Software para o Ensino do Protocolo de Janelas Deslizantes

MARIANE MOREIRA DE SOUZA[1]
RICARDO MARTINS DE ABREU SILVA[2]
HUMBERTO CÉSAR BRANDÃO DE OLIVEIRA[1]

UFLA - Universidade Federal de Lavras
DCC - Departamento de Ciência da Computação
Cx Postal 37 - CEP 37200-000 Lavras (MG)
[1](mariane.souza,humberto.brandao)@gmail.com
[2]rmas@dcc.ufla.br

Resumo. O papel da Informática como ferramenta educativa tem crescido de maneira significativa nos últimos anos. Com o uso de uma ferramenta computacional, o ato de se aprender pode se tornar mais simples e prazeroso, e o estudante, motivado, pode vir a aprimorar ainda mais o seu conhecimento. O principal objetivo deste trabalho é construir um *software* na área de Redes de Computadores, complementar ao texto sobre o Protocolo de Janelas Deslizantes da Camada de Enlace do livro *"Computer Networking with Internet Protocols and Technology"* de *William Stallings.*

Palavras-Chave: *software* educativo, redes de computadores, janelas deslizantes, *go-back-n, selective-reject*

Software for Sliding-Window Protocol teaching

Abstract. The importance of Informatics as a educative tool has grown significantly in the last years. Using a computational tool the learning act can become simpler and more pleasant and the student, motivated, can come to improve much more his learning. The main objective of this work is to construct a software in the computer networks area, to complement a text from *William Stallings.´*s book *"Computer Networking with Internet Protocols and Technology",* about Sliding-Window Protocol on Data-Link Layer

Keywords: *educative software, computer networks, sliding windows, go-back-n, selective-reject*

(Received February 20, 2005 / Accepted July 21, 2005)

1 Introdução

Este trabalho tem por objetivo a criação de um *software* de interface simples, porém ilustrativa, na área de Redes de Computadores, para servir como auxílio no ensino/aprendizado do protocolo de Janelas Deslizantes, funcionando como complemento ao texto sobre o assunto no livro *'Computer Networking with Internet Protocols and Technology '* de *William Stallings*.

A área de Redes de Computadores apresenta vários processos que podem ser executados na prática pelo uso do computador. O protocolo de Janelas Deslizantes foi escolhido para implementação neste trabalho por ser um assunto que muitas vezes os alunos apresentam uma dificuldade de compreensão necessária à implementação do algoritmo devido aos seus diversos fluxos de controle, além da possibilidade de utilizar o *software* como complemento à explicação do assunto no livro-texto de *William Stallings*. O *software* também poderá servir de base para o estudo do funcionamento de protocolos de camadas superiores à Camada de Enlace, como a Camada de Transporte da pilha de protocolos *TCP/IP*.

2 O protocolo de Janelas Deslizantes

O protocolo de Janelas Deslizantes pode ser descrito da seguinte maneira [Stallings(2003)] [4]: Dadas 2 estações *A* e *B*, tem-se que *A* representa a estação transmissora, capaz de transmitir *w* quadros de uma só vez sem receber nenhum tipo de reconhecimento do recebimento destes quadros, e *B* a estação receptora, capaz de receber até *w* quadros de uma só vez sem enviar nenhum reconhecimento. A estação *B* reconhece um quadro enviando um reconhecimento (*RR*) com o índice do próximo quadro desejado. Esse reconhecimento indica que a estação *B* está pronta para receber os próximos quadros a partir do índice em questão. A estação *A* mantém uma lista com os índices dos quadros que pode enviar, enquanto a estação *B* mantém uma lista com os índices dos quadros que espera receber. Essas listas podem ser vistas como "janelas de quadros", como ilustrado na Figura 1, baseada em ilustrações do livro de *William Stallings* [Stallings(2003)] [4].

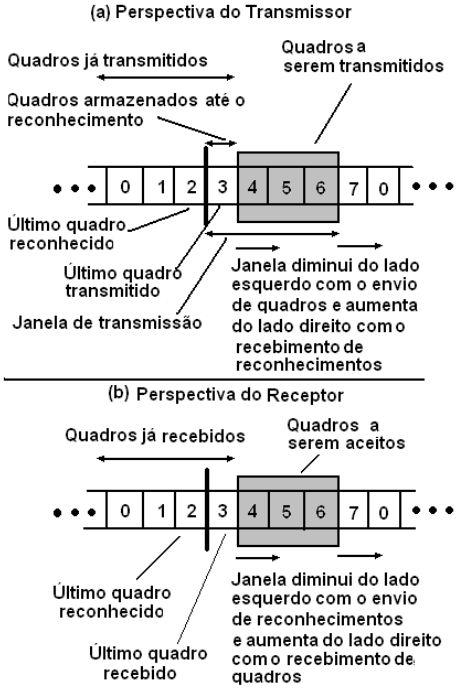

Figura 1: Esquema de Janelas do Protocolo de Janelas Deslizantes

A Figura 1 mostra janelas com os índices de 0 a 7 dos referentes quadros, de maneira que os quadros subsequentes começam a ser referenciados novamente a partir do índice 0. Segundo [Stallings(2003)] [4], para um campo de número de sequência de *K-bit*, resultando numa escala de número de sequência de 2^k, o tamanho máximo possível da janela de quadros é 2^{k-1}. No caso da Figura 1, o valor do campo de número de sequência é *3-bit*, resultando em uma escala de número de sequência de valor 8 (indíces de 0 a 7) e o tamanho máximo da janela permitido é, portanto, 4.

A Figura 1(a), representa a janela sob a perspectiva da entidade transmissora. Neste caso, o retângulo sombreado indica os quadros a serem enviados. A cada vez que um quadro é enviado, o retângulo sombreado diminui da esquerda para a direita, e quando um reconhecimento é recebido, o retângulo cresce da esquerda para a direta. A barra vertical é utilizada para marcar o início da janela de quadros a serem transmitidos, e se move para a direita à medida que reconhecimentos vão sendo recebidos. Os índices anteriores à barra vertical indicam os quadros já transmitidos e que já receberam reconhecimento. Os índices entre a barra vertical e o retângulo sombreado representam os quadros que foram transmitidos, porém ainda não receberam reconhecimento. Esses quadros ainda não reconhecidos devem ser armazenados, no caso da necessidade de retransmissão dos mesmos.

A Figura 1(b) representa a janela sob a perspectiva da entidade receptora, cuja especificação é correspondente à janela da entidade transmissora, porém de maneira oposta. Por exemplo, enquanto na janela do transmissor a parte sombreada indica os quadros a serem enviados, na janela do receptor isso representa os quadros a serem recebidos, e assim sucessivamente.

Observe que o protocolo de Janelas Deslizantes fornece uma forma de controle de fluxo: a entidade receptora (*B*), precisa estar apta a receber apenas *w* quadros de uma só vez, garantindo que não haja sobrecarga na transmissão dos dados. Além disso, existe a vantagem com relação a outros protocolos como o *Stop-And-Wait* que é a maior eficiência na utilização do meio, uma vez que não é necessário aguardar o reconhecimento de cada quadro para que o seu subsequente possa ser enviado.

O protocolo implementa também mecanismos de controle de erros de transmissão. São eles: *Go-back-N ARQ* e *Selective-reject ARQ*, ambos implementados no *software* apresentado neste artigo.

Conforme mencionado em [4] [Stallings(2003)], o *Go-back-N ARQ* é o mecanismo de controle de erro mais comum e utilizado. Caso ocorram erros na transmissão de quadros ou reconhecimentos podem acontecer os seguintes casos (2):

- Ao receber um quadro com erro, a estação receptora mandará um quadro de rejeição (*REJ*) para aquele quadro. Neste caso, a estação transmissora deverá retransmitir o quadro com erro e todos os subsequentes.

- A estação transmissora determina um tempo para que o quadro enviado seja reconhecido. Caso esse tempo expire, e a estação receptora não tenha enviado nenhum reconhecimento ou quadro de rejeição, ou esses se perderam durante a transmissão, a estação transmissora envia um quadro de "aviso"(*RR PBit*), indicando que a estação receptora deve enviar um reconhecimento, ou rejeição de quadro. O tempo de reconhecimento é setado até um determinado número de vezes. Caso não haja resposta, a estação transmissora reinicializa a transmissão.

- Uma vez que o reconhecimento é cumulativo, caso ocorra erro na transmissão do reconhecimento de um quadro pela estação receptora, antes mesmo de receber o reconhecimento do quadro em questão, a estação transmissora envia o próximo quadro e este é reconhecido pela estação receptora, anulando a perda do primeiro reconhecimento.

O *Selective-reject ARQ* é outro mecanismo de controle de erros, no qual apenas os quadros que foram enviados com erro são retransmitidos (*SREJ*). Para isso, a estação receptora mantém armazenados todos os quadros recebidos de forma correta após a chegada do quadro com erro, porém fora de ordem e os ordena quando o quadro danificado for recebido novamente. No caso de erros na transmissão do reconhecimento, os mecanismos são os mesmos utilizados no *Go-back-N ARQ*.

A Figura 3 ilustra um exemplo de transmissão de quadros posteriores (subsequentes) ao quadro que chegou com erro, utilizando o mecanismo de controle de erros *Selective-reject*. Como na Figura 2, o erro ocorre na transmissão do quadro número quatro. Porém, neste caso, a estação receptora armazena os próximos quadros enviados pela estação transmissora (quadros cinco e seis), e esta, ao receber um *SREJ*4, retransmite apenas o quadro recebido com erro (quatro), e continua normalmente a transmissão dos quadros posteriores (sete, zero e um).

O *Selective-reject ARQ* minimiza a quantidade de retransmissão de quadros, porém, o fato da estação receptora ter que manter armazenados os quadros recebidos fora de ordem e ter um mecanismo lógico de ordenação para os mesmos, faz com que o *Go-back-N ARQ* seja o mecanismo mais utilizado na prática atualmente [Stallings(2003)] [4].

3 O Projeto

O *software* apresentado neste trabalho realiza a transmissão de quadros entre duas entidades pelo protocolo de Janelas Deslizantes, com mecanismo de controle de

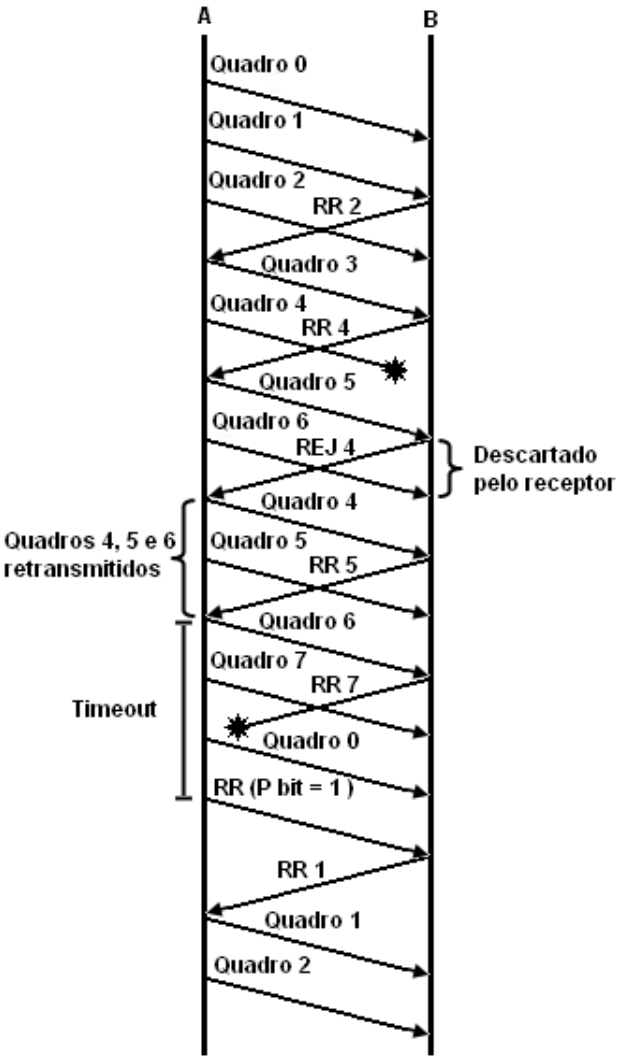

Figura 2: Transmissão de quadros utilizando o mecanismo de controle de erros *Go-back-N ARQ*

erros *Go-back-N ARQ* e *Selective-reject ARQ*. A implementação do protocolo e dos mecanismos de controle de erros, bem como a interface que mostra o funcionamento do protocolo através da notação gráfica descrita na seção 2, foram baseadas em ilustrações e conceitos abordados no livro *'Computer Networking with Internet Protocols and Technology '* de *William Stallings*.

O projeto de desenvolvimento do *software* foi baseado no modelo Orientado à Objeto (*OO*). O *software* possui ao todo quatorze classes, sendo:

- Quatro classes de interface com o usuário: HostAView, HostBView, HostAPresentation, HostBPresentation.

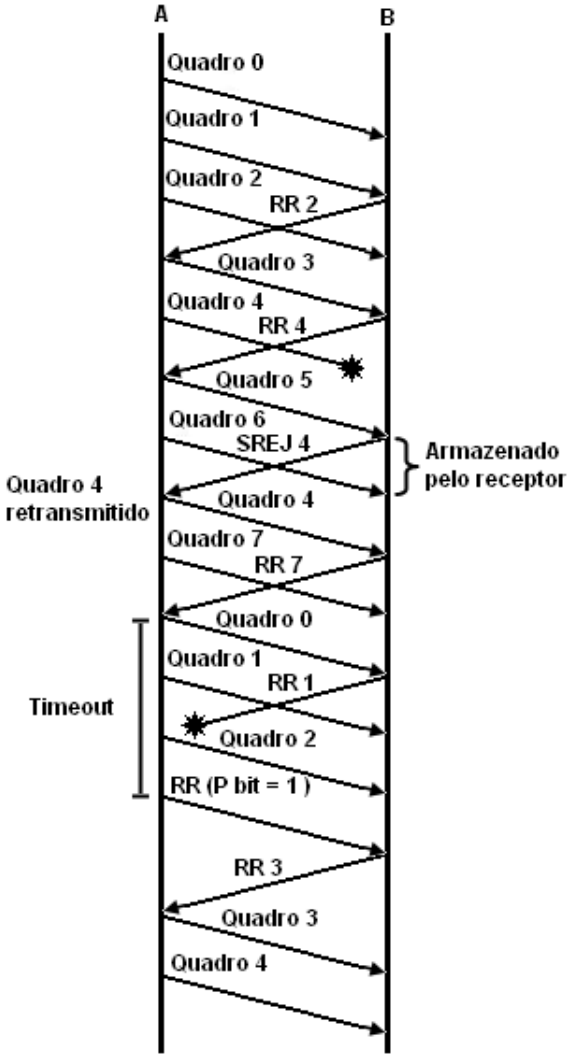

Figura 3: Transmissão de quadros utilizando o mecanismo de controle de erros *Selective-reject ARQ*

- Seis classes representando objetos do protocolo: Window, Frame, ACK, REJ, SREJ, RRPBit.
- Uma classe de configuração com parâmetros passados pelo usuário: SlidingWindowConfig.
- Uma classe responsável por diferenciar os dados de saída exibidos de maneira gráfica ou textual: InterfaceResult.
- Duas classes representando as entidades transmissora e receptora, com a implementação do protocolo e mecanismos de controle de erros: HostB e HostA

A arquitetura de comunicação utilizada no projeto foi a Cliente-Servidor, na qual existem estações responsáveis por solicitar conexões (estações transmissoras ou clientes) e outra por atendê-las (estação receptora ou servidora)(4). A comunicação entre estações clientes e servidoras pode ser visualizada através da estação monitora.

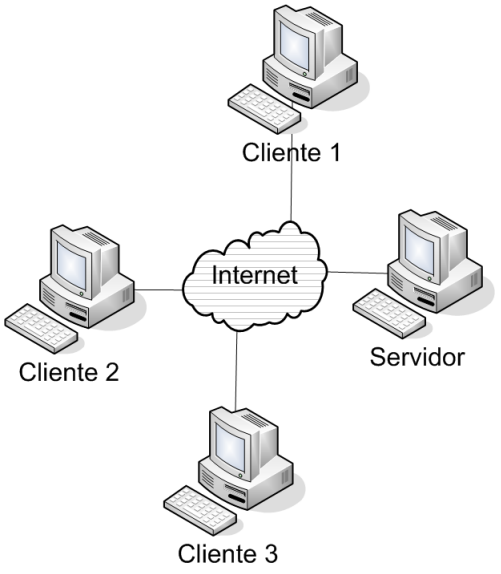

Figura 4: Arquitetura Cliente-Servidor

Utilizando-se essa arquitetura, o *software* pode ser executado em todas as combinações possíveis entre a parte monitora, servidora e cliente, indo desde a configuração com todos situados na mesma máquina, até a configuração totalmente distribuída com cada qual em máquinas distintas (Figura 5). Todas essas possíveis configurações foram utilizadas durante a realização dos testes nos laboratórios do Departamento de Ciência da Computação da Universidade Federal de Lavras (UFLA) em Minas Gerais, Brasil.

O estilo de comunicação utilizado entre as duas estações cliente e servidora é a comunicação via *sockets* [Stevens(1992)] [3], na qual a estação servidora é inicializada e permanece na espera de conexões da estação cliente para que se possa iniciar efetivamente a transmissão de dados.

Para a implementação do *software* utilizou-se a linguagem de programação Java, mais especificamente a plataforma J2SE [J2SE(2002)] [1], devido a sua alta portabilidade, orientação à objetos e interface amigável. O ambiente de desenvolvimento escolhido foi a IDE Net-Beans [NETBEANS(2000)] [2], pela facilidade de lidar com códigos e interface gráfica.

O *software* foi desenvolvido no ambiente Windows,

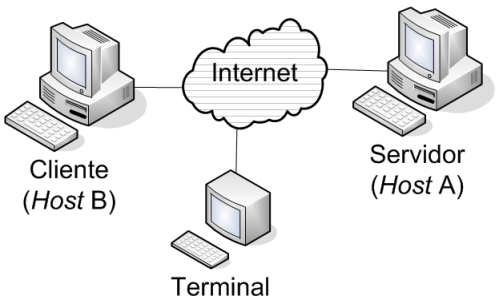

Figura 5: Arquitetura Cliente Servidor: cliente, servidor e monitor em máquinas distintas

porém os testes foram realizados nos ambientes Windows e Linux, demonstrando resultados satisfatórios em ambos os casos, em termos de velocidade de execução.

O programa está dividido em duas partes: a primeira parte representa a estação transmissora de quadros (servidor) e a segunda a estação receptora de quadros (cliente). Primeiro, inicializa-se a parte do programa que representa a estação servidora, responsável pelo recebimento de conexões (*Host A*). Para isso, basta clicar no botão com o rótulo "Inicializar"situado na interface do programa tal como ilustra a Figura 6. Depois disso, inicializa-se o programa com a parte que representa a estação cliente (*Host B*). Para que se inicie a comunicação, é necessário que o usuário preencha (na parte da estação cliente) alguns campos, como ilustrado na Figura 7:

- IP do servidor com o qual ele deseja se comunicar;
- Campo do Número de Sequência, que indica a escala máxima a ser utilizada na configuração dos quadros (2^k). Indica a amplitude. No caso da Figura 1, esse número é 3 (*3-bit*), resultando em uma escala máxima de 8 (0 .. 7);
- O tamanho da janela, que indica o número máximo de quadros a serem enviados ou recebidos de uma vez. Para que não ocorram erros, o tamanho da janela deve ser no máximo 2^{k-1}, onde *k* representa o campo do número de sequência, definido acima;
- Tipo de mecanismo de controle de erros: *Go-back-N ARQ* ou *Selective-reject ARQ*;
- Porcentagem de erro (quadros danificados e erros de propagação de quadros) desejada para a simulação do processo. A vantagem neste caso é que se pode atribuir porcentagens de erros de quadro e reconhecimento, ou erros na própria propagação dos mesmos.

A simulação da ocorrência de erros foi implementada utilizando-se a função *random* da biblioteca matemática *Math* da própria linguagem de programação *Java*. A implementação foi feita da seguinte maneira: as porcentagens de erro de propagação de quadro e de quadro com erro são escolhidas pelo usuário e armazenadas em duas variáveis. Um quadro só será transmitido se o valor da variável que guarda a porcentagem de erro de propagação for menor ou igual ao número aleatório gerado pela função *random* naquela vez. Caso o quadro tenha sido transmitido com sucesso, o programa verifica se o mesmo não apresenta erros. Um quadro não apresentará erros se o valor da variável que guarda a porcentagem de erro de quadro for menor ou igual ao número aletório gerado pela função *random* naquela vez.

Após o preenchimento dos dados, o usuário clica em um botão com o rótulo "Conectar"(Figura 7) e é iniciado o processo de simulação da transmissão de quadros de dados, segundo o protocolo de Janelas Deslizantes, como mostra a Figura 8, com telas geradas a partir da execução do *software*. Desta maneira, o usuário pode visualizar diversas instâncias do processo, inclusive situações que dificilmente seriam previstas em uma leitura da teoria subjacente.

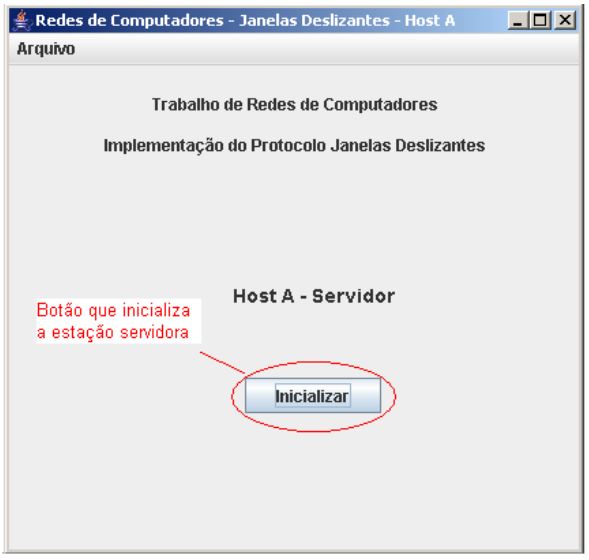

Figura 6: Tela inicial do *software* representando a estação *Host A*

Duas situação bastante comuns que podem ocorrer durante a simulação do protocolo é o erro na transmissão de um quadro e o erro na transmissão de um reconhecimento. Observou-se, por exemplo, a situação na qual a estação receptora de quadros (*Host A*), ao ter recebido um quadro com erro (quadro dois), enviou um

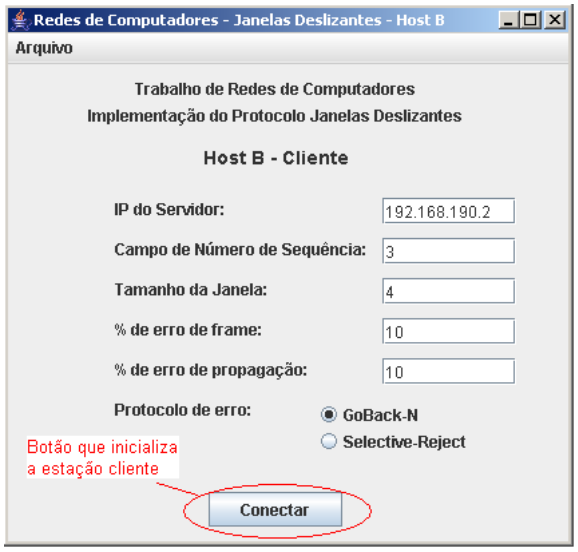

Figura 7: Tela inicial do *software* representando a estação *Host B*

quadro de rejeição (*REJ*2), descartando os quadros subsequentes (quadro três). Quando a estação transmissora recebeu o quadro de rejeição, esta reenviou o quadro dois e todos os subsequentes, conforme a explicação do protocolo com mecanismo de controle de erros por *Go-back-N*, descrita na seção 2. A saída exibida ao usuário é semelhante à seguinte:

Estação Transmissora de Quadros:
... ***Frame 2 enviado***
Janela: ...0 1 \| 2 [3] 4 5 6 7 ...
Frame 3 enviado
Janela: ...0 1 \| 2 3 [] 4 5 6 7 ...
REJ 3 recebido
Janela: ...0 1 \| [2 3] 4 5 6 7 ...
Frame 2 enviado
Janela: ...0 1 \|2 [3] 4 5 6 7 ...
Frame 3 enviado
Janela: ...0 1 \|2 3 [] 4 5 6 7 ...
RR 4 recebido
Janela: ...0 1 2 3 \|[4 5] 6 7 ...
Estação Receptora de Quadros:
... ***Frame 2 recebido com erro!***
Frame inesperado recebido: Frame 3
REJ 3 enviado
Frame 2 recebido
Janela: ...0 1 \|2 [3] 4 5 6 7 ...
Frame 3 recebido
Janela: ...0 1 \|2 3 [] 4 5 6 7 ...
RR 4 enviado
Janela: ...0 1 2 3 \| [4 5] 6 7 ...

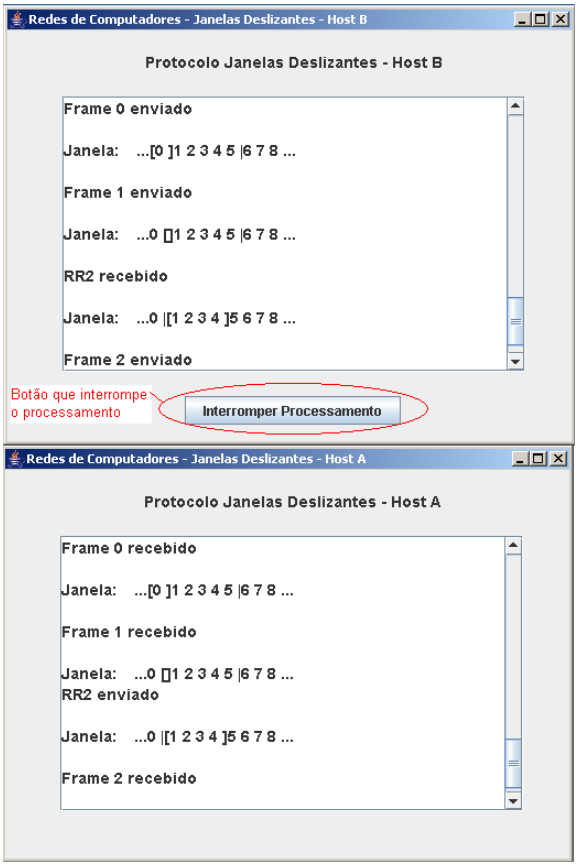

Figura 8: Saída (*Output*) do *software* para o protocolo de Janelas Deslizantes sem ocorrência de erros

Também foi observada a ocorrência do erro durante a transmissão de um reconhecimento, por exemplo, o reconhecimento do quadro cinco (*RR*6), pela estação receptora, fazendo com que a estação transmissora enviasse um quadro de aviso (*RRPBit*), pedindo um reconhecimento de quadro, uma vez que o tempo de reconhecimento já havia se esgotado e sua janela estava vazia, não podendo enviar novos quadros sem ter recebido reconhecimentos. A estação receptora recebeu o quadro de aviso e enviou novamente o reconhecimento para o quadro cinco (*RR*6) (Figura 9).

Uma facilidade extra do *software* é que o processo em execução, pode ser interrompido a qualquer momento, bastando que o usuário dê um clique no botão com o rótulo "Interromper Processamento"(Figura 8), para que se faça uma análise mais profunda sobre determinado trecho de saída do *software*, permitindo que retorne à sua execução em qualquer momento, bastando apenas um novo clique naquele mesmo botão (neste momento com o rótulo "Continuar Processamento- Figura 9). O *software* desenvolvido ilustra a teoria sobre

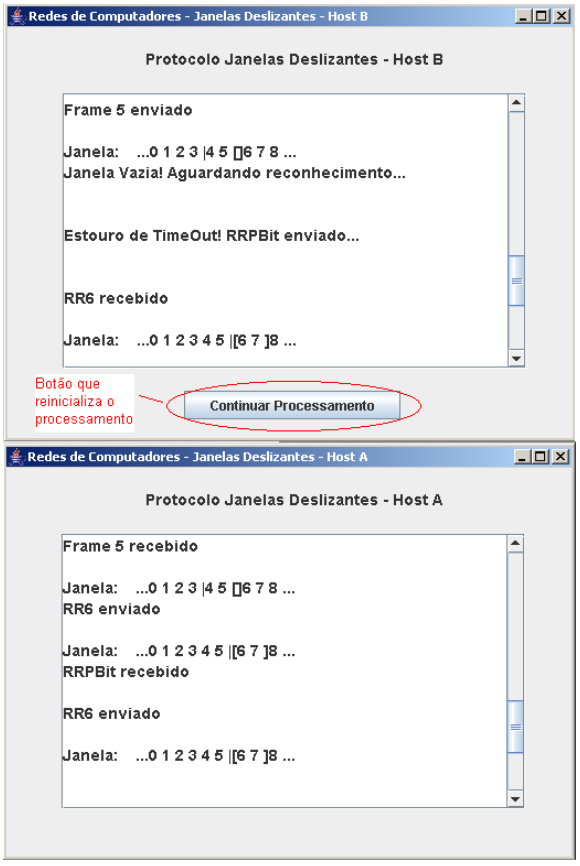

Figura 9: Saída (*Output*) do *software* para o protocolo de Janelas Deslizantes sem ocorrência de erros na transmissão de reconhecimentos

o Protocolo de Janelas Deslizantes do livro de *William Stallings*, permitindo que se faça uma associação entre aquilo estudado na teoria e o seu real comportamento analisado na prática por meio da utilização de uma ferramenta computacional.

4 Conclusão

Este trabalho teve como objetivo principal, a construção de um *software* na área de Redes de Computadores, para servir como auxílio no processo de ensino/aprendizado do protocolo de Janelas Deslizantes, funcionando como complemento ao texto sobre o assunto no livro *"Computer Networking with Internet Protocols and Technology"* de *William Stallings*.

A princípio, o *software* aponta para resultados aparentemente satisfatórios relativos à velocidade de execução, que deverão ser avaliados posteriormente de forma mais rigorosa.

As principais dificuldades encontradas durante a implementação do *software* ocorreram devido aos diversos tipos de fluxo de controles do protocolo. Outra dificuldade foi devido à implementação do *trace* na interface, representado pelo botão com o rótulo "Interromper Processamento", para que o usuário pudesse interromper o processo e analisar melhor os detalhes da transmissão dos quadros. A dificuldade se deu devido ao uso de *threads* (vários processos em execução) na implementação do protocolo, dificultando a interrupção do processamento.

O *software* apresentado é um protótipo e seu aperfeiçoamento é sugerido para futuros trabalhos, incluindo a criação de uma versão gráfica do mesmo. Porém, é importante destacar que o *software* funciona como auxílio no estudo do Capítulo referente à Camada de Enlace de Dados do livro de *William Stallings*. Como outras sugestões para futuros trabalhos, tem-se a utilização do *software* durante as aulas práticas da disciplina Redes de Computadores do Departamento de Ciência da Computação da Universidade Federal de Lavras (UFLA), com o intuito de testar sua usabilidade educacional, verificando sua eficiência de forma mais prática e rigorosa, corrigindo possíveis erros que possam vir a surgir. Outra sugestão seria a expansão do *software* para implementação de outros simuladores de protocolos oriundos da pilha de protocolos *TCP/IP*.

O *software* educativo pode ser adquirido no endereço: www.comp.ufla.br/∼mary/redes/src.zip

Referências

[1] Shafer, G. *'Java 2 Standard Edition'*. http://java.sun.com, Março de 2002.

[2] NetBeans. *'Welcome to the NetBeans Community'. http://www.netbeans.org/about/index.html*, Junhode2000.

[3] Stevens, W. R. *'Advanced Programming in the UNIX Environment'*, 1992.

[4] Stallings, W. *'Computer Networking with Internet Protocols and Technology'*, Setembro de 2003.