

Laboratório: Comunicação em Canal Confiável Ideal

Disciplina: Redes de Computadores II

Professora: Angelita Rettore de Araujo

Introdução

Em redes de computadores, a transmissão de dados é um processo complexo, muitas vezes sujeito a desafios como perdas de pacotes e atrasos. No entanto, para compreender a necessidade e o funcionamento de protocolos de transporte mais sofisticados, é fundamental primeiro analisar o comportamento de um cenário ideal: um **canal de comunicação perfeitamente confiável**.

Neste laboratório, você explorará uma simulação que representa esse cenário ideal, onde a transmissão de dados ocorre de forma contínua e sem falhas impostas pelo canal. Usaremos os agentes UDP (User Datagram Protocol) e a aplicação CBR (Constant Bit Rate) para simular um fluxo de dados simples, permitindo a observação do tráfego em suas condições mais básicas, antes de introduzirmos as complexidades de um canal real.

Objetivos

1. Compreender e observar o comportamento da transmissão de dados em um canal de comunicação considerado "perfeitamente confiável".
2. Analisar o fluxo de pacotes quando não há mecanismos de controle de confiabilidade intrínsecos ao protocolo de transporte (UDP) e quando o canal não induz perdas.
3. Estabelecer uma base de conhecimento para comparar com cenários mais realistas de redes, que incluem perdas e atrasos.

Material Fornecido

- Script TCL para simulação de comunicação em canal confiável: `01-rdt1-canal-confiavel.tcl`
- Ferramentas para análise de tráfego de rede: `NAM (Network Animator)`.

Simulação 01: Comunicação Contínua em Canal Confiável

Passo 1: Configuração e Execução Inicial

1. O script `01-rdt1-canal-confiavel.tcl` simula um cenário onde um emissor (**Emissor**) envia dados para um receptor (**Receptor**) através de um canal sem perdas e sem atrasos adicionais (apenas o atraso de propagação inerente ao meio físico).
2. Execute o script utilizando os comandos abaixo no seu Terminal.

Execução no Linux

```
xhost +
docker run --rm -it -e DISPLAY=$DISPLAY -v $PWD:/ns2 -v /tmp/.X11-
unix:/tmp/.X11-unix gelirettore/ns2
```

```
ns 01-rdt1-canal-confiavel.tcl
```

Execução no macOS

Obs: Certifique-se de que o **XQuartz** esteja instalado e configurado no seu macOS para permitir conexões de rede (**Allow connections from network clients**) e que o comando **xhost +** tenha sido executado no Terminal antes de iniciar o contêiner.

```
xhost +  
docker run --rm -it -e DISPLAY=host.docker.internal:0 -v $PWD:/ns2  
gelirettore/ns2  
ns 01-rdt1-canal-confiavel.tcl
```

Passo 2: Análise do Comportamento

1. Após a execução, o NS2 gerará um arquivo **.nam** (**01-rdt1-canal-confiavel.nam**) que será automaticamente aberto pelo NAM. Observe a animação do fluxo de pacotes entre o Emissor e o Receptor.
2. O script também gera um arquivo de rastreamento (**01-rdt1-canal-confiavel.tr**). Você pode abri-lo com um editor de texto para uma análise detalhada dos eventos (envio e recebimento de pacotes).

Para Refletir e Discutir:

- Descreva o padrão de envio e recebimento de pacotes que você observou no NAM. Os pacotes são enviados em sequência? Há alguma interrupção no fluxo ou eles parecem fluir continuamente?
- Nesta simulação, o protocolo de transporte utilizado no emissor é o **UDP**, e a aplicação é do tipo **CBR (Constant Bit Rate)**. Qual a principal característica do UDP em relação à confiabilidade? Como o CBR se comporta nesse contexto ideal?
- Você observou o uso de Acknowledgments (ACKs) ou retransmissões de pacotes nesta simulação? Por que esses mecanismos não são observados aqui?
- Como as configurações de **queue-limit** (limite da fila do link) e **rate_** (taxa de envio da aplicação CBR) contribuem para o fato de não haver perdas de pacotes nesta simulação?
- Se o "canal confiável" representasse uma rede real e o Emissor enviasse dados a uma taxa muito maior do que a capacidade do link ou se o **queue-limit** fosse muito pequeno, o que poderia acontecer com os pacotes?

Reflexão Final

Esta simulação apresentou um cenário ideal de comunicação, onde o canal não introduz erros ou perdas. Na prática, redes de computadores estão sujeitas a diversos desafios que afetam a confiabilidade da entrega de dados, como congestionamento, interferências, falhas de hardware, entre outros.

Pense nos cenários reais que você já estudou ou observou em sua vida diária na Internet. Um protocolo de transporte "confiável" precisa ser capaz de superar as imperfeições do canal para garantir que os dados cheguem corretamente ao seu destino.

