

Simulação de Interferência em Redes *Ad Hoc* comparando MATLAB e C (GSL)

Fagner P. de Araújo e Renato M. de Moraes

Departamento de Sistemas Computacionais

Universidade de Pernambuco

Recife, PE 50720-001, Brasil

Email: {fpa,renato}@dsc.upe.br

Resumo—Este trabalho apresenta um estudo comparativo do comportamento da relação sinal-ruído e interferência (SNIR) em redes *ad hoc* sem fio usando plataformas de simulações distintas. Primeiramente implementamos em C usando GNU Scientific Library (GSL), uma biblioteca robusta e eficiente para grande processamento de dados. Depois, utilizamos MATLAB que mostrou-se não ser tão eficiente, porém apresenta simplicidade no desenvolvimento das simulações, pois apresenta uma tratamento automático de vetores e matrizes. Verificamos que o tempo de simulação em C com GSL é bem menor se comparado ao tempo necessário no MATLAB para os estudos de interferência em redes *ad hoc*. Portanto, a biblioteca GSL é uma excelente alternativa ao MATLAB para simulações não só de redes *ad hoc*, como também outros sistemas de telecomunicações.

Palavras-Chave—Simulação, Interferência, GSL, MATLAB.

Abstract—This work presents a comparative study of the behavior of the signal to noise and interference ratio (SNIR) in wireless *ad hoc* networks using different simulation testbeds. In the first case, we implemented C using GNU Scientific Library (GSL), a robust and efficient library for large data processing. Secondly, we used MATLAB which is not so efficient, however it presents simplicity in the simulation designs since it provides a straightforward treatment for arrays and matrices. We showed that the simulation time in C with GSL is much shorter than the MATLAB simulation time for the interference analysis performed in our study of *ad hoc* networks. Therefore, the GSL library is an excellent alternative to MATLAB for simulations of not only *ad hoc* networks, but also other telecommunications systems.

Keywords— Simulation, Interference, GSL, MATLAB.

I. INTRODUÇÃO

Os efeitos de interferência previamente analisados em Moraes e Araújo [1] confirmam que a comunicação de um nó com seus vizinhos mais próximos é viável quando o número total de nós n cresce ao infinito. O modelo apresentado em [1] pode ser aplicado para redes *ad hoc* sem fio móveis e estáticas. Entretanto, ao utilizarmos simulações para validarmos o modelo, elas se tornam computacionalmente inviáveis quando n é muito grande, devido a enorme quantidade de tempo que é consumida na execução. Por esta razão, é necessário um meio para aumentar a velocidade de simulação. A GNU Scientific Library (GSL) [2] é uma biblioteca numérica para C/C++ que contém funcionalidades embutidas para números randômicos, distribuições estatísticas, matemática de matrizes, números complexos e muito mais, além de ser código aberto e gratuito. Essa biblioteca cumpre a necessidade de uma ferramenta poderosa, facilmente adaptável na análise de telecomunicações. De outro lado, temos o MATLAB [3] com um conteúdo abrangente

possuindo vastas bibliotecas com funções matemáticas pré-definidas provendo um tratamento de vetores e matrizes automático, entretanto essa não é uma ferramenta disponibilizada gratuitamente, além de possuir baixo desempenho computacional.

II. DESEMPENHO E ANÁLISE DA SIMULAÇÃO

Para se ter êxito com algoritmos complexos que possuem grande quantidade de cálculos é necessário uma plataforma de alto desempenho que possa obter resultados num curto prazo. Alguns estudos vêm demonstrando que o desempenho do MATLAB diminui substancialmente quando a ordem de processamento cresce. Almasi e Padua [4] apontam que a principal fraqueza do MATLAB é sua tipagem fraca, pois há um interpretador no ambiente de desenvolvimento que verifica os tipos de dados no tempo de execução, resultando num grande atraso. Outro grande problema no MATLAB são as operações de alto-nível baseada em matrizes tais como operações de memória e análise de limites de arrays (vetores). Isso causa uma grande sobrecarga (overhead) que resulta em baixo desempenho. Menon e Pingali [5] propõem e analisam alternativas que podem ser usadas no lugar do MATLAB ou auxiliá-lo.

III. MODELO E ANÁLISE DE INTERFERÊNCIA

O problema de modelagem que utilizamos neste estudo é de uma rede *ad hoc* sem fio com os nós sendo fixos ou móveis. Nosso modelo consiste de uma rede circular (ou disco) com área normalizada ao valor de uma unidade, e que contém n nós móveis [1]. Assume-se que os nós deslocam-se de acordo com o modelo de mobilidade uniforme que resulta uma distribuição uniforme dos nós na rede [6]. Uma fração θ do total do número de nós n na rede, $n_S = \theta n$, que representa os nós transmissores, é aleatoriamente escolhida, enquanto o restante dos nós, $n_R = (1 - \theta)n$, operam como possíveis nós receptores. Um nó j é capaz de comunicar-se a uma taxa de transmissão de W bits/seg com nó i , no tempo t se a seguinte relação é satisfeita [7], [8], [1]

$$SNIR = \frac{\frac{P_i(t)}{(r_{ij}(t))^\alpha}}{N_0 + \underbrace{\frac{1}{L} \sum_{k \neq i} \frac{P_k(t)}{(r_{kj}(t))^\alpha}}_I} = \frac{\frac{P}{r_{ij}^\alpha}}{N_0 + I} \geq \beta, \quad (1)$$

onde $P_i(t)$ é a potência de transmissão do nó i , $r_{ij}(t)$ é a distância do nó i em relação ao nó j , β é o nível SNIR mínimo necessário para a viabilidade da comunicação, N_0 é potência do ruído, L é o ganho de processamento do sistema, α é o fator de atenuação com a distância e I é a interferência total

recebida no nó j . Assumimos que não há processamento do sistema, i.e., $L = 1$, e que $P_i(t) = P \forall i$.

O objetivo de nossa análise foi saber como a SNIR medida por um receptor comunicando com um vizinho próximo comporta-se à medida que o total de número de nós n na rede cresce ao infinito. Conforme desmostrado em [1], para o caso $\alpha > 2$, a Eq. (1) é reduzida a

$$SNIR_{r'}(n) = \frac{P}{\frac{N_0}{(\theta n \pi)^{\frac{\alpha}{2}}} + \frac{2P}{\alpha-2} \cdot \frac{1}{1 - \frac{1}{\pi^{\frac{\alpha}{2}} (\theta n)^{\frac{\alpha-2}{2}}} f_{\alpha}(r')}}}, \quad (2)$$

onde $f_{\alpha}(r')$ é dada por

$$f_{\alpha}(r') = \int_0^{\pi} \frac{d\gamma}{\left[\sqrt{\frac{1}{\pi} - (r' \sin \gamma)^2} - r' \cos \gamma \right]^{\alpha-2}}. \quad (3)$$

IV. SIMULAÇÃO MONTE-CARLO

Para validar nosso modelo de análise de interferência, desenvolvemos uma simulação Monte-Carlo para o estudo descrito acima. Nossa simulação foi implementada primeiro em C usando GSL, e depois em MATLAB, conforme o algoritmo Monte-Carlo descrito em [1]. No MATLAB, usamos a função `betarnd(2,1)` [3] para obter a distribuição uniforme dos nós, conforme exigido em [1]. O vetor de transmissores escolhidos a partir de uma fração do total dos nós são aleatoriamente selecionados. No MATLAB, esta tarefa pode ser feita usando a função `randsample` [3]. Em GSL, usamos a função `gsl_ran_beta(rd,2,1)` [2] para gerar uma distribuição uniforme (rd é a estrutura de dados `const gsl_rng *rd` que descreve uma instância do gerador de números randômicos dessa biblioteca). Já para escolher os transmissores é usada a função `gsl_ran_choose` [2].

V. RESULTADOS

A Fig. 1 exibe o comportamento de SNIR, Eq. (2), em função de n para $\alpha = 3$ e $\theta = \frac{1}{3}$, e as simulações para o MATLAB e C (GSL), portanto confirmando que os códigos estão corretos e gerando resultados semelhantes. Observa-se que a relação sinal-ruído e interferência decresce à medida que n aumenta, porém tende a uma constante para n grande, indicando que a comunicação entre dois nós vizinhos numa rede é possível desde que $\alpha > 2$ [1].

Por outro lado, a Fig. 2 ilustra que o tempo de execução do MATLAB cresce rapidamente em relação ao tempo de execução em C (GSL) no qual há também um crescimento, porém bem menor (o que resulta a curva quase constante na escala apresentada). Isso demonstra um alto desempenho do código C em relação ao código MATLAB. Observa-se, assim, uma enorme diferença entre uma simulação MATLAB, que dura horas, e uma em C que dura apenas alguns minutos.

VI. CONCLUSÕES

Um tempo de execução baixo é essencial numa simulação em redes *ad hoc*, dada a importante economia de tempo que se pode ter num estudo de desempenho para muitos nós na rede. Por outro lado, é bastante comum em simulações numéricas aplicadas haver uma grande quantidade de processamento de dados, gerando um baixo desempenho no código MATLAB. Nossos resultados mostram que a biblioteca em C GSL pode

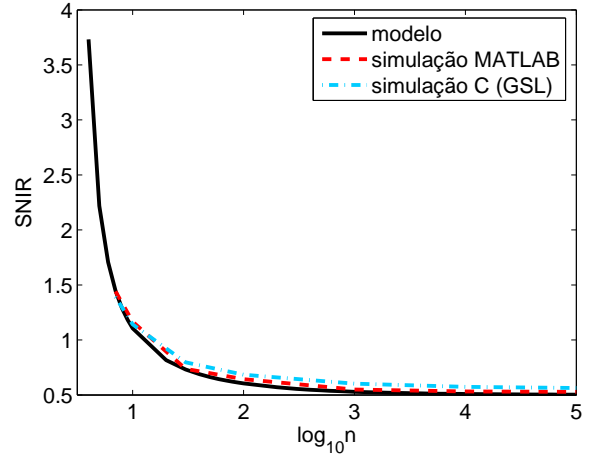


Fig. 1. Curvas SNIR em função de n para $\alpha = 3$, $\theta = \frac{1}{3}$, $P = 1$ e $N_0 = 0$ quando o nó receptor está localizado no centro da rede para o modelo e as simulações MATLAB e C. Na legenda, *modelo* é usado para Eq. (2), enquanto *simulação* é a simulação Monte-Carlo para Eq. (1).

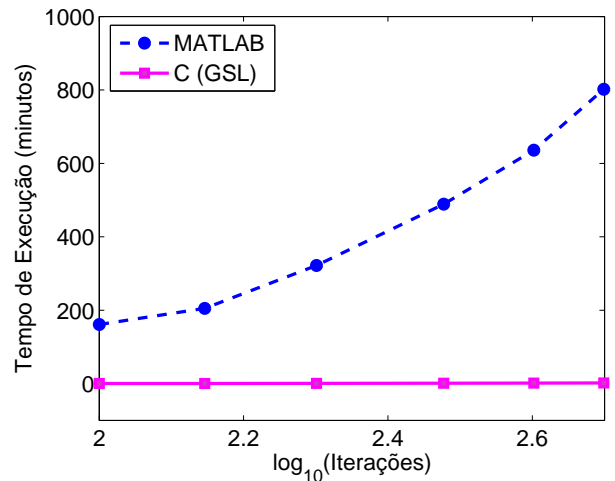


Fig. 2. Curvas do tempo de execução em função do número de iterações para as simulações MATLAB e C (GSL).

muito bem ser utilizada nesses tipos de simulações, reduzindo drasticamente o tempo de execução em relação ao MATLAB, e pode também beneficiar outros estudos de sistemas de telecomunicações que necessitem de uma ferramenta veloz de simulação.

REFERÊNCIAS

- [1] R. M. de Moraes and F. P. de Araújo, "Interference Analysis in Wireless Ad Hoc Networks", in *Proc. of IEEE International Conference on Telecommunications (ICT)*, Penang, Malaysia, May 2007.
- [2] GNU Scientific Library (GSL), <http://www.gnu.org/software/gsl/>
- [3] Mathworks Inc. homepage. <http://www.mathworks.com>
- [4] G. Almasi and D. A. Padua. "MAJIC: Compiling MATLAB for Speed and Responsiveness", in *Proc. of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, June 2002.
- [5] V. Menon and K. Pingali. "High-level semantic optimization of numerical codes", in *Proc. of ACM Conference on Supercomputing*, ACM SIGARCH, June 1999.
- [6] N. Bansal and Z. Liu, "Capacity, delay and mobility in wireless ad-hoc networks," in *Proc. of IEEE Infocom*, San Francisco, California, March 2003.
- [7] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, March 2000.
- [8] M. Grossglauser and D. Tse, "Mobility increases the capacity of wireless ad-hoc networks," in *Proc. of IEEE Infocom*, Anchorage, Alaska, March 2001.