_3(X)] \\ &= P_1(X^3) + X P_2(X^3) + X^2 P_3(X^3) \\ &= 1 + X + X^4 + X^5 + X^9 + X^{11} + X^{12} + X^{14} + X^{19} + X^{20} + X^{21} + X^{22} \end{aligned}$$

A partir da sequência  $\tilde{P}(X)$  correspondente ao intercalamento das três sequências recebidas:  $\tilde{P}_1(X)$ ,  $\tilde{P}_2(X)$  e  $\tilde{P}_3(X)$ , será feita a decodificação. Seja  $\tilde{I}_1(X)$  e  $\tilde{I}_2(X)$  as duas sequências decodificadas referentes à  $I(X)$  da primeira e segunda transmissões respectivamente. Após a decodificação, as síndromes de  $\tilde{I}_1(X)$  e  $\tilde{I}_2(X)$  são verificadas pelo código detector de erro  $C_0$ . Se ambas as síndromes forem zero, o receptor aceita  $\tilde{I}_1(X)$  e  $\tilde{I}_2(X)$ . Se ambas as síndromes forem diferentes de zero,  $\tilde{P}_1(X)$  é descartada, e  $\tilde{P}_2(X)$  e  $\tilde{P}_3(X)$  são armazenadas. Então, o receptor envia um NAK para o transmissor. Este NAK, indica que o transmissor deve enviar novamente  $P_1(X)$ . As retransmissões de  $P_1(X)$ ,  $P_2(X)$  e  $P_3(X)$  continuarão ciclicamente até que as síndromes de  $\tilde{I}_1(X)$  e  $\tilde{I}_2(X)$ , sejam ambas iguais a zero. Se apenas uma síndrome for zero, pode-se subtrair a sequência decodificada de  $\tilde{P}_3(X)$  e tentar decodificar a sequência resultante baseada nos códigos de taxa 1/2  $C_1$  ou  $C_2$ . A figura 4.4 mostra o throughput para o esquema ARQ híbrido tipo-II com código convolucionacional de taxa 2/3, considerando a memória do codificador igual a 6. Observa-se uma melhoria significativa em termos de throughput do esquema ARQ com taxa 2/3 sobre o esquema que utiliza código convolucionacional de taxa 1/2, especialmente para altas taxas de erro de bit (maior que  $10^{-2}$ ).

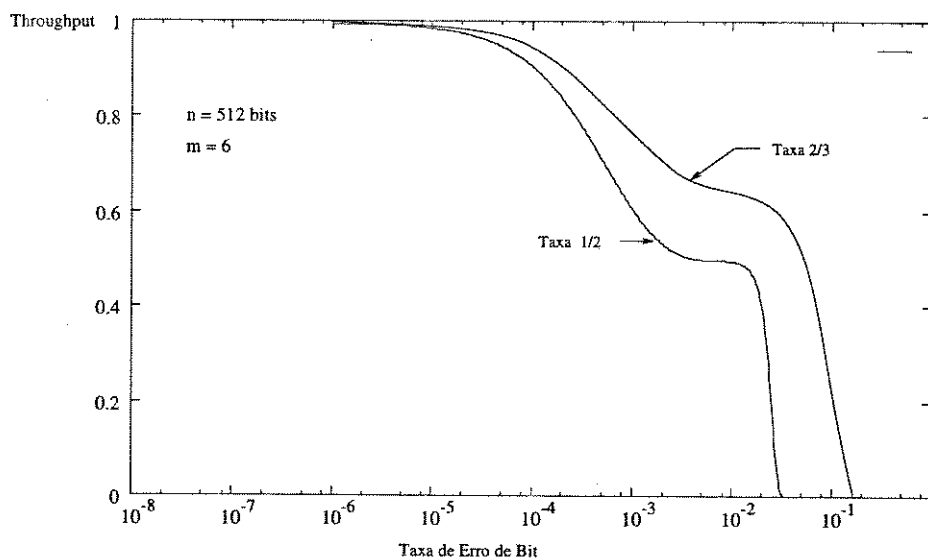


Figura 4.4: Throughput do esquema ARQ híbrido tipo-II com taxa 2/3 comparado com esquema ARQ híbrido tipo-II com taxa 1/2, considerando  $m = 6$ .



## 4.4 Esquema ARQ Híbrido Tipo-II com Combinação de Códigos

O procedimento de codificação do esquema ARQ híbrido tipo-II com combinação de códigos, é semelhante ao esquema ARQ híbrido tipo-II que utiliza código convolucional de taxa 1/2. A diferença está no procedimento de detecção/decodificação de erro no receptor.

Este esquema utiliza dois códigos; um código de bloco para o processo de detecção de erro e um código convolucional  $(2, 1, m)$  de taxa 1/2 e memória  $m$ , denotado por  $C_1$ , para a processo de correção de erro. Seja  $(G_1(X), G_2(X))$  os dois polinômios geradores de  $C_1$ . Quando a mensagem de informação estiver pronta para a transmissão, primeiramente ela é codificada em uma palavra-código  $I(X)$  baseada no código  $C_0$ . A palavra-código  $I(X)$  é transformada em duas sequências baseada no código  $C_1$ :  $P_1(X) = I(X)G_1(X)$  e  $P_2(X) = I(X)G_2(X)$ . O transmissor envia alternadamente estas duas sequências.

No processo de correção de erro, o decodificador trabalha com uma combinação de todas as sequências recebidas relativas a uma dada palavra de informação. O receptor, ao contrário do esquema híbrido com código convolucional de taxa 1/2, não descarta as sequências recebidas com erro, ele as armazena, para serem utilizadas no procedimento de decodificação. Se forem detectados erros nas duas primeiras sequências  $\tilde{P}_1^{(1)}(X)$  e  $\tilde{P}_2^{(1)}(X)$ , a sequência resultante do intercalamento de  $\tilde{P}_1^{(1)}(X)$  e  $\tilde{P}_2^{(1)}(X)$  será decodificada baseada no código convolucional de taxa meio, usando um decodificador de Viterbi. Se for detectado erro na sequência decodificada, o receptor solicita uma nova retransmissão, e armazena estas duas sequências que foram recebidas com erro (lembrando que no esquema híbrido tipo-II que utiliza apenas o código convolucional de taxa 1/2, o receptor armazena apenas a última sequência que foi recebida com erro, descartando a sequência anterior).

A próxima sequência retransmitida será  $P_1(X)$ . Considere  $\tilde{P}_1^{(2)}(X)$  a segunda sequência recebida referente à  $P_1(X)$ . Se não for detectado a presença de erros em  $\tilde{P}_1^{(2)}(X)$ , obtém-se a mensagem de informação  $I(X)$  a partir de  $\tilde{P}_1^{(2)}(X)$ , descartando  $\tilde{P}_1^{(1)}(X)$  e  $\tilde{P}_2^{(1)}(X)$  do buffer do receptor. Mas se houver erros em  $\tilde{P}_1^{(2)}(X)$ , as sequências  $\tilde{P}_1^{(1)}(X)$ ,  $\tilde{P}_2^{(1)}(X)$  e  $\tilde{P}_1^{(2)}(X)$  são intercaladas, formando uma sequência combinada que é decodificada por um código de

repetição  $(3, 1, m)$  com polinômios geradores  $(G_1(X), G_2(X), G_1(X))$ . A sequência combinada é processada pelo decodificador de Viterbi. Se ainda houver erros na sequência resultante da decodificação, o receptor armazena  $\tilde{P}_1^{(2)}(X)$  juntamente com  $\tilde{P}_1^{(1)}(X)$  e  $\tilde{P}_2^{(1)}(X)$  e solicita uma nova retransmissão. A próxima sequência a ser retransmitida é  $P_2(X)$ . Seja  $\tilde{P}_2^{(2)}(X)$  a sequência recebida correspondente à  $P_2(X)$ . O mesmo procedimento de verificação de erro é aplicado à  $\tilde{P}_2^{(2)}(X)$ . Se não houver erro na decodificação, a mensagem é liberada para o usuário. Caso contrário, forma-se uma sequência combinada a partir de  $\tilde{P}_1^{(1)}(X)$ ,  $\tilde{P}_2^{(1)}(X)$ ,  $\tilde{P}_1^{(2)}(X)$  e  $\tilde{P}_2^{(2)}(X)$ , que é decodificada por um código de repetição  $(4, 1, m)$ , com polinômios geradores  $(G_1(X), G_2(X), G_1(X), G_2(X))$ . Esta sequência combinada é novamente processada pelo decodificador de Viterbi. Este procedimento continua até decodificar a mensagem corretamente.

As repetições de  $P_1(X)$  e  $P_2(X)$  e a operação de combinação de códigos, resultam em uma família de códigos de repetição com taxas  $1/(2+i)$ ,  $i = 1, 2, \dots$ . Estes códigos de repetição são obtidos a partir do código de taxa  $1/2$ , acrescentando alternadamente, os polinômios geradores  $G_1(X)$  e  $G_2(X)$ . Pode-se obter duas famílias de códigos de repetição, dependendo de qual polinômio gerador será utilizado na terceira retransmissão. Por exemplo, considere um código convolucional  $(2, 1, 6)$  de taxa  $1/2$ , tendo polinômios geradores<sup>2</sup>  $(133, 171)$ . Os polinômios geradores das duas famílias de códigos de repetição com taxas  $1/(2+i)$ ,  $i = 1, 2$  e  $3$ , são mostrados na tabela 4.1. Esta tabela mostra as distâncias livres ( $d_{free}$ ) dos códigos [Lin, 1983]. e observa-se que os códigos de repetição de taxas  $1/3$  e  $1/5$  obtidos repetindo primeiramente o polinômio gerador 133, têm distâncias livres maiores que os seus correspondentes que foram obtidos repetindo primeiramente o polinômio gerador 171.

Para um melhor desempenho desta estratégia, deve-se observar qual das duas famílias de códigos de repetição apresentam maior  $d_{free}$ , para determinar qual das sequências  $P_1(X)$  e  $P_2(X)$  será transmitida primeiramente.

O throughput deste esquema é dado por [Kallel, 1990-A]:

$$\eta_{CC} = \frac{1}{T_{CC}} \frac{k}{n+m} \quad (4.7)$$

<sup>2</sup>Os polinômios geradores estão representados pelo sistema de numeração octal. Portanto, como exemplo, o polinômio gerador 133 é dado por  $G_1(X) = 1 + X^2 + X^3 + X^5 + X^6$ .

Taxa	Polinômios Geradores	$d_{free}$	Polinômios Geradores	$d_{free}$
1/2	(133,171)	10	(133,171)	10
1/3	(133,171,133)	13	(133,171,171)	12
1/4	(133,171,133,171)	20	(133,171,171,133)	20
1/5	(133,171,133,171,133)	23	(133,171,171,133,171)	22

Tabela 4.1: Distâncias livres ( $d_{free}$ ) de duas famílias de códigos de repetição obtidos a partir de um código convolucional (2,1,6) com polinômios geradores (133, 171).

Onde  $T_{CC}$  é o número médio de sequências transmitidas para uma dada palavra-código, sendo dado por:

$$1 + P_d + \sum_{i=1}^{\infty} (P_d)^{i+1} \prod_{j=1}^i P(D_d^{(j)}) \leq T_{CC} \leq 1 + P_d + \sum_{i=1}^{\infty} P(D_d^{(i)}) \quad (4.8)$$

Lembrando que  $P_d$  é a probabilidade de detectar um erro na sequência recebida, sendo obtida por:

$$P_d = 1 - P_c - P_e \approx 1 - P = 1 - (1 - p)^{n+m} \quad (4.9)$$

Considere agora, as seguintes probabilidades:

- $P(D_c^{(i)})$  é a probabilidade de que a sequência decodificada não contenha erro;
- $P(D_d^{(i)})$  é a probabilidade de que a sequência decodificada contenha um erro detectável;
- $P(D_e^{(i)})$  é a probabilidade de que a sequência decodificada contenha um erro não detectável.

Então, tem-se que:

$$P(D_c^{(i)}) + P(D_d^{(i)}) + P(D_e^{(i)}) = 1 \quad (4.10)$$

Lembrando que a operação de detecção é realizada pelo código detector de erro  $C_0$ . O valor de  $i$  corresponde a um determinado código de repetição de taxa  $1/(2 + i)$ . Para  $i = 0$ , tem-se o código de taxa 1/2. A probabilidade  $P(D_c^{(i)})$  é limitada inferiormente por:

$$P(D_c^{(i)}) \geq (1 - P(E^{(i)}))^n \quad (4.11)$$

Portanto, a probabilidade  $P(D_d^{(i)})$  pode ser aproximada para:

$$P(D_d^{(i)}) \simeq 1 - P(D_c^{(i)}) \geq 1 - (1 - P(E^{(i)}))^n \quad (4.12)$$

onde  $P(E^{(i)})$  é a probabilidade de ocorrer um erro de decodificação pelo algoritmo de Viterbi, sendo limitada por [Viterbi, 1971]:

$$P(E^{(i)}) \leq \sum_{d=d_{free}^{(i)}}^{\infty} a_d^{(i)} Pr_d \quad (4.13)$$

onde  $d_{free}^{(i)}$  e  $a_d^{(i)}$  são, respectivamente, a distância livre e o espectro de distância do código  $(2+i, 1, m)$ , e  $Pr_d$  é a probabilidade de selecionar um caminho errado na distância  $d$ . Para um canal binário simétrico (BSC) com probabilidade de transição  $p$ , tem-se:

$$Pr_d = \begin{cases} \sum_{j=\frac{d+1}{2}}^d \binom{d}{j} p^j (1-p)^{d-j} & \text{para } d \text{ ímpar} \\ \sum_{j=\frac{d}{2}+1}^d \binom{d}{j} p^j (1-p)^{d-j} + \frac{1}{2} \binom{d}{d/2} [p(1-p)]^{d/2} & \text{para } d \text{ par} \end{cases} \quad (4.14)$$

A figura 4.5 mostra o throughput do esquema ARQ híbrido tipo-II com combinação de códigos de repetição, considerando a memória do decodificador  $m$  igual a 6. Observa-se que o desempenho em termos de throughput do esquema ARQ híbrido tipo-II com combinação de códigos, é superior ao selective-repeat ideal, especialmente para taxas de erro de bit maiores que  $10^{-2}$ .

## 4.5 Esquema ARQ Híbrido tipo-II com Códigos Convolucionais Puncionados

O ideal para um sistema ARQ seria utilizar vários códigos de diferentes taxas de tal forma que este sistema se adapte às condições do canal. Como consequência, se o canal estiver muito ruidoso, utiliza-se um código com uma maior capacidade de correção de erro, caso contrário,

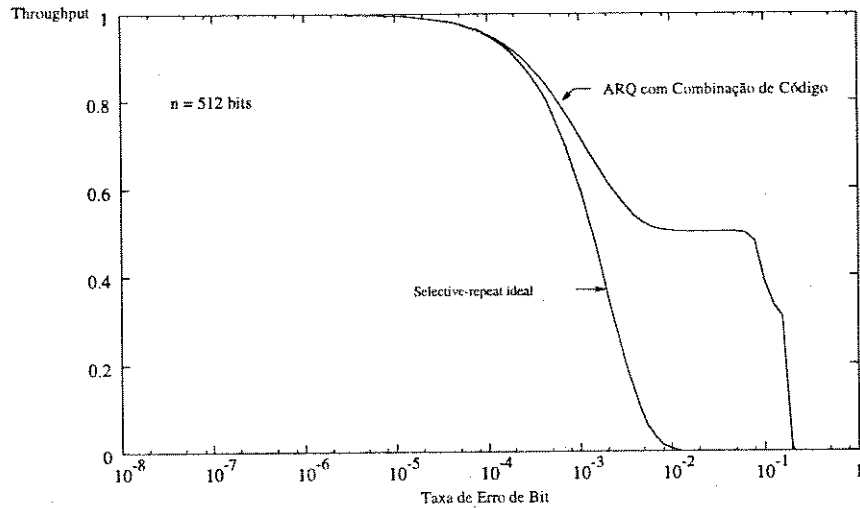


Figura 4.5: Throughput do esquema ARQ híbrido tipo-II com combinação de códigos de repetição considerando o buffer do receptor infinito e utilizando um código convolucional (2, 1, 6), com polinômios geradores (133,171).

utiliza-se um código com uma capacidade menor (código de alta taxa). A desvantagem de tal sistema é que para cada código de um sistema ARQ seria necessário um conjunto codificador/decodificador. Na prática, entretanto, o sistema ARQ por simplicidade de implementação deve utilizar apenas um conjunto codificador/decodificador ao invés de vários conjuntos de codificadores/decodificadores. Isto pode ser conseguido através do puncionamento do código convolucional utilizado ([Hagenauer, 1988] e [Kallel, 1990-B]). O puncionamento do código convolucional consiste em não transmitir alguns bits da sequência codificada.

Os códigos puncionados gerados a partir de um dado código convolucional, apresentam algumas características de compatibilidade de taxa. Isto é, todos os bits que foram descartados em um código puncionado de alta taxa, são utilizados pelos códigos de baixas taxas. Portanto, os códigos de taxas maiores estão "embutidos" nos códigos de taxas menores. Estes códigos são denominados de códigos Convolucionais Puncionados Compatíveis em Taxa (códigos CPCT).

A obtenção de um código puncionado, consiste em descartar alguns bits (ou seja, deixar de transmitir) da sequência gerada a partir de um codificador de taxa  $1/N$ . Isto resulta em códigos de altas taxas de codificação. O puncionamento é feito periodicamente, com período  $P$ . O puncionamento na sequência codificada, é feito a partir de uma tabela denominada de

tabela de puncionamento, que consiste em uma matriz  $N \times P$ , onde  $N$  é o número de saídas do codificador. Esta matriz é composta por dois símbolos: o "1" indicando que o bit codificado será enviado e o "0" que o bit não será enviado. Para uma melhor compreensão, considere o exemplo baseado na figura 4.6, onde  $N = 2$ .

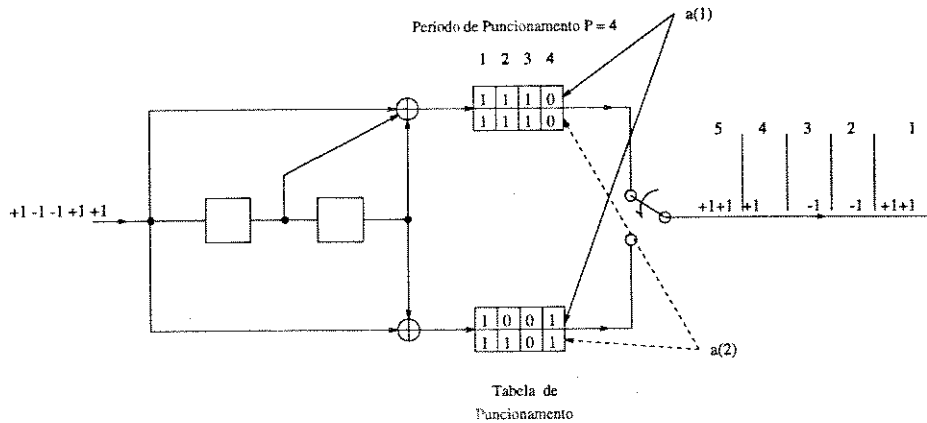


Figura 4.6: Exemplo de um codificador convolucional com duas tabelas de puncionamento compatíveis em taxa e período de puncionamento igual a 4.

Os bits de informação e os bits da sequência codificada são representados por  $\pm 1$  para evitar confusão com os símbolos da tabela de puncionamento. O período de puncionamento  $P$  é igual a 4 e a matriz de puncionamento é dada por:

$$a(1) = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad (4.15)$$

Ao invés de transmitir  $2.P = 8$  bits, envia-se primeiramente,  $P + 1 = 5$  bits por  $p$  símbolos de informação, resultando em um código convolucional puncionado de taxa  $4/5$ .

Suponha que este código de taxa  $4/5$  não seja suficiente para corrigir os erros introduzidos pelo canal. Então, pode-se utilizar códigos mais poderosos com taxas menores:  $4/6$ ,  $4/7$  ou  $4/8$ . Para obter estes códigos, basta substituir adequadamente os "0"s da matriz de puncionamento  $a(1)$  por "1"s, resultando as seguintes matrizes:

$$a(2) = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad (4.16)$$

$$a(3) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad (4.17)$$

$$a(4) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (4.18)$$

As matrizes  $a(2)$ ,  $a(3)$  e  $a(4)$ , correspondem às matrizes de puncionamento referentes aos códigos de taxas  $4/6$ ,  $4/7$  e  $4/8 = 1/2$ , respectivamente. Portanto, têm-se uma família de códigos compatíveis em taxa, obtida a partir de um código de taxa  $1/2$ . Um problema a ser considerado, é se estes códigos são bons ou se existem alguns deles que são catastróficos. Esta questão não será abordada neste trabalho, e para uma maior compreensão ver referências [Hagenauer, 1988], [Begin, 1989] e [Haccoun, 1989].

No processo de transmissão, primeiramente a mensagem de  $k$  bits de informação é codificada em uma palavra de  $n$  bits baseada no código de detecção de erro  $C_0$ . O segundo passo, consiste em codificar esta palavra por meio de um código convolucional de taxa  $1/N$ . Em seguida, apenas alguns bits são transmitidos de acordo com a matriz de puncionamento, gerando um código de taxa  $P/(P+1) = P/(P+1)$ . No receptor, a sequência recebida formada a partir de um código CPCT de taxa  $P/(P+1)$ , é decodificada por meio de um decodificador de Viterbi. Após o processo de decodificação, a síndrome da sequência decodificada é verificada baseada em  $C_0$ . Se a síndrome for zero, assume-se que a sequência não possui erro, sendo aceita pelo receptor que envia um ACK para o transmissor. Caso contrário, o receptor envia um NAK para o transmissor solicitando uma retransmissão. O transmissor, após o recebimento deste NAK, incrementa o valor da variável  $l$  de uma unidade, na expressão da taxa de codificação  $P/(P+l)$ , resultando em uma taxa de codificação igual a  $P/(P+2)$  e transmite a próxima sequência. O processo de retransmissão consiste em incrementar o valor de  $l$  e como resultado, transmitir uma nova palavra-código baseada em um código com taxa menor (com maior capacidade de correção de erro), até atingir uma taxa igual a  $1/N$ .

Se a sequência formada a partir do código de taxa  $1/N$  não for decodificada corretamente, pode-se utilizar dois procedimentos na transmissão da próxima sequência: o primeiro consiste na combinação de códigos, utilizando códigos de repetição de taxa  $1/N + i, i = 1, 2, 3 \dots$

[Kallel, 1990-B]. O segundo procedimento, utiliza-se o esquema com código convolucional de taxa 1/2, que opera somente com as duas últimas sequências recebidas.

O throughput deste esquema pode ser calculado através da expressão:

$$\eta_{CP} = \frac{P}{P + 1 + h(T_{CP} - 1)} \frac{k}{n + m} \quad (4.19)$$

onde  $h$  é o número de bits que são adicionados em cada retransmissão e o número médio  $T_{CP}$  de transmissões para uma dada palavra é limitado por:

$$1 + \sum_{n=1}^{\infty} \prod_{j=1}^n P(D_d^{(j)}) \leq T_{CP} \leq 1 + \sum_{j=1}^{\infty} P(D_d^{(j)}) \quad (4.20)$$

A figura 4.7 mostra o throughput do esquema ARQ híbrido tipo-II com códigos convolucionais puncionados com diferentes taxas de código iniciais, considerando  $m = 6$  e  $h = 1$ . Observa-se que quanto maior a taxa inicial do código, melhor será o desempenho do esquema com códigos convolucionais puncionados. Para altas taxas de erro de bit (acima de  $10^{-3}$ ), é interessante utilizar o esquema ARQ híbrido tipo-II com códigos puncionados pois apresenta um alto desempenho em termos de throughput.

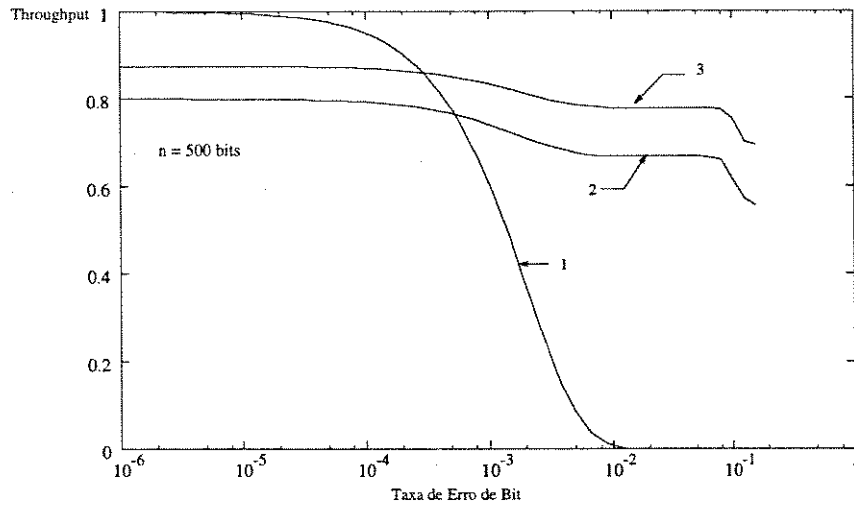


Figura 4.7: Throughput do esquema ARQ híbrido tipo-II com códigos puncionados com diferentes taxas iniciais: (1) Selective-repeat ideal; (2) e (3) Esquema ARQ híbrido tipo-II com taxas iniciais iguais a 4/5 e 7/8, respectivamente.

A figura 4.8 mostra o throughput do esquema ARQ híbrido tipo-II, iniciando com um código convolucional puncionado de taxa 7/8 e  $m = 6$ , e utilizando diferentes valores de  $h$ .



Observa-se que, pequenos valores de  $h$  implica em um maior número de retransmissões para uma dada palavra, e conseqüentemente, um melhor desempenho do esquema ARQ com códigos puncionados.

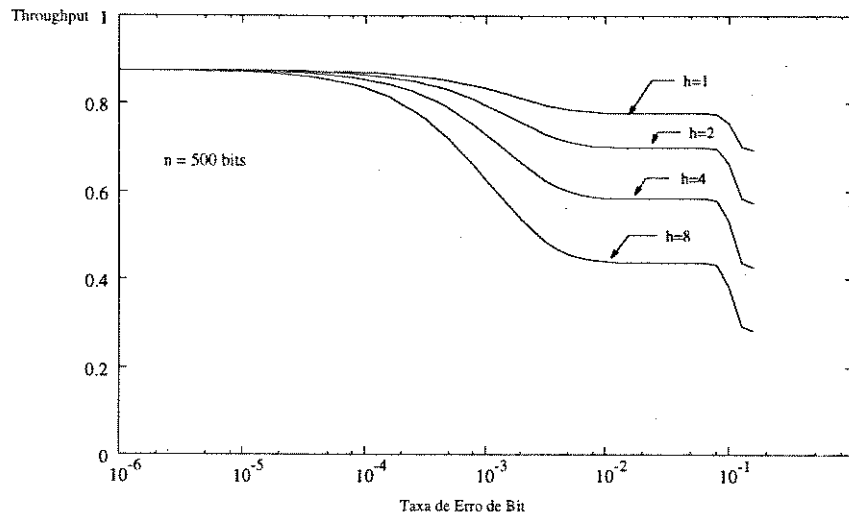


Figura 4.8: Throughput do esquema ARQ híbrido tipo-II com códigos convolucionais puncionados para diversos valores de  $h$ .

## 4.6 Comparações Entre os Diversos Esquemas ARQ Híbridos Tipo-II Selective-Repeat que Utilizam Códigos Convolucionais

Nesta seção, são feitas várias comparações em termos de throughput dos esquemas ARQ híbridos tipo-II, que utilizam códigos convolucionais no processo de correção de erro. Para todos os esquemas, é assumido que o buffer do receptor tem tamanho infinito.

A figura 4.9, compara os esquemas ARQ híbridos tipo-II que utilizam códigos convolucionais de taxa  $1/2$  com os esquemas de combinação de códigos. Todos os dois esquemas apresentam um desempenho em termos de throughput superior ao esquema ARQ selective-repeat ideal para taxas de erro de bit acima de  $10^{-3}$ . Para uma taxa de erro de  $10^{-2}$ , os dois esquemas híbridos apresentam um throughput em torno de 0,5. A vantagem do esquema ARQ que utiliza

combinação de códigos é que ele ainda mantém um throughput em torno de 0.5 para uma taxa de erro até  $10^{-1}$ , ao passo que, para esta taxa, o throughput do esquema ARQ que utiliza código convolucionacional de taxa 1/2 é igual a zero.

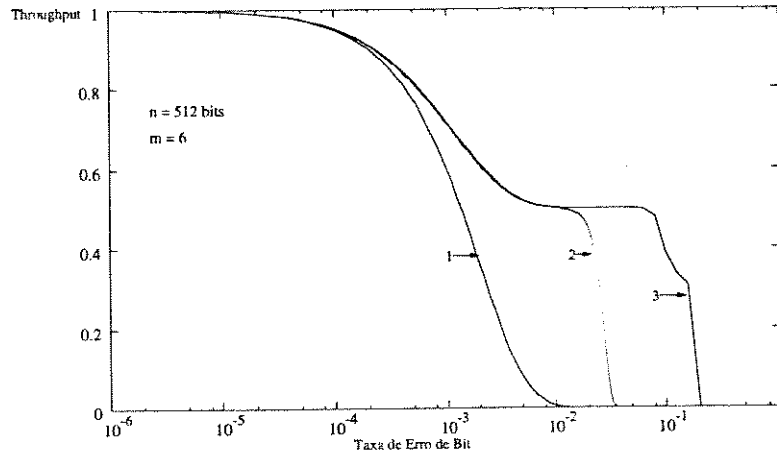


Figura 4.9: Comparação do throughput: (1) Esquema ARQ selective-repeat ideal; (2) Esquema ARQ híbrido tipo-II com código convolucionacional de taxa 1/2 e  $m = 6$  e (3) Esquema ARQ híbrido tipo-II com combinação de código e  $m = 6$ .

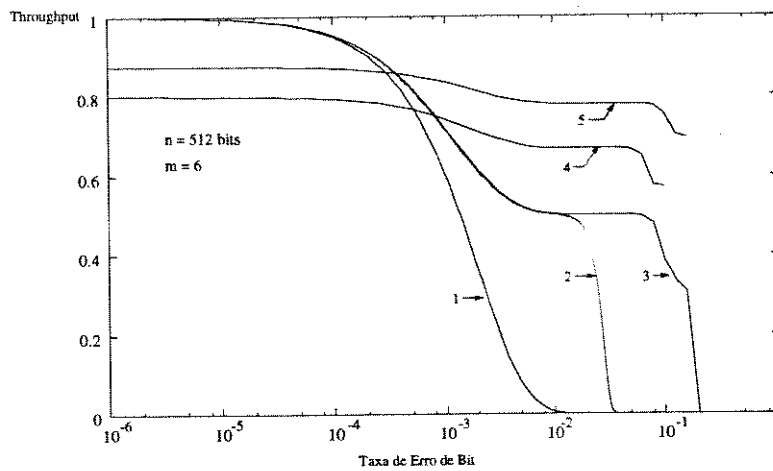


Figura 4.10: Comparação do throughput: (1) Esquema ARQ selective-repeat ideal; (2) Esquema ARQ híbrido tipo-II com código convolucionacional de taxa 1/2; (3) Esquema ARQ híbrido tipo-II com combinação de código; (4) e (5) Esquemas ARQ híbridos tipo-II com códigos puncionados com  $h = 1$  e taxas iniciais iguais a  $4/5$  e  $7/8$ , respectivamente.

A figura 4.10 mostra o esquema ARQ com códigos puncionados de taxas iniciais de  $7/8$  e  $4/5$ , ambos com memória  $m = 6$  e  $h = 1$ , os esquemas ARQ com combinação de códigos e o esquema que utiliza código convolucional de taxa  $1/2$ . Devido ao acréscimo de bits para a correção de erro nas primeiras transmissões de todas as sequências no esquema ARQ com códigos puncionados, os esquemas com combinação de código e com código convolucional de taxa  $1/2$ , apresentam um desempenho superior para taxas de erro de bit menores que  $10^{-3}$ . Acima desse valor, o desempenho do esquema com códigos puncionados ainda mantém-se alto, ao passo que, o throughput dos outros dois esquemas cai mais rapidamente.

A figura 4.11 mostra o esquema ARQ híbrido com códigos puncionados de taxa inicial  $7/8$  e  $h = 1, 2$  e  $4$ , os esquemas ARQ híbridos com combinação de códigos e código convolucional de taxa  $1/2$ . Para taxas de erros acima de  $10^{-3}$ , os esquemas com códigos convolucionais puncionados, mesmo com  $h > 1$ , possuem um bom desempenho em termos de throughput, sendo superior aos outros dois esquemas.

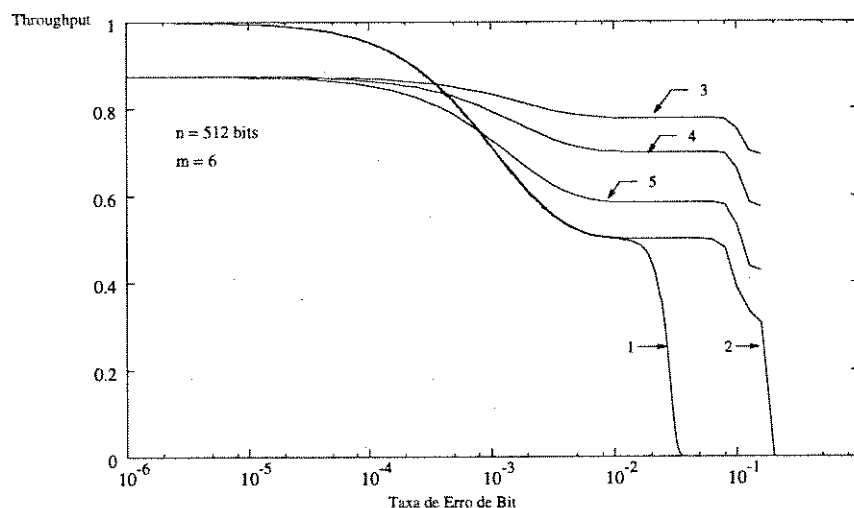


Figura 4.11: Comparação do throughput: (1) Esquema ARQ híbrido tipo-II com código convolucional de taxa  $1/2$ ; (2) Esquema ARQ híbrido tipo-II com combinação de código; (3), (4) e (5) Esquemas ARQ híbridos tipo-II com códigos puncionados de taxas iniciais iguais a  $7/8$  e  $h = 1, 2$  e  $4$ , respectivamente.

A figura 4.12 mostra todos os esquemas ARQ híbridos tipo-II analisados neste capítulo. Observa-se que o esquema ARQ com código convolucional de taxa  $2/3$ , tem um desempenho

superior aos esquemas ARQ que utilizam combinação de códigos e código convolucional de taxa 1/2. Para uma taxa de erro maior que  $10^{-2}$ , os esquemas que utilizam códigos puncionados, apresentam um desempenho superior ao esquema com código de taxa 2/3. Para uma taxa de erro de bit menor que  $10^{-2}$ , o esquema ARQ com código convolucional de taxa 2/3 tem um throughput superior ao esquema ARQ com códigos puncionados utilizando taxas de códigos iniciais de 4/5 ou 7/8.

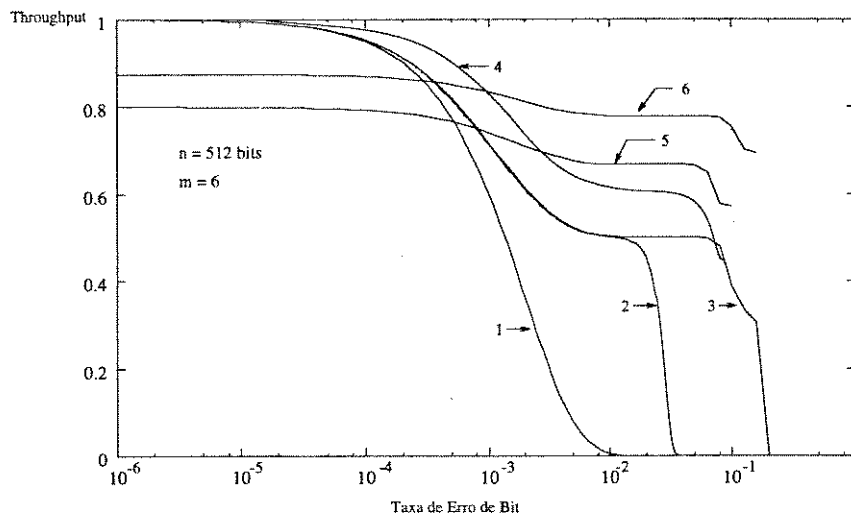


Figura 4.12: Comparação do throughput: (1) Esquema ARQ selective-repeat ideal; (2) Esquema ARQ híbrido tipo-II com código convolucional de taxa 1/2; (3) Esquema ARQ híbrido tipo-II com combinação de código; (4) Esquema ARQ híbrido tipo-II com código convolucional de taxa 2/3; (5) e (6) Esquemas ARQ híbridos tipo-II com códigos puncionados com  $h = 1$  e taxas iniciais iguais a 4/5 e 7/8, respectivamente.

## 4.7 Conclusão

O uso de códigos convolucionais no processo de correção de erro em sistemas ARQ híbridos tipo-II, tem-se mostrado bastante eficiente, especialmente quando a taxa de erro de bit do canal é muito alta, proporcionando sistemas com um desempenho significativo em termos de throughput.

O esquema ARQ que utiliza código convolucional de taxa  $1/2$  é o mais simples de implementar e trabalha apenas com as duas últimas sequências recebidas no processo de decodificação. Um parâmetro importante a ser considerado neste esquema, é a memória do codificador que, quanto maior, melhor será o desempenho, conforme mostram as figuras 4.1 e 4.2. Por outro lado, no receptor, o aumento da memória implica em um aumento da complexidade do decodificador.

No esquema ARQ com código convolucional de taxa  $2/3$ , os códigos de detecção e correção de erro são combinados em uma forma similar ao esquema com código de taxa  $1/2$ . O esquema com código de taxa  $2/3$ , trabalha com até três sequências para uma dada palavra-código no processo de retransmissão. Como consequência, o processo de decodificação só será feito se a terceira sequência recebida estiver com erro. Como a probabilidade de ocorrer um erro diminui a medida que o número de retransmissões referentes a uma dada palavra aumenta, o desempenho em termos de throughput é superior ao esquema ARQ com código convolucional de taxa  $1/2$ .

O esquema ARQ com combinação de códigos é semelhante ao esquema ARQ com código convolucional de taxa  $1/2$ . A diferença está no processo de decodificação que trabalha com todas as sequências recebidas com erro. Com isso, o número de bits de redundância aumenta, reduzindo a probabilidade de ocorrer um erro de decodificação e como resultado, há o aumento do throughput, especialmente se o canal estiver muito ruidoso.

O esquema ARQ com códigos puncionados é um esquema adaptativo, pois trabalha com códigos puncionados de diversas taxas de codificação, que possibilita o ajuste de acordo com as condições do canal; se o canal estiver muito ruidoso, utiliza-se códigos de taxas menores que são mais poderosos no processo de correção de erro, mantendo um valor de throughput significativo, ao passo que, o throughput dos demais esquemas deteriora-se rapidamente. Este esquema se difere dos demais pois o processo de decodificação é feito em cada sequência recebida.

## Capítulo 5

# CONCLUSÕES FINAIS

### 5.1 Comentários e Comparações entre os Esquemas ARQ

O desempenho dos esquemas ARQ híbridos analisados neste trabalho ( capítulos 3 e 4 ), mostraram ser superiores ao esquema ARQ selective-repeat ideal, que é o esquema que apresenta a melhor performance entre os esquemas ARQ básicos. Altos desempenhos em termos de throughput dos esquemas híbridos, se deve ao fato de que, nestes esquemas, alguns padrões de erros são corrigidos, ocasionando assim, redução no número de retransmissões.

No capítulo 3, a ênfase foi dada aos esquemas ARQ híbridos que utilizam códigos de bloco no processo de correção de erro. Primeiramente, foi mostrado o procedimento de codificação de duas categorias de esquemas ARQ híbridos: tipo-I e tipo-II. O esquema ARQ híbrido tipo-I emprega apenas um código que executa simultaneamente, os processos de detecção e correção de erro. Como consequência, um número maior de bits de paridade é adicionado em cada palavra transmitida (lembrando que nos esquemas ARQ puros, o código é utilizado apenas para o processo de detecção de erro e, por isso, o número de bits de paridade usado é menor que no esquema ARQ híbrido tipo-I). Este acréscimo de bits de paridade, resulta na redução da taxa do código ( $k/n$ ), e conseqüentemente, um menor desempenho, em termos de throughput, especialmente para baixas taxas de erro de bit do canal. Se os níveis de ruídos e interferências do canal puderem ser previstos, o esquema ARQ híbrido tipo-I pode ser uma boa opção, pois pode-se projetar um código corretor de erro de acordo com as condições do canal, obtendo

melhores desempenhos neste esquema híbrido.

Por outro lado, para um canal onde a taxa de erro de bit não é estacionária, é conveniente utilizar um esquema cuja estratégia de transmissão/retransmissão se enquadre às condições do canal no momento do processo de transmissão das palavras-códigos. Os esquemas ARQ híbridos tipo-II, apresentam esta solução. Estes esquemas, de maneira básica, utilizam dois códigos: um para o processo de detecção de erro e o outro para o processo de correção de erro. Portanto, quando a taxa de erro de bit do canal for pequena, são adicionados a cada mensagem de informação a ser transmitida, somente os bits de paridade usados no processo de detecção de erro. Se as condições do canal piorarem, ou seja, se a taxa de erro de bit do canal aumentar, pode-se usar um maior número de bits de paridade para tentar corrigir alguns padrões de erro, resultando em uma redução no número de retransmissões.

Uma característica importante no esquema ARQ híbrido tipo-II, é que o código corretor de erro é inversível, ou seja, a partir do bloco de paridade gerado por este código, é possível obter a mensagem de informação através de um processo de inversão. Como resultado, o número de retransmissões pode ser ainda menor, pois o receptor tentará primeiro recuperar a mensagem de informação a partir da palavra de paridade pelo processo de inversão, e só então, tentará obtê-la através do procedimento de correção de erro, utilizando a palavra de informação transmitida anteriormente mais a palavra de paridade. Uma outra vantagem, é que o número de bits de paridade contidos na palavra de paridade a ser utilizada no processo de retransmissão, é o mesmo usado na palavra-código formada pelo código detector de erro, uma vez que os  $k$  bits de paridade formados pelo código corretor de erro, são codificados em  $n$  bits pelo código detector de erro.

Diversas estratégias empregadas nos esquemas ARQ híbridos tipo-II foram discutidas no capítulo 3. Foi mostrado na seção 3.4.1 um tratamento para evitar overflow no buffer do receptor, considerando um esquema ARQ híbrido tipo-II com buffer de tamanho finito. O esquema ARQ híbrido tipo-II modificado difere do esquema ARQ híbrido tipo-II básico, no procedimento de correção de erro; o código corretor de erro é usado somente no processo de correção, ao contrário do que ocorre no esquema híbrido tipo-II básico, onde o código corretor de erro  $C_1$ , pode ser usado para correção e detecção de erro, o que resulta em uma redução

da capacidade de correção de erro do código. Consequentemente, o desempenho em termos de throughput do esquema modificado, é superior ao esquema ARQ híbrido tipo-II básico.

Em termos de complexidade de implementação, os dois esquemas ARQ híbridos tipo-II básico e modificado, possuem praticamente a mesma, pois ambos apresentam os mesmos procedimentos: detecção de erro baseada no código detector de erro, decodificação baseada no código corretor de erro e, finalmente, o processo de inversão.

O terceiro esquema ARQ híbrido tipo-II discutido, é uma generalização do esquema ARQ híbrido tipo-II modificado, onde o código corretor de erro, é da forma  $(mn, n)$ , sendo o valor de  $m \geq 2$ . Uma característica do código corretor de erro usado neste esquema, é que sua matriz geradora é particionada em  $m$  submatrizes de dimensão  $n \times n$ , e cada submatriz, é uma matriz geradora de um sub-código. Portanto, pode-se gerar uma palavra-código da forma  $c = (c_1|c_2|\dots|c_m)$  onde cada  $c_i$ , ( $i = 1, 2, \dots, m$ ) possui  $n$  bits e é formado a partir de uma submatriz. O processo de retransmissão de uma dada palavra, consiste em enviar cada  $c_i$  à medida que é solicitada uma nova retransmissão. O ideal é que essas submatrizes sejam inversíveis, para tentar obter a palavra de informação, antes de usar o procedimento de decodificação.

A utilização de códigos KM (vide apêndice B) no processo de correção de erro em um esquema ARQ-GH, reduz a complexidade do esquema ARQ híbrido tipo-II generalizado, com relação ao procedimento de decodificação, pois se não utilizasse este código no esquema ARQ-GH, seria necessário um circuito decodificador para cada sub-código, que resultaria em um sistema de implementação complexa, especialmente para grandes valores de  $m$ .

Os esquemas ARQ híbridos tipo-II que utilizam códigos de Hamming em cascata no processo de correção de erro, apresentam uma baixa complexidade, em termos de implementação de seus circuitos codificadores/decodificadores, pelo fato de utilizar códigos de Hamming de pequenos comprimentos. Como este esquema emprega somente um código de Hamming  $(N, K)$  para a correção de erro, é necessário apenas um decodificador no receptor, para efetuar as correções dos erros.

O procedimento de geração das palavras de paridade a partir de um código de Hamming em cascata, consiste em dividir a palavra-código de  $n$  bits, formada após a codificação pelo código detector de erro, em blocos de  $K$  bits, onde cada bloco é codificado pelo código de Hamming,



em blocos de  $N$  bits. Esta etapa, constitui o primeiro nível de cascata. O bloco de  $n(N/K)$  bits obtido após o primeiro nível de codificação, é novamente particionado em blocos de  $K$  bits, que serão codificados em blocos de  $N$  bits pelo mesmo código de Hamming  $(N, K)$ . Este processo de codificação continua até o  $i$ -ésimo nível de cascadeamento. Para que toda divisão de uma palavra por  $K$  resulte em um número inteiro em todos os níveis de cascata, é necessário que o valor  $n$  da palavra-código seja igual a  $aK^i$ , onde  $a$  é um número inteiro e  $i$  é o número de níveis de cascata empregado.

A transmissão consiste em enviar a palavra de  $n$  bits formada a partir do código detector de erro. Na primeira retransmissão é enviada a palavra formada no primeiro nível de codificação em cascata, na segunda retransmissão é enviada a palavra obtida após o segundo nível de cascata, e assim sucessivamente, até atingir o  $i$ -ésimo nível de cascata. Se alcançar o  $i$ -ésimo nível e a mensagem ainda não for obtida corretamente, a próxima retransmissão será novamente a palavra-código de  $n$  bits. Observa-se que cada palavra enviada em cada retransmissão, tem tamanhos diferentes. Como consequência, o receptor deve tratar com palavras de comprimentos variáveis, o que ocasiona um aumento da complexidade de implementação, especialmente com relação ao algoritmo de armazenamento dessas palavras no buffer do receptor.

O esquema ARQ híbrido tipo-II com códigos de Hamming em cascata, deve possuir um circuito adicional para fazer o entrelaçamento dos bits e que não existe nos outros esquemas híbridos tipo-II anteriores. Este procedimento é necessário para que os blocos obtidos sejam independentes nos diversos níveis de cascata, para que erros não sejam propagados para os níveis superiores de cascadeamento, conforme mostra a figura 3.13.

Uma consideração importante ainda com relação a este esquema, é na escolha do código detector de erro, que tem que ser da forma  $(aK^i, k)$ . Caso o valor de  $n$  não seja exatamente igual a  $aK^i$ , pode-se utilizar os conceitos de códigos encurtados ou estendidos para adequar o valor de  $n$  [Berlekamp, 1968].

Todos os esquemas ARQ híbridos tipo-II seguem basicamente o mesmo procedimento; utiliza dois códigos de bloco, um para o processo de detecção e outro para o processo de correção de erro. O esquema ARQ híbrido tipo-II com códigos de Hamming em cascata, foi o esquema que apresentou o melhor desempenho em termos de throughput, especialmente utilizando códigos de

Hamming (15,11) e (31,26), com capacidade de correção de erro  $t = 1$ . Foram feitas simulações neste esquema, utilizando códigos BCH com maiores capacidades corretoras de erro do que os códigos de Hamming, mas os resultados obtidos utilizando códigos BCH, foram inferiores aos esquemas que utilizam códigos de Hamming (15,11) e (31,26).

As figuras 5.1 e 5.2 comparam os esquemas ARQ híbridos tipo-II utilizando códigos de Hamming em Cascata com os esquemas ARQ híbridos tipo-II utilizando códigos BCH com capacidades corretoras de erro  $t > 1$ .

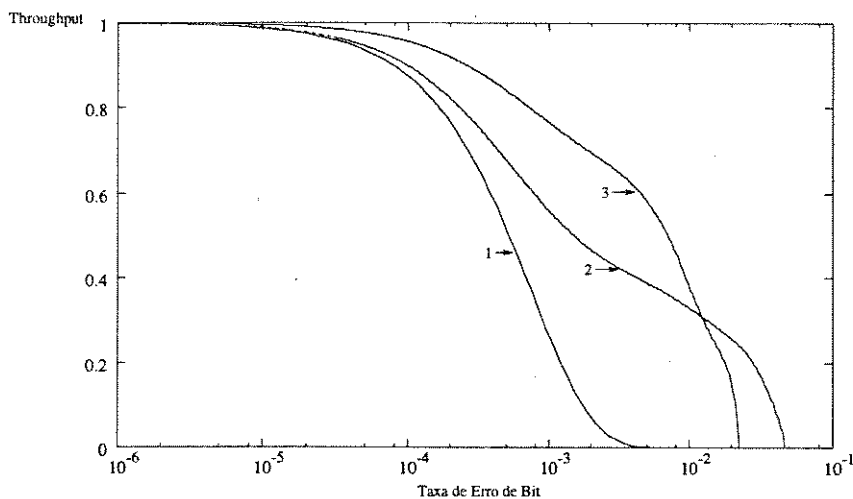


Figura 5.1: Throughput dos esquemas ARQ híbridos tipo-II com códigos em Cascata: (1) Esquema ARQ Selective-repeat ideal; (2) Código BCH (15,7) com  $t = 2$  e  $n = 1029$ ; (3) Código de Hamming (15,11) com  $t = 1$  e  $n = 1331$ .

Observa-se pela figura 5.1 que o desempenho do esquema híbrido utilizando código de Hamming (15,11) com capacidade corretora  $t = 1$ , é superior ao esquema híbrido com código BCH (15,7) e  $t = 2$ . Somente para uma taxa de erro de bit maior que  $10^{-2}$ , é que o esquema com código (15,7) supera o esquema com código (15,11).

Utilizou-se na figura 5.2, um esquema ARQ híbrido tipo-II em cascata com código BCH (31,11) e  $t = 5$ , cujo desempenho obtido em termos de throughput, é bem inferior ao esquema que utiliza o código de Hamming (31,26). Conclui-se que o esquema ARQ híbrido tipo-II em cascata, apresenta melhores resultados, quando se utiliza códigos de menores capacidades corretoras de erros, ou seja, um menor número de bits de paridade.

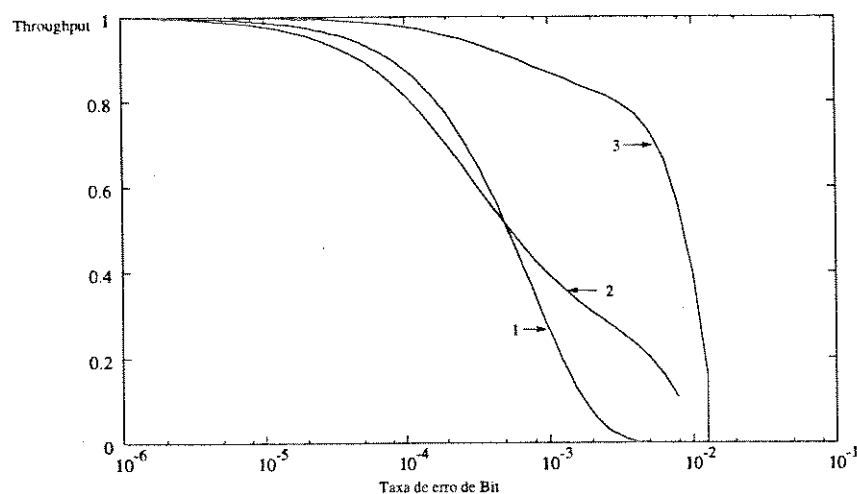


Figura 5.2: Throughput dos esquemas ARQ híbridos tipo-II com códigos em cascata: (1) Esquema ARQ Selective-repeat ideal; (2) Código BCH (31,11) com  $t = 5$  e  $n = 1331$ ; (3) Código de Hamming (31,26) com  $t = 1$  e  $n = 1352$ .

O capítulo 4 mostra a utilização dos códigos convolucionais no processo de correção de erros dos esquemas ARQ híbridos. Pode-se ter algumas vantagens com o emprego de códigos convolucionais. Primeiramente, estes códigos possuem ótimas capacidades corretoras. A segunda vantagem se refere à decodificação que, por ser um processo sequencial, uma retransmissão pode ser solicitada antes mesmo do término de uma decodificação, ou a decodificação de uma palavra pode ser interrompida caso o número de erros tenha superado a capacidade de correção do código utilizado.

Basicamente, o esquema é constituído por um código convolucional de taxa  $1/2$ , que gera duas seqüências que são utilizadas tanto no processo de transmissão, como no processo de retransmissão. Como vantagem, uma mensagem de informação a ser transmitida, é codificada por dois códigos; o código de bloco detector de erro e o código convolucional. Com isso, aumenta a proteção, mas aumenta o número de bits de paridade, para uma dada mensagem.

Um fator a ser considerado nos esquemas ARQ híbridos com código convolucional, é com relação ao tamanho da memória do codificador, que influencia no desempenho do sistema ARQ. Quanto maior for a memória do codificador, melhor será o desempenho. Verifica-se que, para uma memória do codificador de tamanho  $m \geq 10$ , o desempenho em termos de throughput

deste esquema é praticamente o mesmo, resultando em melhorias pouco significativas, ao passo que, para uma memória do codificador de tamanho grande, ocasiona uma maior complexidade de implementação do decodificador. Dependendo da taxa de erro do canal (se for muito baixa), é conveniente utilizar um codificador com um número de memórias de valor pequeno.

O emprego de códigos convolucionais de taxa  $2/3$  nos esquemas ARQ híbridos, resultaram em melhores desempenhos em termos de throughput, se comparados com o esquema ARQ híbrido com código convolucional de taxa  $1/2$ . O codificador do código convolucional  $(3, 2, m)$  no esquema ARQ híbrido tipo-II, funciona como se fosse dois codificadores de taxa  $1/2$  independentes, gerando três sequências codificadas para serem empregadas no processo de transmissão de uma dada palavra. Portanto, a decodificação só será empregada, se as três sequências forem recebidas com erro, ao passo que, no esquema ARQ híbrido tipo-II com código convolucional de taxa  $1/2$ , a decodificação é empregada após falhar a segunda tentativa de transmissão de uma palavra (ou seja, a primeira retransmissão). Como a probabilidade de ocorrer erro diminui a medida que o número de retransmissões aumentam, o desempenho do esquema ARQ com código convolucional de taxa  $2/3$  é superior ao esquema ARQ com código convolucional de taxa  $1/2$ , uma vez que o número de retransmissões para uma dada palavra, é maior para o esquema com código de taxa  $2/3$ .

Um outro esquema ARQ híbrido com códigos convolucionais, denominado de esquema ARQ híbrido tipo-II com combinação de códigos, utiliza o mesmo procedimento de transmissão empregado no esquema ARQ híbrido com código convolucional de taxa  $1/2$ . A diferença está no processo de decodificação que trabalha com todas as sequências recebidas referentes a uma dada palavra. Portanto, a cada retransmissão, a decodificação é processada por um código de repetição de taxa  $1/(2 + i)$ ,  $i = 1, 2, \dots$ , ocasionando uma redução da probabilidade de erro de decodificação e resultando numa melhoria em termos de throughput se comparar com o esquema ARQ híbrido tipo-II com código convolucional de taxa  $1/2$ .

Pelo fato do receptor armazenar todas as sequências recebidas com erro referentes a uma dada palavra, é necessário um buffer no receptor com grande capacidade de armazenamento, para tratar todas as sequências recebidas referentes a uma palavra. Se não houver uma lógica adequada para tratar com todas as sequências armazenadas, certamente ocorrerá overflow no

buffer do receptor.

O uso de códigos puncionados nos esquema ARQ híbridos tipo-II, são interessantes pelo fato de utilizar diversos códigos com diferentes taxas ( $k/n$ ) e apenas um conjunto de codificador/decodificador. Como resultado, pode-se utilizar um número reduzido de bits de paridade quando a taxa de erro de bit do canal estiver baixa, e um número maior de bits de paridade para altas taxas de erro de bit.

A figura 5.3 mostra as curvas de throughput de esquemas ARQ híbridos tipo-II que utilizam códigos de bloco e códigos convolucionais. Não se pode fazer comparações entre os esquemas que utilizam códigos de bloco com os que utilizam códigos convolucionais, pois não há parâmetros em comum para efetuar a comparação. Entretanto, observando a figura 5.3, pode-se inferir que os esquemas ARQ híbridos com códigos convolucionais, são mais apropriados para altas taxas de erro de bit, acima de  $10^{-2}$ , ao passo que, os esquemas ARQ híbridos com códigos de bloco, apresentam desempenhos superiores quando a taxa de erro de bit do canal estiver entre  $10^{-5}$  e  $10^{-3}$ .

Trabalhos recentes na área de ARQ, se concentram nos esquemas ARQ híbridos tipo-II, pois apresentam melhores desempenhos que os esquemas ARQ básicos, uma vez que, o desempenho de qualquer esquema ARQ básico, pode no máximo, se igualar ao desempenho do esquema ARQ selective-repeat com buffer do receptor de tamanho infinito. Diferentes esquemas ARQ híbridos foram analisados, tanto para códigos de bloco como para convolucionais, mostrando qual esquema melhor se enquadra às condições do canal, ou que melhor se adapte à situações que não podem ser previstas.

## 5.2 Futuros Trabalhos

Seguindo procedimento análogo ao esquema ARQ híbrido tipo-II com código convolucional de taxa  $2/3$ , pode-se empregar códigos convolucionais de taxas maiores. Como exemplo, pode ser utilizado um código convolucional de taxa  $3/4$  no esquema ARQ híbrido tipo-II no processo de correção de erro. A matriz geradora deste código pode ser definida como:

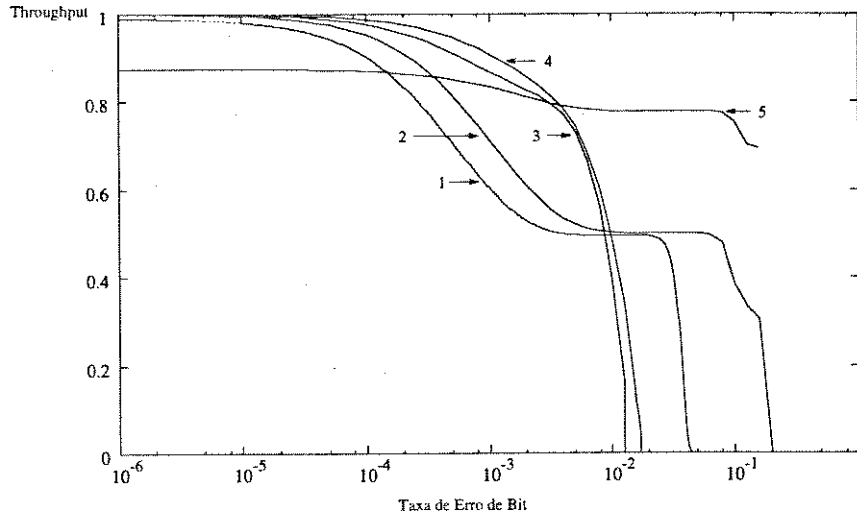


Figura 5.3: Throughput de vários esquemas ARQ híbridos tipo-II: (1) esquema ARQ híbrido tipo-II com código convolucional de taxa 1/2, memória do codificador  $m = 10$  e  $n = 512$  bits; (2) Esquema ARQ híbrido tipo-II com combinação de códigos, memória  $m = 10$  e  $n = 512$  bits; (3) e (4) Esquemas ARQ híbridos tipo-II com código de Hamming (31,26) e  $n = 1352$  e  $676$  respectivamente; (5) Esquema ARQ híbrido tipo-II com códigos convolucionais puncionados com  $h = 1$ , taxa inicial  $7/8$  e  $n = 512$ .

$$G(X) = \begin{bmatrix} G_1(X) & 0 & 0 & G_{31}(x) \\ 0 & G_2(X) & 0 & G_{32}(X) \\ 0 & 0 & G_3(X) & G_{33}(X) \end{bmatrix} \quad (5.1)$$

Empregando um código convolucional de taxa 3/4 no esquema ARQ híbrido tipo-II, pode-se utilizar no procedimento de transmissão/retransmissão de uma dada palavra, quatro sequências geradas a partir deste código:  $P_1(X) = I(X)G_1(X)$ ,  $P_2(X) = I(X)G_2(X)$ ,  $P_3(X) = I(X)G_3(X)$  e  $P_4(X) = I(X)G_{31}(X) + I(X)G_{32}(X) + I(X)G_{33}(X)$ .

Uma outra proposta para futuros trabalhos, seria explorar as mensagens de retorno enviadas pelo receptor, para transmitir informações sobre a natureza dos erros ocorridos na palavra recebida. Nestes blocos, podem estar contidas, além de informações sobre o estado da palavra recebida pelo receptor, informações sobre o estado do canal de comunicação, dado pelo

número de erros contidos nas palavras transmitidas para o receptor. Esta estratégia pode ser interessante, em esquemas que trabalhem com diversos códigos com diferentes capacidades de correção.

Pode-se ainda, de acordo com as condições do canal enviadas pelo receptor, projetar um sistema com vários códigos de blocos (códigos BCH por exemplo) que apresentem diferentes capacidades de correção. Como resultado, a exemplo do esquema ARQ híbrido tipo-II com códigos puncionados, pode-se ter um sistema que se adapte às condições do canal. Para um sistema que trabalhe com vários códigos de diferentes capacidades, mas de mesmo comprimento ( $n$ ), a mensagem de informação a ser codificada terá tamanhos variados. Como resultado, não há necessidade de uma lógica para tratar o armazenamento de palavras-código de tamanho diferentes no receptor. Se as mensagens de informação tiverem o mesmo tamanho, o sistema deverá ser capaz de tratar com palavras-códigos de diversos tamanhos, pois os códigos usados tem diversas capacidades corretoras e com isso, comprimentos variados.

Uma outra proposta de trabalho pode ser projetar um esquema ARQ híbrido tipo-II que trabalhe com códigos de bloco e convolucionais no processo de correção de erro, uma vez que o desempenho dos esquemas ARQ híbridos tipo-II que utilizam códigos de bloco cai rapidamente para taxas de erros acima de  $10^{-3}$ , ao passo que o throughput do esquema com códigos convolucional não deteriora tão rapidamente para esta taxa de erro de bit.

Como o esquema de Weldon [Weldon, 1982] apresenta desempenhos próximos ao selective-repeat ideal, pode-se utilizar um esquema ARQ híbrido baseado nesta estratégia, especialmente para altas taxas de erro de bit, onde a transmissão de várias cópias de uma dada palavra, ocasiona uma melhoria com relação ao throughput.

Por fim, uma outra sugestão seria a utilização no esquema ARQ híbrido tipo-II em cascata de códigos de Hamming de maiores comprimentos, como exemplo um código de Hamming (63,57), que pode resultar num melhor throughput, pois o esquema ARQ híbrido tipo-II que utiliza códigos de Hamming em cascata (31,26), apresenta resultados superiores a estes esquemas com códigos de Hamming (15,11) e (7,4).

### 5.3 Conclusão Final

Este trabalho é uma análise comparativa das estratégias ARQ mais relevantes encontradas na literatura, mostrando quando utilizar um determinado esquema, de acordo com as condições do canal de comunicação, da capacidade de armazenamento do buffer do receptor, e também com relação à complexidade de implementação dos algoritmos de transmissão/retransmissão. Por exemplo, se a taxa de erro do canal for muito baixa, não há necessidade de utilizar um esquema ARQ híbrido pois, o desempenho deste esquema será praticamente o mesmo de um esquema ARQ básico, que é muito menos complexo em termos de implementação.

A motivação para a realização deste trabalho foi a não existência na literatura, de um trabalho que tratasse exclusivamente de esquemas ARQ, mostrando quais esquemas são mais significativos e quais proporcionam melhores resultados de acordo com os parâmetros fixados.

Uma característica deste trabalho, foi a realização de comparações entre os diversos esquemas ARQ (especialmente em termos de throughput) sob o ponto de vista dos mesmos parâmetros, uma vez que os trabalhos existentes na área não apresentam tais comparações. Como resultado, pode-se escolher qual esquema ARQ será mais conveniente e fornecerá melhores resultados para uma determinada situação, seja o estado do canal, a complexidade de implementação dos circuitos, a lógica, etc.

As comparações se concentraram nos esquemas ARQ híbridos por estes proporcionarem melhores resultados em termos de throughput. E devido ao avanço tecnológico atual, implementar circuitos para codificação/decodificação de códigos poderosos e buffers capazes de armazenar uma grande quantidade de dados, se tornaram triviais, ao passo que, no final da década de 70 e início da década de 80, a complexidade de implementação e o tamanho da memória, eram fatores limitantes.

As comparações entre os diversos esquemas ARQ, foram feitas através de programas computacionais para o cálculo do desempenho, em termos de throughput, destes esquemas.

A tendência dos trabalhos mais recentes na área, estão na elaboração de novos esquemas ARQ híbridos tipo-II adaptativos, utilizando tanto códigos de blocos como códigos convolucionais. Além disso, existem esforços para modelar e analisar o comportamento dos diversos



esquemas ARQ em canais de comunicação com desvanescimento, como exemplo, os canais de comunicação móvel. Uma outra linha de estudo é a utilização de esquemas ARQ em modulação codificada. Enfim, os sistemas de controle de erro ARQ, tem-se expandido em diversas áreas da telecomunicação e este trabalho serve como uma referência inicial para quem deseja começar um trabalho na área de ARQ.

# Apêndice A

## Processo de Inversão

O processo de inversão pode ser conseguido por um circuito sequencial linear. Seja  $C$  um código cíclico  $(n, k)$  [Lin, 1983] com  $n - k \leq k$ . Seja  $g(X)$  o polinômio gerador de  $C$ :

$$g(X) = 1 + g_1X + g_2X^2 + \cdots + g_{n-k-1}X^{n-k-1} + X^{n-k} \quad (\text{A.1})$$

E

$$v(X) = v_0 + v_1X + v_2X^2 + \cdots + v_{n-1}X^{n-1} \quad (\text{A.2})$$

um polinômio código (ou palavra-código). Na forma sistemática os  $k$  coeficientes de maior ordem  $v_{n-k}, v_{n-k+1}, \dots, v_{n-1}$  são idênticos aos  $k$  dígitos de informação, os  $n - k$  coeficientes  $v_0, v_1, \dots, v_{n-k-1}$  são os dígitos de paridade. Considere o conjunto dos palavras-código em  $C$  cujos  $2k - n$  componentes de maior ordem são zeros. Há no código  $C$ ,  $2^{n-k}$  palavras com esta característica. Se os  $2k - n$  componentes de maior ordem iguais a zero forem removidos destas palavras-código, obtém-se um conjunto de  $2^{n-k}$  palavras de tamanho  $2(n - k)$ . Estas palavras formam um código cíclico encurtado  $C_1$  de taxa meio  $(2(n - k), n - k)$ [ver referencia shu lin,livro]. Este código obtido, tem a mesma capacidade de correção do código  $C$ .

Agora, será mostrado a propriedade de inversão do código  $C_1$ . Seja

$$u(X) = u_0 + u_1X + u_2X^2 + \cdots + u_{n-k-1}X^{n-k-1} \quad (\text{A.3})$$

a mensagem de informação a ser codificada. Dividindo  $X^{n-k}u(X)$  pelo polinômio gerador  $g(X)$ . tem-se:

$$X^{n-k}u(X) = a(X)g(X) + b(X) \quad (\text{A.4})$$

onde  $\mathbf{a}(X)$  e  $\mathbf{b}(X)$  são o quociente e o resto, respectivamente. A palavra-código referente à mensagem  $\mathbf{u}(X)$  é:

$$\mathbf{w}(X) = \mathbf{b}(X) + X^{n-k}\mathbf{u}(X) \quad (\text{A.5})$$

onde o polinômio  $\mathbf{b}(X)$  é a parte de verificação de paridade. O teorema abaixo, prova a propriedade de inversão do código  $C_1$ .

**Teorema 1.** Sejam quaisquer duas palavras-código pertencentes ao código inversível de taxa meio  $C_1$ , estas palavras-código nunca terão os mesmos dígitos de paridade.

*Prova.* Sejam duas mensagens distintas  $\mathbf{u}_1(X)$  e  $\mathbf{u}_2(X)$ . Dividindo  $X^{n-k}\mathbf{u}_1(X)$  e  $X^{n-k}\mathbf{u}_2(X)$  pelo polinômio gerador  $\mathbf{g}(X)$ , obtém-se:

$$X^{n-k}\mathbf{u}_1(X) = \mathbf{a}_1(X)\mathbf{g}(X) + \mathbf{b}_1(X), \quad (\text{A.6})$$

$$X^{n-k}\mathbf{u}_2(X) = \mathbf{a}_2(X)\mathbf{g}(X) + \mathbf{b}_2(X), \quad (\text{A.7})$$

Portanto, as palavras-código para  $\mathbf{u}_1(X)$  e  $\mathbf{u}_2(X)$  são:

$$\mathbf{w}_1(X) = \mathbf{b}_1(X) + X^{n-k}\mathbf{u}_1(X), \quad (\text{A.8})$$

$$\mathbf{w}_2(X) = \mathbf{b}_2(X) + X^{n-k}\mathbf{u}_2(X), \quad (\text{A.9})$$

respectivamente. Suponha que

$$\mathbf{b}_1(X) = \mathbf{b}_2(X) = \mathbf{b}(X). \quad (\text{A.10})$$

Adicionando a equação A.6 a A.7 e usando a equação A.10, obtém-se:

$$[\mathbf{u}_1(X) + \mathbf{u}_2(X)]X^{n-k} = [\mathbf{a}_1(X) + \mathbf{a}_2(X)]\mathbf{g}(X). \quad (\text{A.11})$$

Como  $\mathbf{g}(X)$  e  $X^{n-k}$  são primos entre si,  $\mathbf{u}_1(X) + \mathbf{u}_2(X)$  deve ser divisível por  $\mathbf{g}(X)$ . Entretanto, isto é impossível pois  $\mathbf{u}_1(X) + \mathbf{u}_2(X) \neq 0$  e seu grau é menor que  $n - k$ , mas o grau de  $\mathbf{g}(X)$  é  $n - k$ . Portanto,  $\mathbf{b}_1(X) \neq \mathbf{b}_2(X)$

C.Q.D.

Como o resto  $\mathbf{b}(X)$  resultante da divisão da equação de  $X^{n-k}\mathbf{u}(X)$  por  $\mathbf{g}(X)$  é único, o teorema 1 mostra que há uma correspondência 1 para 1 entre uma mensagem  $\mathbf{u}(X)$  e seu

polinômio de verificação de paridade  $\mathbf{b}(X)$ . Portanto, conhecendo  $\mathbf{b}(X)$ ,  $\mathbf{u}(X)$  é unicamente determinado.

Para se obter a mensagem  $\mathbf{u}(X)$  do polinômio de paridade  $\mathbf{b}(X)$ , multiplica-se ambos os lados da equação A.4 por  $X^k$ :

$$X^n \mathbf{u}(X) = \mathbf{a}(X)\mathbf{g}(X)X^k + \mathbf{b}(X)X^k. \quad (\text{A.12})$$

Rearranjando a equação A.12, obtém-se:

$$\mathbf{u}(X)(X^n + 1) + \mathbf{u}(X) = \mathbf{a}(X)\mathbf{g}(X)X^k + \mathbf{b}(X)X^k. \quad (\text{A.13})$$

Como  $\mathbf{g}(X)$  é um fator de  $X^n + 1$ , a equação A.13 pode ser reescrita como:

$$\mathbf{b}(X)X^k = [\mathbf{u}(X)\mathbf{h}(X) + \mathbf{a}(X)X^k] \mathbf{g}(X) + \mathbf{u}(X), \quad (\text{A.14})$$

onde  $\mathbf{h}(X) = (X^n + 1)/\mathbf{g}(X)$ . Da equação A.14, tem-se que a mensagem  $\mathbf{u}(X)$  é simplesmente o resto da divisão de  $\mathbf{b}(X)X^k$  pelo polinômio gerador  $\mathbf{g}(X)$ . Para obter  $\mathbf{u}(X)$ , usa-se um circuito de divisão com conexão de feedback baseado em  $\mathbf{g}(X)$ . Este processo para determinar  $\mathbf{u}(X)$  a partir dos dígitos de paridade  $\mathbf{b}(X)$  é denominado de processo de inversão.

Um circuito mais rápido, para inverter  $\mathbf{b}(X)$  pode ser implementado da seguinte forma. Dividindo  $X^k$  por  $\mathbf{g}(X)$ , tem-se:

$$X^k = \mathbf{c}(X)\mathbf{g}(X) + \mathbf{r}(X) \quad (\text{A.15})$$

onde o resto

$$\mathbf{r}(X) = r_0 + r_1X + \dots + r_{n-k-2}X^{n-k-2} + r_{n-k-1}X^{n-k-1}. \quad (\text{A.16})$$

Multiplicando ambos os lados da equação A.15 por  $\mathbf{b}(X)$  e utilizando a equação A.14, obtém-se:

$$\mathbf{b}(X)\mathbf{r}(X) = [\mathbf{u}(X)\mathbf{h}(X) + \mathbf{a}(X)X^k + \mathbf{b}(X)\mathbf{c}(X)] \mathbf{g}(X) + \mathbf{u}(X). \quad (\text{A.17})$$

A expressão acima mostra que a mensagem  $\mathbf{u}(X)$  é obtida multiplicando  $\mathbf{b}(X)$  por  $\mathbf{r}(X)$  e dividindo este produto por  $\mathbf{g}(X)$ . A figura A.1 mostra como o circuito pode ser implementado.

Por exemplo, considere um código BCH (1023, 523). Este código tem distância mínima igual a 111. Este código pode ser usado simultaneamente para corrigir todas as combinações

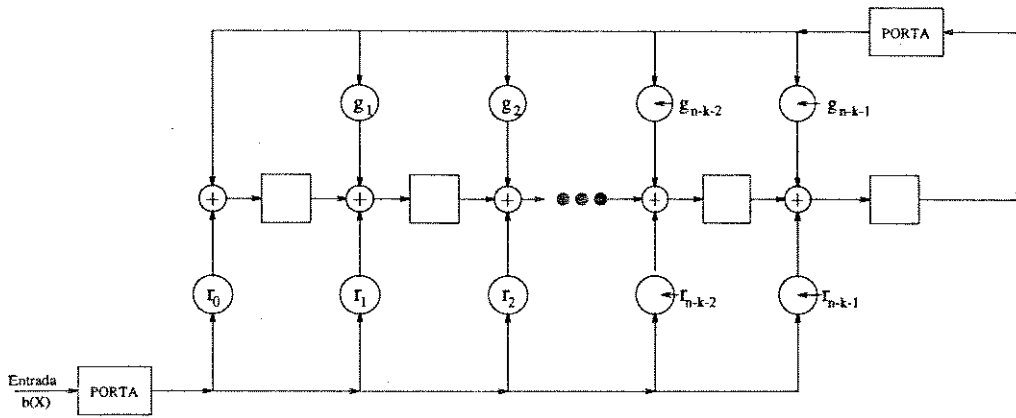


Figura A.1: Circuito de Inversão

de 5 erros ou menos e detectar todas as combinações de 105 ou menos erros. Neste caso, a probabilidade de não se detectar um erro é extremamente pequena. Reduzindo este código em 23 dígitos, obtém-se um código inversível (1000,500).

# Apêndice B

## Descrição dos Códigos $KM$

Considere uma convolução aperiódica de duas sequências  $z_j, j = 0, 1, \dots, k - 1$  de tamanho  $k$  e  $y_l, l = 0, 1, \dots, d - 1$  de tamanho  $d$ . Seja a sequência resultante  $\Phi_i, i = 0, 1, \dots, N - 1$ , onde  $N = k + d - 1$ . Se  $P[Qz \times Ry]$  for o algoritmo para a convolução aperiódica, então a matriz  $Q^T$  representa a matriz geradora de um código linear  $(n, k, \bar{d})$  onde  $\bar{d} \geq d$ . As matrizes  $P, Q$  e  $R$  tem dimensões  $(N \times n), (n \times k)$  e  $(n \times d)$ , respectivamente. A multiplicação de sequências é feita componente a componente. O tamanho  $n$  do código, é igual à complexidade multiplicativa do algoritmo. As palavras coluna  $z$  e  $y$  são definidas como as palavras  $(z_0, z_1, \dots, z_{k-1})^T$  e  $(y_0, y_1, \dots, y_{d-1})^T$ , respectivamente. O polinômio gerador de uma sequência  $x_r$ , pode ser representada como:

$$X(u) = \sum_{r=0}^{n-1} x_r u^r \quad (\text{B.1})$$

Seja  $Z(u), Y(u)$  e  $\Phi(u)$  os polinômios das sequências  $z_j, y_l$  e  $\Phi_i$ , respectivamente. Então a convolução aperiódica pode ser obtida como o produto:

$$\Phi(u) = Z(u)Y(u) \quad (\text{B.2})$$

Se  $N - 1 = k + d - 2$ , então  $\Phi(u)$  pode ser definido módulo um polinômio  $P(u)$  de grau pelo menos  $N$ , isto é,

$$\Phi(u) = Z(u)Y(u) \text{ mod } P(u), \text{ grau } [P(u)] \geq N. \quad (\text{B.3})$$

Se  $P(u)$  for o produto de  $t$  polinômios primos  $P_i(u), i = 1, 2, \dots, t$ , isto é,  $P(u) = \prod_{i=1}^t P_i(u)$ , então  $\Phi(u)$  pode ser computado, primeiramente reduzindo os polinômios  $Z(u)$  e  $Y(u)$  modulo

$P_i(u)$ ,

$$\begin{aligned}Z_i(u) &\equiv Z(u) \pmod{P_i(u)} \\Y_i(u) &\equiv Y(u) \pmod{P_i(u)}\end{aligned}\tag{B.4}$$

para  $i = 1, 2, \dots, t$ . O polinômio  $\Phi(u)$  é então obtido computando os  $t$  produtos polinomiais  $Z_i(u)Y_i(u) \pmod{P_i(u)}$  e usando o Teorema Chinês do Resto pra reconstruir  $\Phi(u)$  dos produtos módulo  $P_i(u)$  [Krishna, 1987].

## Referências

- [Adachi, 1989] F. Adachi, S. Ito and K. Ohno, "Performance Analysis of a Time Diversity ARQ in Land Mobile Radio," *IEEE Transactions on Communications*, VOL.COM-37, n<sup>o</sup>. 2, February 1989.
- [Anagnostou, 1986] Miltiades E. Anagnostou and Emmanuel N. Protonotarios, "Performance Analysis of the Selective Repeat ARQ Protocol," *IEEE Transactions on Communications*, VOL.COM-34, n<sup>o</sup>. 2, February 1986.
- [Begin, 1989] Guy Begin and David Haccoun, "High-Rate Punctured Convolutional Codes: Structure Properties and Construction Technique," *IEEE Transactions on Communications*, VOL. 37, n<sup>o</sup>. 12, November 1989.
- [Benelli, 1985] Giuliano Benelli, "An ARQ Scheme with Memory and Soft Error Detectors," *IEEE Transactions on Communications*, VOL. COM-33, n<sup>o</sup>. 3, March 1985.
- [Benelli, 1987] Giuliano Benelli, "An ARQ Scheme with Memory and Integrated Modulation," *IEEE Transactions on Communications*, VOL. COM-35, n<sup>o</sup>. 7, July 1987.
- [Benice, 1964] R. J. Benice and A. H. Frey, JR., "An Analysis of Retransmission Systems," *IEEE Transactions on Communication Technology*, December 1964.
- [Berlekamp, 1968] E. R. Berlekamp, *Algebraic Coding Theory*. New York; McGraw-Hill, 1968.
- [Biersack, 1993] Ernst W. Biersack, "Performance of the IEEE 802.2 Type-2 Logical Link Protocol with Selective Retransmission," *IEEE Transactions on Communications*, VOL. 41, n<sup>o</sup>. 2, February 1993.



- [Bruneel, 1986] Herwig Bruneel and Marc Moeneclaey, "On the Throughput Performance of Some Continuous ARQ Strategies with Repeated Transmissions," *IEEE Transactions on Communications*, VOL. COM-34, n<sup>o</sup>. 3, March 1986.
- [Burton, 1972] H. O. Burton and D. D. Sullivan, "Error and Error Control," *Proceedings of the IEEE*, VOL. 60, n<sup>o</sup>. 11, November 1972.
- [CCITT, 1980] CCITT: Recommendation X.25, "Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode on public data networks," with Plenary Assembly, Doc, n<sup>o</sup>. 7, Geneva, 1980.
- [Chandran, 1992] S. Ram Chandran and Shu Lin, "Selective-Repeat ARQ Schemes for Broadcast Links," *IEEE Transactions on Communications*, VOL. 40, n<sup>o</sup>. 1, January 1992.
- [Chang, 1984] Yet Chang and Cyril Leung, "On Weldon's Strategy," *IEEE Transactions on Communications*, VOL. COM-32, n<sup>o</sup>. 3, March 1984.
- [Chang, 1993] Jin-Fu Chang and Tzu-Hui Yang, "Multichannel ARQ Protocols," *IEEE Transactions on Communications*, VOL. 41, n<sup>o</sup>. 4, April 1993.
- [Conan, 1984] Jean Conan, "The Weight Spectra of Some Short Low-Rate Convolutional Codes," *IEEE Transactions on Communications*, VOL. COM-32, n<sup>o</sup>. 9, September 1984.
- [Drukarev, 1982] Alexander Drukarev and Daniel J. Costello JR., "A Comparison of Block and Convolutional Codes in ARQ Error Control Schemes," *IEEE Transactions on Communications*, VOL. COM-30, n<sup>o</sup>. 11, November 1982.
- [Drukarev, 1983] Alexander Drukarev and Daniel J. Costello JR., "Hybrid ARQ Error Control Using Sequential Decoding," *IEEE Transactions on Information Theory*, VOL. 29, n<sup>o</sup>. 4, July 1983.
- [Du, 1988] Du Jun Du, Masao Kasahara and Toshihiko Namekawa, "Separable Codes on Type-II Hybrid ARQ Systems," *IEEE Transactions on Information Theory*, VOL. 36, n<sup>o</sup>. 10, October 1988.

- [Easton, 1981] Malcolm C. Easton, "Design Choices for Selective-Repeat Retransmission Protocols," *IEEE Transactions on Communications*, VOL. COM-29, n<sup>o</sup>. 7, July 1981.
- [Fantacci, 1990] Romano Fantacci, "Performance Evaluation of Efficient Continuous ARQ Protocols," *IEEE Transactions on Communications*, VOL. 38, n<sup>o</sup>. 6, June 1990.
- [Fantacci, 1991] Romano Fantacci, "Performance Evaluation of Some ARQ Schemes Using Efficient Modulation Techniques and Noncoherent Detection," *IEEE Transactions on Communications*, VOL. 39, n<sup>o</sup>. 3, March 1991.
- [Haccoun, 1989] David Haccoun and Guy Begin, "High-Rate Punctured Convolutional Codes for Viterbi and Sequential Decoding," *IEEE Transactions on Communications*, VOL. 37, n<sup>o</sup>. 11, November 1989.
- [Hagenauer, 1988] Joachim Hagenauer, "Rate-Compatible Punctured Convolutional Codes (RCPC Codes) and their Applications," *IEEE Transactions on Communications*, VOL.36, n<sup>o</sup>. 4, April 1988.
- [Kallel, 1988] Samir Kallel and David Haccoun, "Sequential Decoding with ARQ and Code Combining: A Robust Hybrid FEC/ARQ System," *IEEE Transactions on Communications*, VOL. COM-36, n<sup>o</sup>. 7, July 1988.
- [Kallel, 1990-A] Samir Kallel, "Analysis of a Type II Hybrid ARQ Scheme with Code Combining," *IEEE Transactions on Communications*, VOL. COM-38, n<sup>o</sup>. 8, August 1990.
- [Kallel, 1990-B] Samir Kallel and David Haccoun, "Generalized type II Hybrid ARQ Scheme Using Punctured Convolutional Coding," *IEEE Transactions on Communications*, VOL. COM-38, n<sup>o</sup>. 11, November 1990.
- [Kallel, 1991] Samir Kallel and David Haccoun, "Sequential Decoding with an Efficient Partial Retransmission ARQ Strategy," *IEEE Transactions on Communications*, VOL. 39, n<sup>o</sup>. 2, February 1991.

- [Kallel, 1992] Samir Kallel, "Analysis of Memory and Incremental Redundancy ARQ Schemes Over a Nonstationary Channel," *IEEE Transactions on Communications*, VOL. 40, n<sup>o</sup>. 9, September 1992.
- [Kallel, 1994] Samir Kallel, "Efficient Hybrid ARQ Protocols with Adaptive Forward Error Correction," *IEEE Transactions on Communications*, VOL. 42, n<sup>o</sup>. 2/3/4, February/March/April 1994.
- [Kasami, 1982] T. Kasami, T. Klove, and S. Lin, "On the Probability of Undetected Error of Linear Block Codes," GLOBECOM '82, Conf. Rec., E7.4, Miami, FL, Nov. 29-Dec. 2, 1982.
- [Kim, 1992] S. R. Kim and C. K. Un, "Throughput Analysis for Two ARQ Schemes Using Combined Transition Matrix," *IEEE Transactions on Communications*, VOL. 40, n<sup>o</sup>. 11, November 1992.
- [Konheim, 1980] Alan G. Konheim, "A Queueing Analysis of Two ARQ Protocols," *IEEE Transactions on Communications*, VOL. COM-28, n<sup>o</sup>. 7, July 1980.
- [Korzihik, 1965] V. I. Korzhik, "Bounds on undetected error probability and optimum group codes in a channel with feedback," *Radio-Technika*, n<sup>o</sup>. 20, VOL. 1, pp 27-33, 1965.
- [Kousa, 1991] Maan A. Kousa and Mushfiqur Rahman, "An Adaptive Error Control System Using Hybrid ARQ Scheme," *IEEE Transactions on Communications*, VOL 29, n<sup>o</sup>. 10, July 1991.
- [Krishna, 1987] Hari Krishna and Salvatore D. Morgera, "A New Error Control Scheme for Hybrid ARQ Systems," *IEEE Transactions on Communications*, VOL. COM-35, n<sup>o</sup>. 10, October 1987.
- [Lau, 1986] C. Lau and C. Leung, "Performance Analysis of a Memory ARQ Scheme with Soft Decision Detectors," *IEEE Transactions on Communications*, VOL. COM-34, n<sup>o</sup>. 8, August 1986.

- [Leung, 1981] Cyril S. K. Leung and Albert Lam, "Forward Error Correction for an ARQ Scheme," *IEEE Transactions on Communications*, VOL. COM-29, n<sup>o</sup>. 10, October 1981.
- [Leung, 1988] C. H. C. Leung, Y. Kikumoto and S. A. Sorensen, "The Throughput Efficiency of the Go-Back- $N$  ARQ Scheme Under Markov and Related Error Structures," *IEEE Transactions on Communications*, VOL. COM-36, n<sup>o</sup>. 2, February 1988.
- [Lin, 1982] Shu Lin and Philip. S. Yu, "A Hybrid ARQ Scheme with Parity Retransmission for Error Control of Satellite Channels," *IEEE Transactions on Communications*, VOL. COM-30, n<sup>o</sup>. 7, July 1982.
- [Lin, 1983] Shu Lin and Daniel J. Costello, Jr, *Error Control Coding*, Prentice-Hall, Inc. Englewood Cliffs, New Jersey 07632, 1983.
- [Lin, 1984] Shu Lin, Daniel J. Costello, Jr and Michael J. Miller, "Automatic-Repeat-Request Error-Control Schemes," *IEEE Communications Magazine*, VOL. 22, n<sup>o</sup>. 12, December, 1984.
- [Lin, 1991] Mao-Chao Lin and Mao-Yuan Guu, "The Performance Analysis of a Concatenated ARQ Scheme Using Parity Retransmission," *IEEE Transactions on Communications*, VOL. 39, n<sup>o</sup>. 12, December 1991.
- [Lu, 1989] Deng-Lin Lu and Jin-Fu Chang, "Analysis of ARQ Protocols via Signal Flow Graphs," *IEEE Transactions on Communications*, VOL. 37, n<sup>o</sup>. 3, March 1989.
- [Lugand, 1989] Laurent R. Lugand, Daniel J. Costello, JR. and Robert H. Deng, "Parity Retransmission Hybrid ARQ Using Rate 1/2 Convolutional Codes on a Nonstationary Channel," *IEEE Transactions on Communications*, VOL. 37, n<sup>o</sup>. 7, July 1989.
- [Mac, 1977] F.J. Mac Williams and N. J. A. Sloane, *Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.
- [Martins, 1990] Joaquim Arnaldo C. Martins and Jorge de Carvalho Alves, "ARQ Protocols with Adaptive Block Size Perform Better Over a Wide Range of Bit Error Rates," *IEEE Transactions on Communications*, VOL. 38, n<sup>o</sup>. 6, June 1990.

- [Mase, 1983] Kenichi Mase, Toyofumi Takenaka, Hisao Yamamoto and Masaaki Shinohara, "Go-Back- $N$  ARQ Schemes for Point-to-Multipoint Satellite Communications," *IEEE Transactions on Communications*, VOL.COM-31, n<sup>o</sup>. 4, April 1983.
- [Metzner, 1985] John J. Metzner and Deshan Chang, "Efficient Selective Repeat ARQ Strategies for Very Noisy and Fluctuating Channels," *IEEE Transactions on Communications*, VOL.COM-33, n<sup>o</sup>. 5, May 1985.
- [Miller, 1981] Michael J. Miller and Shu Lin, "The Analysis of Some Selective-Repeat ARQ Schemes with Finite Receiver Buffer," *IEEE Transactions on Communications*, VOL. COM-29, n<sup>o</sup>. 9, September 1981.
- [Moeneclaey, 1986] Marc Moeneclaey, Herwig Bruneel, Ignace Bruyland and Doo-Young Chung, "Throughput Optimization for a Generalized Stop-and-Wait ARQ," *IEEE Transactions on Communications*, VOL. COM-34, n<sup>o</sup>. 2, February 1986.
- [Morgera, 1989] Salvatore D. Morgera and Vitalice K. Oduol, "Soft-Decision Applied to the Generalized Type-II Hybrid ARQ Scheme," *IEEE Transactions on Communications*, VOL. 37, n<sup>o</sup>. 4, April 1989.
- [Morris, 1978] Joel M. Morris, "On Another Go-Back- $N$  ARQ Technique for High Error Rate Conditions," *IEEE Transactions on Communications*, VOL. COM-26, n<sup>o</sup>. 1, January 1978.
- [Oduol, 1993] Vitalice K. Oduol and Salvatore D. Morgera, "Performance Evaluation of the Generalized Type-II Hybrid ARQ Scheme with Noisy Feedback on Markov Channels," *IEEE Transactions on Communications*, VOL. 41, n<sup>o</sup>. 1, January 1993.
- [Proakis, 1983] J. G. Proakis, *Digital Communications*. New York:McGraw-Hill, 1983.
- [Pursley, 1989] Michael B. Pursley and Stuart D. Sandeberg, "Delay and Throughput for Three Transmission Schemes in Packet Radio Networks," *IEEE Transactions on Communications*, VOL. 37, n<sup>o</sup>. 12, December 1989.

- [Pursley, 1992] Michael B. Pursley and Stuart D. Sandeberg, "Variable-Rate Hybrid ARQ for Meteor-Burst Communications," *IEEE Transactions on Communications*, VOL. 40, n<sup>o</sup>. 1, January 1992.
- [Raychaudhuri, 1984] Dipankar Raychaudhuri, "ALOHA with Multipacket Messages and ARQ-Type Retransmission Protocols - Throughput Analysis," *IEEE Transactions on Communications*, VOL. COM-32, n<sup>o</sup>. 2, February 1984.
- [Rice, 1992] Michael D. Rice and Stephen B. Wicker, "Modified Majority Logic Decoding of Cyclic Codes in Hybrid-ARQ Systems," *IEEE Transactions on Communications*, VOL. 40, n<sup>o</sup>. 9, September 1992.
- [Rosberg, 1990] Zvi Rosberg and Moshe Sidi, "Selective-Repeat ARQ: The Joint Distribution of the Transmitter and the Receiver Resequencing Buffer Occupancies," *IEEE Transactions on Communications*, VOL. 38, n<sup>o</sup>. 9, September 1990.
- [Sastry, 1975] A. R. K. Sastry, "Improving Automatic Repeat-Request (ARQ) Performance on Satellite Channels Under High Error Rate Conditions," *IEEE Transactions on Communications*, n<sup>o</sup>. 4, April 1975.
- [Shacham, 1983] Nachum Shacham, "Performance of ARQ with Sequential Decoding Over One-Hop and Two-Hop Radio Links," *IEEE Transactions on Communications*, VOL. COM-31, n<sup>o</sup>. 10, October 1983.
- [Shacham, 1984] Nachum Shacham, "ARQ with Sequential Decoding of Packetized Data: Queueing Analysis," *IEEE Transactions on Communications*, VOL. COM-32, n<sup>o</sup>. 10, October 1984.
- [Shacham, 1991] Nachum Shacham and Don Towsley, "Resequencing Delay and Buffer Occupancy in Selective Repeat ARQ with Multiple Receivers," *IEEE Transactions on Communications*, VOL. 39, n<sup>o</sup>. 6, June 1991.
- [Shacham, 1992] Nachum Shacham and Byung Cheol Shin, "A Selective-Repeat-ARQ Protocol

- for Parallel Channels and Its Resequencing Analysis," *IEEE Transactions on Communications*, VOL. 40, n<sup>o</sup>. 4, April 1992.
- [Shiozaki, 1991] Akira Shiozaki, Kiyoshi Okuno, Katsufumi Suzuki and Tetsuro Segawa, "A Hybrid ARQ Scheme with Adaptive Forward Error Correction for Satellite Communications," *IEEE Transactions on Communications*, VOL. 39, n<sup>o</sup>. 4, April 1991.
- [Therrian, 1984] C. W. Therrian and K. Fukunaga, "Propertiers of Separable Covariance Matrices and their Asociated Gaussian Random Proceses," *IEEE Transactions Pattern Anal. machine Intell.*, VOL. PAMI-6, September 1984.
- [Towsley, 1981] Don Towsley, "A Statistical Analysis of ARQ Protocols Operating in a Nonindependent Error Environment," *IEEE Transactions on Communications*, VOL. com-29, n<sup>o</sup>. 7, July 1981.
- [Turney, 1981] P. F. Turney, "An Improved Stop-and-Wait ARQ Logic for Data Transmission in Mobile Radio Systems," *IEEE Transactions on Communications*, VOL. COM-29, n<sup>o</sup>. 1, January 1981.
- [Viterbi, 1971] A. J. Viterbi, "Convolutcional Codes and their Performance in Communications Systems," *IEEE Transactions on Communications*, VOL. COM-19, October 1971.
- [Wang, 1983] Yu-Ming Wang and Shu Lin, "A Modified Selective-Repeat Type-II Hybrid ARQ System and Its Performance Analysis," *IEEE Transactions on Communications*, VOL. COM-31, pp. 593-607, May 1983.
- [Weldon, 1982] E. J. Weldon, Jr, "An Improved Selective-repeat ARQ Strategy," *IEEE Transactions on Communications*, VOL. COM-30, n<sup>o</sup>. 3, March 1982.
- [Wicker, 1991] Stephen B. Wicker, "Adptive Rate Error Control Through the Use of Diversity Combining and Majority-Logic Decoding in a Hybrid-ARQ Protocol," *IEEE Transactions on Communications*, VOL. 39, n<sup>o</sup>. 3, March 1991.

- [Yamamoto, 1980] Hirosuke Yamamoto and Kohji Itoh, "Viterbi Decoding Algorithm for Convolutional Codes with Repeat Request," *IEEE Transactions on Information Theory*, VOL. 26, n<sup>o</sup>. 5, September 1980.
- [Yu, 1981] P. S. Yu and Shu Lin, "An Efficient Selective-Repeat ARQ for Satellite Channels and Its Throughput Analysis," *IEEE Transactions on Communications*, VOL. COM-29, pp. 353-363, March 1981.