

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA**

André José Silveira

**CÓDIGO FONTANAL DIGITAL COM JANELA
DESLIZANTE SUAVE PARA TRANSMISSÃO DE CONTEÚDOS
MULTIMÍDIA**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina para a obtenção do grau de Mestre em Engenharia Elétrica. Orientador: Prof. Bartolomeu Ferreira Uchôa Filho, Ph.D .

Florianópolis

2016

**Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.**

Silveira, André José

Código fontanal digital com janela deslizante suave para transmissão de conteúdos multimídia / André José Silveira ; orientador, Bartolomeu Ferreira Uchôa Filho - Florianópolis, SC, 2016.

70 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Elétrica.

Inclui referências

1. Engenharia Elétrica. 2. Códigos Fontanis. 3. Luby Transform. 4. LT. I. Uchôa Filho, Bartolomeu Ferreira. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

André José Silveira

**CÓDIGO FONTANAL DIGITAL COM JANELA
DESLIZANTE SUAVE PARA TRANSMISSÃO DE CONTEÚDOS
MULTIMÍDIA**

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica, área de concentração Comunicações e Processamento de Sinais, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.

Florianópolis, 09 de dezembro de 2016.

Prof. Marcelo Lobo Heldwein, Ph.D.
Coordenador do Programa de Pós-Graduação em Engenharia
Elétrica

Banca examinadora

Prof. Bartolomeu Ferreira Uchôa Filho, Ph.D.
Universidade Federal de Santa Catarina

Prof. Danilo Silva, Ph.D.
Universidade Federal de Santa Catarina

Prof. Roberto Wanderley da Nóbrega, Dr.
Instituto Federal de Santa Catarina

Stevan Grubisic, Dr.
Agência Nacional de Telecomunicações - ANATEL

Para minha mãe Rosa e minhas filhas Elisa, Helena e Ana Sophia.

Agradecimentos

Quero reservar este espaço para agradecer e expressar meu reconhecimento às pessoas que estiveram comigo ao longo desta jornada, apoiando, sendo paciente, e nos momentos mais difíceis dando uma palavra de conforto ou um abraço. E, em especial:

a minha querida mãe por seu amor, carinho e apoio incondicional, e por sempre acreditar na educação.

aos meus grandes amores, minhas filhas. Gratidão pela alegria e pela luz que vocês são na minha vida. Peço perdão pelos momentos que me fiz ausente. Vocês estão sempre no meu coração.

a minha família de criação. Mãe, irmãos, irmãs e sobrinhos pelo seu apoio e ajuda nos momentos difíceis durante este trabalho.

ao meu professor, orientador e amigo Bartolomeu Ferreira Uchôa Filho. Gratidão pelos ensinamentos, conselhos, conversas e por sua paciência. Foi uma honra e um privilégio ser seu aluno e orientado. Mais do que os ensinamentos acadêmicos, levo para minha vida um exemplo de ser humano a ser seguido.

aos colegas da Agência Nacional de Telecomunicações–ANATEL, por suprirem minha ausência nas licenças e afastamentos. Ao Gerente da Unidade Operacional de Florianópolis, Estevão Hobold, e ao Coordenador da Fiscalização, Jonir Alfredo dos Santos, pelo apoio e liberação para participação no mestrado. Ao amigo Stevan Grubisic pelo incentivo para que eu iniciasse o mestrado.

a todos os demais professores e servidores do programa de pós-graduação que contribuíram para o meu processo de desenvolvimento no mestrado.

aos colegas de laboratório Rodrigo Farias, Ricardo Bohaczuk Venturelli, Paulo Ricardo Branco Silva, Roberto Wanderley da Nóbrega, José Clair Menezes Junior, Simara Sonaglio, Juliana Camilo Inácio, Maria Cláudia de Almeida Castro, Márcio Henrique Doniak e

Robinson Pizzio pela amizade. Ao amigo Bruno Fontana da Silva pela ajuda em algumas simulações no MATLAB.

a todos os demais amigos e familiares que estiveram ao meu lado durante este ciclo tão importante na minha vida. Agradeço a todos de coração. Muito obrigado.

"A mente que se abre a uma nova ideia, jamais voltará ao seu tamanho original."

Albert Einstein

Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica

CÓDIGO FONTANAL DIGITAL COM JANELA DESLIZANTE
SUAVE PARA TRANSMISSÃO DE CONTEÚDOS MULTIMÍDIA

André José Silveira

09 de dezembro de 2016

Orientador: Bartolomeu Ferreira Uchôa Filho

Área de concentração: Comunicação e Processamento de Sinais

Palavras-chave: Códigos Fontanais, Luby Transform, LT

Número de páginas: 87

Técnicas de transmissão de áudio, vídeo e imagem por difusão através de um canal que pode ser modelado como um canal com apagamento têm evoluído muito na comunidade científica. O uso de códigos *Luby Transform (LT)*, *Raptor* e *Tornado*, pertencentes à família dos códigos fontanais, são um exemplo desta evolução. Também tem havido um grande esforço por parte da comunidade científica para melhorar a qualidade de experiência dos usuários no momento da transmissão de áudio, vídeo e imagem. Possibilitar, em tempo de transmissão, que os usuários ouçam o áudio, ou vejam o vídeo ou a imagem que estão sendo transmitidos é um grande desafio para os atuais sistemas de transmissão. O código LT padrão (tradicional) proposto por Luby em 2002 só permite que a informação ou o conteúdo de mídia seja acessado após recuperar toda a informação da fonte. Esta característica

se deve ao fato de os K pacotes da fonte serem selecionados para compor o pacote de paridade conforme uma distribuição uniforme. Logo, no decodificador não haverá um sequenciamento ou ordenamento dos pacotes recuperados, impossibilitando assim o acesso parcial ao conteúdo de mídia, por exemplo. Em 2007, Bogino et al. Propuseram um esquema de transmissão para o código LT utilizando uma janela deslizante. O objetivo do sistema proposto por Bonino et al. é fazer com que somente pacotes da fonte que estejam contidos na janela de comprimento W sejam selecionados para compor pacotes de paridade. Isto garante um certo sequenciamento ou ordenamento dos pacotes na decodificação, possibilitando o acesso ao conteúdo em tempo de transmissão. Após a transmissão de uma quantidade predeterminada de paridades na janela W , é feito um deslocamento de S pacotes da fonte de modo que haja uma sobreposição de $W-S$ pacotes na nova janela de mesmo comprimento W formada. Esta sobreposição proporciona algumas melhorias em relação a um sistema tradicional de janela fixa. Este trabalho propõe um esquema de transmissão utilizando o código LT e uma janela deslizante suave para seleção dos pacotes da fonte objetivando uma redução na taxa de pacotes da fonte não recuperados, comparada com o esquema que utiliza janela deslizante de Bogino et al. Resultados de simulação computacional comprovam o melhor desempenho do esquema proposto.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering

DIGITAL FOUNTAIN CODE WITH SOFT SLIDING WINDOW FOR STREAMING OF MULTIMEDIA CONTENTS

André José Silveira

December09th, 2016

Advisor: Bartolomeu Ferreira Uchôa Filho

Area of concentration: Communications and Signal Processing

Keywords: Fountain Codes, Luby Transform, LT

Number of pages: 87

Broadcast transmission techniques for audio, video and image through a channel that can be modeled as an erasure channel have greatly evolved within the scientific community. The use of LT, Raptor, and Tornado codes, all belonging to the family of fountain codes, are an example of this evolution. There has also been a great effort, by the scientific community, to improve the quality of user experience when streaming audio, video, and image. Enabling, at time of transmission, users to listen to audio or watch video or image that are being transmitted is a major challenge for current transmission systems. The standard LT code proposed by Luby in 2002 allows information or media content to be accessed only after recovering all the information from the source. This feature is due to the fact that K source packets are uniformly selected to compose the parity packet. Thus, the decoder will not have a sequencing or ordering of recovered packets, thereby

preventing partial access to such a media content, for example. In 2007, Bogino et al. proposed a transmission scheme with the LT code using a sliding window. The aim of the proposed system by Bogino et al. is to have only the source packets which fall within in the window of width W selected to comprise parity packets. This ensures a certain sequencing or ordering of packets in decoding, enabling access to content at the time of transmission. After transmitting a predetermined number of parities within the window, a displacement S of source packets is made so that there is an overlap of $W-S$ packets in the new length- W window. This overlap provides some improvements in comparison with a traditional system of fixed window. This paper proposes a transmission scheme using the LT code and a soft sliding window for the source package selection aiming at a reduction in the rate of non-recovered packet sources, compared with a system using sliding window. Simulation results show the improved performance achieved by the proposed scheme.

1 Sumário

1.	Introdução.....	17
1.1.	Motivação.....	17
1.2.	Modelo de canais com apagamentos	18
1.3.	Fontanasdigitais.....	19
1.4.	Objetivos	21
1.5.	Organização da dissertação	22
2.	Códigos LT.....	23
2.1.	O processo LT	24
2.2.	Codificação	27
2.3.	Decodificação.....	30
2.4.	Análise e projeto da distribuição de graus.....	35
2.4.1.	Distribuição Sóliton Ideal.....	35
2.4.2.	Distribuição Sóliton Robusta	37
2.5.	Escolha dos parâmetros da distribuição Sóliton Robusta 40	
3.	Transmissão usando a janela deslizante	45
3.1.	Introdução	45
3.2.	Janela Deslizante.....	46
3.3.	Análise da taxa de símbolos da fonte não recuperados	52
4.1.	Introdução	59

4.2.	Janela deslizante suave.....	60
4.3.	Análise da taxa de símbolos da fonte não recuperados	65
4.4.	Análise do sequenciamento da decodificação dos símbolos da fonte	72
5.	Conclusões e Sugestões para trabalhos futuros	79
5.1.	Conclusões	79
5.2.	Sugestão de trabalhos futuros	80
6.	Bibliografia	83

1. Introdução

1.1. Motivação

A transmissão de conteúdo multimídia por difusão através de um canal com apagamento tem merecido um grande esforço da comunidade científica. O código LT (*Luby Transform Codes*), pertencente à família dos códigos fontanais, foi um grande avanço para o cenário descrito acima. Porém, o código LT padrão apresentado em [1, 2] não permite que o conteúdo multimídia sendo transmitido seja acessado durante o tempo de transmissão. É necessário aguardar a recepção de uma determinada quantidade de pacotes de paridades para que o decodificador recupere toda a informação da fonte, e só assim o conteúdo multimídia poderá ser assistido. Diversas técnicas de codificação vêm sendo propostas pela comunidade científica para transmissão de conteúdo multimídia, como IDNC (*Instantly Decodable Network Coding*) [3-9]. A codificação de rede [10] e a codificação de índice [11] também abriram as portas para muitos algoritmos como os propostos em [12-14]. Com o objetivo de contornar os problemas do código LT para transmissão de conteúdo multimídia, em [15] é apresentado um sistema de transmissão utilizando uma janela deslizante (*Sliding Window - SW*) para o código LT. Vislumbrando uma possível redução na taxa de pacotes da fonte não recuperados, este trabalho propõe o uso de uma janela deslizante suave para o código LT.

1.2. Modelo de canais com apagamentos

Canais com apagamentos, ou mais conhecidos como *Binary Erasure Channel* – BEC, foram introduzidos em [16]. Este modelo de canal é muito importante, pois pode modelar, por exemplo, pacotes UDP na internet que podem ser recebidos sem erros ou não recebidos. Canais ruidosos também podem ter pacotes apagados quando o código corretor de erro falhar na decodificação e o decodificador desconsiderar o pacote recebido. Outro cenário que pode ser modelado como um canal com apagamento é a gravação de dados em um sistema de disco rígido (*storage*) [2].

Podemos definir um canal BEC como um canal com alfabeto de entrada binário, $A = \{a_1, a_2\}$, e um alfabeto de saída ternário, $B = \{b_1, b_2, b_3\}$, que se relaciona probabilisticamente com a entrada por meio de uma probabilidade condicional. A Figura 1.1 ilustra o modelo de canal apresentado. A chamada matriz probabilidade de transição associada ao canal é dada por[17]:

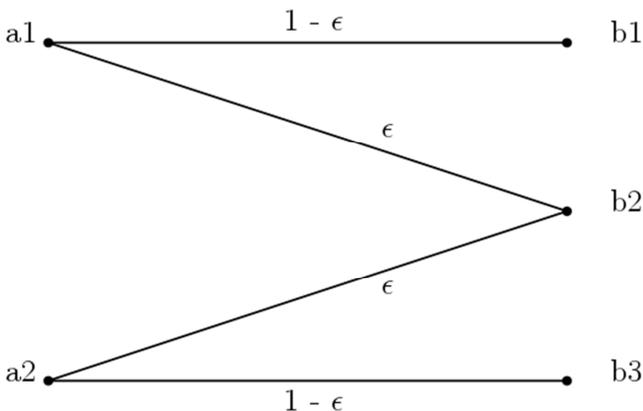


Figura 1.1 – Canal BEC

$$Q = \begin{pmatrix} q_{1|1} & q_{2|1} & q_{3|1} \\ q_{1|2} & q_{2|2} & q_{3|2} \end{pmatrix} = \begin{pmatrix} 1 - \varepsilon & \varepsilon & 0 \\ 0 & \varepsilon & 1 - \varepsilon \end{pmatrix} \quad (1.1)$$

O símbolo de saída b_2 é chamado de apagamento, já que nada pode ser dito, em relação ao símbolo de entrada, ao observá-lo na saída. Porém, ao observar-se na saída os símbolos b_1 ou b_3 , pode-se afirmar com probabilidade um de acerto que o símbolo de entrada correspondente é a_1 ou a_2 , respectivamente [17].

Neste trabalho, consideramos que o canal é conhecido para a correta definição da sobrecarga (*overhead*) a ser utilizado na transmissão.

1.3. Fontanas digitais

Nesta seção vamos entender a ideia dos códigos fontanais, ou de fontanas digitais introduzidos em [18, 19]. Suponha que você tenha um arquivo, composto de K pacotes, e queira transmiti-lo para U usuários através de um canal com apagamento. No método convencional de transmissão *Automatic Repeat Request* - ARQ [20], o transmissor iria transmitir sequencialmente pacote por pacote e, em seguida, receberia dos U usuários por um canal de retorno a informação sobre quais pacotes foram apagados, para serem retransmitidos. Percebe-se intuitivamente a ineficiência deste sistema de transmissão uma vez que a mesma informação poderá ter que ser retransmitida inúmeras vezes. Para contornar este problema, algumas alternativas eficientes para o

protocolo ARQ são propostas em [21-23], baseando-se na ideia de recuperação de apagamento.

Os sistemas de transmissão que passaram a usar códigos corretores de erros *Forward Error Correction* - FEC obtiveram um avanço considerável no desempenho da transmissão de informação. Nestes sistemas, um código corretor de erro é utilizado permitindo ao receptor não apenas detectar erros, mas também corrigi-los. A filosofia por trás dos códigos FEC em geral é justamente permitir a transmissão de informação sem a necessidade de os U usuários informarem por um canal de retorno os pacotes que foram perdidos para serem retransmitidos. A ideia dos códigos fontanais é desenvolver um esquema de codificação em que a partir do momento que o receptor receba um número suficientemente grande de pacotes, geralmente um pouco maior do que K , o receptor possa decodificar e recuperar a mensagem originalmente transmitida. Nos códigos fontanais, os pacotes transmitidos são paridades aleatórias dos K pacotes da fonte. Os códigos fontanais podem ser comparados a um chafariz (que jorra um número arbitrariamente grande de gotas de água), a partir do qual os pacotes de paridade são pulverizados para todos os U usuários. Os U usuários recebem aleatoriamente $n=K+E$ pacotes de paridades, e com isto conseguem recuperar a informação original [2]. Logo, a quantidade de símbolos de paridade recebidos no receptor dever ser $n = (1+\alpha)K$, onde K é o número de símbolos da fonte, e α a sobrecarga do código.

Os códigos fontanais são considerados sem taxa, uma vez que a quantidade de pacotes de paridade gerados pode ser ilimitada ou definida em tempo real. Logo, independentemente da estatística do

canal, podemos transmitir a quantidade de pacotes de paridades necessária para o decodificador recuperar a informação da fonte [2].

Neste trabalho temos a intenção de utilizar uma das implementações dos códigos fontanais, o código LT, para transmitir conteúdo multimídia e acessar a informação em tempo de transmissão. Como o conteúdo multimídia tem um certo sequenciamento dos dados da fonte, não é possível transmitir estes dados indefinidamente. Ou seja, uma parte dos dados do conteúdo multimídia estará disponível para compor pacotes de paridade somente durante uma parte do tempo, e logo em seguida esta será substituída por outra parte do conteúdo multimídia. Isto faz com que percamos uma característica dos códigos fontanais que é ser considerado sem taxa, pois no início da transmissão de uma fração do conteúdo multimídia teremos que definir a sobrecarga que será utilizada.

1.4. Objetivos

Com base no que foi mencionado nas seções anteriores e na revisão do Capítulo 2, percebe-se que os códigos fontanais em sua proposta original não permitem que conteúdos multimídia sejam acessados em tempo de transmissão. Ou seja, é necessário que o receptor receba um subconjunto qualquer de $K+E$ pacotes de paridade para que a informação da fonte seja recuperada e aí então o conteúdo multimídia seja acessado. Este trabalho objetiva propor um esquema de transmissão utilizando o código LT, pertencente à família dos códigos fontanais, que permita a decodificação e, conseqüentemente, o acesso à mídia sendo transmitida em tempo de transmissão.

O Esquema de transmissão proposto neste trabalho, batizado de Janela Deslizante Suave (*Soft Sliding Window - SSW*), possibilita um certo sequenciamento na decodificação dos pacotes da fonte permitindo que o conteúdo multimídia seja acessado em tempo de transmissão. Este esquema proposto é uma variação do esquema de janela deslizante [15].

1.5. Organização da dissertação

O trabalho está organizado da seguinte forma: No Capítulo 1 foram apresentadas a motivação para este trabalho bem como aspectos básicos sobre canal com apagamento e fontanas digitais. Também, na última seção, foram apresentados os objetivos do trabalho. No Capítulo 2 é feita uma revisão dos códigos LT. No Capítulo 3 é apresentado o sistema de janela deslizante proposto em [15]. No Capítulo 4, apresentamos a proposta deste trabalho que é o sistema de janela deslizante suave. No Capítulo 5, apresentam-se uma conclusão e sugestões para trabalhos futuros.

2. Códigos LT

Os códigos LT (*Luby Transform Codes*), pertencentes à família dos códigos fontanais, e propostos por Michael Luby em 2002 [1], vêm sendo amplamente estudados e aplicados em sistemas de transmissão por difusão no qual o canal utilizado pode ser modelado como um canal com apagamento. Os códigos LT foram os primeiros a considerar a ideia de fontanas digitais. Anteriormente, Michael Luby em 1998 [24] propôs o código tornado que é um código linear aleatório. Uma outra implementação de fontanas digitais é apresentada no código raptor [25], em que é feito o uso do código LT juntamente com um código LDPC (*Low-Density Parity-Check*) [26,27]. Em [2] é feita uma excelente revisão dos principais códigos fontanais, dentre eles o código fontanal aleatório linear, o código LT e o código raptor. Neste trabalho nos concentraremos no código LT.

No processo de codificação e decodificação LT, a informação da fonte é dividida em K pacotes menores com comprimento ℓ . O comprimento ℓ destes K pacotes da fonte podem ser escolhidos como desejado, e não tem qualquer influência sobre a teoria. Geralmente, o valor de ℓ depende do comprimento da carga útil do pacote do protocolo que está sendo utilizado.

Então, no codificador LT, os dados da fonte de comprimento Z são divididos em $K = Z/\ell$ símbolos de entrada, isto é, cada símbolo de entrada possui comprimento ℓ . Cada símbolo de codificação, também de comprimento ℓ , é gerado conforme descrito na Seção 2.2. Já no

decodificador LT, dado um conjunto de símbolos codificados e alguma representação de como estes símbolos foram gerados, o decodificador recupera iterativamente os símbolos da fonte conforme descrito na Seção 2.3.

2.1. O processo LT

A análise do código LT é diferente da análise do código tornado [28-30]. Primeiramente se faz necessária uma descrição do processo LT para termos posteriormente um bom entendimento do codificar e decodificador LT, e do projeto e análise da distribuição de graus. O processo LT é classicamente demonstrado em [1],[2] como uma nova generalização do processo de jogar bolas em um balde. Nesta análise, os baldes são associados aos símbolos de entrada e as bolas, às arestas ou às conexões para os símbolos de saída. O processo é bem sucedido se no final todos os símbolos de entrada são cobertos, conforme ilustra a Figura 2.1.

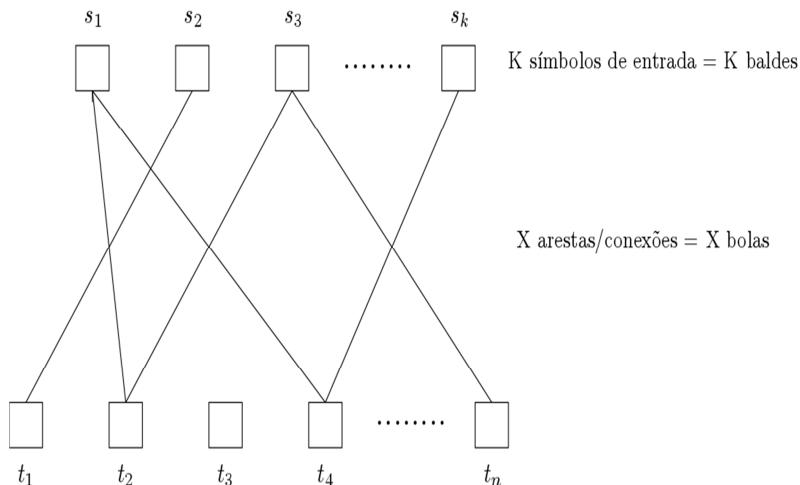


Figura 2.1 – Processo LT

Ao jogar uma bola, a probabilidade de acertar um dos K baldes é $1/K$, e a probabilidade de um balde específico não ser acertado é $\left(1 - \frac{1}{K}\right)$. Então, ao final do processo de jogar bolas a probabilidade de que um balde em particular esteja vazio depois que X bolas tenham sido jogadas é [2]:

$$\left(1 - \frac{1}{K}\right)^X \sim e^{-X/K} \quad (2.1)$$

E o número esperado de baldes vazios, δ , para qualquer quantidade X de bolas jogadas, é obtido multiplicando-se a equação (2.1) pela quantidade de baldes [2]:

$$\delta = Ke^{-X/K} \quad (2.2)$$

Ou seja, este número esperado de baldes vazios δ é um valor muito pequeno, e probabilisticamente representa aproximadamente que todos os baldes tenham pelo menos uma bola seja $(1 - \delta)$ apenas se[2]:

$$X > K \log_e \frac{K}{\delta} \quad (2.2)$$

Nas próximas seções será feito um estudo do algoritmo de codificação e decodificação do código LT, e faremos uma análise da escolha dos parâmetros da distribuição de probabilidade Sóliton Robusta.

2.2. Codificação

No código LT, o grau d_n , ou seja, a quantidade de pacotes que irão compor os pacotes de paridade, segue uma distribuição de probabilidades chamada Distribuição Sóliton Robusta $\mu(d)$. Esta distribuição possibilita que o receptor decodifique a informação original com uma probabilidade de falha δ . A escolha dos parâmetros da Distribuição Sóliton Robusta é de fundamental importância para o processo de codificação e decodificação do código LT. Nas seções seguintes, iremos abordar com mais detalhes esta distribuição.

Uma vez sorteado o grau d_n , os d_n pacotes da fonte que irão compor o pacote de paridade são selecionados conforme uma distribuição uniforme. O uso de uma distribuição uniforme garante uma aleatoriedade ao processo e possibilita que todos os pacotes da fonte tenham a mesma probabilidade de serem selecionados para compor um pacote de paridade a ser transmitido.

O processo de codificação no código LT é apresentado abaixo, no qual cada pacote de paridade t_n é composto por pacotes de informação da fonte: $S_1, S_2, S_3, \dots, S_K$ da seguinte forma:

1. Selecione aleatoriamente o grau d_n da Distribuição Sóliton Robusta $\mu(d)$;
2. Selecione de uma distribuição uniforme d_n diferentes pacotes da fonte, e faça t_n igual à soma bit a bit módulo 2 de todos os d_n pacotes de informação selecionados.

Este processo de seleção do grau d_n e dos pacotes que irão compor as paridades nada mais é do que um processo de formação de uma matriz geradora \mathbf{G} , semelhante a um código de bloco. Logo, os pacotes de paridades t_n podem ser obtidos por meio da seguinte equação:

$$t_n = \sum_{k=1}^K S_k G_{kn} \quad (2.4)$$

Podemos observar que, a cada iteração do processo de formação de um pacote de paridade t_n , o grau d_n selecionado nada mais é do que o peso da coluna da matriz \mathbf{G} no instante de tempo n .

O processo de codificação do código LT também pode ser representado em forma de grafo. Vamos considerar uma entrada composta por 6 símbolos: $s_1s_2s_3s_4s_5s_6 = 100110$. Para simplificar e facilitar o entendimento, consideramos o comprimento da mensagem de apenas 1 bit. Apesar de o código LT permitir que um número ilimitado de símbolos de paridade sejam formados, vamos considerar que apenas 7 símbolos de saída são gerados, sendo eles $t_1t_2t_3t_4t_5t_6t_7$. Com isto, o processo de codificação é realizado em 7 etapas, uma para cada símbolo de paridade de saída, conforme a Tabela 2.1.

Símbolo	Grau (d_n)	Vizinhos	Valor
t_1	2	s_4s_5	$1 \oplus 1 = 0$
t_2	4	$s_1s_3s_4s_5$	$1 \oplus 0 \oplus 1 \oplus 1 = 1$
t_3	4	$s_1s_2s_4s_5$	$1 \oplus 0 \oplus 1 \oplus 1 = 1$
t_4	2	s_2s_6	$0 \oplus 0 = 0$
t_5	2	s_3s_5	$0 \oplus 1 = 1$
t_6	1	s_3	$0 = 0$
t_7	6	$s_1s_2s_3s_4s_5s_6$	$1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$

Tabela 2.1 – Processo de codificação LT

O processo de codificação LT descrito acima resulta no grafo bipartido apresentado na Figura 2.2. O grafo bipartido é formado pelos símbolos codificados t_n que representam os nós de saída, e suas respectivas conexões s_k , que representam os nós de entrada.

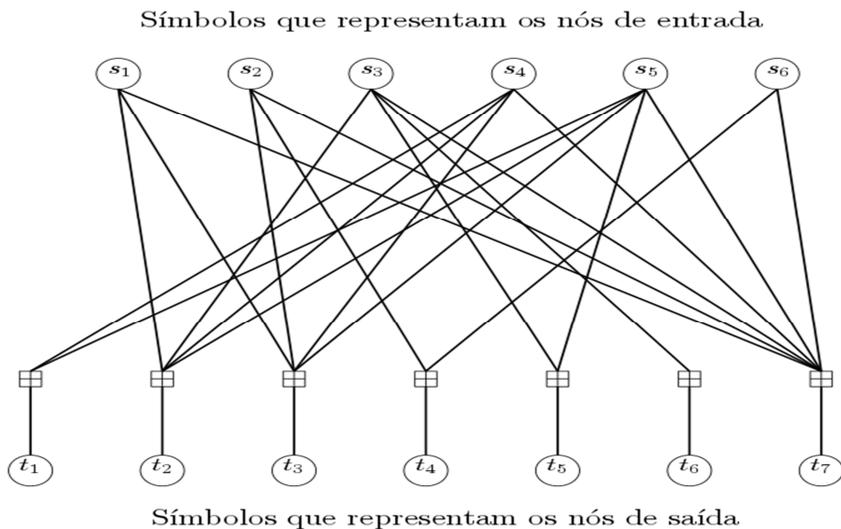


Figura 2.2 – Grafo bipartido resultante do processo de codificação LT

2.3. Decodificação

No processo de decodificação do código LT, considera-se que o decodificador de alguma forma tem conhecimento da quantidade de pacotes (grau d_n) e também quais foram esses pacotes da fonte que compuseram o pacote de paridade t_n quando da sua geração no transmissor. A função do decodificador é recuperar \mathbf{s} de $\mathbf{t} = \mathbf{sG}$, em que \mathbf{G} é a matriz gerada no codificador, passada por mensagem ao decodificador. O processo de decodificação LT também pode ser analisado na forma de um grafo bipartido, uma vez que é de conhecimento do decodificar todas as conexões que compõem os nós de saída t_n . O algoritmo de decodificação é apresentado abaixo[2]:

1. Encontrar um nó de saída t_n que esteja conectado a apenas um símbolo de entrada s_k ; caso não exista tal nó de saída, o processo de decodificação falha, sendo interrompido neste ponto. Com isto, não é possível recuperar os símbolos da fonte, sendo necessário receber mais símbolos codificados antes de fazer uma nova tentativa de decodificação.

- a) Fazer $s_k = t_n$,

- b) Fazer a soma modulo 2 de s_k com todos os nós t_n que estejam conectados a s_k . Com isso, a informação de s_k é retirada de t_n , restando somente a informação das demais conexões,

- c) Remover todas as arestas conectadas ao símbolo da fonte s_k . Com isso, o grau dos nós t_n conectados a s_k são reduzidos de um,

2. Repetir (1) até que todos os s_k símbolos da fonte sejam recuperados.

Na Figura 2.3 (a até l) é apresentado o processo de decodificação correspondente à codificação apresentada na Tabela 2.1 e Figura 2.2. Pode-se observar no grafo que existem 6 nós superiores que representam os símbolos da fonte, que são, no início do algoritmo, desconhecidos. Também se pode observar os 7 nós inferiores que representam os símbolos de saída $t_1t_2t_3t_4t_5t_6t_7$ com valor, no início do algoritmo, igual a 0110101, respectivamente. Abaixo descrevemos como o decodificador percorre o grafo, conforme o algoritmo de decodificação apresentado, para recuperar os símbolos da fonte:

1. Na primeira iteração do algoritmo, verifica-se que t_6 é o único nó de saída conectado a apenas um único nó de entrada, s_3 (Figura 2.3-a),
2. Fazer $s_3 = t_6$ e eliminar o nó de saída, t_6 (Figura 2.3-b),
3. Adicionar o valor de s_3 aos nós de saída que estão conectados ao mesmo, e em seguida desconectar s_3 do grafo (Figura 2.3-c),
4. Na segunda iteração, t_5 é o único nó de saída conectado a apenas um único nó de entrada, s_5 (Figura 2.3-c),
5. Fazer $s_5 = t_5$ e eliminar o nó de saída t_5 (Figura 2.3-d),
6. Adicionar o valor de s_5 aos nós de saída que estão conectados ao mesmo, e em seguida desconectar s_5 do grafo (Figura 2.3-e),
7. Na terceira iteração, t_1 é o único nó de saída conectado a apenas um único nó de entrada s_4 (Figura 2.3-e),

Então o algoritmo continua até que todos os símbolos da fonte sejam recuperados.

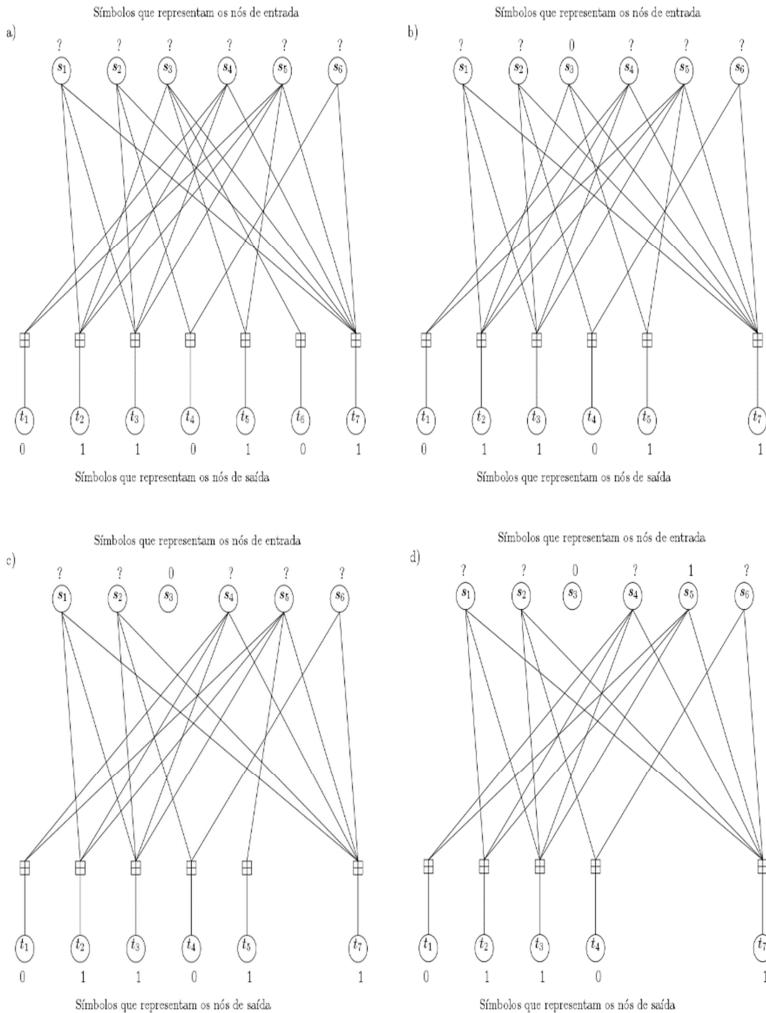


Figura 2.3 (a até d) – Grafos resultantes do processo de decodificação LT

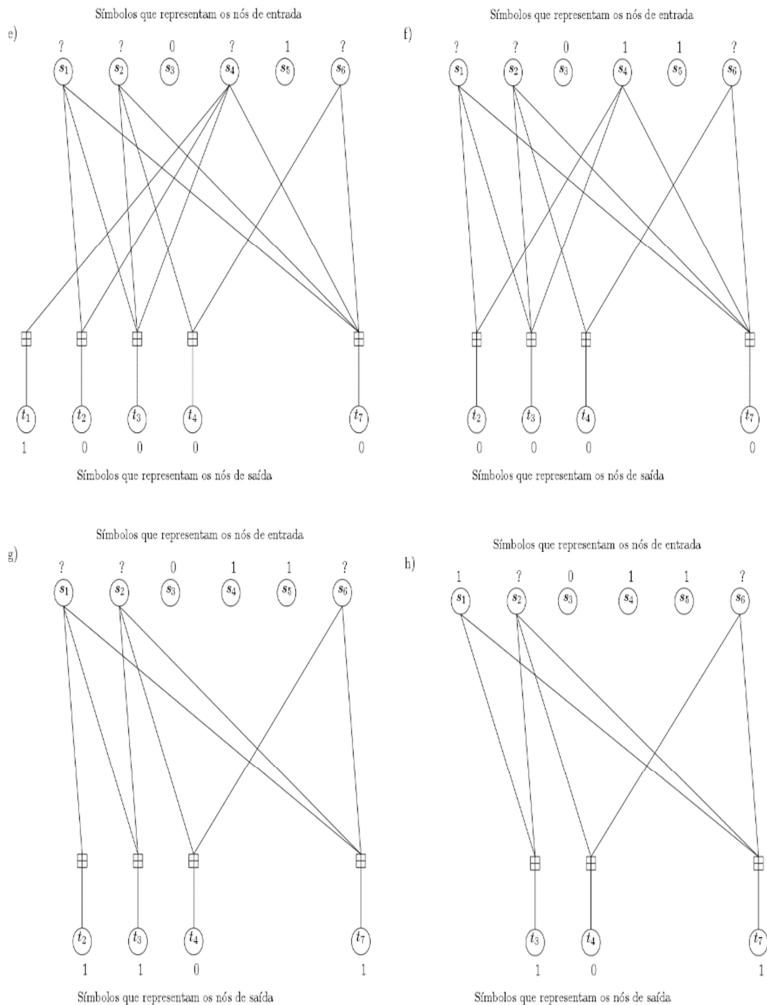


Figura 2.3 (e até h) – Grafos resultantes do processo de decodificação LT

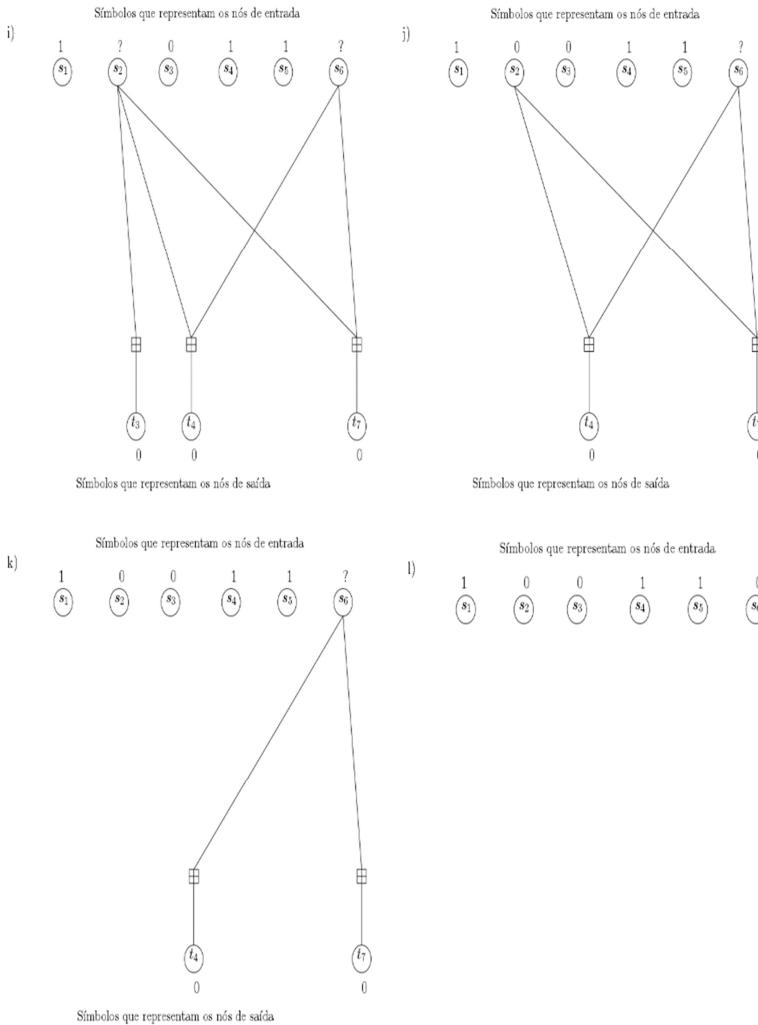


Figura 2.3 (i até l) – Grafos resultantes do processo de decodificação LT

2.4. Análise e projeto da distribuição de graus

A definição da distribuição de probabilidades dos graus é uma parte crítica do projeto. Para garantir que nenhum pacote da fonte fique sem conexão, o grau de alguns pacotes codificados devem possuir valor alto, isto é, o valor de d próximo ao valor de K . Por outro lado, alguns pacotes devem ter o valor de graus baixos, de modo que o processo de decodificação possa começar e se manter até o final. Além de garantir está eficiência do processo de codificação e decodificação, essa distribuição de valores altos e baixos dos graus é fundamental para que o número total de operações de adições envolvidas na codificação e decodificação seja mantido pequeno [2].

2.4.1. Distribuição Sóliton Ideal

A distribuição Sóliton Ideal é definida com:

$$\rho(1) = \frac{1}{K} \tag{2.5}$$

$$\rho(d) = \frac{1}{d(d-1)} \quad \text{para } d = 2, 3, \dots, K$$

A Figura 2.4 apresenta o gráfico da distribuição Sóliton Ideal para $K = 10000$.

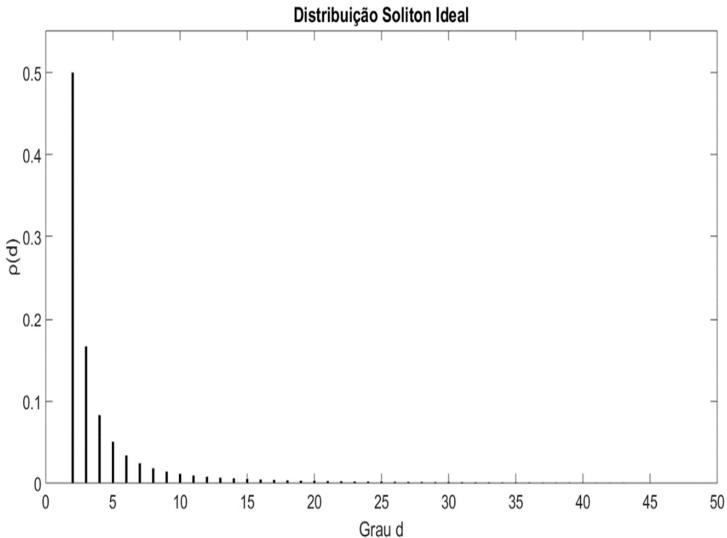


Figura 2.4 – Distribuição Sóliton Ideal

A distribuição Sóliton Ideal atende em partes as necessidades do processo LT ao garantir uma grande parte da massa de distribuição de probabilidade para os graus de valores baixos. Porém, muito pouco da massa de distribuição de probabilidade é aplicada aos graus de valores altos. Isto faz com que durante o processo de codificação alguns pacotes da fonte não sejam selecionados para compor um pacote de paridade. Além disto, a massa de probabilidade do grau de valor 1 tende a um valor muito baixo à medida que K aumenta. Como visto anteriormente, o processo de decodificação necessita de pacotes de paridade de grau 1 para dar início ao processo. Logo, uma massa de probabilidade muito pequena para o valor de grau 1 impossibilitaria o início do processo de decodificação.

2.4.2. Distribuição Sóliton Robusta

Para contornar os problemas apontados na distribuição Sóliton Ideal, Luby[1] sugeriu a utilização da distribuição Sóliton Robusta [2]:

$$\mu(d) = \frac{\rho(d) + \tau(d)}{Z} \quad (2.6)$$

em que Z é uma constante de normalização:

$$Z = \sum_d \rho(d) + \tau(d) \quad (2.7)$$

A função positiva $\tau(d)$ é definida como:

$$\tau(d) = \begin{cases} \frac{P}{K} \frac{1}{d} & \text{para } d = 1, 2, \dots, (K/P) - 1 \\ \frac{P}{K} \log \frac{P}{\delta} & \text{para } d = K/P \\ 0 & \text{para } d > K/P \end{cases} \quad (2.8)$$

O parâmetro P representa o valor esperado de grau 1, e é definido como:

$$P \equiv c \log_e \left(\frac{K}{\delta} \right) \sqrt{K} \quad (2.9)$$

O valor da constante c costuma ser na prática ligeiramente menor que 1. Já o parâmetro δ é um limitante da probabilidade de que o processo de decodificação irá falhar após receber uma determinada quantidade de pacotes de paridade [2].

Na Figura 2.5 é apresentado o gráfico da distribuição Sóliton Ideal, $\rho(d)$, juntamente com o da função positiva $\tau(d)$, para $K = 10000$, $c = 0,1$, $\delta = 0,5$, $P = 99$. Podemos observar nesta figura que a distribuição Sóliton Ideal $\rho(d)$ garante que uma certa quantidade de pacotes de paridade possua grau baixo. Esta característica é muito importante para garantir o início e a continuidade do processo de decodificação. Já observando a função positiva $\tau(d)$, fica evidente que a mesma garante que uma certa quantidade de pacotes de paridades tenha grau alto, definido em $K/P \cong 101$. Esta característica também é muito importante para garantir que todos os pacotes da fonte sejam selecionados para compor os pacotes de paridade.

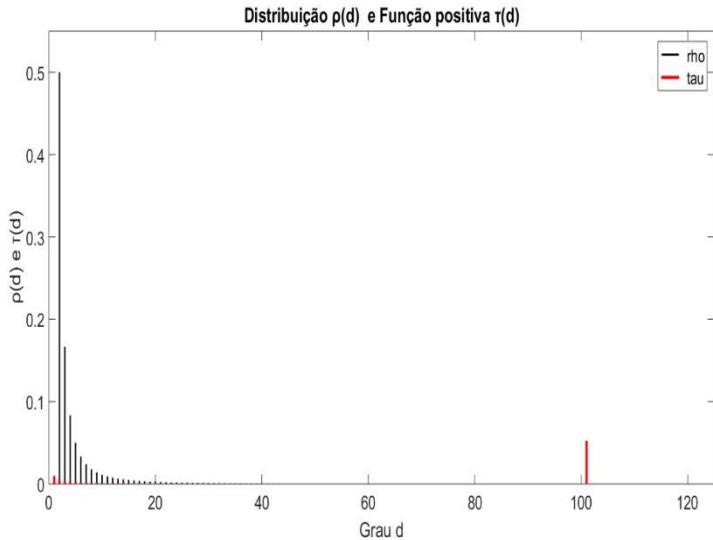


Figura 2.5 – Distribuição Sóliton Ideal $\rho(d)$ e a função positiva $\tau(d)$

Luby mostrou que o número de pacotes de paridade necessários na recepção para garantir que o processo de decodificação seja executado até o fim com probabilidade $(1 - \delta)$ é $K' = KZ$. Ou seja, independentemente da taxa de apagamento do canal BEC, o receptor deve receber pelo menos K' pacotes de paridade para que o processo de decodificação seja executado com sucesso com probabilidade $(1 - \delta)$, recuperando todos os pacotes da fonte. A Figura 2.6 apresenta o gráfico da distribuição Sóliton Robusta, $\mu(d)$, para $K=10000$, $c=0,1$, e $\delta=0,5$.

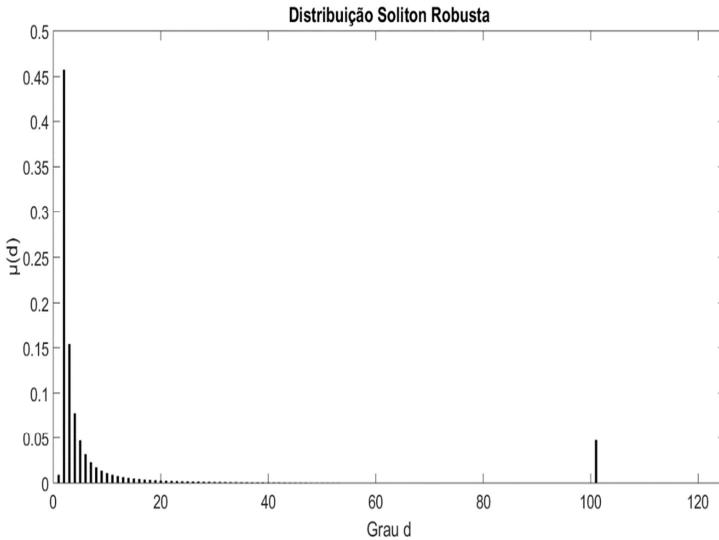


Figura 2.6 – Distribuição Sóliton Robusta

2.5. Escolha dos parâmetros da distribuição Sóliton Robusta

A escolha dos parâmetros c e δ da distribuição Sóliton Robusta possibilita a obtenção de comportamentos diferentes no processo de codificação e decodificação. Conforme visto anteriormente, o parâmetro δ é um limitante da probabilidade de que o processo de decodificação irá falhar após receber K' pacotes de paridade. Para efeitos de simulação, fixamos $\delta = 0,5$. Já o parâmetro c exerce influência sobre o valor de P , que é o valor esperado de graus 1, no valor de Z , que é a quantidade de sobrecarga necessária para o processo de decodificação ser executado com probabilidade de sucesso $(1 - \delta)$, e no parâmetro K/P , que representa a probabilidade de pacotes de paridade de grau K/P . Esta influência de c nestes parâmetros faz com que o histograma de pacotes

de paridades recebidos necessários para o sucesso do processo de decodificação tenha um espalhamento diferente de acordo com o valor escolhido de c . Para uma melhor visualização deste comportamento, os histogramas das Figuras 2.7 (a, b e c) ilustram o resultado de 1000 simulações para $K = 10000$, $\delta = 0,5$ e $c(a) = 0,01$, $c(b) = 0,05$, $c(c) = 0,1$.

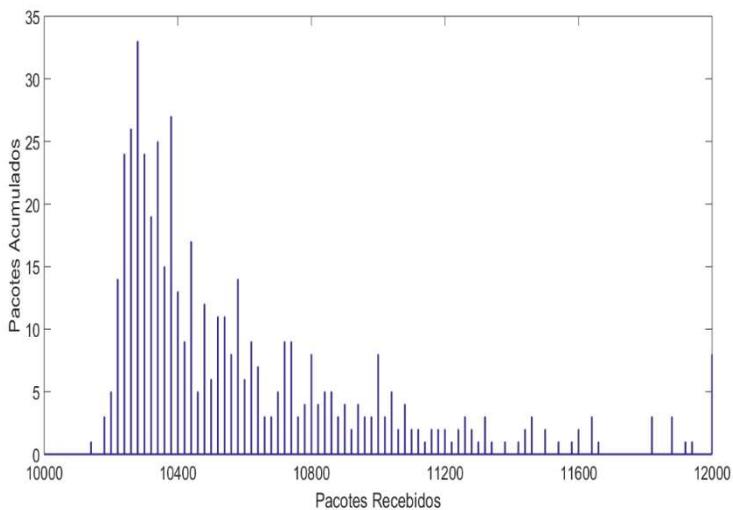


Figura 2.7 (a) – Histograma de 1000 pacotes recebidos para $K = 10000$, $\delta = 0,5$ e $c(a) = 0,01$. Valor de $P \approx 10$, $Z \approx 1,01$ e $K/P \approx 1010$

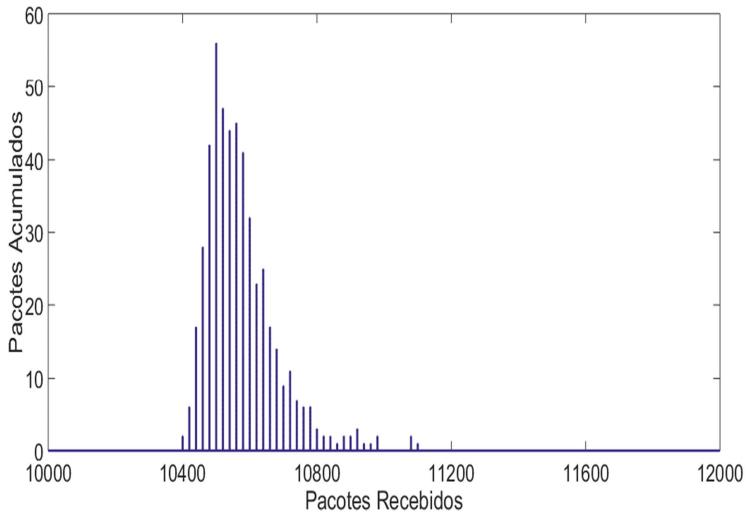


Figura 2.7 (b) – Histograma de 1000 pacotes recebidos para $K = 10000$, $\delta = 0,5$ e $c(b) = 0,05$. Valor de $P \approx 49$, $Z \approx 1,05$ e $K/P \approx 202$

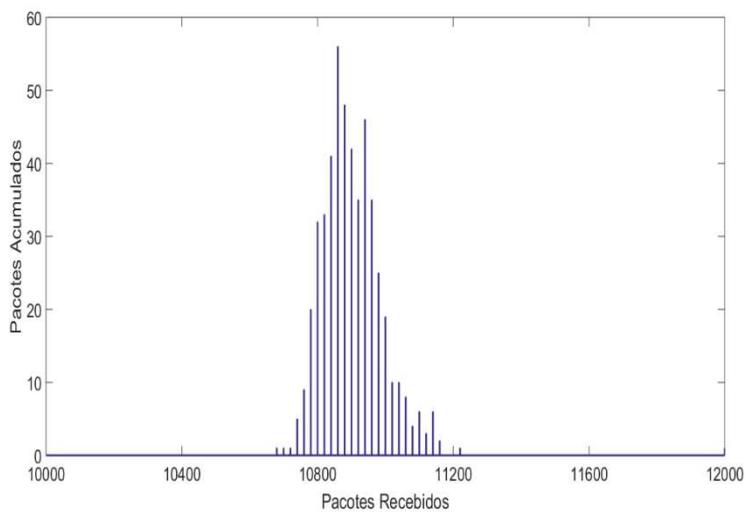


Figura 2.7 (c) – Histograma de 1000 pacotes recebidos para $K = 10000$, $\delta = 0,5$ e $c(c) = 0,1$. Valor de $P \approx 99$, $Z \approx 1,1$ e $K/P \approx 101$

3. Transmissão usando a janela deslizante

3.1. Introdução

Como foi visto nos capítulos anteriores, os códigos fontanais foram desenvolvidos para operar em canais com apagamento, e a sua principal propriedade é que os dados da fonte podem ser recuperados a partir de qualquer subconjunto dos pacotes codificados, dado que este subconjunto de pacotes recebidos seja suficiente [15]. Ou seja, o transmissor pode ficar indefinidamente pulverizando pacotes de paridade no canal, e uma vez que o receptor receba uma quantidade suficiente de pacotes codificados, toda a informação da fonte pode ser recuperada.

A utilização dos códigos fontanais em aplicações multimídia pode ser muito interessante, principalmente em situações nas quais um canal de retorno entre os usuários e o transmissor não seja possível, e nem a ordenação de recepção dos pacotes é garantida. Porém, a utilização dos códigos fontanais em aplicações multimídia encontrou algumas barreiras. Uma delas é a necessidade de os usuários receberem um subconjunto suficiente de pacotes codificados para então rodar o algoritmo de decodificação e ter acesso ao conteúdo multimídia. Ou seja, não é possível que o usuário acesse o conteúdo multimídia enquanto o transmissor ainda realiza a transmissão dos demais pacotes codificados. Outro grande complicador está relacionado à quantidade de memória utilizada, uma vez que todo o subconjunto de pacotes codificados necessita ser armazenado no dispositivo.

Em [15], Boginoet *al.* propuseram um esquema de transmissão denominado *Sliding Fountain (SF)* que busca contornar as fragilidades dos códigos fontanais em aplicações multimídia. Neste esquema de transmissão de conteúdo multimídia utilizando código fontanal é feito uso de uma técnica denominada *Sliding Window (SW)*. O objetivo é utilizar janelas de transmissão que se sobrepõem durante o processo de codificação possibilitando aos dados da fonte serem estendidos virtualmente. Esta extensão virtual dos pacotes da fonte permite uma melhora na eficiência no processo de codificação e decodificação do código LT possibilitando uma taxa de recuperação dos pacotes da fonte superior em relação a um sistema de janela fixa [15].

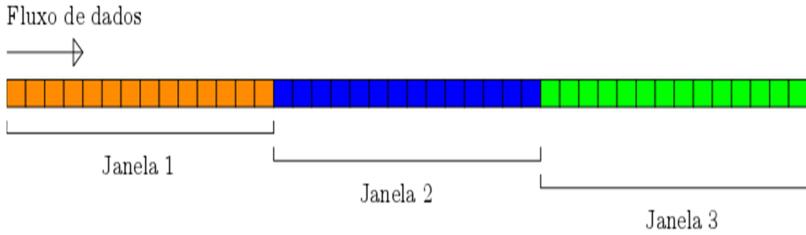
3.2. Janela Deslizante

A ideia de particionar os K pacotes da fonte em janelas de comprimento W permite um sequenciamento temporal do conteúdo multimídia no momento da codificação e decodificação, possibilitando o acesso parcial ao conteúdo multimídia em tempo de transmissão. Assim, para a composição dos pacotes de paridade, o codificador LT seleciona os símbolos da fonte dentre aqueles contidos na janela no momento da transmissão.

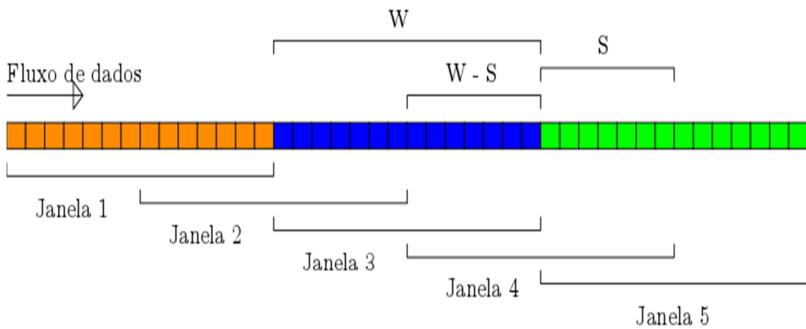
Num sistema de janela, a performance de codificação e decodificação do código LT fica comprometida, pois o comprimento W da janela, na prática, deve ser um valor baixo para garantir a ordem cronológica do conteúdo multimídia. Esta redução na performance do código LT resulta no aumento da sobrecarga ϵ do sistema.

Para contornar esta limitação, Bonino *et al*[15], propuseram um esquema de janela deslizante. A ideia principal é estender virtualmente os dados da fonte, de modo que um certo símbolo da fonte possa ser selecionado para compor um pacote de paridade em vários momentos diferentes durante a transmissão, mas sem que haja perda do sequenciamento cronológico do fluxo do conteúdo multimídia que está sendo transmitido.

A Figura 3.1 (a) apresenta o esquema de transmissão utilizando janela fixa. Na Figura 3.1 (b), tem-se a ilustração do esquema de janela deslizante proposto por Bogino *et al*[15].



(a)



(b)

Figura 3.1 – Sistema de janela fixa tradicional versus o sistema de janela deslizante

Conforme ilustrado na Figura 3.1 (a), no esquema de janela fixa os pacotes da fonte têm a oportunidade de serem selecionados para comporem pacotes de paridade em um único momento, ou seja, somente na ocasião em que a Janela 1, de comprimento W , está sendo transmitida. Assim, ao final da transmissão da Janela 1, passam-se a ser considerados os próximos W símbolos, que constituem a Janela 2, e não

mais consideram-se os W símbolos antigos, da Janela 1. Ou seja, não há sobreposição de dados entre janelas.

Já no esquema de janela deslizante [15] apresentado na Figura 3.1 (b), ao final da transmissão da Janela 1, de comprimento W , é feito um deslocamento de S símbolos, formando a Janela 2, incluindo S novos símbolos da fonte e excluindo apenas os S símbolos mais antigos pertencentes à janela anterior, com $S < W$. Com este movimento, pode-se verificar que há uma região de sobreposição de comprimento $W-S$ entre as janelas. Então, cada símbolo da fonte estará contido em W/S janelas de transmissão. Isto proporciona aos símbolos a oportunidade de serem novamente selecionados para compor pacotes de paridade. Esta sobreposição dos dados da fonte em cada janela de transmissão estende virtualmente os dados da fonte melhorando a performance do codificador e do decodificador do código LT, proporcionando uma redução da sobrecarga ϵ de transmissão.

Num sistema de janela fixa, o número total de janelas N_f é definido por [15]:

$$N_f = K/W \quad (3.1)$$

Sendo K a quantidade de símbolos da fonte e W a quantidade de símbolos contidos na janela.

Já no sistema de janela deslizante, a quantidade total de janelas, N_s , de comprimento W é dada por [15]:

$$N_s = \frac{K - W}{S} + 1 \quad (3.2)$$

Para garantir que haja uma comparação justa entre os dois esquemas, a sobrecarga total em ambos deve ser a mesma. Ou seja, ao final do processo de transmissão o número de símbolos de paridade codificados no sistema de janela deslizante deve ser igual ao número de símbolos de paridade codificados no sistema de janela fixa.

No sistema de janela fixa com uma sobrecarga ε , n símbolos de paridade são gerados dos K símbolos da fonte, conforme apresentado a seguir [15]:

$$\frac{n}{K} = (1 + \varepsilon) \quad (3.3)$$

No esquema de janela deslizante, tem-se a definição de K' , que é o número total de símbolos da fonte virtualmente estendido, devido ao fato de que os símbolos da fonte são processados em W/S vezes, conforme representado a seguir [15]:

$$K' = \frac{W}{S} K \quad (3.4)$$

Para garantir que a sobrecarga será igual nos dois sistemas, pode-se dividir a quantidade de símbolos de paridade gerados no sistema de janela fixa conforme a equação (3.3) pelo total de símbolos da fonte virtualmente estendido definido na equação (3.4). Com isto, determina-se, em média, a quantidade de símbolos de paridades que serão gerados para cada símbolo da fonte que foi virtualmente estendido[15]:

$$\frac{n}{K'} = \frac{(1 + \varepsilon)K}{\frac{W}{S}K} = (1 + \varepsilon)\frac{S}{W} \quad (3.5)$$

Se cada janela do sistema de janela deslizante contém W símbolos da fonte, podemos definir a quantidade de símbolos de paridade n_w que devem ser gerados a partir de cada uma das N_s janelas, bastando multiplicar o resultado da equação (3.5) por W :

$$n_w = (1 + \varepsilon)S \quad (3.6)$$

Então, cada uma das N_s janelas do sistema de janela deslizante transmite n_w símbolos de paridade. A definição do valor do deslocamento S e da sobrecarga ε determinam a velocidade da janela. Estes valores devem ser escolhidos levando-se em consideração o apagamento do canal, os recursos dos dispositivos envolvidos na

transmissão, e a sincronização do fluxo de dados multimídia (*streaming* de dados) [15].

No lado do receptor, o decodificador não tem conhecimento da janela que está sendo processada. O decodificador simplesmente executa o algoritmo de decodificação à medida que os pacotes de paridade são recebidos. Dos pacotes recebidos, alguns pertencem à região de sobreposição e têm mais chances de serem decodificados. Já alguns outros pacotes recebidos, não pertencentes à região de sobreposição possuem uma chance menor de serem decodificados, devendo então ser armazenados em memória para serem novamente processados à medida que novos pacotes de paridade são recebidos.

3.3. Análise da taxa de símbolos da fonte não recuperados

Para uma análise da taxa de símbolos da fonte não recuperados no sistema de janela deslizante usando códigos LT, foram realizadas simulações do sistema variando a velocidade da janela, ou seja, a sobreposição de dados, e a sobrecarga(*overhead*) total do código percebido no lado do decodificador.

A Figura 3.2 mostra o esquema de transmissão utilizado nas simulações para $W=K/10$ e $S=W/2$, ou seja, uma sobreposição de 50%.

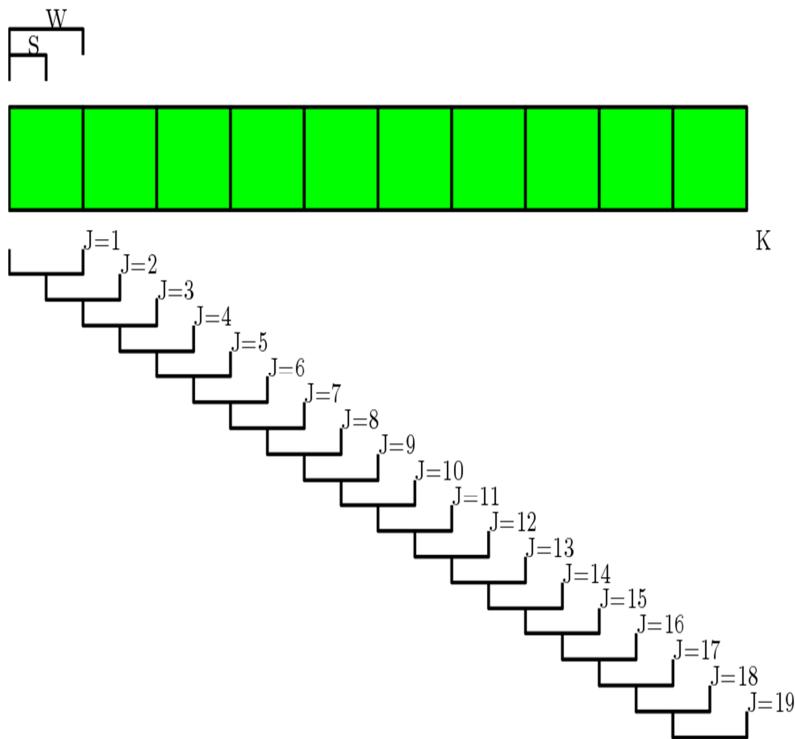


Figura 3.2 – Esquema do sistema de janela deslizante para $W=K/10$ e $S=W/2$

As simulações foram feitas para $K=10000$ e $K=100000$, e o resultado apresentado corresponde à média de 200 realizações. O objetivo da escolha dos dois valores de K é para avaliar o desempenho do sistema de janela deslizante em cenários em que a eficiência do código LT é diferente; à medida que o valor de K aumenta o código LT tem uma melhor eficiência no processo de codificação e decodificação. Os parâmetros do sistema utilizados nas simulações estão definidos na Tabelas 3.1 e 3.2, e os gráficos obtidos nos resultados das simulações, na Figura 3.3, e 3.4, respectivamente.

Parâmetros de simulação		
Parâmetros do código LT	c	0.1
	δ	0.5
Total de símbolos da fonte	K	10000
Comprimento do símbolo	l	1 bit
Comprimento da janela	W	1000
Sobrecarga (<i>Overhead</i>)	ε	0,04 a 0,14
Sobreposição (<i>Overlap</i>) %	N	S
0	10	1000
50	19	500
75	37	250

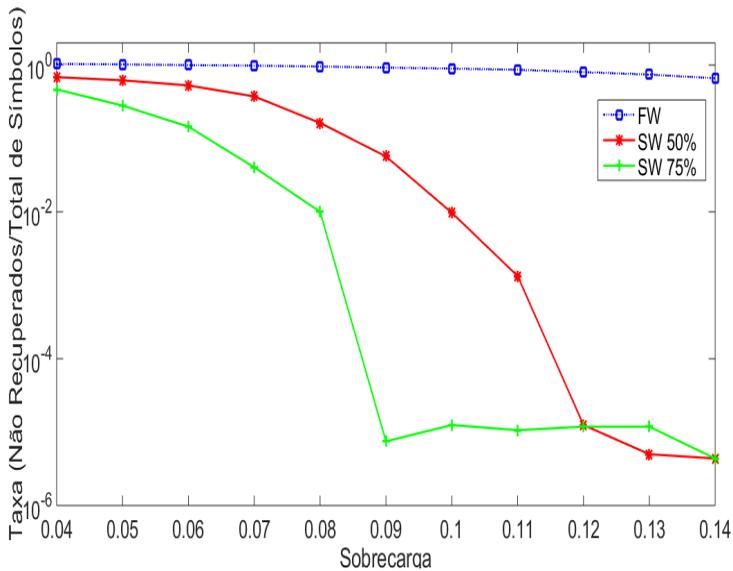
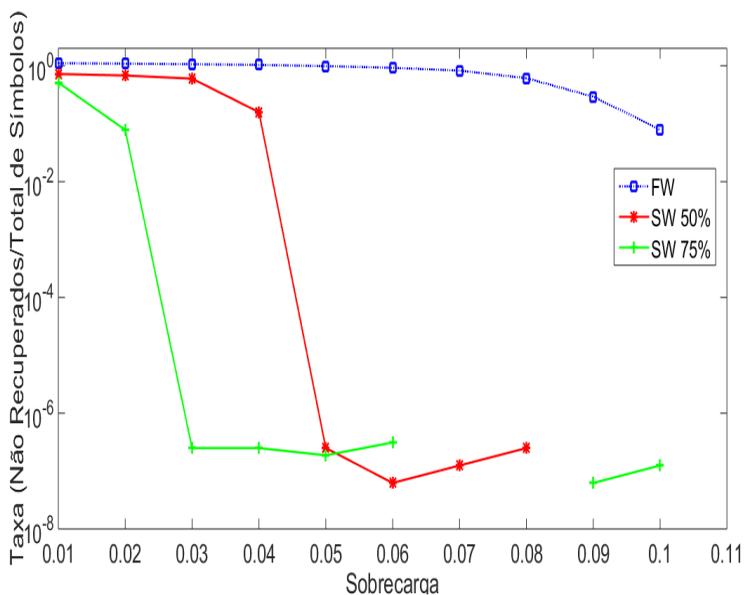
Tabela 3.1 – Parâmetros do sistema para $K = 10000$ 

Figura 3.3 – Taxa de símbolos da fonte não recuperados para

 $K = 10000$

Parâmetros de simulação		
Parâmetros do código LT	c	0.1
	δ	0.5
Total de símbolos da fonte	K	100000
Comprimento do símbolo	l	1 bit
Comprimento da janela	W	10000
Sobrecarga (<i>Overhead</i>)	ε	0,01 a 0,10
Sobreposição (<i>Overlap</i>) %	N	S
0	10	10000
50	19	5000
75	37	2500

Tabela 3.2 – Parâmetros do sistema para $K = 100000$ Figura 3.4 – Taxa de símbolos da fonte não recuperados para $K = 100000$

Os gráficos das simulações apresentados nas Figuras 3.3 e 3.4 foram obtidos observando-se no lado do decodificador os símbolos da fonte que não eram recuperados dividido pela quantidade K de símbolos da fonte, ou seja, representam a taxa de símbolos da fonte não recuperados.

Outra forma de verificar a eficiência do codificador e decodificador de um sistema seria observar a taxa de símbolos de paridades não decodificados dividido pela sobrecarga percebida pelo decodificador, ou seja, representam a taxa de símbolos de paridades não decodificados. Porém, esta taxa acaba representando uma falsa *performance* do sistema, pois existem situações em que toda a matriz geradora \mathbf{G} pode ser resolvida pelo decodificador, mas alguns símbolos da fonte não foram recuperados. Isto pode acontecer devido a estes símbolos da fonte não serem selecionados para compor símbolos de paridade na codificação. Outra situação que daria um falso *desempenho* do sistema é quando se tem um símbolo de paridade não decodificado, ou seja, uma coluna da matriz \mathbf{G} não foi resolvida pelo decodificador. Este símbolo de paridade não decodificado pode representar vários símbolos da fonte não recuperados. Com o objetivo de ter uma visão mais real do desempenho do sistema, neste trabalho optou-se por observar a taxa de símbolos não recuperados no decodificador.

A não recuperação de um símbolo da fonte no código LT pode-se dar por dois motivos. O primeiro pode ser o fato de o símbolo não ter sido selecionado nenhuma vez para compor um símbolo de paridade na transmissão. O segundo motivo pode ser que mesmo sendo selecionado

para compor um símbolo de paridade na transmissão aquele símbolo de paridade ou aquela coluna da matriz \mathbf{G} não foi completamente resolvida.

Pode-se observar pelos resultados das simulações que o sistema de janela deslizante reduz significativamente a taxa de símbolos da fonte não recuperados para determinados valores de sobrecarga percebido pelo receptor, se comparados com o sistema de janela fixa. Também pode-se observar no resultado das simulações que a sobrecarga necessária para se ter uma baixa taxa de símbolos da fonte não recuperados vai caindo a medida que K aumenta, e que a sobreposição (*overlap*) também aumenta.

4. Transmissão usando a janela deslizante suave

4.1. Introdução

O sistema de janela deslizante apresentado no capítulo anterior possibilitou uma grande redução na taxa de símbolos da fonte não recuperados comparado com um sistema de janela fixa para uma mesma sobrecarga. A ideia de estender virtualmente os pacotes da fonte dá a estes símbolos a chance de serem selecionados para formar símbolos de paridades com um conjunto maior de outros símbolos da fonte, melhorando assim a eficiência na codificação e na decodificação.

Porém, observou-se durante as simulações que à medida que a quantidade K de pacotes da fonte vai caindo, é necessário que a sobrecarga aumente para mantermos uma baixa taxa de símbolos da fonte não recuperados. Isto é inerente ao código LT, pois sua eficiência na codificação e decodificação é diretamente proporcional à quantidade K de pacotes da fonte.

Uma forma de tentar reduzir esta sobrecarga é alterando os valores c e δ da distribuição Sóliton Robusta do código LT, de forma a reduzirmos o valor do parâmetro Z , que representa a sobrecarga necessária aproximadamente para termos sucesso na decodificação do código LT com probabilidade de erro δ . Um dos efeitos colaterais de alterar os valores c e δ da distribuição Sóliton Robusta do código LT para reduzir o valor do parâmetro Z é que acaba ocorrendo um espalhamento maior da sobrecarga necessária para ter-se uma

decodificação eficiente, conforme pode-se observar nas Figuras 2.7 (a, b, c) da Seção 2.5.

Na próxima seção será introduzida a ideia do sistema utilizando uma janela deslizante suave. As simulações apresentam uma taxa de não recuperação de pacotes da fonte menor que a do sistema de janela deslizante em [15] para uma quantidade K de pacotes da fonte menores, e para determinados valores de c e δ da distribuição Sóliton Robusta do código LT.

4.2. Janela deslizante suave

O sistema de janela deslizante trouxe a ideia de estender virtualmente os pacotes da fonte. Esta extensão virtual possibilitou a combinação dos pacotes da fonte com um conjunto maior de outros pacotes da fonte para a formação dos símbolos de paridade de transmissão. Esta possibilidade de outras combinações melhorou a eficiência da codificação e da decodificação, reduzindo a taxa de símbolos da fonte não recuperados.

Com o objetivo de estender virtualmente ainda mais os pacotes da fonte, possibilitando ainda mais outras combinações de seleção dos pacotes da fonte para formarem símbolos de paridade na codificação, e consequentemente reduzir a taxa de símbolos da fonte não recuperados, na Figura 4.1 é apresentada a ideia da janela deslizante suave.

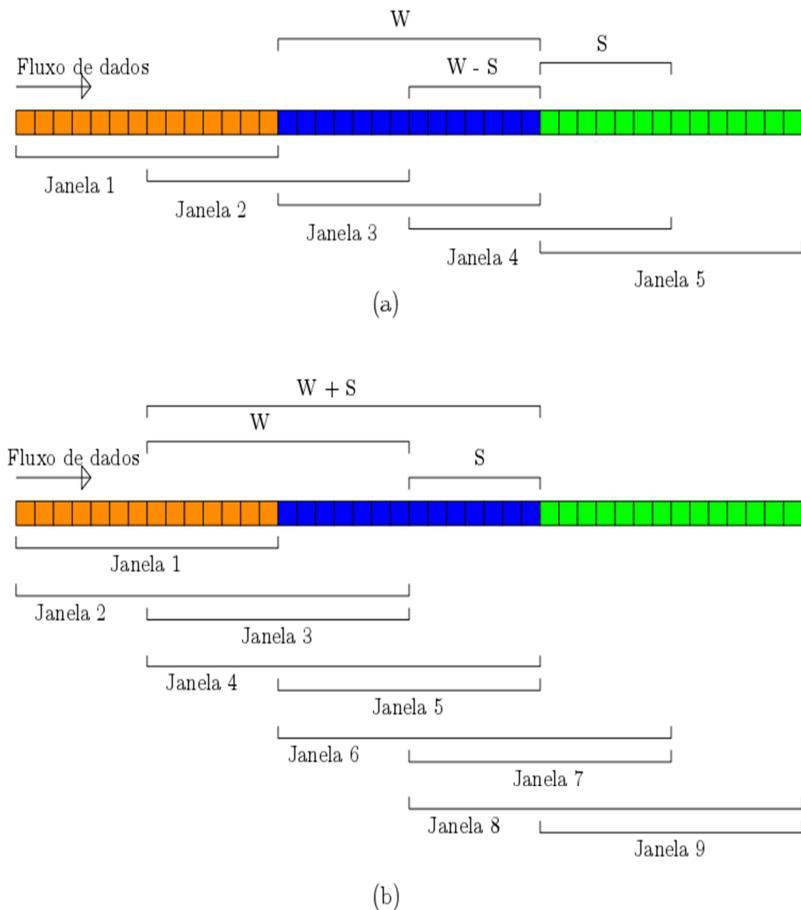


Figura 4.1 – Sistema de janela deslizante versus o sistema de janela deslizante suave

A ideia do sistema de janela deslizante suave é fazer com que a janela de transmissão deslize de forma mais lenta, e permita que os S novos símbolos adicionados à janela tenham a oportunidade de formar símbolos de paridades com os S símbolos que deixarão de compor a janela de transmissão no momento seguinte. Com isto, o tamanho da

janela de transmissão será W durante um momento da transmissão, e será $W+S$ no momento seguinte, quando S novos símbolos da fonte são adicionados à janela, e novamente W quando S símbolos antigos da fonte deixam de compor a janela.

Observando a Figura 4.1, pode-se definir intuitivamente a quantidade N_s de janelas do sistema como:

$$N_s = \left(\left(\frac{K - W}{S} + 1 \right) * 2 \right) - 1 \quad (4.1)$$

Podemos observar que a quantidade de janelas, N_s , do sistema SSW praticamente dobra comparado ao sistema SW. Logo, a quantidade n_w de símbolos codificados para cada janela do sistema SSW deve ser a metade da quantidade n_w do sistema SW. Então a quantidade n_w de símbolos codificados para cada janela do sistema SSW pode ser calculada da seguinte forma:

$$n_w = (1 + \varepsilon) \frac{S}{2} \quad (4.2)$$

A Figura 4.2 apresenta outra forma de visualização e comparação dos sistemas de janela fixa, janela deslizante e janela deslizante suave. A ideia aqui é ilustrar como o sistema de janela deslizante suave estende

virtualmente mais os símbolos da fonte que o sistema de janela deslizante. A Figura 4.2 também mostra que a quantidade de transmissões total é mantida igual nos três sistemas, além de ilustrar a região de sobreposição no sistema de janela deslizante e no sistema de janela deslizante suave. Para o sistema de janela fixa foram definidos a quantidade de símbolos da fonte, $K=6$ (ABCDEF GHIJKL) e duas janelas de comprimento $W=6$ (“ABCDEF”; “GHIJKL”). Para o sistema de janela deslizante foram definidos a quantidade de símbolos da fonte, $K=6$ (ABCDEF GHIJKL), três janelas de comprimento $W=6$ (“ABCDEF”; “DEFGHI” e “GHIJKL”) e o deslocamento $S=3$. Já para o sistema de janela deslizante suave foram definidos a quantidade de símbolos da fonte, $K=6$ (ABCDEF GHIJKL), três janelas de comprimento $W=6$ (“ABCDEF” ; “DEFGHI” e “GHIJKL”), duas janelas de comprimento $W=9$ (“ABCDEF GHI” e “DEFGHIJKL”) e o deslocamento $S=3$. Podemos verificar em vermelho a região de sobreposição do sistema de janela deslizante, e em verde a sobreposição do sistema de janela deslizante suave.

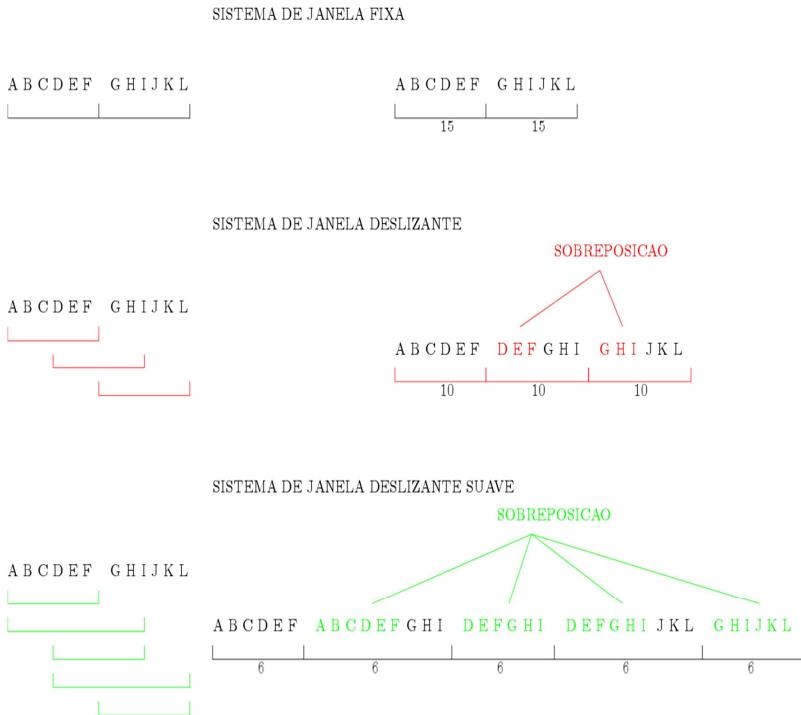


Figura 4.2 – Sistema de janela fixa, de janela deslizante, e de janela deslizante suave

Pode-se observar que a quantidade total de transmissões dos 3 sistemas foi mantida para efeitos de comparação. Ou seja, o sistema de janela fixa possui 2 janelas de transmissão com 15 transmissões em cada janela. O sistema de janela deslizante apresenta 3 janelas de transmissão com 10 transmissões em cada janela, e por último o sistema de janela deslizante suave proposto nesta dissertação apresenta 5 janelas de transmissão com 6 transmissões em cada janela.

4.3. Análise da taxa de símbolos da fonte não recuperados

Da mesma forma que foi feito na Seção 3.3, para uma análise da taxa de símbolos da fonte não recuperados no sistema de janela deslizante suave usando códigos LT, foram realizadas simulações do sistema variando a velocidade da janela, ou seja, a sobreposição de dados, e a sobrecarga (*overhead*) total do código percebido no lado do decodificador.

A Figura 4.3 mostra o esquema de transmissão do sistema de janela deslizante suave utilizado nas simulações para $W=K/10$ e $S=W/2$, ou seja, uma sobreposição de 50%.

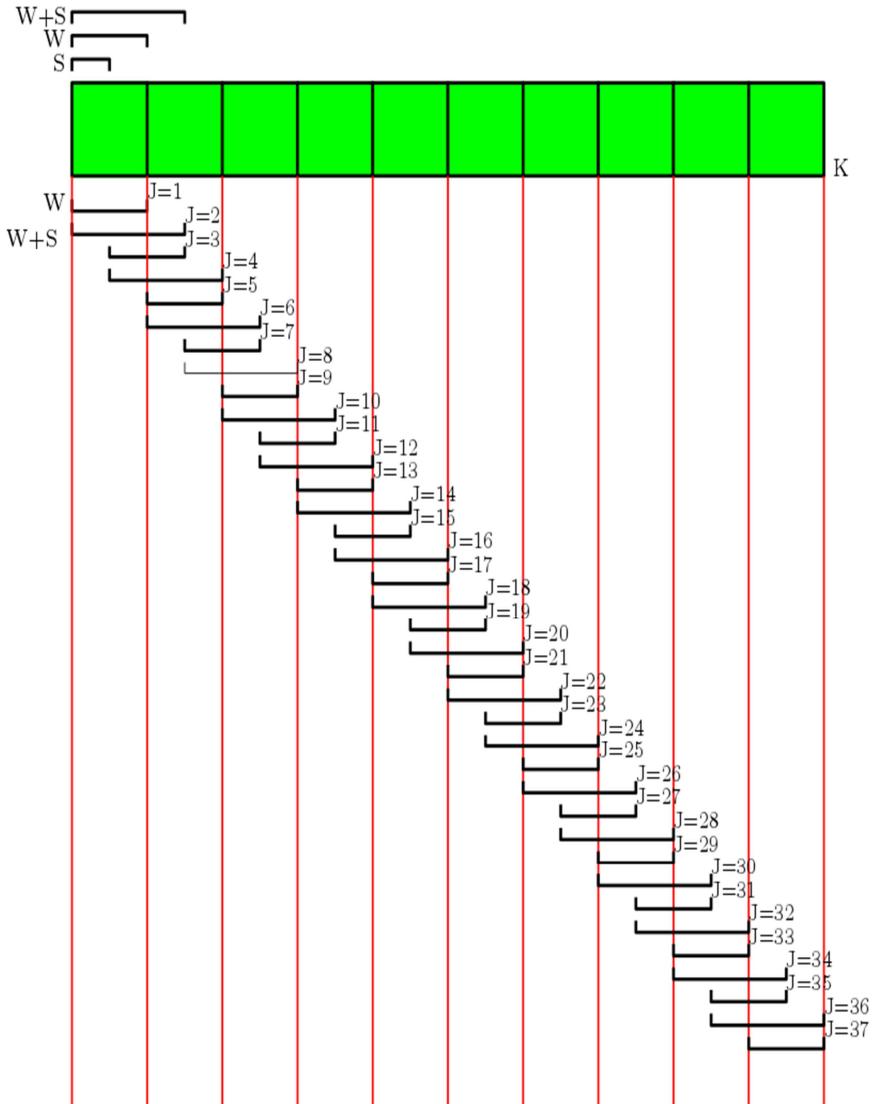


Figura 4.3 – Esquema do sistema de janela deslizante suave para

$$W=K/10 \text{ e } S=W/2$$

Para facilitar a visualização da taxa de símbolos da fonte não recuperados nos sistemas de janela fixa, janela deslizante, e janela deslizante suave, os gráficos gerados nas simulações foram feitos com os 3 sistemas juntos. As simulações foram realizadas variando-se a velocidade da janela, ou seja, a sobreposição de dados, e a sobrecarga total do código percebido no lado do decodificador para $K=10000$ e $K=100000$, e correspondem à média de 200 realizações. Os parâmetros do sistema utilizados nas simulações estão definidos nas Tabelas 4.1 e 4.2, e o gráfico obtido nos resultados das simulações nas Figuras 4.5 e 4.6, respectivamente.

Parâmetros de simulação				
	Sistema SW		Sistema SSW	
Parâmetros do código LT	c	0,1	c	0,1
	δ	0,5	δ	0,5
Total de símbolos da fonte	K	10000	K	10000
Comprimento do símbolo	ℓ	1	ℓ	1
Comprimento da janela	W	1000	W	1000
Sobrecarga (<i>Overhead</i>)	ε	0,04 a 0,14	ε	0,04 a 0,14
Sobreposição (<i>Overlap</i>) %	N	S	N	S
0	10	1000	10	1000
50	19	500	37	500
75	37	250	73	250

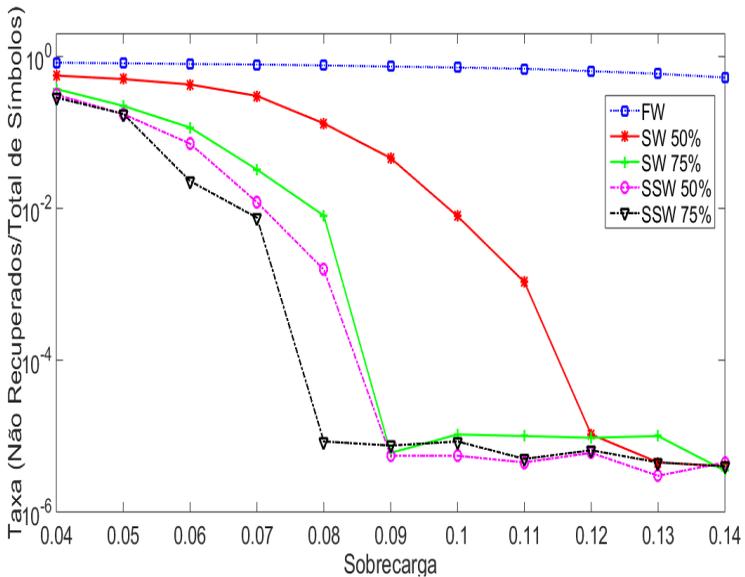
Tabela 4.1 – Parâmetros do sistema para $K=10000$ 

Figura 4.5 – Taxa de símbolos da fonte não recuperados para

 $K=10000$

Parâmetros de simulação				
	Sistema SW		Sistema SSW	
Parâmetros do código LT	c	0,1	c	0,1
	δ	0,5	δ	0,5
Total de símbolos da fonte	K	100000	K	100000
Comprimento do símbolo	ℓ	1	ℓ	1
Comprimento da janela	W	10000	W	10000
Sobrecarga (<i>Overhead</i>)	ε	0,04 a 0,10	ε	0,04 a 0,10
Sobreposição (<i>Overlap</i>) %	N	S	N	S
0	10	10000	10	10000
50	19	5000	37	5000
75	37	2500	73	2500

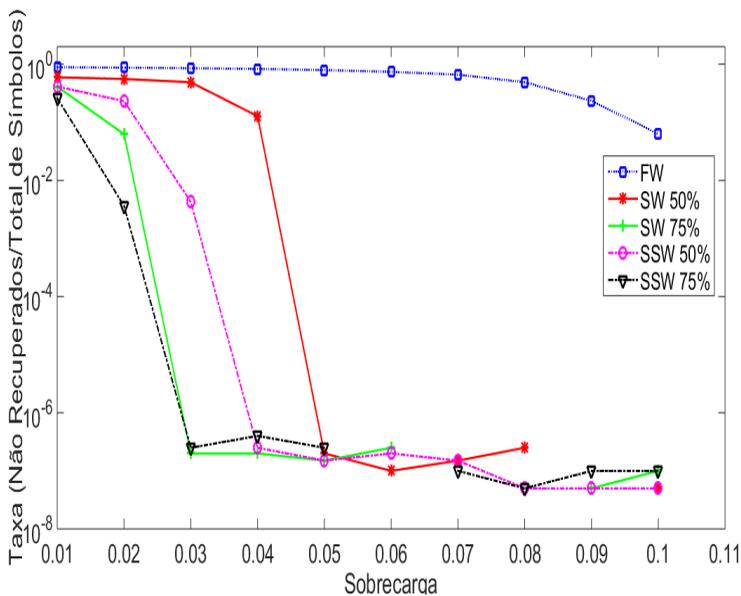
Tabela 4.2 – Parâmetros do sistema para $K=100000$ 

Figura 4.6 – Taxa de símbolos da fonte não recuperados para $K=100000$

Para as configurações dos sistemas de janela deslizante e janela deslizante suave mostradas nas Tabelas 4.1 e 4.2, observou-se uma redução na taxa de não recuperação de símbolos da fonte no sistema de janela deslizante suave. Para $K=10000$, o sistema SSW apresentou um resultado melhor tanto para a sobreposição de 75% quanto para a sobreposição de 50% em relação ao sistema SW. Por exemplo, a taxa de 10^{-5} foi atingida pelo sistema SSW com uma sobrecarga a partir de 0,08 para a sobreposição de 75%, enquanto que a mesma taxa e sobreposição foi atingida pelo sistema SW com sobrecarga a partir de 0,09. Já para a sobreposição de 50% no sistema SSW, a taxa de 10^{-5} foi atingida com uma sobrecarga de 0,09, enquanto que para o sistema SW a mesma taxa foi atingida com sobrecarga de 0,12. Para $K=100000$, a taxa de 10^{-7} foi atingida pelo sistema SSW e SW com uma sobrecarga a partir de 0,03 com a sobreposição de 75%. Porém, pode-se observar no gráfico que a curva para o sistema SSW começa a declinar primeiro em relação ao sistema SW. Com uma sobreposição de 50% a taxa de 10^{-7} foi atingida pelo sistema SSW com sobrecarga a partir de 0,04, e pelo sistema SW com sobrecarga a partir de 0,5.

Observa-se que à medida que a quantidade K de símbolos da fonte vai diminuindo, a sobrecarga necessária para se manter uma baixa taxa de não recuperação de símbolos da fonte vai aumentando. Conforme foi visto na Seção 2.5, os parâmetros c e δ da distribuição Sóliton Robusta do código LT podem ser ajustados, possibilitando uma redução da sobrecarga do sistema. A alteração dos parâmetros c e δ afetam o valor de Z , que representa aproximadamente a sobrecarga necessária para que o decodificador LT recupere os símbolos da fonte com probabilidade δ de falha no processo de decodificação.

Por se tratar de sistemas diferentes, com esquemas de transmissão diferentes, o ajuste nos parâmetros c e δ da distribuição Sóliton Robusta do código LT podem ocasionar comportamentos diferentes nos sistemas SW e SSW. A Figura 4.7 apresenta os resultados das simulações para $K=10000$, com os parâmetros da distribuição Sóliton Robusta do código LT $c=0,01$ e $\delta=0,1$. Como este valor do parâmetro c faz com que a eficiência do codificador e decodificador LT seja reduzida, o resultado das simulações corresponde à média de 800 realizações.

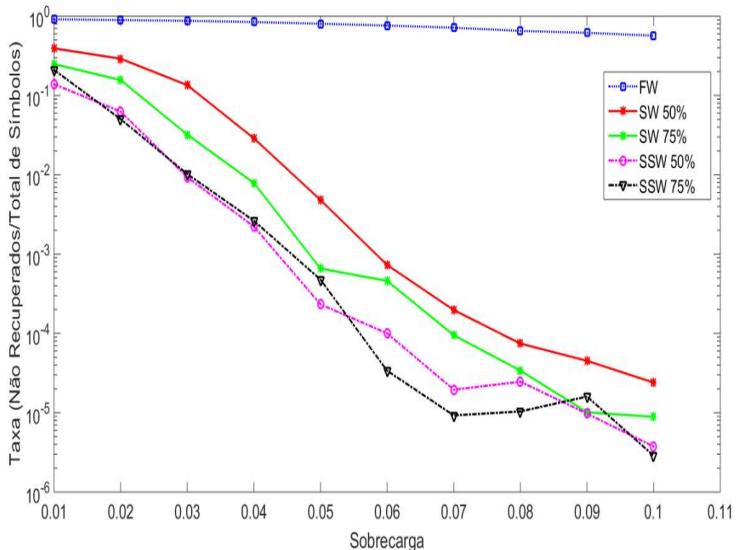


Figura 4.7 – Taxa de símbolos da fonte não recuperados para $K=10000$, $c=0,01$ e $\delta=0,1$

Nestas simulações, pode-se observar que o sistema SSW começou a apresentar uma taxa de não recuperação dos símbolos da fonte na ordem de 10^{-5} com uma sobrecarga de 0,07 para a sobreposição de 75%. Esta taxa com a mesma sobreposição foi atingida no sistema SSW na Figura 4.5 com uma sobrecarga de 0,08. Pode-se observar também que as curvas da Figura 4.5 vão declinando mais suavemente quando comparadas com as curvas da Figura 4.5. Este efeito faz com que o sistema em geral fique menos vulnerável a uma variação da sobrecarga.

4.4. Análise do sequenciamento da decodificação dos símbolos da fonte

Uma das preocupações que surgiram durante a implementação e a simulação do sistema de janela deslizante suave foi com relação ao sequenciamento da decodificação dos pacotes, que permitisse aos usuários acesso ao conteúdo multimídia durante a transmissão. Não foi encontrado nenhum trabalho ou referência acadêmica do sistema de janela deslizante para efeito de comparação com o sistema de janela deslizante suave.

Para realizar as simulações definiu-se um percentual de aceitação de perda de pacotes para um conteúdo de vídeo. Ou seja, definimos um valor de Qualidade de Serviço para uma mídia de vídeo. Valores para os parâmetros de Qualidade de Serviço aproximados, para diferentes tipos de mídias, são encontrados em[31-33], os quais serviram como referência de utilização neste trabalho. Definiu-se como aceitável, para uma mídia de vídeo, uma perda de até 1% dos símbolos da fonte.

Com o objetivo de reduzir o tempo de simulação e recursos de memória, no lado do transmissor o codificador foi implementado codificando blocos de símbolos de paridade referentes à transmissão de uma janela W completa. Da mesma forma, no lado do receptor, a decodificação foi implementada decodificando blocos de símbolos de paridade referentes à transmissão de uma janela W completa. Com isto, o resultado da simulação acaba representando o pior caso, pois considera sempre o processamento múltiplo de n_w símbolos de paridade na transmissão e recepção. Ainda no lado do receptor, a análise da sequência da decodificação dos símbolos da fonte foi feita agrupando grupos de 10% da quantidade K de símbolos da fonte. Para cada simulação foi definida uma sobrecarga fixa ϵ , a sobreposição S , e valor dos parâmetros c e δ da distribuição Sóliton Robusta do código LT.

A Figura 4.8 mostra a simulação para $K=10000$, $\epsilon=0,10$, $S=50\%$, $c=0,1$ e $\delta=0,5$. A Figura 4.9 mostra a simulação para $K=10000$, $\epsilon=0,10$, $S=75\%$, $c=0,1$ e $\delta=0,5$. Em ambos os casos o resultado corresponde à média de 200 realizações.

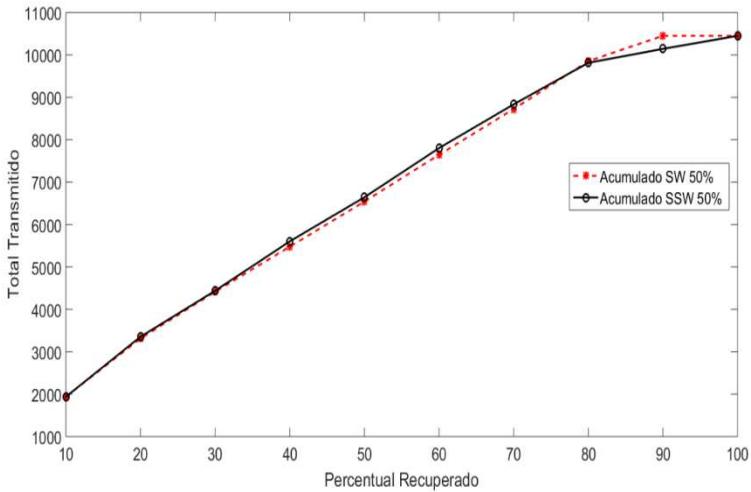


Figura 4.8 – Sequenciamento da decodificação dos símbolos da fonte para $K=10000$, $\varepsilon=0,10$, $S=50\%$, $c=0,1$ e $\delta=0,5$

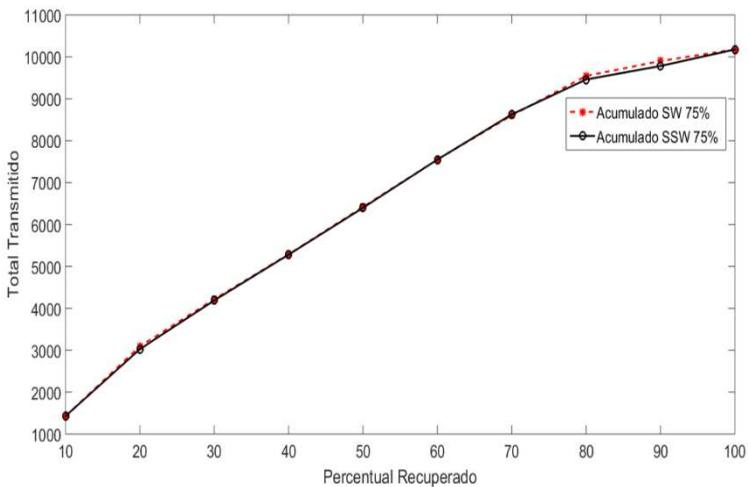


Figura 4.9 – Sequenciamento da decodificação dos símbolos da fonte para $K=10000$, $\varepsilon=0,10$, $S=75\%$, $c=0,1$ e $\delta=0,5$

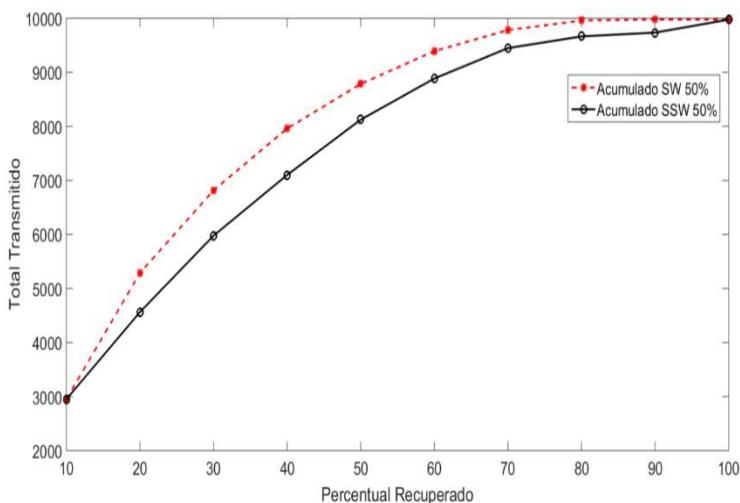


Figura 4.10 – Sequenciamento da decodificação dos símbolos da fonte para $K=10000$, $\varepsilon=0,05$, $S=50\%$, $c=0,1$ e $\delta=0,5$

Pode-se verificar nas Figuras 4.8 e 4.9 que os símbolos da fonte são recuperados sequencialmente nos dois sistemas praticamente ao mesmo tempo tanto para uma sobreposição de 50% quanto para a sobreposição de 75%. Porém na Figura 4.10 pode-se observar que, à medida que a sobrecarga ε diminui, o sistema SW vai apresentando um atraso maior na decodificação sequenciada dos símbolos da fonte, enquanto que o sistema SSW se mostra mais robusto a esta variação.

Este mesmo comportamento foi observado nas simulações para $K=100000$ apresentadas nas Figuras 4.11, 4.12 e 4.13, e também correspondem à média de 200 realizações.

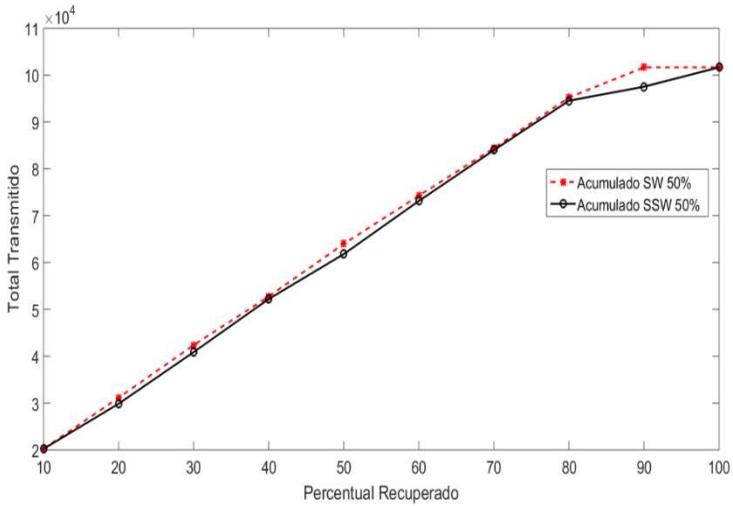


Figura 4.11 – Sequenciamento da decodificação dos símbolos da fonte para $K=100000$, $\varepsilon=0,07$, $S=50\%$, $c=0,1$, e $\delta=0,5$

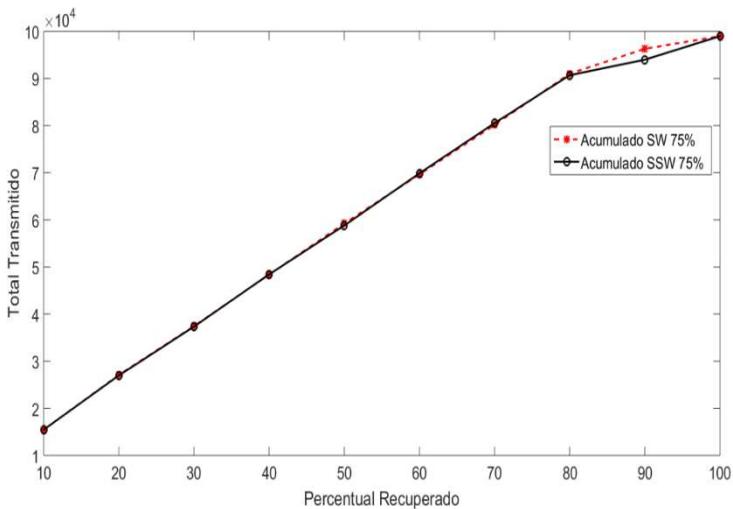


Figura 4.12 – Sequenciamento da decodificação dos símbolos da fonte para $K=100000$, $\varepsilon=0,07$, $S=75\%$, $c=0,1$, e $\delta=0,5$

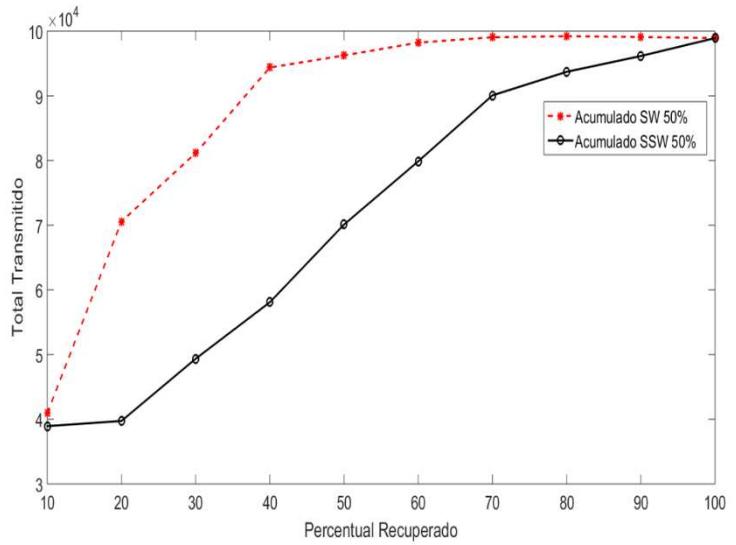


Figura 4.13 – Sequenciamento da decodificação dos símbolos da fonte para $K=100000$, $\varepsilon=0,04$, $S=50\%$, $c=0,1$, e $\delta=0,5$

5. Conclusões e Sugestões para trabalhos futuros

Neste capítulo são apresentadas as conclusões obtidas durante o desenvolvimento do presente trabalho, fazendo uma análise crítica dos resultados obtidos. Indicaremos também algumas sugestões de trabalho futuro.

5.1. Conclusões

Esta dissertação apresentou um estudo dos códigos fontanais, especificamente o código LT, para aplicação em sistemas multimídia. Após uma revisão no Capítulo 2 do código LT, foi feita no Capítulo 3 uma apresentação de um sistema de janela deslizante para aplicação multimídia utilizando o código LT. O resultado das simulações apresentaram uma redução da taxa de não recuperação de símbolos da fonte do sistema de janela deslizante comparado com o sistema de janela fixa para diferentes valores de K e de sobreposição S . Em seguida, no Capítulo 4 foi apresentada uma proposta de sistema utilizando uma janela deslizante suave, juntamente com os resultados da simulação do sistema e os resultados da simulação do sequenciamento da decodificação dos pacotes da fonte. Os resultados das simulações apresentaram uma redução da taxa de não recuperação de símbolos da fonte do sistema de janela deslizante suave comparado com o sistema de janela fixa e com o sistema de janela deslizante para diferentes valores de K e de sobreposição S . As simulações também demonstraram que tanto o sistema de janela deslizante quanto o sistema de janela deslizante suave apresentam um sequenciamento da decodificação dos pacotes da

fonte, permitindo que o conteúdo multimídia seja acessado durante a transmissão da informação. Porém observou-se nas simulações que o sistema de janela deslizante suave é mais robusto a reduções da sobrecarga (*overhead*) ϵ .

5.2. Sugestão de trabalhos futuros

A ideia do sistema de janela deslizante suave proposta neste trabalho de mestrado abre as portas para um aprofundamento maior em diversas outras linhas de pesquisa. A seguir são propostas algumas sugestões para trabalhos futuros a serem realizados:

- Pesquisar uma distribuição de graus para códigos LT, ou uma alteração na distribuição Sóliton Robusta com a finalidade de conseguir melhores resultados na taxa de não recuperação de símbolos da fonte;
- Equacionamento dos parâmetros c , δ e Z da distribuição Sóliton Robusta do código LT com o sistema de janela deslizante e janela deslizante suave possibilitando o cálculo mais preciso da sobrecarga;
- Pesquisar a utilização do código Raptor no sistema de janela deslizante suave;
- Pesquisar a utilização de proteção desigual de erro para o sistema de janela deslizante suave;
- Implementar toda a solução do sistema de janela deslizante e do sistema de janela deslizante suave em um protocolo

com características de canal BEC, por exemplo UDP, para uma comparação prática dos dois sistemas;

6. Bibliografia

- [1] Luby, M.: “*LT codes*”. Proc. 43rd Ann. IEEE Symp. On Foundations of Computer Science, 16–19 November 2002, pp. 271–282.
- [2] D.J.C. Mackay, “*Fountain Codes*” IEEE Proc. Doi. 20050237 M pp.1062-1068, May 2005.
- [3] L. Keller, E. Drinea, and C. Pragouli, “*Online Broadcasting with Network Coding,*” in Fourth Workshop on Network Coding, Theory and Applications, 2008. NetCod 2008., Hong Kong, Jan. 2008, pp. 1–6.
- [4] S. Sorour and S. Valaee, “*A Network Coded ARQ Protocol for Broadcast Streaming over Hybrid Satellite Systems,*” in Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on, Sep. 2009, pp. 1098 –1102.
- [5] S. Sorour, S. Valaee, and N. Alagha, “*Joint Control of Delay and Packet Drop Rate in Satellite Systems using Network Coding,*” in Advanced satellite multimedia systems conference (asms) and the 11th signal processing for space communications workshop (spsc), 2010 5th, Sep.2010, pp. 46 –53.
- [6] S. Sorour and S. Valaee, “*On Minimizing Broadcast Completion Delay for Instantly Decodable Network Coding,*” in Communications (ICC), 2010 IEEE International Conference on, May 2010, pp. 1 –5.

- [7] P. Sadeghi, D. Traskov, and R. Koetter, “*Adaptive Network Coding for Broadcast Channels*,” in *Network Coding, Theory, and Applications*, 2009. NetCod ’09. Workshop on, June 2009, pp. 80 – 85.
- [8] P. Sadeghi, R. Shams, and D. Traskov, “*An Optimal Adaptive Network Coding Scheme for Minimizing Decoding Delay in Broadcast Erasure Channels*,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, 2010.
- [9] M. Muhammad, M. Berlioli, G. Liva, and G. Giambene, “*Instantly decodable network coding protocols with unequal error protection*,” in *Proc. of IEEE International Conference on Communications (ICC’2013)*, Sydney, Australia, June 2013, pp. 5120–5125
- [10] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “*Network Information Flow*,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.
- [11] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, “*Index Coding With Side Information*,” *Information Theory, IEEE Transactions on*, vol. 57, no. 3, pp. 1479 –1494, Mar. 2011.
- [12] L. Keller, E. Drinea, and C. Pragouli, “*Online Broadcasting with Network Coding*,” in *Fourth Workshop on Network Coding, Theory and Applications*, 2008. NetCod 2008., Hong Kong, Jan. 2008, pp. 1–6.

- [13] J. K. Sundararajan, D. Shah, and M. Medard, “*ARQ for Network Coding*,” in IEEE International Symposium on Information Theory, 2008. ISIT 2008., Toronto, ON, Jul. 2008, pp. 1651–1655.
- [14] A. Tehrani, A. Dimakis, and M. Neely, “*Bipartite Index Coding*,” in Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on, Jul. 2012, pp. 2246–2250.
- [15] M. Bogino, P. Cataldi, M. Grangetto, E. Magli, G. Olmo, “*Sliding-Window Digital Fountain Codes for Streaming of Multimedia Applications*,” in IEEE International Symposium on Circuits and Systems (ISCAS), New Orleans, USA, May 2007.
- [16] Elias, P. “*Coding for two noisy channels*”. Information Theory, Third London Symposium, p. 61-76, 1955.
- [17] Finamore, Weiler A., “*Avanços em Códigos Fontanais*”. XXIX Simpósio Brasileiro de Telecomunicações - SBrT’11, 02-05 de outubro de 2011, Curitiba, PR.
- [18] J.W. Byers, M. Luby, M. Mitzenmacher. “*A Digital Fountain Approach to Asynchronous Reliable Multicast*”. IEEE J. on Selected Areas in Communications, Special Issue on Network Support for Multicast Communications, 2002.
- [19] J.W. Byers, M. Luby, M. Mitzenmacher, A. Rege. “*A Digital Fountain Approach to Reliable Distribution of Bulk Data*”. Proceedings of ACM SIGCOMM ’98, pages 56–67, Vancouver, September 1998.

- [20] D. Bertsekas and R. Gallager, “*Data Networks*”. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1987.
- [21] J. Metzner, “*An Improved Broadcast Retransmission Protocol*,” IEEE Transactions on Communications, vol. 32, no. 6, pp. 679 – 683, Jun.1984.
- [22] J. Nonnenmacher, E. Biersack, and D. Towsley, “*Parity-based Loss Recovery for Reliable Multicast Transmission*,” IEEE/ACM Transactions on Networking, vol. 6, no. 4, pp. 349 –361, Aug. 1998.
- [23] B. Adamson, C. Bormann, M. Handley, and J. Macker, “*NACK-Oriented Reliable Multicast Protocol*,” IETF RFC, Nov. 2004.
- [24] M. Luby, “*Tornado Codes: Practical Erasure Codes Based on Random Irregular Graphs*”, RANDOM 1998.
- [25] Shokrollahi, A.: “*Raptor codes*”. Technical report, Laboratoire d’algorithmique, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2003. Available from algo.epfl.ch/.
- [26] Richardson, T., Shokrollahi, M.A., and Urbanke, R.: “*Design of capacity-approaching irregular low-density parity check codes*”, IEEE Trans. Inf. Theory, 2001, 47, (2), pp. 619–637.
- [27] Richardson, T., and Urbanke, R.: “*Efficient encoding of low-density parity-check codes*”, IEEE Trans. Inf. Theory, 2001, 47, (2), pp. 638–656.

- [28] M. Luby, M. Mitzenmacher, A. Shokrollahi. "*Analysis of Random Processes via And-Or Tree Evaluation*". Proceedings of 9th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California, January 25–27, 1998.
- [29] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, V. Stemann. "*Practical Loss-Resilient Codes*". Proceedings of 9th Annual ACM Symposium on Theory of Computing, 1997.
- [30] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman. "*Efficient Erasure Correction Codes*". IEEE Trans. on Information Theory, Special Issue on Codes and Graphs and Iterative Algorithms, Vol. 47, No. 2, Feb. 2001.
- [31] Schulzrinne, Henning. "*Audio and Video Over Packet Networks — Issues, Architecture and Protocols*", Interop'94, Paris, France, 1994.
- [32] Schulzrinne, Henning, et al. "*RTP: A Transport Protocol for Realtime Applications*", Internet RFC 1889, Internet Engineering Task Force, Jan.1996.
- [33] Campbell, Andrew, Coulson, G., Hutchison, D. "*Flow Management in a Quality of Service Architecture*", In: IFIP International Conference on High Performance Networking, 5, 1994, Grenoble, France. Proceedings ... Grenoble, 1994.