

**INSTITUTO FEDERAL**  
**SANTA CATARINA**  
Campus São José

**Mini Projeto ADS - Avaliação de Desempenho do Protocolo  
TCP em condições de congestionamento: cubic x reno**

**Disciplina: Avaliação de Desempenho de Sistemas**

**Discente: Bruno Martins do Nascimento e Daniel Valdeley  
Marques**

**Docente: Eraldo Silveira E Silva**

**Curso: Engenharia de telecomunicações – IFSC.**

# Introdução

O protocolo TCP (Transmission Control Protocol) é utilizado na comunicação de dados, muito usado em situações de congestionamento, onde a rede pode se tornar sobrecarregada. Dois dos algoritmos de controle de congestionamento mais comuns são Cubic e Reno, cada um com suas próprias abordagens e características distintas.

O algoritmo Reno é uma das implementações mais antigas e amplamente adotadas. Ele opera em duas fases principais: aumento lento e prevenção de congestionamento. Na fase de aumento lento, a taxa de transmissão é aumentada de forma linear até que um limite de janela de congestionamento seja atingido. Após isso, o Reno entra na fase de prevenção de congestionamento, onde a janela de congestionamento é aumentada linearmente. No entanto, o Reno pode não se adaptar tão bem a redes modernas de alta largura de banda e latência variável, como redes de longa distância e alta velocidade.

Por outro lado, o algoritmo Cubic foi projetado para enfrentar esses desafios. Em vez de usar um aumento linear, o Cubic ajusta sua janela de congestionamento de forma cúbica. Isso permite uma adaptação mais rápida às mudanças na largura de banda e latência da rede, tornando-o mais adequado para redes de alta velocidade e latência variável.

## Experimento

- Tráfego de background (médio, alto). A ser definido no experimento:

médio = 500 Mbps

alto = 900 Mbps

- Tráfego de background somente UDP:

pc3 e pc4

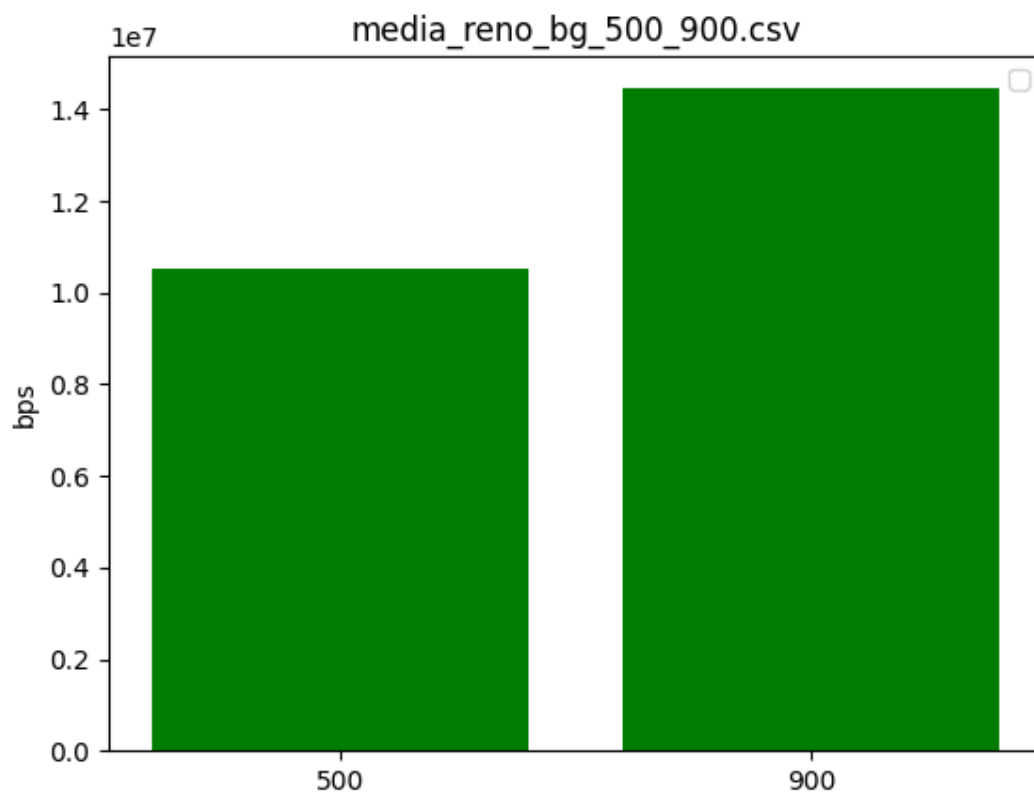
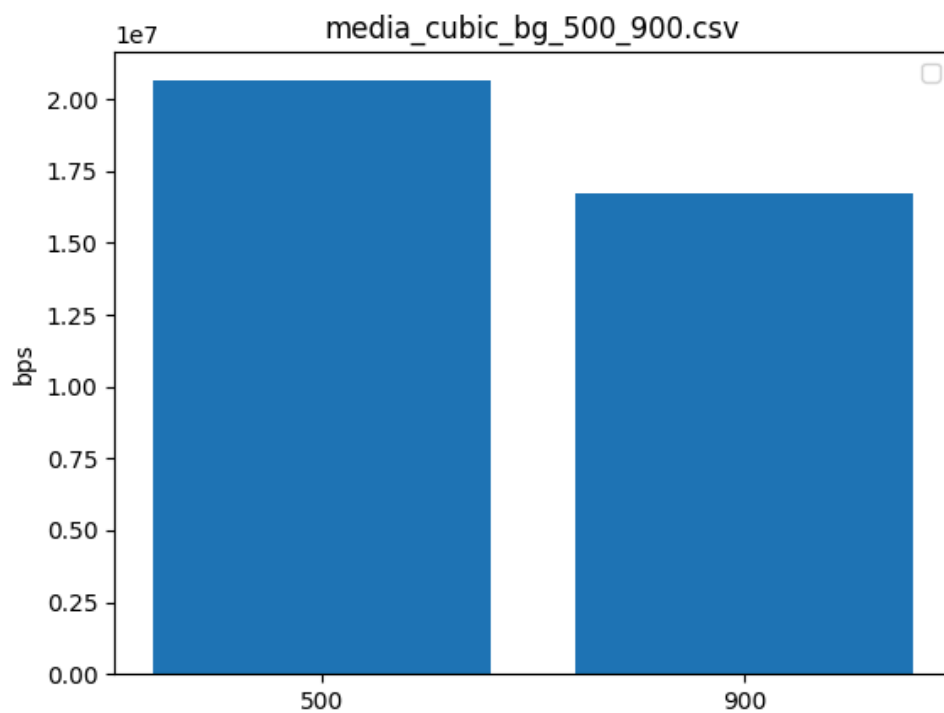
- Cenário com um fluxo TCP:

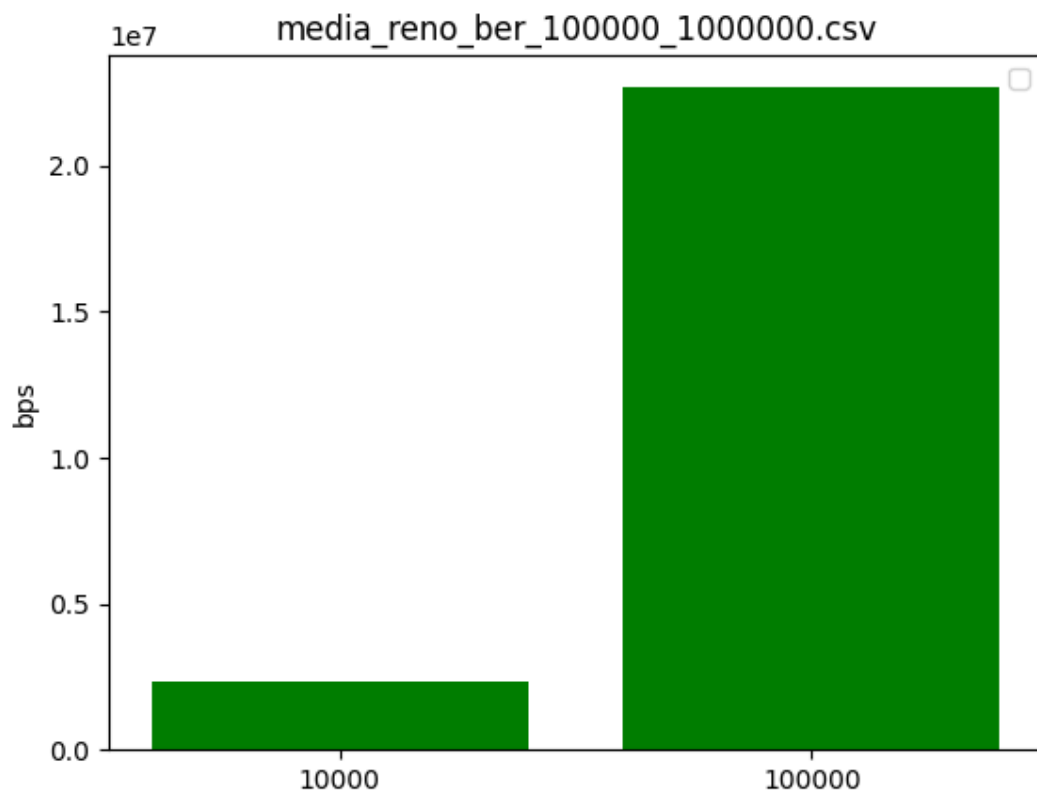
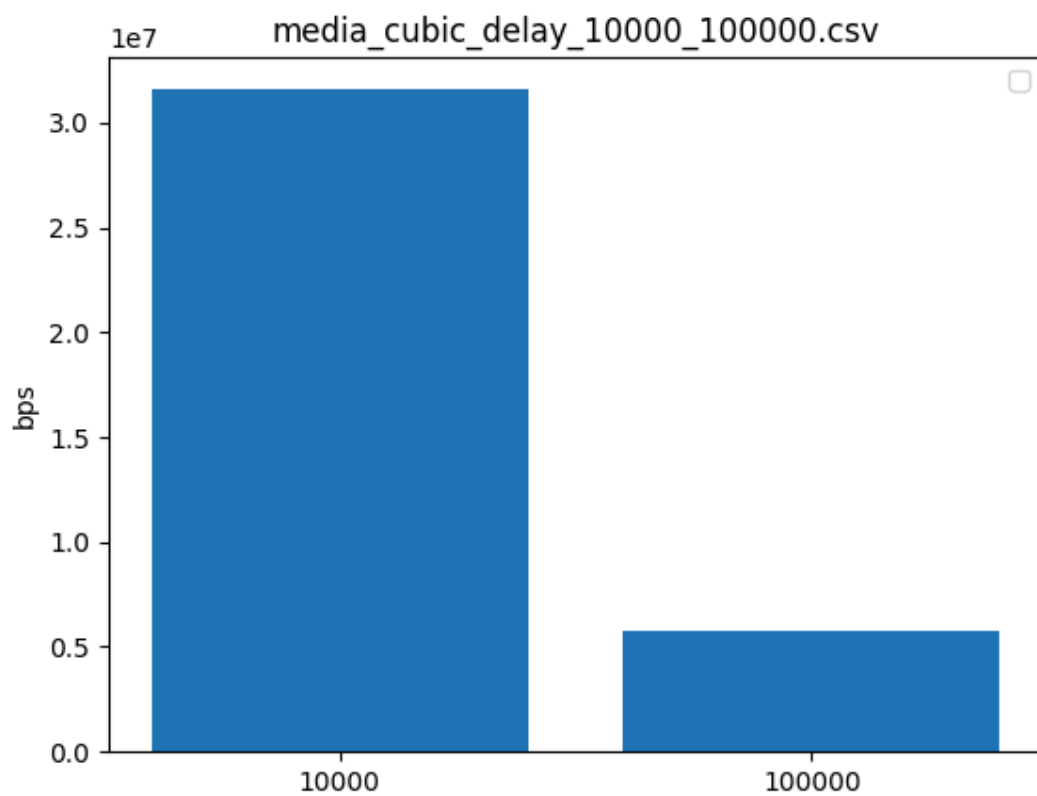
pc1 e pc2

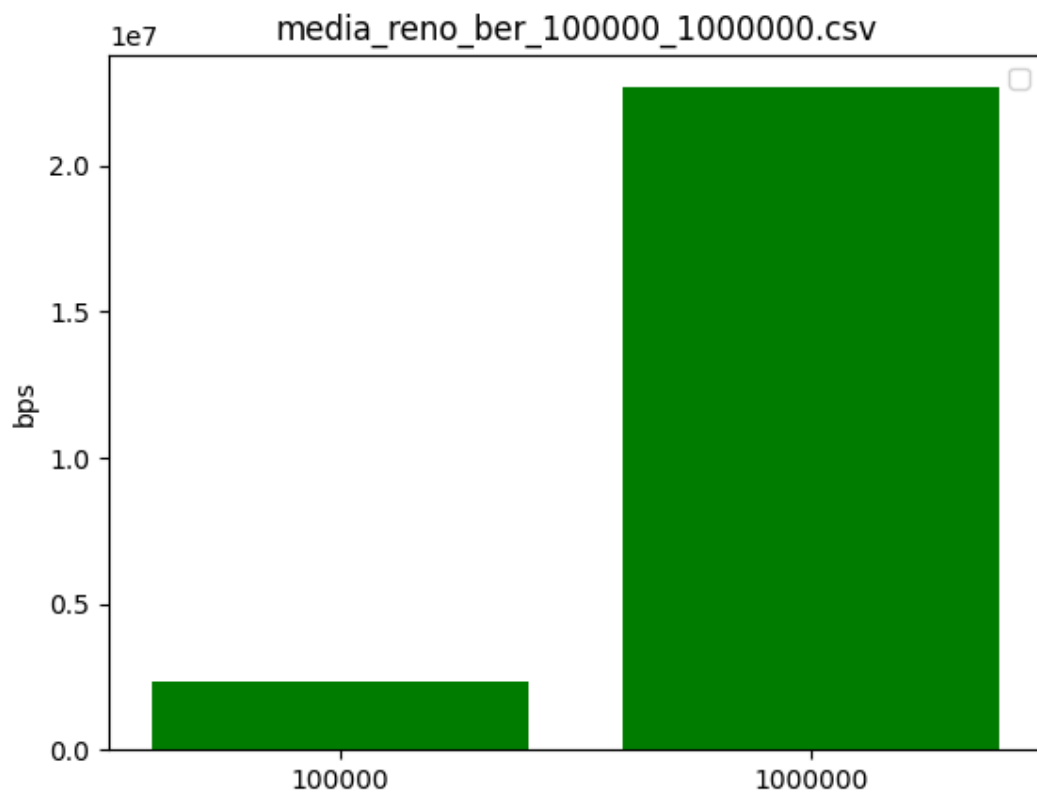
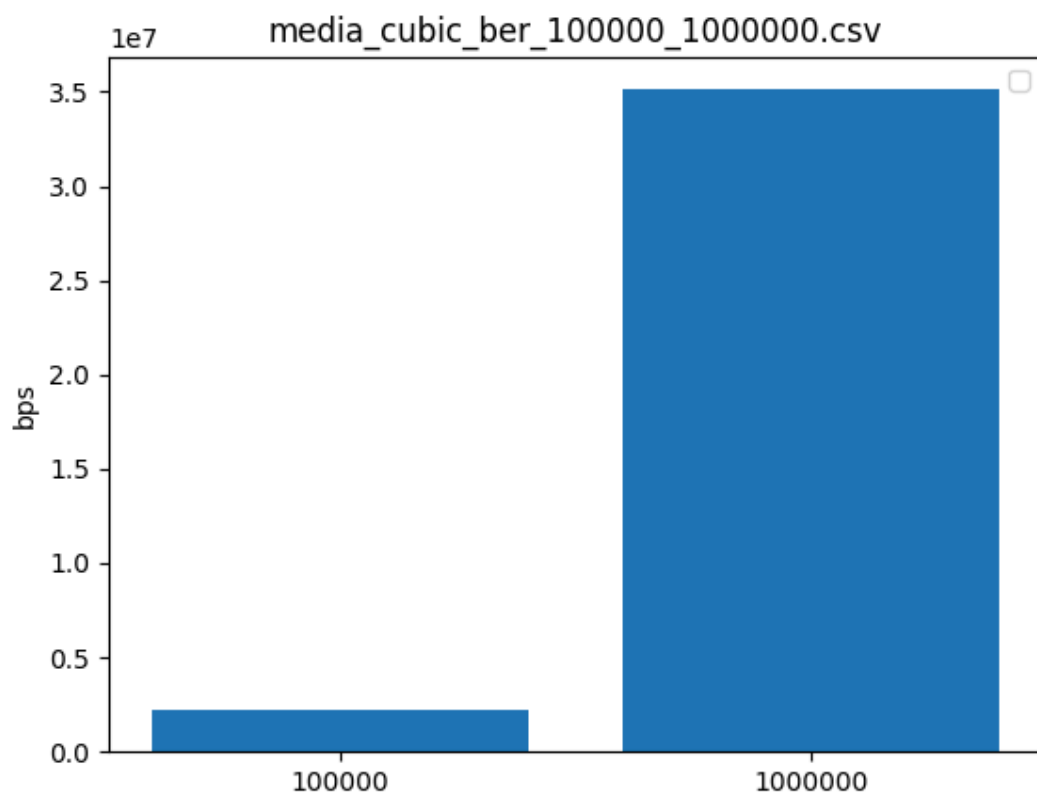
- Algoritmo de congestionamento: cubic e reno
- Links são sempre ethernet 1Gbps. Considere todos links com BER 0 mas no link entre roteadores com BER 1/1000000 e BER 1/100000
- Retardo forçado no link entre roteadores: 10ms e 100ms
- Usar imunes e IPERF
- Usar janela default do TCP:  
pc1 e pc2

- Comparação de resultados usando gráficos de barras
- Automatização e coleta de dados para a geração de gráficos

## Análise dos resultados







Link para os resultados do experimento:

<https://github.com/DanielValdeley/mini-projeto-imunes.git>