

Modelo Paralelo de Traçado de Raios 3D para Predição de Propagação de Ondas Eletromagnéticas

André Mendes Cavalcante, Marco José de Sousa, João Crisóstomo Weyl Albuquerque Costa,
Carlos Renato Lisboa Francês e Gervásio P. dos Santos Cavalcante
Universidade Federal do Pará, Laboratório de Eletromagnetismo Aplicado, Belém, PA, 66.075-900, Brasil

Resumo— Neste artigo um modelo computacional paralelo de traçado de raios 3D aplicado na predição de rádio-propagação é apresentado. Esta abordagem considera que as principais tarefas de uma técnica de traçado de raios 3D podem ser avaliadas de forma independente e/ou paralela. A distribuição da carga de trabalho entre os nós participantes do computador paralelo (*cluster*) é realizada através de uma designação aleatória dos raios iniciais e dos pontos de recepção para os mesmos. Simulações foram realizadas para validar e avaliar o desempenho do modelo proposto.

Palavras-chaves— Computação paralela, *cluster* de PC's, traçado de raios 3D.

I. INTRODUÇÃO

Nos últimos anos, o expressivo crescimento das comunicações sem-fio e a necessidade de provisão de novos serviços móveis vêm requerendo o uso ou o desenvolvimento de metodologias de avaliação rápidas e precisas para modelos de canais de rádio que consideram a complexidade do ambiente, para possibilitar um eficiente planejamento desses sistemas de comunicação. A predição eficiente do canal de propagação imprime atualmente um importante papel na determinação de vários parâmetros da rede, tais como, cobertura, taxa de dados transmitidos, localização ótima dos rádio-transmissores, diagramas de irradiação das antenas, etc. Neste contexto, modelos de predição baseados em técnicas de traçado de raios vêm se mostrando como soluções promissoras, principalmente nos modernos sistemas de rádio [1]-[5]. Apesar da abordagem por traçado de raios ser muito útil no projeto, análise e desenvolvimento de sistemas sem-fio, é reconhecido que tais modelos são computacionalmente dispendiosos e requerem consideráveis recursos (memória e processamento) para obterem resultados de predição com razoáveis níveis de precisão [2],[4].

Várias abordagens têm sido propostas para reduzir o tempo computacional gasto por modelos de predição baseados em traçado de raios. As principais estão relacionadas ao pré-processamento da base de dados do cenário [2], técnicas de filtros de dados [5] e métodos de aproximação de procedimentos [6]. Todas essas abordagens têm um compromisso comum: “negociar” precisão de predição com tempo de processamento. Uma forma natural de melhorar este compromisso é empregar técnicas de processamento paralelo e/ou

distribuído para acelerar o tempo de execução do modelo, enquanto se mantém a precisão intacta [4]. A viabilização física dessa abordagem é particularmente atrativa com o uso de *clusters* de PC's, devido tais configurações de computadores já estarem acessíveis a vários grupos de pesquisa.

Recentemente, algumas estratégias computacionais com uso de processamento paralelo têm sido propostas para se reduzir o tempo computacional requerido sem afetar o requisito de precisão de predição [4],[7]. Em [4], a estratégia de paralelização proposta é muito complexa e de difícil implementação. Esta abordagem foi inicialmente aplicada em um modelo de traçado de raios bi-dimensional (2D), sendo criado posteriormente, um protótipo para aplicação em um tipo de modelo de traçado de raios 3D que têm algumas restrições no mecanismo de difração (modelo *Vertical Plane* - [2]). Ambas as versões são muito dependentes da implementação do algoritmo de traçado de raios adotado. Em [7], a estratégia de paralelização foi aplicada em um modelo de traçado de raios totalmente 3D. Entretanto, tal abordagem foi apenas implementada no estágio de processamento de raios. O modelo proposto neste artigo é esquematizado para ser aplicado no processo inteiro, sendo de fácil implementação computacional e podendo ser aplicado facilmente em modelos de traçado de raios tridimensionais no sentido pleno (*Full 3D*) sem qualquer restrição no mecanismo de difração (se desejado). Esta nova abordagem permite reduzir ou até mesmo eliminar várias restrições anteriormente impostas em modelos de traçado de raios por razões práticas (custo computacional alto), favorecendo para o aumento da precisão e a possibilidade de incorporação de novos mecanismos de propagação, como espalhamento difuso [4]-[8] e propagação em ambientes com vegetação [9]. Adicionalmente, isto permite analisar estruturas (cenários) mais complexos.

O artigo está organizado da seguinte forma: A Seção II discute os algoritmos de traçado de raios usados na rádio-propagação. A Seção III descreve o modelo computacional paralelo proposto. A Seção IV apresenta algumas simulações para validar e avaliar o modelo de traçado de raios paralelo. As conclusões são feitas na Seção V.

II. TÉCNICAS DE TRAÇADO DE RAIOS

Existem basicamente duas abordagens para se implementar uma técnica de traçado de raios aplicada na rádio-propagação. A primeira abordagem é baseada na Teoria das Imagens (*IT - Image Theory*) [10]. Esta abordagem é fortemente dependente da dimensão (tamanho) e da complexidade do ambiente, sendo mais usada em ambientes pequenos e simples envolvendo

André Mendes Cavalcante, amc@ufpa.br, Marco José de Sousa, marcojs@ufpa.br, João Crisóstomo Weyl Albuquerque Costa, jweyl@ufpa.br, Carlos Renato Lisboa Francês, rfrances@ufpa.br, Gervásio P. dos Santos Cavalcante, gervasio@ufpa.br. Tel/Fax: +55 91 3183-1634.

Este artigo foi parcialmente pelo CNPq e projeto Ericsson (UFA01).

apenas o mecanismo da reflexão [10]. Apesar de ser possível na IT a incorporação de algoritmos de busca por pontos de transmissão e difração, sua limitação intrínseca anteriormente mencionada permanece. Por outro lado, o método SBR (*Shoot-and-Bouncing-Rays*), referida algumas vezes como “Força-Bruta”, é a abordagem de traçado de raios mais adequada para grandes e complexos ambientes [10], podendo envolver qualquer combinação das interações básicas (reflexão, transmissão e difração). A limitação intrínseca dessa abordagem é o alto tempo de processamento gasto para avaliar todos os percursos de raios.

Independente da abordagem considerada para o traçado de raios, o tempo computacional gasto na execução do programa pode alcançar valores elevadíssimos [4], principalmente para ambientes enquadrados nas modernas configurações de sistemas de rádio (por exemplo, com múltiplas fontes, com interações *outdoor-indoor*, com vegetação, com espalhadores, etc). O primeiro esforço no sentido de se reduzir esse tempo é otimizar os procedimentos de testes de sombreado (testes de interseção raio-objeto). Existem diversas abordagens relacionadas a esta otimização, tais como a BSP (*Binary Space Partition*), a SVP (*Space Volumetric Partition*), a do algoritmo *Angular Z-buffer*, a dos volumes envolventes (BV - *Bounding volumes*) entre outros [10]. Adicionalmente, esforços têm sido feitos para a paralelização do código do programa que implementa os algoritmos de traçado de raios [4],[7]. Na IT, esta paralelização do código não é trivial, pois a estrutura de dados adotada (árvore de imagens) é totalmente concatenada, dificultando a divisão das tarefas e o balanceamento de carga entre os processadores. A técnica SBR por sua concepção já é intrinsecamente paralelizável, pois os raios lançados pela antena transmissora são independentes. Podendo ser aplicável, portanto, diretamente no contexto de programação paralela. Pelo exposto, o modelo paralelo desenvolvido foi baseado na técnica SBR.

III. MODELO PARALELO

O modelo paralelo proposto para algoritmos SBR foi esquematizado em estágios conforme ilustrado na Fig.1. A idéia básica deste modelo é que após uma fase de pré-processamento de dados, a carga de processamento total pode ser dividida entre os nós que compõem o computador paralelo (*cluster*) através de uma distribuição aleatória entre os mesmos, dos raios iniciais a serem lançados e dos pontos de recepção a serem avaliados. A eficiência dessa abordagem é obtida através da independência das entidades envolvidas (raios e pontos de recepção) e da forma de distribuição (aleatória). A abordagem aleatória tende a ser mais eficiente no aspecto de balanceamento de carga quanto maior for o número total de raios emitidos e de pontos de recepção considerados, exatamente o caso quando mais se justifica o uso da computação paralela [7]. Através desta estratégia, a carga de processamento de um *cluster* homogêneo é balanceado através da distribuição de um número igual de raios e pontos de recepção (aleatoriamente escolhidos) para cada nó. Para um *cluster* heterogêneo, o número de raios e de pontos de recepção de cada nó devem ser proporcionais a sua capacidade de processamento.

Evidentemente, descobrir a capacidade de processamento de computadores é algo que se pode fazer antecipadamente, sendo até possível estimá-la com base em características de *hardware* e *software*.

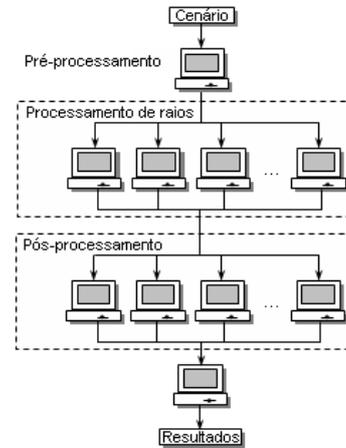


Fig. 1. Modelo de paralelização do algoritmo SBR 3D.

No contexto da computação paralela e distribuída, o modelo proposto pode ser enquadrado no paradigma SPMD (*Single Program Multiple Data*) [11], pois dado um determinado cenário, cada nó executa o mesmo programa SBR sobre dados (arquivos de entrada) distintos. A estratégia de comunicação inicial entre os nós para fornecer os arquivos de entrada através da rede poderia ser implementada, por exemplo, com o uso da biblioteca padrão de comunicação por passagem de mensagens MPI (*Message Passing Interface*) [12]. Entretanto, uma estratégia mais simples foi implementada, onde arquivos de entrada personalizados para cada nó são previamente criados e distribuídos na rede através de um sistema de arquivos de rede, como o NFS (*Network File System*), usado em sistemas UNIX [13]. Após a geração e carregamento dos arquivos de entrada (estágio de pré-processamento), cada nó executará o processamento dos raios definidos para si. Quando a simulação de raios estiver concluída, cada processo isolado (nó) pode enviar seus resultados através da rede usando MPI, ou disponibilizando-os na forma de arquivos locais compartilhados através do NFS (neste artigo a estratégia usando NFS foi a adotada). O resultado do processamento de raios é um relatório de todos os raios que alcançaram os pontos de recepção considerados no cenário. A recepção, organização e avaliação dos resultados providos por cada nó consiste do estágio de pós-processamento. A seguir, cada estágio do modelo paralelo será brevemente detalhado, sendo adotada a terminologia de que se o programa for executado de forma serial, ele será chamado de “modo serial”, em caso contrário, ele será chamado de “modo *cluster*”:

- 1) **Estágio de Pré-processamento:** Definição e criação de arquivos de entrada específicos para cada nó. Após esta etapa, dois arquivos para cada nó, denominados de “arquivo de *setup*” e “arquivo de raios”, são definidos. A função desses arquivos será explicada nos procedimentos subsequentes listados abaixo:

- a) **Carregamento do arquivo de setup:** Leitura de um arquivo de *setup* pré-determinado. O arquivo de *setup* provê informações sobre os parâmetros de simulação (potência transmitida, tipo das antenas, localização dos pontos de recepção, número máximo de interações, etc) e dos nomes dos arquivos de cenário (base de dados dos prédios) e de raios que devem ser carregados.
- b) **Carregamento do arquivo de raios:** Leitura do arquivo de raios definido no arquivo de *setup*. O arquivo de raios contém os diretores dos raios iniciais que foram aleatoriamente designados para um determinado nó. No modo serial, todos os diretores são designados para um único nó.
- c) **Geração de arquivos de tabelas:** Geração de um arquivo de tabela específico para cada nó. Os arquivos de tabela contém a lista dos pontos de recepção que estarão sob responsabilidade de um determinado nó. Esta designação de pontos é feita de forma aleatória e de tal maneira que cada arquivo contenha aproximadamente o mesmo número de pontos de recepção. Estes arquivos serão úteis na paralelização do estágio de pós-processamento (detalhes adicionais serão dados na seção específica). Como proposta de implementação no modo *cluster*, elege-se um determinado nó para ser o responsável por esta geração de arquivos (incluindo o dele próprio). No modo serial, todos os pontos de recepção são designados para um único nó.
- 2) **Estágio de Processamento de raios:** Responsável em executar o algoritmo SBR propriamente dito. Cada nó executa este estágio apenas para os raios definidos em seu arquivo de raios.
- 3) **Estágio Pós-processamento:** Consiste basicamente no salvamento de arquivos de percurso e a avaliação dos resultados de predição. Três etapas são definidas neste estágio, conforme listadas abaixo:
- a) **Salvamento de arquivos de percurso:** Após a etapa de processamento de raios, um arquivo de percurso para cada ponto de recepção é criado. Cada arquivo de percurso contém todos os percursos de raios que foram considerados como recebidos por um determinado ponto de recepção. No modo *cluster*, cada nó participante gera seus próprios arquivos, requerendo assim, de um procedimento adicional para montagem de um único arquivo por ponto de recepção. Tal procedimento foi chamado de “Agrupamento de arquivos” e será descrito adiante.
- b) **Sincronização Salvamento/Agrupamento:** Para iniciar a etapa de agrupamento de arquivos é necessário que todos os nós já tenham concluído o procedimento de salvamento. Para cada nó indicar para os outros que sua etapa de salvamento está concluída, um arquivo de *status* é criado. Tal arquivo não contém qualquer informação e é usado somente para verificação do *status* do procedimento de salvamento (por exemplo, se o arquivo de *status* de um determinado nó foi criado, então isto indica que o mesmo está apto a iniciar o procedimento de agrupamento de arquivos, em caso contrário, não). Quando todos os nós estiverem prontos, eles começam a executar a etapa de agrupamento.
- c) **Agrupamento de arquivos:** Cada nó é responsável pelo agrupamento dos arquivos de percurso de todos os nós apenas para os pontos de recepção definidos no seu arquivo de tabela. É importante observar que esta etapa se faz necessária apenas no modo *cluster*.
- d) **Sincronização Agrupamento/Avaliação:** Para executar a etapa de avaliação de resultados é necessário que todos os nós já tenham concluído o procedimento de agrupamento de arquivos. Da mesma forma que foi realizado para a etapa de sincronização anterior, cada nó cria um arquivo de *status* para indicar para os outros nós que sua etapa de agrupamento está concluída. Quando todos os nós estiverem prontos, eles executam a etapa de Avaliação de resultados.
- e) **Avaliação de resultados:** Esta etapa é responsável pela avaliação dos resultados de predição (campo elétrico, potência recebida, ângulo de chegada de cada raio, etc) e geração de arquivos de saída com tais informações para cada ponto de recepção. No modo *cluster*, cada nó é responsável por avaliar resultados apenas para os pontos de recepção definidos em seu arquivo de tabela.

IV. RESULTADOS

Para validar o modelo paralelo proposto, foi considerado como estudo de caso um cenário *outdoor* na cidade de Ottawa (Canadá). O cenário adotado está dentro da área mostrada na Fig.2. O transmissor foi localizado na posição rotulada como “Tx” a uma altura de 8.5 m e os pontos de recepção foram posicionados ao longo da rua Laurier a uma altura de 3.65 m. Todas as antenas foram consideradas verticalmente polarizadas. Como algoritmo de traçado de raios foi considerado o modelo clássico SBR 3D juntamente com a Teoria Uniforme da Difração (UTD), ambos brevemente descritos em [10]. Os campos foram calculados para uma frequência de 910 MHz com percursos envolvendo até 4 reflexões e 2 difrações. Os efeitos de percursos que difratam sobre os topos dos prédios foram desprezados devido as antenas transmissora e receptoras estarem bem abaixo das alturas dos prédios (como relatado em [14]). Nesses casos, tais percursos apresentam níveis de potência muito baixos comparados com os outros percursos que propagam entre os prédios [9]. Apesar desta suposição, o modelo SBR 3D foi ainda executado. Os dados geométricos dos prédios para os cálculos foram extraídos diretamente dos mapas apresentados em [15]. Seguindo a sugestão apresentada em [15], foi adotado um valor para a permissividade relativa de todas as paredes dos prédios de 6, e uma condutividade de 0.5 S/m. Uma permissividade relativa de 15 e uma condutividade de 0.05 S/m foram usadas para o solo.

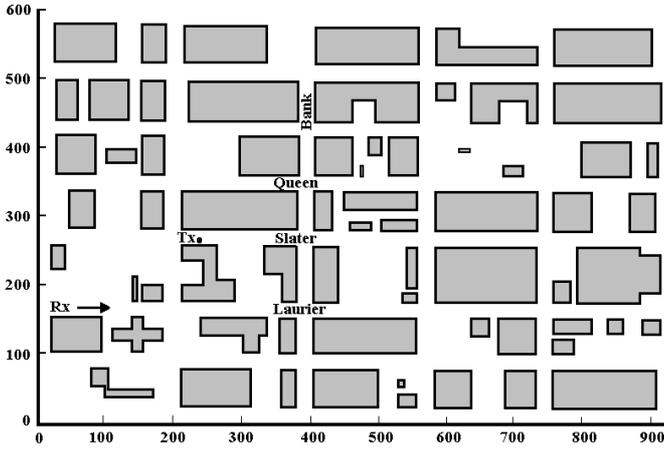


Fig. 2. Mapa da cidade de Ottawa

As simulações foram executadas em um *cluster* consistindo de quatro (04) nós equipados com processadores Pentium IV HT 3.2 GHz e memórias principais de 1.0 GB. Todos os códigos computacionais foram implementados usando a linguagem de programação orientada à objetos C++. O compilador usado foi o g++ versão 3.3.5 20050117 (*pre-release*) sob um sistema operacional baseado em Linux. O paradigma mestre/escravo (*master/slave*) foi usado para implementar o sistema de arquivos de rede Unix (NFS - *Network File System*). Arquivos de entrada (arquivo de *setup* e arquivo de raios) personalizados para cada processo (nó) foram previamente criados. Todos esses arquivos juntamente com o arquivo de cenário foram distribuídos na rede através do NFS. Para avaliar o desempenho do modelo paralelo proposto, foi usado principalmente métricas como *speedup*, taxa de expansão de carga de trabalho (*workload expansion ratio*) e taxa de utilização de recursos (*resource utilization ratio*). Considerando que T_{seq} é o melhor tempo de execução alcançado quando apenas uma máquina for usada (modo serial), t_i é o tempo de execução para o i -ésimo nó quando uma configuração de *cluster* de n -nós for usada, T_{max} e T_{avg} são os tempos de execução máximo e médio, entre os n nós, enquanto T_{sum} é a somatória dos tempos de execução de todos os nós, então, $T_{max} = \max_{i=1}^n t_i$, $T_{sum} = \sum_{i=1}^n t_i$, e $T_{avg} = T_{sum}/n$. O *speedup* S_n , a expansão da carga de trabalho W_n , e a utilização de recursos U_n podem ser calculadas respectivamente por $S_n = T_{seq}/T_{max}$, $W_n = T_{sum}/T_{seq}$, e $U_n = T_{avg}/T_{max}$. Para medir mais precisamente a escalabilidade do modelo proposto, foi empregado também a métrica de eficiência, que é definida como $E_n = S_n/n$ [4].

Conforme mostrado na Fig.3, os fatores de *speedup* obtidos para o caso com 655362 raios lançados pela fonte (i.e., uma separação angular média entre raios vizinhos no espaço 3D $\bar{\alpha} \approx 0.27^\circ$) apresentaram um comportamento acima do caso linear. Esta situação é referida na literatura como “*speedup* super-linear”.

A Tabela I mostra o número de raios processados por cada nó em diversas configurações de *cluster* para 655362 raios lançados pela fonte, apresentando uma diferença máxima de raios processados relacionada com o valor médio (n° de raios

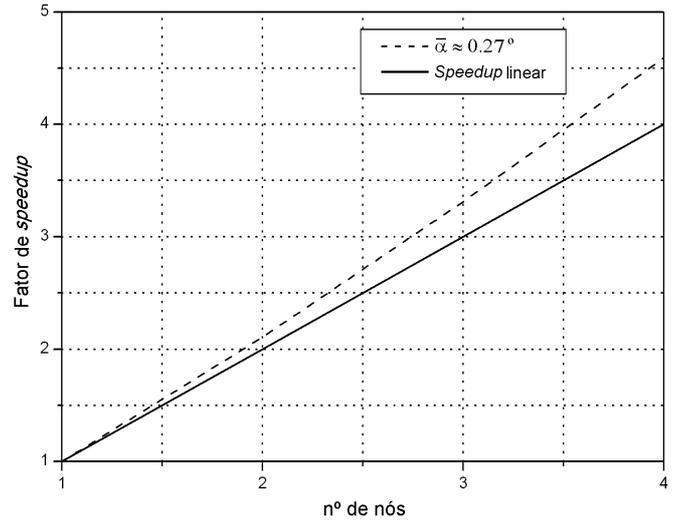


Fig. 3. Fatores de *speed-up* para 655362 raios lançados pela fonte

processados no modo serial/ n° de nós) próximo de 1.12%. Adicionalmente, na Tabela II são mostrados os tempos médios de processamento gastos por cada nó. Apesar de cada nó processar um número diferente de raios, o maior tempo de processamento obtido para cada configuração de *cluster* foi sempre abaixo do valor do tempo médio linear (tempo de execução serial / n° de nós).

TABELA I
RAIOS PROCESSADOS

Nó	Serial ($n = 1$)	Cluster ($n = 2$)	Cluster ($n = 3$)	Cluster ($n = 4$)
0	124310173	62325589	41107549	30881464
1	-	61984584	41899821	31214554
2	-	-	41302803	31154001
3	-	-	-	31060154

TABELA II
TEMPO DE PROCESSAMENTO

Nó	Serial ($n = 1$)	Cluster ($n = 2$)	Cluster ($n = 3$)	Cluster ($n = 4$)
0	11226.838 s	5304.846 s	3380.315 s	2439.285 s
1	-	5329.656 s	3390.986 s	2431.279 s
2	-	-	3379.247 s	2445.582 s
3	-	-	-	2432.800 s

A Tabela III mostra as métricas de avaliação de desempenho consideradas para o modelo paralelo proposto. Conforme mostrado na mesma, as taxas de expansão de carga de trabalho obtidas em todos os casos foram sempre abaixo do caso ideal ($W_n = 1.0$), decrescendo seu valor com o aumento do número de nós. Isto caracteriza que é possível esperar uma boa escalabilidade do modelo. As taxas de utilização de recursos obtidas são muito próximas da taxa de utilização ideal, indicando que todos os nós gastam pouco tempo em estado ocioso (*idle status*). A eficiência do modelo melhorou com o aumento do número de nós, apresentando valores acima da eficiência ideal em todas as configurações de *cluster* consideradas.

TABELA III
MÉTRICAS DE AVALIAÇÃO DE DESEMPENHO

Cluster de n -nós	S_n	W_n	U_n	E_n [%]
1	1.000	1.000	1.000	100.0
2	2.106	0.947	0.998	105.3
3	3.310	0.904	0.998	110.3
4	4.591	0.868	0.996	114.7

Como a única diferença entre os processos de cada nó é o volume de dados dos arquivos de entrada, estes resultados da avaliação de desempenho mostram que o tempo de processamento das tarefas executadas por cada nó apresentam uma taxa de redução acima do linear se comparada com a taxa de redução do volume de dados manipulados, principalmente em procedimentos relacionados a varredura das estruturas de dados utilizadas e alocação de memória. Isto implica que se o algoritmo SBR for particionado (i.e., distribuição dos raios iniciais e dos pontos de recepção em diversos arquivos de entrada) e for estruturado para ser executado de forma serial, ainda assim será mais atrativo que sendo executado de forma serial sem particionamento algum. A escalabilidade do modelo é naturalmente garantida dado a independência dos raios iniciais e pontos de recepção. Entretanto, a super-eficiência apresentada pelo modelo apenas será mantida enquanto o ganho de *speedup* obtido no estágio de processamento de raios de cada nó em uma determinada configuração de *cluster*, for suficientemente grande e compensar as perdas de *speedup* geradas nos outros procedimentos (principalmente no procedimento de agrupamento de arquivos). Esta condição pode ser obtida aumentando-se a complexidade da base de dados do cenário e/ou aumentando a resolução dos raios iniciais a serem lançados. Além de aumentar a eficiência do modelo, o aumento dessas entidades (complexidade do cenário e resolução dos raios) tornam o algoritmo SBR mais preciso.

Para dar uma indicação de intensidade de campo, a perda de propagação (usando um modelo SBR 3D clássico) ao longo da rua Laurier é comparado com medidas relatadas em [14]. Os resultados são mostrados na Fig.4.

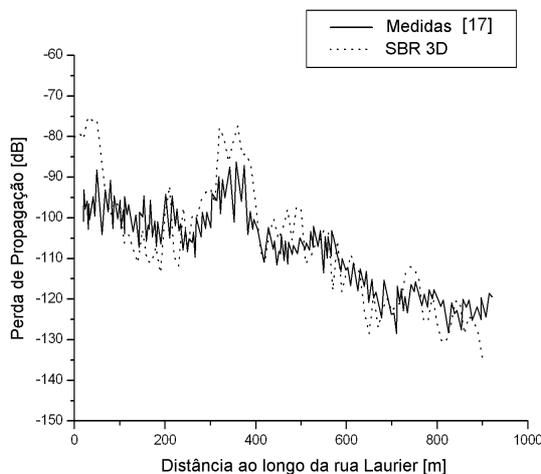


Fig. 4. Perda de propagação ao longo da rua Laurier.

A concordância é boa considerando a simplicidade da base de dados do cenário e a falta de informações sobre as propriedades elétricas dos materiais que compõem as edificações. É interessante comentar uma das mais notáveis diferenças com as medidas. O erro no início da rua Laurier é inesperado dado que existe uma configuração quase de visada (LOS) com o transmissor. Uma explicação é que provavelmente existam algumas árvores ou outras obstruções que espalham o sinal nesta área.

V. CONCLUSÕES

Neste artigo foi apresentado um modelo computacional paralelo para técnicas de traçado de raios 3D aplicado na previsão de rádio-propagação. Esta abordagem é baseada na independência das tarefas em um algoritmo SBR para eficientemente distribuir a carga de trabalho total (através de uma distribuição aleatória de raios iniciais e de pontos de recepção) entre os nós do computador paralelo (*cluster*). Diversos aspectos relacionados com a implementação prática do modelo paralelo foram descritos. Alguns resultados de simulações foram apresentados no sentido de avaliar o desempenho do modelo proposto.

REFERÊNCIAS

- [1] Richter, J., Al-Nuaimi, M.O.; Ivrisstmitz, L.P., "Optimization of radio coverage in urban microcells using a UTD based ray-tracing model," *2004 Antennas and Propagation, IEE Proceedings*, vol. 151, Issue: 3, 21 June 2004, pp. 187 - 192.
- [2] Z. Chen, A. Delis, H.L. Bertoni, "Building footprint simplification techniques and their effects on radio propagation predictions," *Comput. J.*, 47 (1) (January 2004) 103-133.
- [3] Kipp, R.A.; Miller, M.C., "Shooting-and-bouncing ray method for 3D indoor wireless propagation in WLAN applications," *Antennas and Propagation Society Symposium, 2004*, vol. 2, 20-25 June 2004, pp 1639 - 1642.
- [4] Z. Chen, A. Delis, H.L. Bertoni, "Radio-wave propagation predictions using ray-tracing techniques on a network of workstations (NOW)," *J. Parallel Distrib. Comput.*, 64 (2004) 1127-1156.
- [5] T. Kurner, A. Méier, "Prediction of outdoor and outdoor-to-indoor coverage in urban areas at 1.8 GHz," *IEEE J. Selected Areas Comm.*, 20 (3) (April 2002) 496-506.
- [6] Z. Chen, H.L. Bertoni, A. Delis, "Progressive and approximate techniques in ray-tracing based radio wave propagation prediction models," *IEEE Trans. Antennas Propagation*, 52 (1) (2004) 240-251.
- [7] Cavalcante, A. M., Sousa, M. J., Sales Jr, C. S., Costa, J. C. W. A., Francês, C. R. L., and Cavalcante, G. P. S., "Computational parallelization strategy applied in full 3D ray-tracing wireless channel modeling," *International Microwave and Optoelectronic Conference - IMOC'2005*, Brasília - DF, Brazil, July 2005.
- [8] Degli-Esposti, V.; Guiducci, D.; de'Marsi, A.; Azzi, P.; Fuschini, F.; "An advanced field prediction model including diffuse scattering," *IEEE Transactions on Antennas and Propagation*, vol. 52, Issue: 7, July 2004, pp. 1717 - 1728.
- [9] Bertoni, H. L., *Radio propagation for modern wireless systems*, Prentice-Hall - Wireless Communications Series, 2000.
- [10] M. F. Catedra and J. P. Arriaga, *Cell Planning for Wireless Communications*, Artech House - Mobile Communications Series, 1999.
- [11] Y. Foster, K. Kennedy, J. Dongarra, G. Fox, *Sourcebook of Parallel Computing*, Morgan Kaufman Pub, 2002.
- [12] W. Gropp, E. Lusk, A. Skjellum, R. Thakur, *Using MPI : Portable Parallel Programming with the Message Passing Interface (Scientific and Engineering Computation)*, 2nd Edition, MIT Press, 1999.
- [13] Tanenbaum, A.S., *Computer Networks*, 4th Ed., Prentice Hall, 2002.
- [14] J. H. Whitteker, "Measurements of path loss at 910 MHz for proposed microcell urban mobile systems," *IEEE Trans. Veh. Technol.*, vol. 37, no. 3, pp. 125-129, Aug. 1988.
- [15] S.Y. Tan and H.S. Tan, "Propagation model for microcellular communications applied to path loss measurements in Ottawa city streets," *IEEE Trans. Veh. Technol.*, vol. 44, no. 2, pp. 313-317, Aug. 1995.