

# Redes de computadores 1 – Trabalho 3

Leonardo Becker de Oliveira – 20211779

## Janelas Deslizantes

### Introdução

Uma transmissão de dados se baseia em uma comunicação entre dois dispositivos, onde ambos conseguem receber todas as mensagens de forma correta sempre no menor tempo possível. Para que tenha certeza sobre a certidão e por conseguinte continuar com a comunicação, é necessário receber constantes confirmações sobre o recebimento correto ou incorreto das informações. Com isso nesse trabalho será abordado as janelas deslizantes volta N, método responsável por agilizar a transmissão de dados entre dois dispositivos baseando-se na confirmação de recebimento.

### Janela deslizante volta N

Imagina-se primeiro duas ocasiões, uma em que cada pacote só é enviado após receber a confirmação de que foi recebido corretamente, e outro em que todos os pacotes são enviados de uma única vez, recebendo a confirmação somente no final do envio de todos eles.

Na primeira forma, a troca de mensagens seria muito lenta, uma vez que para enviar um pacote levaria um certo tempo, e para receber uma confirmação levaria outra quantidade de tempo, ou seja, a forma conhecida como “para e espera” não é ágil o suficiente para a comunicação atual, em que o tamanho dos dados crescem constantemente.

Já na segunda forma, todos os pacotes seriam enviados de uma só vez, o que levaria certo tempo, mas não haveria no meio termo esse tempo para esperar uma confirmação, ou seja, todos os pacotes seriam enviado no menor tempo o possível. Porém, caso um pacote venha a ser enviado incorretamente, a resposta de negar o envio dos dados viria também só no final de tudo, exigindo que seja reenviado todos os pacotes enquanto todos não conseguirem ser bem recebidos de uma única vez tornando-o também uma forma pouco eficiente de transmissão.

Visto isso, o conceito de janela deslizante tem como objetivo trabalhar entre os dois extremos anteriormente descritos. Seu conceito se da por enviar N pacotes e esperar uma resposta do outro dispositivo, que pode vir ao final ou não do envio desses N pacotes. Caso seja recebido corretamente, é feito a liberação da janela enviando mais N pacotes. Caso seja recebido incorretamente, é necessário reenviar somente os últimos N pacotes enviados, e não todos os pacotes desde o início.

O nome volta N significa justamente esse conceito anteriormente descrito, em que os pacotes são transmitidos sempre em janelas de 0 até N pacotes, enquanto perdurar os dados.

Para realizar a transmissão dos pacotes via raw socket, aplicando o conceito de janela deslizante, dividiu-se cada etapa em uma função:

**Atualiza ponteiro:** Essa função tem por objetivo receber a sequência onde obteve-se a resposta de ACK ou NACK, e a partir disso tratar dos outros ponteiros, para que todos andem em sincronismo até o fim da sequência.

Caso o retorno do pacote seja um ACK (acknowledge), verifica quantos pacotes foram enviados em relação a resposta anteriormente obtida, e libera essa quantidade para ser transmitido com novos pacotes, retornando a transmissão a partir da resposta atual.

Caso a resposta seja um NACK (not acknowledge), verifica quantos pacotes foram enviados entre a resposta anteriormente obtida e a resposta atual, liberando essa quantidade de pacotes, mas diferente de quando se obtêm ACK, é necessário a retransmissão de todos os pacotes a partir da resposta atual.

```

// Atualiza o ponteiro que indica a sequencia do pacote recebido seja ACK ou NACK
void atualizaPonteiros(int *ack, int *pontosinal, int *pontosinalant, int *pontlocal){
    if (*ack == 1){
        // Caso a sequencia do ack seja maior que o ACK anterior, mas respeitando o limite da janela
        if (*pontosinal > (*pontosinalant % TAM_JANELA))
        {
            // ponteiro sinal + quantas janelas se passaram ate esse ponteiro ( para ele nao retornar ao inicio )
            *pontosinal = *pontosinal + (((*pontosinalant - 1) / TAM_JANELA) * TAM_JANELA) + 1;
        }

        if (*pontosinal < (*pontosinalant % TAM_JANELA))
        {
            // ponteiro sinal + quantas janelas se passaram ate esse ponteiro e por ele ser menor, isso quer dizer que ele ja
            // esta na próxima janela
            *pontosinal = *pontosinal + ((*pontosinalant / TAM_JANELA) * TAM_JANELA) + TAM_JANELA + 1;
        }

        if (*pontosinal == (*pontosinalant % TAM_JANELA))
        {
            *pontosinal = *pontosinalant + 1;
        }
    }

    // nack pode ser visto como um ack do valor anterior dependendo das condições
    if (*ack == 0) {
        if (*pontosinal > (*pontosinalant % TAM_JANELA))
        {
            *pontosinal = *pontosinal + (((*pontosinalant - 1) / TAM_JANELA) * TAM_JANELA);
        }
        if (*pontosinal < (*pontosinalant % TAM_JANELA))
        {
            if (*pontosinal == 0)
                *pontosinal = TAM_JANELA + ((*pontosinalant / TAM_JANELA) * TAM_JANELA);
            else
                *pontosinal = (*pontosinal - 1) + ((*pontosinalant / TAM_JANELA) * TAM_JANELA) + TAM_JANELA;
        }
        if ((*pontosinal == (*pontosinalant % TAM_JANELA)))
        {
            *pontosinal = *pontosinalant;
            *pontlocal = *pontosinal;
        }
    }
    *pontosinalant = *pontosinal; }

```

**Analisa devolução:** Função responsável apenas por receber o pacote de resposta, e redistribuir os dados entre os parâmetros anteriormente descritos.

```
void analisaDevolucao(pacote_t packdevolve, int  
pontosinal, int *pontosinal, int *ack)  
{  
    if (packdevolve.tipo == 0x00)  
    {  
        *ack = 0;  
        if ((packdevolve.sequencia == 0) && (pontosinalant ==  
0))  
            *pontosinal = 0;  
        else if (packdevolve.sequencia == 0)  
            *pontosinal = TAM_JANELA - 1;  
        else  
            *pontosinal = packdevolve.sequencia;  
    }  
    else if (packdevolve.tipo == 0x0A)  
    {  
        *ack = 1;  
        *pontosinal = packdevolve.sequencia;  
    }  
}
```

Além dessas duas funções, cabe ao processamento principal do programa liberar o envio dos pacotes, onde para cada nova resposta, as duas funções anteriormente descritas devem ser acionadas, atualizando os delimitadores e prosseguindo assim com a transmissão de dados de acordo com o método de janelas deslizantes.

## Conclusão

Conclui-se então que janela deslizante volta N, é um método importante a ser aplicado em uma comunicação, pois agilizam o envio e recepção de pacotes, o que permite que dados cada vez maiores e em menos tempos sejam enviados, lidando com os erros de uma forma eficiente.