

Análise de Algoritmos de Roteamento - Sinalgo

Pedro Guimarães Jambo
Instituto de Computação - UFRJ

Abstract—O artigo aborda a análise de algoritmos de roteamento de pacotes em redes de computadores. O experimento envolve a realização de diversas simulações envolvendo os algoritmos de roteamento implementados pelo software Sinalgo, levando à posterior análise de robustez de cada algoritmo, com base nos resultados obtidos.

I. INTRODUÇÃO

As redes de computadores desempenham um papel crucial na conectividade global, sustentando a comunicação instantânea e a transferência eficiente de dados em escala global. No contexto dessa complexidade de interconexões, o roteamento é essencial para otimizar o fluxo de informações. Este artigo visa realizar uma análise abrangente do roteamento em redes de computadores, explorando a eficácia e robustez de diferentes algoritmos através do software Sinalgo.

II. PRINCÍPIOS BÁSICOS DO ROTEAMENTO DE PACOTES

O roteamento de pacotes envolve o processamento de rotas para a transmissão eficiente de pacotes de um sistema final para outro. Em uma analogia com carros, onde os pacotes são os veículos, o roteamento é como o sistema de sinalização e semáforos que orientam cada "carro" do ponto A ao ponto B de maneira eficiente. Os roteadores são os dispositivos responsáveis por assegurar as rotas para que os pacotes atinjam o sistema final de destino. Esses dispositivos, através dos enlaces, se comunicam e utilizam algoritmos para calcular e garantir as rotas adequadas.

III. FUNDAMENTOS TEÓRICOS

A. Intervalos de Timeout (TimeoutInterval)

Em redes de computadores, o *TimeoutInterval* é crucial em protocolos de comunicação: ele determina o tempo máximo de espera para o sucesso de uma operação, como (e principalmente) o tempo máximo de espera para uma confirmação do recebimento de um pacote de dados. No contexto do experimento, esse valor será calculado e plotado para permitir comparações mais detalhadas durante a execução dos algoritmos na simulação usando o software Sinalgo. O cálculo envolve valores como *EstimatedRTT* (estimativa do tempo médio de ida e volta) e *devRTT* (desvio padrão dos tempos de ida e volta), como demonstrado nas seguintes fórmulas:

$$\text{EstimatedRTT} = (1 - \alpha) \times \text{EstimatedRTT} + \alpha \times \text{SampleRTT} \quad (1)$$

$$\text{DevRTT} = (1 - \beta) \times \text{DevRTT} + \beta \times |\text{SampleRTT} - \text{EstimatedRTT}| \quad (2)$$

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 \times \text{DevRTT} \quad (3)$$

B. Tipos de Algoritmos de Roteamento

Os algoritmos de roteamento de pacotes podem ser divididos em dois grupos: Algoritmos de Estado de Enlace e Algoritmos de Vetor de Distância. No primeiro grupo, cada nó mantém informações abrangentes sobre a condição atual de todos os enlaces, compartilhando esses dados na rede para construir tabelas de estado de enlace. No segundo grupo, cada nó mantém uma tabela de vetor de distância, estimando as distâncias para destinos conhecidos e os próximos saltos para alcançá-los, atualizando essas informações ao trocar periodicamente informações com seus vizinhos.

Diante disso, os três algoritmos a serem experimentados pertencem ao grupo de Algoritmos de Estado de Enlace: armazenam informações sobre o estado de todos os nós no mapeamento de roteadores na simulação.

IV. EXPERIMENTO

Este artigo apresenta um experimento utilizando o software Sinalgo para avaliar três algoritmos de roteamento por meio de simulações. Após cada round de simulação, o software retorna: custo de rede da execução, tempo de execução do algoritmo (RTT) e tempo total (custo + RTT).

Cada algoritmo é testado em 30 execuções, utilizando o mesmo mapa de roteadores. O roteador de origem será fixado em 43, enquanto os destinos serão: 119 na primeira simulação e 44 na segunda. Por fim, calcularemos o valor do *TimeoutInterval* (sendo o valor de "SampleRTT" o RTT daquela iteração) para cada round, com $\alpha = 0,125$ e $\beta = 0,25$, e analisaremos o gráfico de performance resultante.

A. Pacote enviado do roteador 43 para o roteador 119

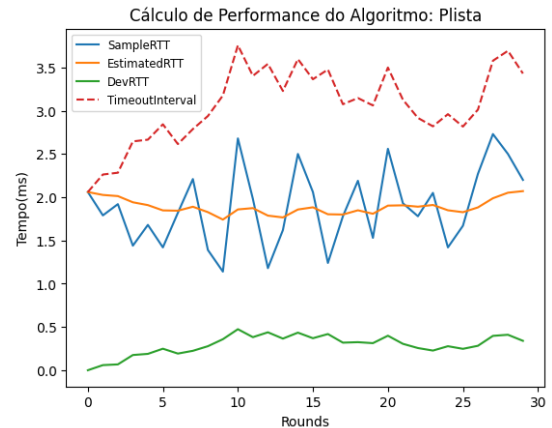


Fig. 1. Performance do algoritmo Plista; Roteador de destino: 119.

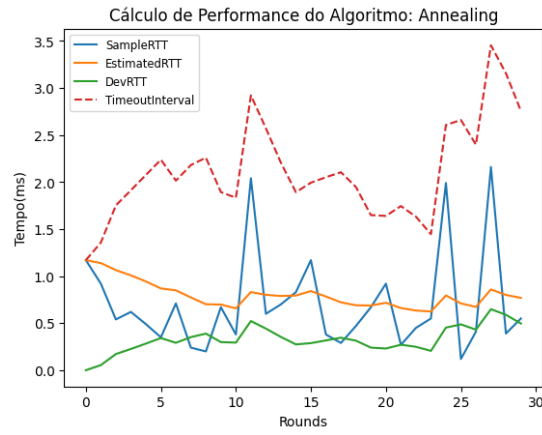


Fig. 2. Performance do algoritmo Annealing; Roteador de destino: 119.

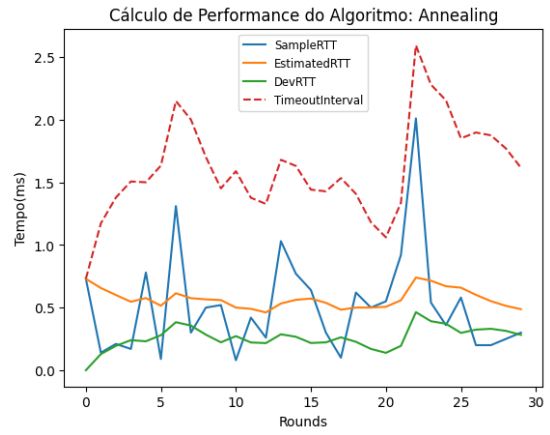


Fig. 5. Performance do algoritmo Annealing; Roteador de destino: 44.

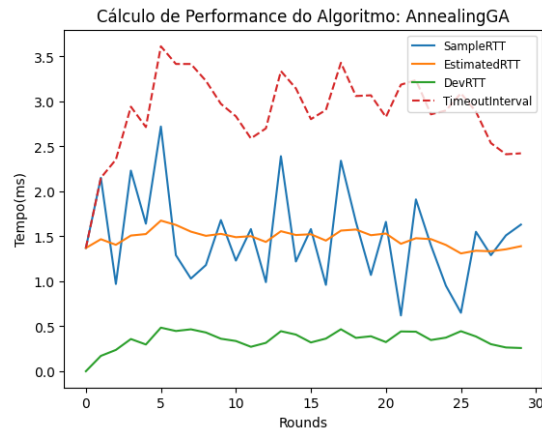


Fig. 3. Performance do algoritmo AnnealingGA; Roteador de destino: 119.

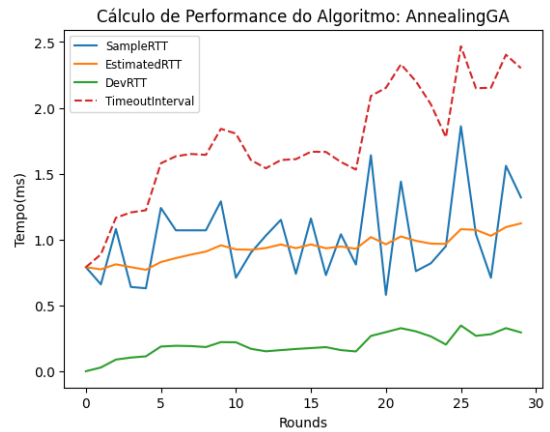


Fig. 6. Performance do algoritmo AnnealingGA; Roteador de destino: 44.

B. Pacote enviado do roteador 43 para o roteador 44

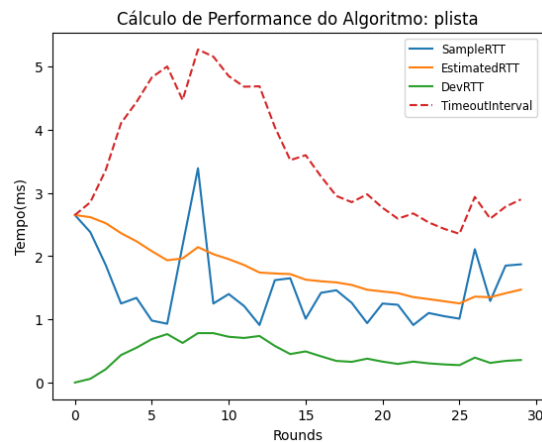


Fig. 4. Performance do algoritmo Plista; Roteador de destino: 44.

V. CONCLUSÃO

Roteador de Destino	Média de TimeoutInterval (ms)		
	Plista	Annealing	AnnealingGA
119	3,0591069557	2,1172198842	2,879382746897
44	3,5368416179	1,6082965450	1,742896484173

O algoritmo de Annealing obteve a menor média de TimeoutInterval em ambas as simulações, enquanto o Plista obteve a maior. No entanto, apesar das melhores médias de tempo, a análise dos gráficos revelou uma elevada taxa de variação para o Annealing. Nesse contexto, o destaque vai para o algoritmo AnnealingGA, que mostrou a menor variação nos gráficos, mantendo médias de timeout próximas ao Annealing. Com médias decentes e baixa variação, o AnnealingGA emerge como o algoritmo mais robusto, sendo a escolha ideal para possível implementação em um Sistema Autônomo (AS).

REFERENCES

- [1] Fonseca, D. S., Wanner, E. F., Marcelino, C. G., Silva, G. P., Jimenez-Fernandez, S., and Salcedo-Sanz, S., "Active GA accelerated by Simulated Annealing to solve SPP in packet networks."