

# Uma Nova Abordagem Paralela para Técnicas de Traçado de Raios 3D Aplicadas na Predição de Radiopropagação

André Mendes Cavalcante, Marco José de Sousa, João Crisóstomo Weyl Albuquerque Costa, Carlos Renato Lisboa Francês e Gervásio Protásio dos Santos Cavalcante

**Resumo:**— Neste artigo, um modelo computacional paralelo de traçado de raios 3D aplicado na predição de radiopropagação é apresentado. A 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 da arquitetura paralela (*cluster* de PC's) é realizada por meio de uma designação aleatória dos raios iniciais e dos pontos de recepção para os mesmos. Simulações são realizadas para validar e avaliar o desempenho do modelo proposto. Os resultados apresentados mostram que o desempenho e a escalabilidade do modelo são mais efetivos para casos complexos e com ostensivo processamento de raios. Essas características favorecem a melhoria da precisão de predição pelo aumento da densidade de raios lançados e pela possibilidade de incorporação de novos mecanismos de propagação.

**Palavras chave:**— Processamento paralelo, Radiopropagação, Traçado de raios.

## I. INTRODUÇÃO

Nos últimos anos, o expressivo crescimento das comunicações sem fio e a necessidade de novos serviços móveis vêm requerendo o desenvolvimento de novas metodologias de avaliação rápidas e precisas para modelos de canais de rádio. A predição eficiente do canal de propagação imprime um importante papel na determinação de vários parâmetros de rede, tais como, cobertura, taxa de dados transmitidos, localização ótima dos rádios-transmissores, diagramas de irradiação das antenas, etc. Nesse 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 de a 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, requerendo consideráveis recursos para obtenção de predições suficientemente precisas [2],[4].

Várias abordagens têm sido propostas para se 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 filtragens de dados [5] e métodos de simplificação de procedimentos [6]. Todas essas abordagens têm um compromisso comum: “negociar” precisão de predição com tempo de processamento. Uma forma natural de melhorar esse compromisso é empregar técnicas de processamento paralelo e/ou distribuído para reduzir 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

Algumas estratégias computacionais aplicadas em técnicas de traçado de raios, com o uso de processamento paralelo, têm sido propostas [4],[7]. Em [4], a estratégia de paralelização faz uso de bibliotecas de passagem de mensagens e de paradigmas de computação, como o de mestre/escravo e o de fases paralelas (*parallel phases*), para distribuir a carga de trabalho entre os nós de um *cluster* de estações de trabalho (COW – *Cluster of Workstations*). Experimentos realizados na mesma referência mostraram que essa abordagem apresenta um *speedup* próximo do linear para a maioria dos casos simulados. Entretanto, ela utiliza sempre um nó adicional (denominado de mestre) para fazer a gerência do processo de paralelização inteiro. Além disso, a complexidade estrutural da mesma faz com que sua implementação computacional não seja trivial. Essa abordagem apresenta também uma forte dependência com os algoritmos de traçado de raios adotados. Em [7], a estratégia de paralelização proposta foi aplicada em um modelo de traçado de raios totalmente 3D. Entretanto, tal abordagem foi avaliada somente no estágio de processamento de raios. Simulações realizadas com essa abordagem em um ambiente *outdoor* simples exibiram um *speedup* próximo do linear para casos com uma alta densidade de raios lançados.

O modelo proposto neste artigo é a extensão da proposta apresentada pelos autores em [7], sendo aplicada em todos os estágios do processo. A nova abordagem apresenta as seguintes vantagens: a) fácil implementação computacional, sendo baseada apenas no sistema de arquivos de rede da arquitetura paralela adotada, não necessitando de bibliotecas de passagem de mensagens e/ou de paradigmas da computação paralela mais complexos; b) pode ser facilmente adaptada para a maioria dos algoritmos baseados em

Este trabalho foi parcialmente financiado pelo CNPq e pela Capes (bolsa PDEE).

Os autores são do Laboratório de Eletromagnetismo Aplicado (LEA), Universidade Federal do Pará (UFPA), CEP: 60075-900, Belém-PA, Brasil (e-mails: amc@ufpa.br, marcojs@ufpa.br, jweyl@ufpa.br, rfrances@ufpa.br e gervasio@ufpa.br).

lançamento de raios, 2D ou 3D, e para ambientes *outdoor* ou *indoor*; c) permite reduzir ou até mesmo eliminar várias restrições impostas em outros modelos de traçado de raios devido ao custo computacional; d) permite melhorar a precisão de predição por meio do uso de técnicas totalmente 3D e da incorporação de novos mecanismos de propagação, como espalhamento difuso [4],[8] e propagação em ambientes com vegetação [9]; e e) possibilita a análise de estruturas mais complexas (cenários com maiores dimensões e com muitos objetos).

O artigo está organizado da seguinte forma: Seção II discute os algoritmos de traçado de raios usados na radiopropagação, sendo apresentado também o modelo de traçado de raios 3D desenvolvido. 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 radiopropagação. A primeira é baseada na Teoria das Imagens (IT - *Image Theory*), a qual é fortemente dependente da dimensão e da complexidade do ambiente, sendo mais usada em ambientes com pequenas dimensões e com poucos objetos, envolvendo apenas o mecanismo da reflexão [10]. Apesar de ser possível a incorporação de algoritmos de busca por pontos de transmissão e difração na IT, essa limitação intrínseca permanece. Por outro lado, o método SBR (*Shoot-and-Bouncing-Rays*), referido algumas vezes como “Força-Bruta”, é a abordagem de traçado de raios mais adequada para ambientes grandes e complexos, podendo envolver qualquer combinação das interações básicas (reflexão, transmissão e difração) [10]. A limitação intrínseca dessa abordagem é o alto tempo de processamento requerido 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 para se reduzir esse tempo foi otimizar os procedimentos de testes de sombreamento (testes de interseção raio-objeto). Existem diversas abordagens relacionadas a essa 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 outras [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, essa 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 da arquitetura

paralela. O método SBR, por sua concepção (independência dos raios lançados pela antena transmissora), já é intrinsecamente paralelizável. Essa característica permite que o mesmo seja aplicável diretamente no contexto de programação paralela. Pelo exposto, o modelo paralelo proposto foi baseado nas prerrogativas do SBR.

A técnica SBR proposta neste trabalho é uma técnica de traçado de raios tridimensional no sentido pleno (*Full 3D*). É importante ressaltar que o uso de técnicas denominadas de *Full 3D* no sentido pleno, sem nenhuma restrição no mecanismo de difração e com a possibilidade de combinação de todos os mecanismos de propagação definidos no modelo, é pouco relatado na literatura justamente por elas requererem recursos computacionais elevadíssimos.

### A. Técnica SBR Full 3D

De acordo com a técnica clássica SBR 3D [10], os raios são lançados pela antena transmissora em todas as direções, cada um representando uma parte da frente-de-onda que se propaga a partir da antena. Métodos SBR são também referidos como métodos diretos (*forward*), devido à simulação ser realizada a partir da antena transmissora, rastreando-se o percurso de cada raio e de seus descendentes (novos raios gerados pela interação com objetos do cenário). Neste trabalho, a técnica SBR 3D foi implementada segundo os três estágios, descritos a seguir:

1) **Lançamento de Raios:** Este estágio é responsável pelo lançamento de raios a partir da fonte (antena transmissora). A sua eficiência é medida pela habilidade do mesmo em gerar um lançamento uniforme de raios a partir da fonte, de tal forma que se consiga subdividir a frente-de-onda transmitida entre os raios em porções com forma e área aproximadamente iguais. Em métodos 2D, esse requisito é perfeitamente obtido, entretanto, em métodos 3D, a subdivisão igualitária não é efetivamente realizável, sendo necessários procedimentos de aproximação especiais. Uma estratégia 3D bastante utilizada na literatura para a modelagem da fonte de raios, capaz de minimizar o problema da subdivisão, é a apresentada em [11]. Essa é a estratégia de lançamento de raios adotada neste trabalho, na qual um icosaedro regular é inscrito em uma esfera de raio unitário que envolve a antena transmissora. Cada vértice do icosaedro representa a direção de lançamento de um raio. Para se prover mais raios, cada face triangular do icosaedro é subdividida em pequenos triângulos, de acordo como ilustrado na Fig. 1. Entretanto, esse método de subdivisão é aproximado, visto que a separação angular  $\alpha$  entre raios vizinhos não é exatamente a mesma, porque apenas os vértices originais do icosaedro tocam a esfera que circunscreve o mesmo. Então, essa estratégia sempre resultará em um valor para a separação angular variando entre  $(\alpha_{\min}, \alpha_{\max})$ , com uma diferença em torno de 20% entre o maior e o menor valor de  $\alpha$  obtido [12].

No estágio de recepção de raios, é necessário adotar um valor fixo para a separação angular de tal forma que seja

possível determinar quais componentes (raios) serão recebidos em cada ponto de recepção. Se for adotado o valor definido por  $\alpha_{\min}$ , alguns componentes de multipercurso que deveriam ser considerados não o serão. Por outro lado, se  $\alpha_{\max}$  for adotado, existirá contagem múltipla de componentes. A estratégia de lançamento adotada neste trabalho sempre considera o valor de  $\alpha_{\max}$  em todos os cálculos, sendo os problemas de contagem múltipla de componentes resolvidos por um procedimento adicional simples descrito no próximo estágio.

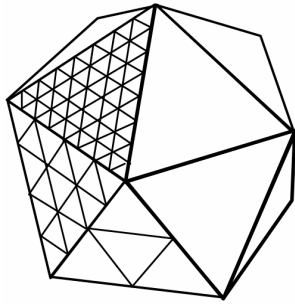


Fig. 1. Subdivisão recursiva das faces de um icosaedro regular.

- 2) **Recepção de Raios:** Este estágio verifica se um determinado raio deve ser considerado como um raio recebido para os pontos de recepção, baseado no conceito de esferas de recepção adaptativas, descrito em [11]. Nessa estratégia, cada ponto de recepção é envolvido por uma esfera de raio variável  $R$ , dado por  $R = \alpha d(3^{-1/2})$ , sendo  $\alpha$  a separação angular adotada e  $d$  a distância *unfolded*, que representa a distância total percorrida pelo raio testado e seus ancestrais em relação ao ponto de recepção. Se um determinado raio interceptar alguma esfera de algum ponto de recepção, este será considerado como recebido, caso contrário, não. A eficiência dessa estratégia será garantida somente se a separação angular  $\alpha$  entre raios vizinhos for exatamente igual para todos os raios lançados, o que não é perfeitamente obtido em uma implementação 3D. Para a separação angular considerada neste trabalho (i.e.,  $\alpha_{\max}$ ), a adoção dessa estratégia de recepção geraria problemas relacionados com contagem múltipla de componentes, caso não fossem resolvidos por meio de um simples procedimento que verifica se um determinado percurso de raio, com os mesmos mecanismos sobre os mesmos objetos, já foi computado na simulação. Em caso afirmativo, apenas o percurso de raio que estiver mais próximo em relação ao ponto de recepção será armazenado. Garante-se, assim, com o uso dessa abordagem, sempre a totalidade e a unicidade de componentes recebidos.
- 3) **Rastreamento de Raios:** Este estágio é responsável por rastrear os percursos de cada raio baseado nas interações do mesmo com os objetos do cenário. Para cada raio lançado pela fonte, um algoritmo recursivo é executado para verificar se o raio intercepta algum objeto do cenário (face ou aresta), ou se o mesmo intercepta algum ponto de recepção (esfera de recepção adaptativa). Se um raio

interceptar uma face, então o mesmo gerará dois novos raios, um refletido e outro transmitido, para os quais o algoritmo recursivo será novamente aplicado. No modelo SBR utilizado neste trabalho, para cenários *outdoor*, o raio transmitido não é considerado. Se um raio interceptar alguma esfera adaptativa de qualquer ponto de recepção, o procedimento de recepção de raios será disparado. No último caso, se um raio interceptar uma aresta, o mesmo gerará diversos raios difratados, para os quais o algoritmo recursivo será novamente aplicado, de acordo com o procedimento descrito adiante. O número de raios difratados gerados dependerá da resolução desejada. No modelo proposto, é considerado que raios difratados são gerados com uma separação angular fixada em  $2\alpha_{\max}$  para a primeira difração,  $3\alpha_{\max}$  para a segunda e assim por diante. Esse decréscimo na resolução dos raios difratados, de acordo com o número de difrações, reduz drasticamente o consumo de memória e de processamento. Observou-se que esse procedimento não afeta substancialmente a precisão de predição do modelo, se for adotada uma resolução inicial muito alta (i.e.,  $\alpha_{\max}$  muito pequeno).

A maioria das técnicas 3D, como a proposta em [4], apresenta restrições no mecanismo de difração, pois geralmente apenas se identificam as arestas no plano 2D e se realizam projeções dos percursos no plano 2D para um plano vertical. Técnicas totalmente tridimensionais apresentam problemas para a identificação de arestas “iluminadas” e para o cálculo do respectivo ponto de difração, que, de um modo geral, poderá se encontrar em algum ponto sobre a aresta no espaço 3D. Neste trabalho, é apresentado o conceito de cilindro de recepção para se realizar essa identificação, permitindo o uso de um modelo de traçado de raios *Full 3D*. Esse cilindro é construído com seu eixo axial coincidindo exatamente sobre a aresta sob teste de iluminação e apresenta duas calotas esféricas, uma no topo e outra na base, de acordo com o esquematizado na Fig. 2. Esse procedimento é equivalente a várias esferas de recepção sobrepostas ao longo da aresta. A dimensão adaptativa  $R$  do cilindro foi considerada igual à do modelo de esfera de recepção adotado. Então, se um raio intercepta o cilindro de recepção de alguma aresta, esta será identificada como uma potencial aresta difratora. Após essa identificação, é executado um procedimento que verifica se a mesma será realmente uma aresta difratora efetiva. Em caso afirmativo, o ponto de difração efetivo é determinado e o percurso de raios, ou seja, o percurso do raio em questão e de seus ancestrais, é corrigido para que possa atingir exatamente o ponto de difração estabelecido. Os procedimentos de determinação do ponto de difração e de correção do percurso de raios dependem do histórico do raio que intercepta o cilindro (i.e., dos mecanismos pelos quais ele foi originado) e são realizados de acordo com o histórico, fazendo-se uso de uma combinação simples de Teoria das Imagens (IT) [10], lei da difração (cone de Keller) [10] e do princípio generalizado de Fermat [12].

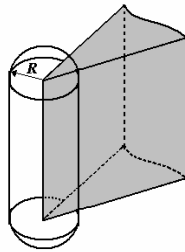


Fig. 2. Conceito de cilindro de recepção para arestas.

### III. MODELO PARALELO

O modelo paralelo proposto para algoritmos SBR foi esquematizado em estágios, conforme ilustrado na Fig.3. A idéia básica desse modelo é que após uma fase de pré-processamento de dados, a carga de processamento total poderá ser dividida entre os nós que compõem a arquitetura paralela (*cluster* de PC's), por meio 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 garantida graças à independência das entidades envolvidas (raios e pontos de recepção) e à 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].

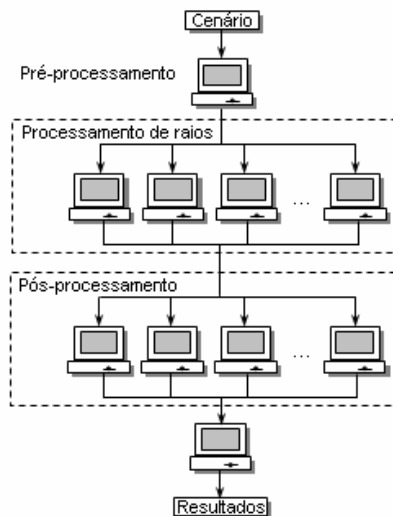


Fig. 3. Modelo de paralelização do algoritmo SBR 3D.

Mediante essa estratégia, a carga de processamento de um cluster homogêneo é balanceada pela distribuição de um número igual de raios iniciais 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 deve ser proporcional à sua capacidade de processamento, o que pode se fazer antecipadamente com base em características de hardware e software.

No contexto da computação paralela e distribuída, o modelo proposto pode ser enquadrado no paradigma SPMD (*Single*

*Program Multiple Data*) [13], pois dado um determinado cenário, cada nó executa o mesmo programa SBR sobre dados (raios iniciais e pontos de recepção) distintos. A estratégia de comunicação inicial entre os nós para fornecer os dados 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*) [14]. Entretanto, uma estratégia mais simples foi implementada, na qual 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 [15]. 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) poderá 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 consistem do estágio de pós-processamento. A seguir, cada estágio do modelo paralelo será brevemente detalhado. Na terminologia adotada, se o programa é executado de forma serial, é chamado de “modo serial”, e, em caso contrário, de “modo *cluster*”:

1) **Estágio de Pré-processamento:** Responsável pela definição, criação e leitura de arquivos de entrada específicos para cada nó. No modelo paralelo implementado, dois arquivos de entrada para cada nó já são previamente definidos, sendo denominados de “arquivo de *setup*” e “arquivo de raios”, cuja função será explicada:

- a) **Carregamento do arquivo de *setup*:** Realiza a leitura de um arquivo de *setup* pré-determinado. Um arquivo de *setup* provê informações sobre os parâmetros de simulação (potência transmitida, tipo das antenas, localização dos pontos de recepção, número máximo de interações, número de transmissores e suas localizações, frequência do sinal, etc), dos nomes dos arquivos de cenário (base de dados do ambiente a ser simulado) e de raios que devem ser carregados.
- b) **Carregamento do arquivo de cenário:** Realiza a leitura do arquivo de cenário definido no arquivo de *setup*. O arquivo de cenário contém as informações geométricas e morfológicas (propriedades elétricas) de todos os objetos pertencentes à base de dados do ambiente onde se deseja realizar as simulações. No modelo paralelo implementado, cada nó realiza a leitura do arquivo de cenário e mantém as informações do mesmo em memória principal até o fim da simulação.
- c) **Carregamento do arquivo de raios:** Realiza a leitura do arquivo de raios definido no arquivo de *setup*. Um 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.

- d) **Geração de arquivos de tabelas:** Responsável pela geração de um arquivo de tabela específico para cada nó. Os arquivos de tabela contêm a lista (apenas os índices) dos pontos de recepção que estarão sob responsabilidade de um determinado nó. Essa designação de pontos é feita de forma aleatória e de tal maneira que cada arquivo contenha aproximadamente o mesmo número de pontos de recepção (considerando um *cluster* homogêneo). Esses arquivos serão úteis na paralelização do estágio de pós-processamento. Como proposta de implementação desse procedimento no modo *cluster*, elege-se um determinado nó para ser o responsável por essa geração de arquivos (incluindo o dele próprio). No modo serial, todos os pontos de recepção são designados para um único nó. É importante ressaltar que os arquivos de tabela poderiam também ser pré-definidos antes da simulação, mas o tempo gasto nesse procedimento, mesmo para um grande número de pontos de recepção, é muito pequeno, e não interfere no desempenho do modelo paralelo se for realizado em tempo de execução.
- 2) **Estágio de Processamento de raios:** Responsável por executar o algoritmo SBR propriamente dito de acordo com o algoritmo SBR descrito na Seção II.A. Cada nó executa este estágio apenas para os raios previamente carregados do seu arquivo de raios.
- 3) **Estágio de Pós-processamento:** Consiste basicamente no salvamento de arquivos de percurso e na avaliação dos resultados de predição. Cinco 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, um procedimento adicional para montagem de um único arquivo por ponto de recepção. Tal procedimento foi chamado de “Agrupamento de arquivos” e será descrito na etapa c.
- 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. Um arquivo de *status* é criado para cada nó indicar para os outros que sua etapa de salvamento está concluída. Tal arquivo não contém qualquer informação e é usado somente para verificação do *status* do procedimento de salvamento. Assim, se o arquivo de *status* de um determinado nó foi criado, então isso indica que o mesmo está apto a iniciar o procedimento de agrupamento de arquivos, em caso contrário, não. Esse procedimento de verificação da existência dos arquivos de *status* de todos os nós é realizado por cada nó por meio de requisições ao sistema de arquivos de rede NFS. A cada verificação sem sucesso, o nó espera um determinado período antes de realizar uma nova verificação. Esse tempo de espera pode ser dimensionado de tal forma que, mesmo com um grande número de nós, as requisições simultâneas sejam evitadas ou minimizadas, não criando assim *overheads* de comunicação ao sistema de arquivos de rede. É importante ressaltar que se o tempo de espera for suficientemente maior que o tempo médio gasto para a verificação dos arquivos de *status*, será improvável que ocorram *overheads* de comunicação, sendo ainda mais improvável que nós tentem acessar o sistema de arquivos simultaneamente. Nos experimentos realizados neste trabalho, foi adotado um tempo espera de 3 segundos acrescidos de um período aleatório entre 0 e 3 segundos. Quando todos os nós obtiverem sucesso em suas verificações (todos os arquivos de *status* criados), eles começarão a executar a etapa de agrupamento. Esta etapa não é realizada no modo serial.
- c) **Agrupamento de arquivos:** Nesta etapa, cada nó faz o carregamento (leitura) de seu arquivo de tabela, o qual foi previamente gerado no estágio de pré-processamento. Nesse procedimento, são definidos quais serão os pontos de recepção que o mesmo deverá agrupar e conseqüentemente quais os arquivos de percurso que deverão ser agrupados por cada nó. O agrupamento não é apenas um procedimento de união dos arquivos gerados por cada nó, ele também realiza a retirada de componentes repetidos nos mesmos, de maneira análoga ao procedimento de eliminação de contagem múltipla de componentes descrito na Seção II.A.2. É importante observar que esta etapa se faz necessária apenas no modo *cluster*, pois no modo serial, já é criado um único arquivo por ponto de recepção.
- d) **Sincronização Agrupamento/Avaliação:** Para avaliação de resultados, é necessário que todos os nós concluam o procedimento de agrupamento de arquivos. Da mesma forma como realizado para a etapa de sincronização anterior (item b), cada nó cria um arquivo de *status* para indicar aos demais que sua etapa de agrupamento está concluída. Esta etapa não é realizada no modo serial.
- e) **Avaliação de resultados:** Responsável pela avaliação dos resultados de predição (campo elétrico, potência recebida, direção de raios, etc) a partir dos arquivos de percurso e pela geração de arquivos de saída com tais informações para cada ponto de recepção. Assim como na etapa de *Agrupamento de arquivos*, cada nó é responsável por avaliar resultados apenas dos

pontos de recepção previamente definidos em seu arquivo de tabela. É importante ressaltar que nesta etapa não é mais necessário carregar novamente os arquivos de tabela, haja vista que tal procedimento já foi realizado na etapa de *Agrupamento de arquivos*. No modo serial, todos os pontos de recepção são avaliados por um único nó.

#### 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. 4. Os dados geométricos dos prédios para os cálculos foram extraídos diretamente dos mapas bidimensionais apresentados em [16]. Para a construção da base de dados 3D, foi adotado uma altura de 30 m para todos os prédios. Seguindo a sugestão apresentada em [17], foi adotado um valor para a permissividade relativa de todas as paredes dos prédios iguais a 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. O transmissor foi localizado na posição rotulada como “Tx” (Fig. 4) a uma altura de 8.5 m e os pontos de recepção foram posicionados ao longo da rua Laurier (Laurier st. na Fig.4) a uma altura de 3.65 m. Todas as antenas foram definidas como dipolos de meia-onda e verticalmente polarizadas.

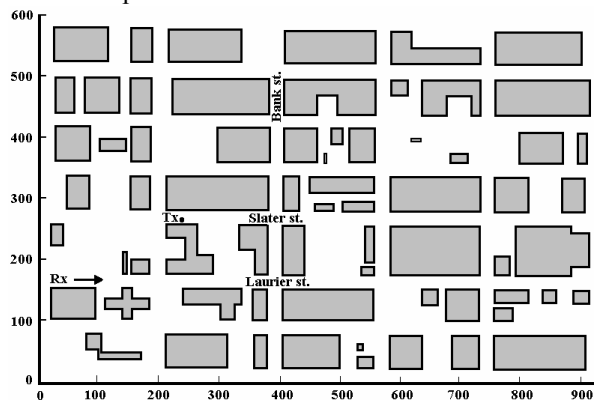


Fig. 4. Mapa de uma área da cidade de Ottawa.

Como algoritmo de traçado de raios foi considerado o modelo SBR *Full 3D* descrito na Seção II.A, juntamente com a Teoria Uniforme da Difração (UTD), brevemente descrita em [10]. Os campos foram calculados para uma frequência de 910 MHz com percursos envolvendo a combinação de até 8 reflexões e 1 difração.

As simulações foram executadas em um *cluster* de PC's do tipo IBM eServer 1350 consistindo de 21 nós computacionais (20 designados como nós escravos e 1 designado como nó mestre). Todos os nós são equipados com um processador Intel Xeon 64Bits e 2 GB de memória principal (RAM), sendo os mesmos interligados por uma rede dual gigabit Ethernet. Os códigos computacionais desenvolvidos foram implementados usando a linguagem de programação orientada a objetos C++. O compilador usado foi o g++ (gcc) versão

3.4.3 20050227 sob um sistema operacional Linux RHEL (*Red Hat Enterprise Linux* v3). É importante ressaltar que o paradigma de computação mestre/escravo é usado no modelo paralelo apenas para implementar o sistema de arquivos de rede Unix (NFS), pois o nó mestre não tem nenhuma gerência sobre as tarefas realizadas pelos nós escravos, podendo o mesmo até ser mais um nó participante das tarefas do modelo paralelo. Arquivos de entrada (arquivo de *setup* e arquivo de raios) personalizados para cada nó foram previamente criados, de acordo com os procedimentos descritos na Seção III.1. Todos esses arquivos, juntamente com o arquivo de cenário, foram distribuídos na rede através do NFS. Para avaliar o desempenho do modelo paralelo proposto, foram usadas principalmente métricas como *speedup*, taxa de expansão de carga de trabalho (*workload expansion ratio*) e taxa de utilização de recursos (*resource utilization ratio*) [4]. 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 o somatório 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 calculados 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 empregada também a métrica de eficiência, que é definida como  $E_n = S_n/n$  [4].

Conforme mostrado na Fig.5, os fatores de *speedup* obtidos para o modelo SBR *Full 3D* com 655362 raios lançados pela fonte (i.e., uma separação angular média  $\bar{\alpha} \approx 0.27^\circ$ ) apresentaram um comportamento superior ao fator de *speedup* de referência (chamado de fator de *speedup* linear [13]) para a maioria das configurações de *cluster*. Esse padrão de comportamento é referido na literatura como “*speedup* superlinear”. É importante ressaltar que esse desempenho foi obtido para uma resolução de lançamento de raios muito elevada, sendo a mesma (para o caso 3D) raramente utilizada na literatura pelo alto esforço computacional necessário.

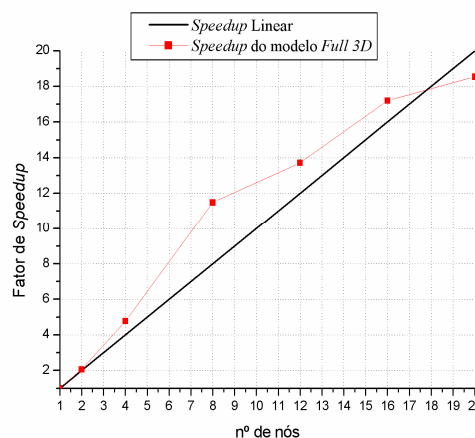


Fig. 5. Fatores de *speed-up* para 655362 raios lançados pela fonte (*Full 3D*).

TABELA I  
RAIOS PROCESSADOS / TEMPO DE PROCESSAMENTO [s]

Nó	Serial	Cluster (n = 2)	Cluster (n = 8)	Cluster (n = 20)
0	41631205 / 130123,2044	20741192 / 63074,16867	5171437 / 11328,83659	2055825 / 6986,809429
1	-	20890013 / 63073,81215	5245772 / 11328,92530	2086088 / 6986,277374
2	-	-	5197687 / 11323,97764	2106759 / 6987,585729
3	-	-	5183565 / 11328,19719	2078755 / 6984,939495
4	-	-	5151656 / 11330,36190	2050920 / 6991,294641
5	-	-	5279294 / 11329,23919	2098711 / 6990,123951
6	-	-	5289232 / 11328,38344	2068666 / 6989,840815
7	-	-	5112562 / 11326,51106	2117228 / 6991,818319
8	-	-	-	2083772 / 6985,491328
9	-	-	-	2110158 / 6986,064325
10	-	-	-	2056878 / 6992,618572
11	-	-	-	2080777 / 6987,034418
12	-	-	-	2044716 / 6987,125064
13	-	-	-	2100475 / 6989,054575
14	-	-	-	2092643 / 6981,424934
15	-	-	-	2088511 / 7016,118563
16	-	-	-	2097588 / 6991,779267
17	-	-	-	2061325 / 6991,395319
18	-	-	-	2091628 / 6983,078678
19	-	-	-	2059782 / 6983,461199

A Tabela I mostra, para quatro (04) configurações de *cluster* utilizadas (serial,  $n=2$ ,  $n=8$ ,  $n=20$ ), o número de raios processados e 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 na grande maioria abaixo do valor do tempo médio linear (tempo de execução serial /  $n$ ).

A Tabela II 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 nos casos com até  $n = 16$  foram sempre abaixo do caso ideal ( $W_n = 1.0$ ), apresentando um comportamento decrescente até  $n = 8$  (caso com melhor desempenho) e crescente para  $n > 8$ , chegando a ultrapassar levemente a unidade em  $n = 20$ . Isso evidencia a boa escalabilidade do modelo para a faixa de nós utilizada. As taxas de utilização de recursos obtidas para todas as configurações de *cluster* são muito próximas da taxa de utilização ideal ( $U_n = 1.0$ ), indicando que todos os nós gastam pouco tempo em estado ocioso (*idle status*). Entretanto, é observado que o aumento do número de nós normalmente gera estados de espera maiores. A eficiência do modelo

apresentou um comportamento acima da eficiência ideal ( $E_n = 100\%$ ) para a maioria das configurações de *cluster* utilizadas, exceto para o caso com  $n = 20$  nós. Neste caso, porém, a eficiência ainda permaneceu acima de 90%.

TABELA II  
MÉTRICAS DE AVALIAÇÃO DE DESEMPENHO

Cluster de $n$ -nós	$S_n$	$W_n$	$U_n$	$E_n$ [%]
1	1,000	1,000	1,000000000	100,00
2	2,063	0,969	0,999997174	103,15
4	4,771	0,838	0,999930402	119,28
8	11,484	0,696	0,999796312	143,55
12	13,726	0,873	0,999530106	114,38
16	17,202	0,929	0,999578948	107,51
20	18,546	1,074	0,996158594	92,73

O desempenho superior ao ideal apresentado pelo modelo paralelo para a maioria das configurações de *cluster* utilizadas pode ser justificado principalmente pelo uso da técnica SBR *Full 3D*, na qual, conforme já mencionado anteriormente, há problemas referentes à contagem múltipla de componentes. Para se combater esse problema, para cada novo percurso de raios recebido é necessário identificar se já existe algum percurso fisicamente semelhante que tenha sido contabilizado para um mesmo ponto de recepção. Para se realizar essa tarefa no contexto de estrutura de dados, os percursos de raios recebidos por cada ponto de recepção são armazenados em uma lista. É necessário varrer essas listas e aplicar o procedimento de eliminação de componentes já contabilizados descritos brevemente na Seção II.A.2. Como os percursos de raios são recebidos de forma arbitrária pelos pontos de recepção e a classificação (ordenação) dos mesmos segundo algum critério não é um procedimento trivial, a varredura das listas deve ser realizada por meio de uma busca linear, que apresenta um tempo total de varredura diretamente proporcional ao tamanho das listas e ao número de buscas realizadas [18]. No modo serial, o número de buscas  $N_{search}$  às listas de percursos é igual ao número de raios tidos como recebidos, e são realizadas sobre listas que podem ser muito grandes. Já no modo *cluster*, os nós apresentam um número de buscas bem menor, igual a  $N_{search} / n$ , e que são realizadas sobre listas menores (inversamente proporcional a  $n$ ). Assim, o ganho de tempo obtido no modo *cluster* para esse procedimento de varredura será proporcional a  $n^2$ , podendo compensar em várias situações as perdas de tempo causadas pelo procedimento de agrupamento (exclusivo do modo *cluster*). Esse efeito pode ser observado através das curvas dos fatores de *speedup* apresentados na Fig.5. Nela, apenas a configuração de *cluster* com  $n = 20$  nós apresenta um fator de *speedup* abaixo do fator linear. Portanto, nessa configuração, o ganho de tempo obtido na varredura de percursos do modo *cluster* não foi suficiente para compensar as perdas de tempo do procedimento de agrupamento de arquivos. É importante ressaltar que todos os outros procedimentos do modelo paralelo de traçado de raios apresentaram uma redução de tempo quase linear com o número de nós utilizados, exceto o

processo de varredura de percursos e o de agrupamento de arquivos. Isso mostra que se fosse utilizado um modelo SBR 2D, que não enfrenta problemas de contagem múltipla, o *speedup* tenderia sempre a estar bem próximo do *speedup* linear, garantindo, assim, a eficiência do modelo para o caso bidimensional (2D).

As características apresentadas pelo modelo paralelo SBR *Full 3D* podem implicar 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, poderá ser em alguns casos mais atrativo que sendo executado de forma serial sem particionamento algum. A escalabilidade do modelo é naturalmente garantida dada a independência dos raios iniciais e pontos de recepção. Entretanto, a eficiência acima de 100% apresentada pelo mesmo 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). Essa 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) torna o algoritmo SBR mais preciso.

Para dar uma indicação de qualidade de predição provida pelo modelo SBR *Full 3D* desenvolvido, a perda de propagação ao longo da Laurier st. obtida foi comparada com medidas relatadas em [16]. Os resultados são mostrados na Fig. 6, observando-se uma boa concordância, considerando a simplicidade da base de dados do cenário e a falta de informações sobre as propriedades elétricas dos materiais que compõem as edificações.

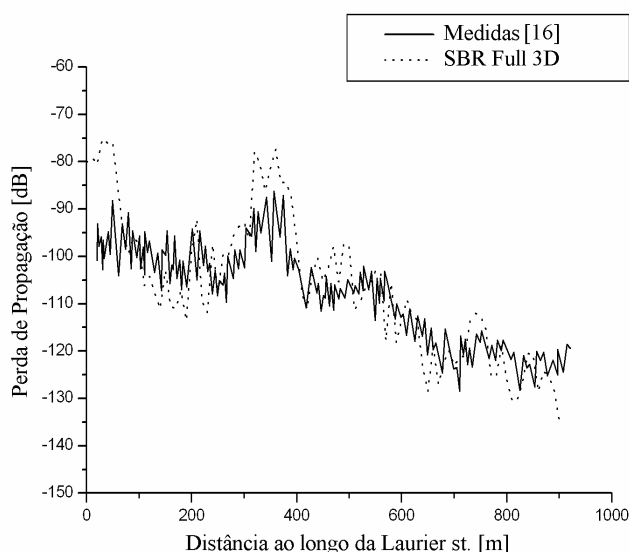


Fig. 6. Perda de propagação ao longo da Laurier st.

## V. CONCLUSÕES

Neste trabalho, foi proposto um modelo computacional paralelo para técnicas de traçado de raios 3D aplicado na predição de radiopropagação. Essa abordagem é baseada na independência das tarefas em um algoritmo SBR para distribuir eficientemente a carga de trabalho total, por meio de uma distribuição aleatória de raios iniciais e de pontos de recepção entre os nós da arquitetura paralela (*cluster* de PC's). Diversos aspectos relacionados com a implementação prática do modelo paralelo foram descritos, enfatizando-se a sua simplicidade de implementação computacional com a necessidade apenas de um sistema de arquivos de rede e a facilidade de adaptação para a maioria dos algoritmos baseados em lançamento de raios, 2D ou 3D, e para ambientes *outdoor* ou *indoor*. Os resultados de simulações apresentados mostraram que a escalabilidade do modelo é naturalmente obtida devido à independência dos processos envolvidos. Adicionalmente, a eficiência do modelo apresentou comportamento acima do ideal para casos com ostensivo processamento de raios. Essas características favorecem o aumento da precisão de predição por meio do aumento da densidade de raios lançados e da possibilidade de incorporação de novos mecanismos de propagação

## VI. AGRADECIMENTOS

Os autores agradecem aos professores Mário Jorge Moreira Leitão e António Augusto de Sousa, ambos do Instituto de Engenharia de Sistemas e Computadores do Porto (INESC-Porto - Portugal), pela disponibilização da infra-estrutura computacional de alto desempenho, pertencente à Faculdade de Engenharia da Universidade do Porto (FEUP), para a execução deste trabalho.

## VII. REFERÊNCIAS

- [1] J. Richter, M.O. Al-Nuaimi, and L.P. Ivrisimtzis, "Optimization of radio coverage in urban microcells using a UTD based ray-tracing model," *Antennas and Propagation, IEE Proceedings*, vol. 151, Issue: 3, 21 June 2004, pp. 187 - 192.
- [2] Z. Chen, A. Delis, and H. L. Bertoni, "Building footprint simplification techniques and their effects on radio propagation predictions," *Comput. J.*, 47 (1) (January 2004) 103-133.
- [3] R. A. Kipp, and M.C. Miller, "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, and 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, and 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, and 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] A. M. Cavalcante, M. J. Sousa, C. S. Sales Jr, J. C. W. A. Costa, C. R. L. Francês, and G. P. S. Cavalcante, "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] V. Degli-Esposti, D. Guiducci, A. de'Marsi, P. Azzi, and F. Fuschini, "An advanced field prediction model including diffuse scattering," *IEEE*



*Transactions on Antennas and Propagation*, vol. 52, Issue: 7, July 2004, pp. 1717 - 1728.

- [9] H. L. Bertoni, *Radio propagation for modern wireless systems*, Prentice-Hall - Wireless Communications Series, 2000.
- [10] M. F. Cátedra, and J. P. Arriaga, *Cell Planning for Wireless Communications*, Artech House - Mobile Communications Series, 1999.
- [11] Seidel, S.Y. and Rappaport, T.S., "Site-specific propagation prediction for wireless in-building personal communication system design," *IEEE Trans. on Veh. Technol.*, vol. 43, no. 4., Nov. 1994.
- [12] Tan, S.Y. and Tan, H.S., "Modelling and measurements of channel impulse response for an indoor wireless communication system," *IEE Proc.-Microw. Antennas Propag.*, vol. 142, no. 5, October 1995.
- [13] Y. Foster, K. Kennedy, J. Dongarra, and G. Fox, *Sourcebook of Parallel Computing*, Morgan Kaufman Pub, 2002.
- [14] W. Gropp, E. Lusk, A. Skjellum, and R. Thakur, *Using MPI: Portable Parallel Programming with the Message Passing Interface (Scientific and Engineering Computation)*, 2nd Edition, MIT Press, 1999.
- [15] A. S. Tanenbaum, *Computer Networks*, 4th Ed., Prentice Hall, 2002.
- [16] 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.
- [17] 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.
- [18] Donald Knuth, *The Art of Computer Programming, Vol. 3: Sorting and Searching*, Third Edition. Addison-Wesley, 1997. ISBN 0-201-89685-0. Section 6.1: Sequential Searching, pp.396-408.

## VIII. BIOGRAFIAS



**André Mendes Cavalcante** recebeu o título de Mestre em Engenharia (Telecomunicações) pela Universidade Federal do Pará (UFPA-Brasil), em 2003. Graduou-se em Engenharia Elétrica na área de Telecomunicações pela mesma instituição em 2001. Foi professor substituto pela Universidade Federal do Pará de 2004 a 2005. Atualmente, está em doutoramento pela UFPA e faz parte do grupo de pesquisa do Laboratório de Eletromagnetismo Aplicado (LEA-UFPA), onde desenvolve vários projetos de pesquisas na área de telecomunicações com empresas associadas. Seus interesses de pesquisa são: modelagem de canais sem fio através de técnicas determinísticas (*ray-tracing*, FDTD), propagação de ondas em ambientes florestais, técnicas de otimização de sistemas, modelos de computação paralela e distribuída e redes de comunicação sem fio.



**Marco José de Sousa** recebeu o título de Mestre em Engenharia (Telecomunicações) pela Universidade Federal do Pará (UFPA-Brasil), em 2003. Graduou-se em Engenharia Elétrica na área de Telecomunicações pela mesma instituição em 2002. Foi professor substituto pela Universidade Federal do Pará de 2004 a 2005. Atualmente, está em doutoramento pela UFPA e faz parte do grupo de pesquisa do Laboratório de Eletromagnetismo Aplicado (LEA-UFPA), onde desenvolve vários projetos de pesquisas na área de telecomunicações com empresas associadas. Seus interesses de pesquisa são: comunicações ópticas, otimização e síntese de filtros ópticos, algoritmos genéticos, inteligência artificial e computacional, técnicas de otimização determinísticas, análise numérica utilizando FDTD e ray-tracing, computação paralela e distribuída, eletrônica e lógica programável.



**João Crisóstomo Weyl Albuquerque Costa** nasceu no Pará, Brasil, em 27 de Janeiro de 1959. Engenheiro Eletricista - Eletrônico, pela Universidade Federal do Pará (UFPA), Brasil, em 1981, pós-graduado em Geofísica, pela Universidade Federal do Pará (UFPA), Brasil, em 1983, Mestre em Engenharia Elétrica - Telecomunicação, em 1989, pela PUC-RJ, Brasil, e Doutor em Engenharia Elétrica em 1994, pela Universidade Estadual de Campinas, São Paulo, Brasil. Trabalhou como engenheiro na Agência de Telecomunicação Brasileira de 1981 a 1982, e na Companhia de Telecomunicações do Pará, de 1984 a 1986. Desde 1994, faz parte do quadro de professores da UFPA, atuando nos laboratórios de Eletromagnetismo Aplicado (LEA) e de Planejamento de Redes de Alto Desempenho (LPRAD) do Departamento de Engenharia Elétrica e de Computação (DEEC). Sua área de interesse é a modelagem de dispositivos e sistemas para comunicações ópticas e móveis, incluindo redes de acesso. É pesquisador do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) desde 1994.



**Carlos Renato Lisboa Francês** recebeu o título de Doutor e Mestre em Ciência da Computação pela Universidade de São Paulo (USP), no Instituto de Ciências Matemáticas e de Computação, Brasil, em 2001 e 1998, respectivamente, e o de Bacharel em Ciência da Computação pela Universidade Federal do Pará (UFPA), Brasil, em 1995. Desde junho de 2002, faz parte do Laboratório de Planejamento de Redes de Alto Desempenho (LPRAD) do Departamento de Engenharia Elétrica e de Computação (DEEC) da UFPA, realizando atividades de ensino e pesquisa. Seus principais interesses são: avaliação de desempenho de sistemas complexos, computação paralela, sistemas distribuídos e mineração de dados.



**Gervásio Protásio dos Santos Cavalcante** nasceu no Pará, Brasil, em 19 de Junho de 1945. Recebeu o título de Engenheiro Eletricista - Eletrônico pela Universidade Federal do Pará (UFPA), Brasil, em 1971, o de Mestre em Engenharia Elétrica (Eletrônica), em 1974, pela Universidade Federal da Paraíba (UFPB), Brasil, e recebeu o grau de Doutor em Engenharia Elétrica (Radiopropagação em ambientes florestais), em 1982, pela Universidade Estadual de Campinas, São Paulo, Brasil. De 1974 até 1986, foi professor Assistente pelo Departamento de Engenharia Elétrica e de Computação (DEEC) da UFPA. Desde 1986, é professor Titular do DEEC-UFPA. Atividades de 1987 até 2001: Chefe do Departamento de Engenharia Elétrica; Coordenador do Programa de Pós-Graduação em Engenharia Elétrica da UFPA; membro do Conselho Superior de Ensino e Pesquisa (CONSEP) da UFPA; membro de Conselho Superior de Administração (CONSAD) da UFPA; membro fundador da Sociedade Brasileira de Microondas e Optoeletrônica (SBMO); membro da Sociedade Brasileira de Telecomunicações (SBrT); membro do Comitê Diretor e Comitê Técnico do III Simpósio Brasileiro de Microondas; membro do Comitê Diretor do VIII Congresso Brasileiro de Automática; Chair do Comitê Técnico e Membro do Comitê Diretor do VI Simpósio Brasileiro de Microondas e Optoeletrônica em Belém, Brasil; Chair do Comitê Técnico e membro da Comissão Organizadora do 1º Encontro de Ciência e Tecnologia para a Amazônia, em Belém do Pará, Brasil. Chair do Comitê Técnico e Membro do Comitê Diretor do Simpósio Internacional de Microondas e Optoeletrônica (IMOC'2001 - International Microwaves and Optoelectronic Symposium), em Belém - Pará, Brasil. Foi Chairman Geral do Simpósio Brasileiro de Telecomunicações, em 2004, em Belém - Pará, Brasil. O Dr. Gervásio Protásio dos Santos Cavalcante é Membro do Instituto de Engenheiros Elétricos e Eletrônicos (IEEE). Tem publicado diversos artigos em periódicos e congressos nacionais e internacionais e orientado diversos estudantes de graduação, de mestrado e de doutorado. Atualmente, suas áreas de interesse são: Radiopropagação em ambientes florestais, sistemas móveis e antenas inteligentes.