

Trabalho Prático I Parte II de Introdução a Sistemas Computacionais

Lucas Ribeiro da Silva
2022055564
Engenharia de Sistemas
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
Email: lucasrsilvak@gmail.com

I. INTRODUÇÃO

Neste relatório consta as informações acerca do Trabalho Prático I Parte II da Disciplina de Introdução a Sistemas Computacionais. Neste trabalho observaremos o comportamento de dois protocolos TCP quando ao controle do congestionamento numa rede, o TCP Cubic e o TCP NewReno. O documento encontra-se separado nas seguintes seções, a primeira, a descrição do dispositivo e sistema operacional, em seguida os itens que foram solicitados no enunciado e por fim a conclusão.

II. DESCRIÇÃO DO AMBIENTE

Esta seção detalha as especificações do hardware e a configuração do ambiente de software utilizado para a instalação e execução do ambiente **NS-3.36.1**.

A. Especificações do Dispositivo

As informações acerca do dispositivo utilizado bem como o Sistema Operacional encontra-se na tabela abaixo.

Tabela I: Especificações de Hardware do Dispositivo

Componente	Detalhes
Dispositivo	Samsung GalaxyBook4
Processador	Intel(R) Core(TM) i5-120U (1.40 GHz)
Memória RAM	16,0 GB (utilizável: 15,7 GB)
Tipo de Sistema	Sistema operacional de 64 bits, processador x64
Sistema Operacional	Windows 11 Home Single Language, Versão 24H2

B. Ambiente de Configuração para NS-3

A instalação e execução do simulador **NS-3** (versão **NS-3.36.1**) foram realizadas através do **Windows Subsystem for Linux (WSL)**, dentro do sistema operacional Windows 11. Os detalhes da configuração do NS-3 são apresentados na Tabela II.

Tabela II: Detalhes do Ambiente de Configuração do NS-3

Configuração	Detalhes
Ambiente Auxiliar	Windows Subsystem for Linux (WSL)
Distribuição Linux no WSL	Ubuntu 22.04.5 LTS (Jammy Jellyfish)
Versão do NS-3 Utilizada	NS-3.36.1

III. TRABALHO PRÁTICO

Nesta seção faremos o que foi proposto nos enunciados. Na primeira parte, criaremos a topologia requisitada e em seguida, utilizaremos os protocolos TCP Cubic e TCP NewReno e comparemos o controle de algoritmo deles.

A. Primeira Parte

Para a primeira parte, definimos o gargalo de 10 Mbps, latência de 100 ms e taxa de erro de 0,001% com um fluxo único, e plotaremos os resultados obtidos para comparar os resultados do TCP Cubic e do TCP NewReno. A seguir a topologia e os resultados:



Figura 1: Topologia da Primeira Parte

Print da execução:

```
lucas@Twistzbook: ~/ns-allinone-3.36.1/ns-3.36.1$ ./ns3 run "
scratch/lab2-part1 --transport_prot=TcpCubic --nFlows=4 --da
taRate=10Mbps --delay=100ms --errorRate=0.00001"
[0/2] Re-checking globbed directories...
ninja: no work to do.

----- Lab 2 Part 1 Goodput (ns3::TcpCubic) -----
Flow 0 (N0->N3): 1257060 bytes received, Goodput: 529288 bps
Flow 1 (N0->N3): 1182600 bytes received, Goodput: 497937 bps
Flow 2 (N0->N3): 2093640 bytes received, Goodput: 881533 bps
Flow 3 (N0->N3): 2349140 bytes received, Goodput: 989112 bps
Average Flow Goodput: 724467 bps

lucas@Twistzbook: ~/ns-allinone-3.36.1/ns-3.36.1$ ./ns3 run "
scratch/lab2-part1 --transport_prot=TcpNewReno --nFlows=4 --
dataRate=10Mbps --delay=100ms --errorRate=0.00001"
[0/2] Re-checking globbed directories...
ninja: no work to do.

----- Lab 2 Part 1 Goodput (ns3::TcpNewReno) -----
Flow 0 (N0->N3): 1004480 bytes received, Goodput: 422939 bps
Flow 1 (N0->N3): 1093540 bytes received, Goodput: 460438 bps
Flow 2 (N0->N3): 1160700 bytes received, Goodput: 488716 bps
Flow 3 (N0->N3): 962140 bytes received, Goodput: 405112 bps
Average Flow Goodput: 444301 bps
```

Figura 2: Print da Execução

1) *Letra A:* Para o primeiro teste, comparamos a janela de congestionamento dos dois protocolos TCP utilizados, conforme o gráfico abaixo.

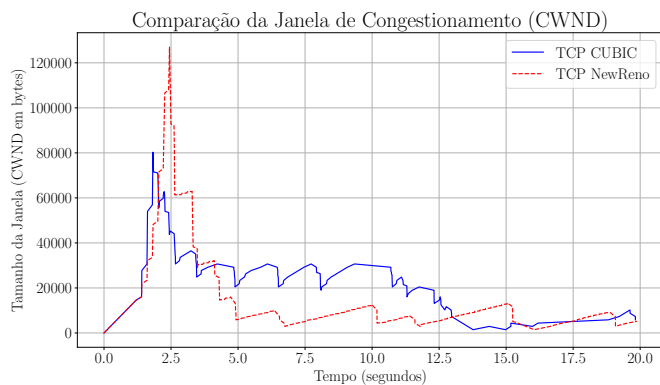


Figura 3: Comparação da Janela de Congestionamento

É possível perceber pelo gráfico o comportamento de partida lenta da janela de congestionamento, onde ela cresce rapidamente para descobrir o gargalo da rede e depois cai pela metade, até encontrar o comportamento linear da rede. Esse comportamento é visto tanto no NewReno quanto no CUBIC, embora o pico do NewReno seja muito maior. Uma diferença notável entre os algoritmos é a capacidade do Cubic de se manter constante ao longo do tempo aproveitando melhor a rede que o NewReno, tendo uma vazão média quase 2x maior que ele.

2) *Letra B:* Para o segundo teste, variamos o número de fluxos para 1, 2 e 4 e aumentamos progressivamente a latência da rede entre 50 e 300ms para analisar os resultados da variação na vazão da rede utilizando os protocolos selecionados.

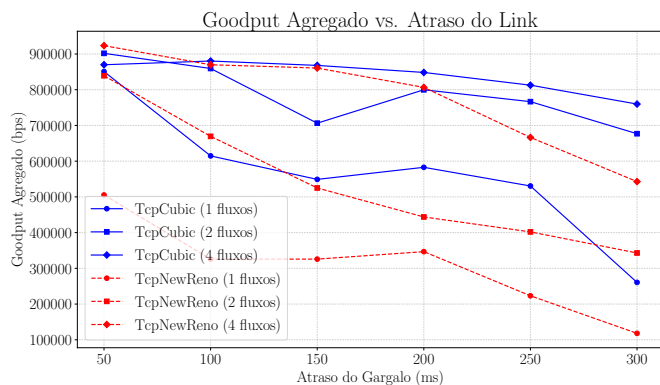


Figura 4: Goodput vs. Atraso

No segundo gráfico, é possível observar como o TCP Cubic mantém uma vazão maior em comparação ao TCP NewReno conforme o aumento da latência do sistema, isso ocorre pois o TCP Cubic tende a desvalorizar os Ack's na configuração da janela de congestionamento, mantendo uma vazão constante mesmo com o aumento do RTT.

3) *Letra C:* Para o terceiro teste, variamos o número de fluxos para 1, 2 e 4 e aumentamos progressivamente a taxa de erro da rede de 0,001% até 0,1% para analisar os resultados da variação na vazão da rede utilizando os protocolos selecionados.

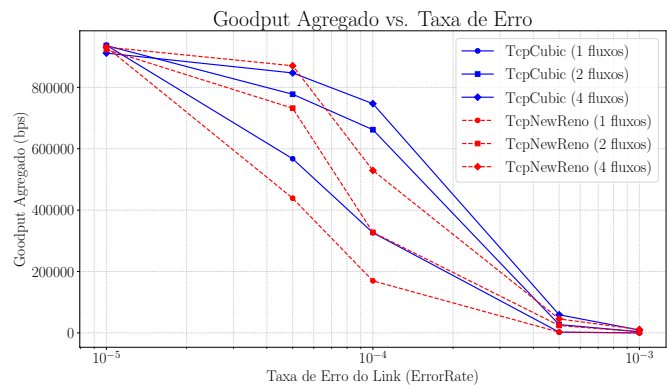


Figura 5: Goodput vs. Taxa de Erro

No terceiro gráfico, é possível observar como o TCP Cubic lida ligeiramente melhor com maiores taxas de erro que o TCP NewReno, mantendo uma vazão mais alta em todas as quantidades de fluxo. Entretanto, conforme a taxa de erros aumenta progressivamente, ambos os protocolos pioram significativamente.

B. Segunda Parte

Para a segunda parte, modificamos a topologia da rede para incluir dois nós de destino com latências significativamente diferentes. O primeiro com latência de 50 ms e o segundo com latência de 0.01 ms.

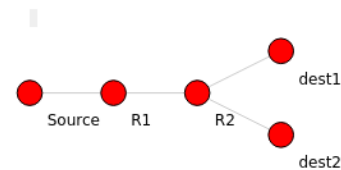


Figura 6: Topologia da Rede

Print da Execução:

```
lucas@Twistzbook: ~/ns-allinone-3.36.1/ns-3.36.1$ ./ns3 run "
scratch/lab2-part2 --transport_prot=TcpNewReno --nFlows=4 --
dataRate=1Mbps --delay=20ms --errorRate=0.00001 --run=0"
[0/2] Re-checking globbed directories...
ninja: no work to do.

----- Lab 2 Part 2 Goodput (ns3::TcpNewReno, Run 0) -----
Flows to dest1 (Short RTT): 2, Avg Goodput: 219768 bps
Flows to dest2 (Long RTT): 2, Avg Goodput: 243743 bps
PARSE_ME, ns3::TcpNewReno, 4, 0, 219768, 243743

lucas@Twistzbook: ~/ns-allinone-3.36.1/ns-3.36.1$ ./ns3 run "
scratch/lab2-part2 --transport_prot=TcpCubic --nFlows=4 --da
taRate=1Mbps --delay=20ms --errorRate=0.00001 --run=0"
[0/2] Re-checking globbed directories...
ninja: no work to do.

----- Lab 2 Part 2 Goodput (ns3::TcpCubic, Run 0) -----
Flows to dest1 (Short RTT): 2, Avg Goodput: 209011 bps
Flows to dest2 (Long RTT): 2, Avg Goodput: 250505 bps
PARSE_ME, ns3::TcpCubic, 4, 0, 209011, 250505
```

Figura 7: Print da Execução

Para a segunda parte, variamos o número de fluxos entre 2 e 8, definimos o gargalo de 1 Mbps, uma latência de 20 ms

no gargalo e uma taxa de erro de 0,001%. Então observamos a vazão para os diferentes números de fluxos para os dois destinos desejados.

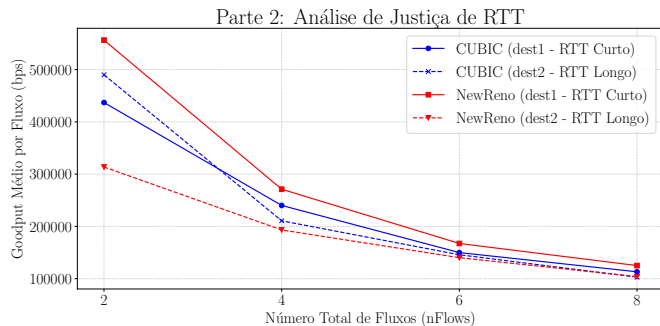


Figura 8: Análise de Justiça

Podemos observar pelo gráfico como o TCP Cubic equilibra melhor o fluxo entre os dois endereços de destino, com um fluxo parecido entre eles, enquanto isso, o TCP NewReno apresenta um comportamento um pouco desigual, priorizando o destino de menor latência, isso ocorre novamente porque o TCP NewReno utiliza-se bastante do uso de Ack's para configurar a janela de transporte, enquanto o Cubic busca reduzir a dependência desse parâmetro.

IV. CONCLUSÕES

O presente trabalho permitiu a compreensão aprofundada do controle de Congestionamento no protocolo TCP. Foi possível avaliar o comportamento de dois protocolos diferentes e observar o comportamento deles em diversas situações de redes diferentes. Em geral, o TCP Cubic mostrou-se um protocolo mais justo e equilibrado que o TCP NewReno, conseguindo aproveitar melhor as condições da rede sem priorizar os destinos de maior tecnologia. Os arquivos foram submetidos por meio do arquivo *Lab2_Lucas_Ribeiro.tar.gz* e estão disponibilizados no Github se necessário <https://github.com/lucasrsilvak/UFMG-Redes/tree/main/TP1%20Parte%202>.