

Redes de sensores móveis: análise da velocidade, comunicação e esforço computacional

Filippe C. Jabour^{1,2}, Eugênia Giancoli^{1,2}, Aloysio C. P. Pedroza¹

¹ UFRJ - PEE/COPPE/GTA - DEL/POLI

² CEFET-MG - Campus Leopoldina

{jabour, eugenia, aloysio}@gta.ufrj.br

Abstract—This work studies three parameters of a proposal for mobility support in wireless sensor networks. The variable parameters are the nodes' mobility rate, radio range and the number of iterations per second for three proposed algorithms. Through simulations, we verify which algorithm is suitable to each subset of the investigated scenarios.

Resumo— Este trabalho faz um estudo sobre três parâmetros de uma proposta de suporte à mobilidade em redes de sensores sem fio. Os parâmetros variáveis são a taxa de mobilidade dos nós, a faixa de alcance do rádio e o número de iterações por segundo para três algoritmos propostos. Através de simulações, nós verificamos quais algoritmos são apropriados a qual subconjunto dos cenários investigados.

I. INTRODUÇÃO

Uma rede de sensores sem fio (RSSF) é uma rede composta por dispositivos autônomos, espacialmente distribuídos, usando sensores para, cooperativamente, monitorar condições físicas ou ambientais, tais como temperatura, som, vibrações, pressão, movimento ou poluentes, em diferentes localizações. [1][2].

Em RSSF móveis, os sensores, os observadores ou o fenômeno são móveis. Sempre que um dos sensores associado ao caminho entre observador e fenômeno se move, esta rota pode falhar. neste caso, o observador ou os sensores envolvidos devem tomar a iniciativa de construir uma nova rota [3]. A mobilidade dos sensores pode advir de influências ambientais, tais como vento ou água, sensores podem estar anexados ou serem carregados por entidades móveis, ou eles podem possuir capacidade automotiva [1]. Alguns cenários com mobilidade são: sensores usados para estudo do movimento de tornados [3]; sensores conectados a veículos, em grandes cidades, para estudo das condições de tráfego e planejamento de rotas [3]; sensores podem estar flutuando em correntes fluviais [4]; existe um projeto da NASA para exploração de Marte, com sensores dirigidos por correntes eólicas [5].

Nós propusemos um esquema em duas camadas para suporte à mobilidade em RSSF [6][7][8]. Tal proposta se baseia em interações locais entre sensores, em tarefas globais desempenhadas por agentes móveis (AM) e em previsões de localização feitas por nós móveis. Os algoritmos propostos apresentaram comportamentos diferentes nos diversos cenários estudados. Foram necessárias novas rodadas de simulações para identificar os parâmetros e as faixas de variação dos mesmos onde cada algoritmo melhor se adapta.

Na seção II é descrito o algoritmo de previsão de localização dos nós. Na seção III são apresentados os três algoritmos usados pelos AM para desempenhar sua tarefa. O simulador desenvolvido e as simulações efetuadas são descritos na seção IV. A análise dos algoritmos, no que se refere à velocidade dos nós, alcance de comunicação e esforço computacional, está na seção V. Na seção VI estão as conclusões e trabalhos futuros.

II. PREVISÃO DE LOCALIZAÇÃO DO NÓ

Um algoritmo de localização computa periodicamente a posição atual P do nó. Com duas destas medidas, $P_i(x_i, y_i)$ e $P_{i-1}(x_{i-1}, y_{i-1})$, o nó computa a inclinação m da reta definida por estas pontos. Na equação 1, m_i representa a direção momentânea do deslocamento do nó; m_{pred_i} (equação 2) é a previsão da direção de deslocamento; $m_{pred_{i-1}}$ é a última previsão feita; e o parâmetro α é usado para dar maior ou menor peso ao último coeficiente angular calculado, m_i , em comparação ao histórico dos valores estimados, $m_{pred_{i-1}}$. Conforme mostrado em [6] e [7], este algoritmo produz previsões suficientes para nortear as tomadas de decisão dos AM.

$$m_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \quad (1)$$

$$m_{pred_i} = \alpha m_i + (1 - \alpha) m_{pred_{i-1}} \quad (0 \leq \alpha \leq 1) \quad (2)$$

Uma vez que a inclinação m_i do movimento é estimada, a distância que o nó percorrerá no próximo passo é estimada da mesma maneira (equação 3). O termo $d_{pred_{i-1}}$ representa a última previsão e d_i é calculado em cada passo baseado na posição “real” do nó. A localização “real” vem de um algoritmo de localização e incorpora um erro intrínseco. O termo γ tem a mesma função que o α .

Tanto a computação de m_{pred_i} quanto a de d_{pred_i} são de complexidade computacional constante.

$$d_{pred_i} = \gamma d_i + (1 - \gamma) d_{pred_{i-1}} \quad (0 \leq \gamma \leq 1) \quad (3)$$

III. ALGORITMO DE DECISÃO DOS AGENTES MÓVEIS

A. Interações locais

Estamos interessados nos cenários descritos na seção I, onde os nós se movem aleatoriamente pelo campo de sensoriamento.

Nós consideramos que os nós executam um protocolo de localização que lhes forneça suas coordenadas em um dado momento, com um certo erro. Os nós estimam repetidamente sua posição futura e a direção de seu movimento (seção II).

Todos os nós mantêm uma lista com seus vizinhos alcançáveis em um dado instante (V_i). Isto pode ser feito com uso de um protocolo como o *Internet MANET Encapsulation Protocol (IMEP)* [9]. Este protocolo incorpora mecanismos de monitoramento do estado dos enlaces e conectividade com vizinhos.

Self-organization Protocols for Wireless Sensor Networks [10] propõe um conjunto de algoritmos distribuídos que fornecem as bases para uma completa auto-organização de uma rede de sensores. A combinação do SMACS e do *Mobile MAC*, apresentadas em [10] atende aos demais requisitos das interações locais propostas em nosso trabalho.

B. Tarefas globais

Genericamente falando, AM é um tipo especial de *software* que pode executar autonomamente. Uma vez despachado, ele pode migrar de nó em nó, executando seu processamento autonomamente [11]. Algumas vantagens do uso de AM são a escalabilidade da RSSF, economia de banda, facilidade de reprogramação e agregação de dados [12]. Muitas aplicações são listadas em [13], onde a implementação de um *middleware* (plataforma) de AM para o sensor MICA2 e o desenvolvimento de algumas aplicações são descritos e avaliados.

No presente trabalho, os AM têm tarefas de sensoriamento a serem executadas. Estas tarefas contêm as coordenadas da região de onde os dados devem ser coletados a as coordenadas da região para onde os dados devem ser trazidos. O destino momentâneo de um AM é chamado de **região alvo**. Esta região é definida por duas coordenadas em vértices opostos de um retângulo.

Quando injetado na rede, o AM inicia uma instância do processo de migração, em busca destino, na direção da região alvo. Periodicamente, o AM consulta a posição atual estimada por seu hospedeiro. Se ela estiver dentro da região alvo, ele coleta os dados de interesse (ou executa a computação programada) e inicia nova instância do processo de migração, em direção à região de retorno. Se o hospedeiro não estiver na região alvo, o AM executa o algoritmo de decisão. Nós propusemos três algoritmos.

1) *Algoritmo somente-distância*: O AM identifica no conjunto de vizinhos V_i (incluindo o hospedeiro atual), qual deles está mais próximo do centro da região alvo. Se for o hospedeiro atual, nenhuma ação é tomada. Se for um dos vizinhos, o AM inicia o processo de migração para este vizinho.

2) *Algoritmo somente-coeficiente-angular*: Enquanto executa o algoritmo de localização (seção II), o nó calcula a cada iteração um parâmetro que chamamos de **tipo de movimento**. Este parâmetro assume os valores S (parado), U (para cima), D (para baixo), R (para a direita) ou L (para a esquerda). U e D são direções sobre o eixo Y ou muito próximas a ele. Estes casos são isolados para tratamento dos infinitos da tangente. R

e L estão em todo o restante do primeiro e quarto quadrantes (R) e segundo e terceiro quadrantes (L).

O AM computa o mesmo parâmetro (tipo de movimento) relacionado à reta definida pela localização de seu hospedeiro e o centro da região alvo, isto é, a inclinação da **trajetória ótima**. Este valor é chamado de **tipo de movimento ótimo**. A inclinação da trajetória ótima é chamada de **coeficiente angular ótimo**. Se houver, dentre os vizinhos e o hospedeiro (conjunto V_i), um ou mais nós com o mesmo tipo de movimento que o da trajetória ótima, o AM identifica aquele cuja inclinação da trajetória esteja mais próxima do coeficiente angular ótimo. Se for o seu hospedeiro, nenhuma ação é tomada. Se for um dos vizinhos, o AM inicia o processo de migração para este nó.

3) *Algoritmo distância-e-coeficiente-angular*: Se houver no conjunto V_i um ou mais nós com o mesmo tipo de movimento que o da trajetória ótima, o AM executa o algoritmo **somente-coeficiente-angular** (seção III-B2). Se todos os nós em V_i tiverem tipos de movimento diferentes do ótimo, o algoritmo **somente-distância** é usado.

4) *Características comuns aos três algoritmos*: Nenhum algoritmo de roteamento é necessário e nenhum estado precisa ser armazenado na rede. Decisões instantâneas e baseadas em valores não cumulativos existentes nos nós, permitem aos AM executarem as tarefas necessárias. Os três algoritmos são de complexidade linear $O(k)$, onde k é o número de vizinhos e $k \leq n$ (n é o número total de nós da rede).

IV. DESCRIÇÃO DO SIMULADOR E DAS SIMULAÇÕES

Foi desenvolvido um simulador de eventos discretos usado para validar e comparar os algoritmos propostos. Trata-se de uma aplicação Java [14] que lê um arquivo de entrada no formato do simulador *ns-2* [15]. Este arquivo contém as instruções de movimentação dos nós da RSSF. Todos estes nós são configurados com o mesmo alcance de comunicação de rádio (em metros) e o mesmo erro percentual do algoritmo de localização. O simulador é configurado com uma precisão de relógio que vai de zero a doze casas decimais para cada passo da simulação. Cada nó, cada AM e o relógio da simulação são executados como linhas de execução independentes e concorrentes (*threads*). Somente quando todas as *threads* terminam a computação de um passo, o relógio avança para o próximo (segundo a precisão do relógio). Alguns AM são injetados na rede com alguns parâmetros: hospedeiro inicial, instante inicial de processamento, coordenadas da região de destino, coordenadas da região de retorno e tipo de algoritmo (seções III-B1, III-B2 ou III-B3) a ser usado.

Considera-se que um algoritmo de localização está em execução e que uma nova estimativa de localização é gerada para cada nó, em cada passo do relógio da simulação. O conjunto de vizinhos V_i é considerado disponível e é fornecido pelo simulador quando demandado. Quando o AM decide migrar do seu hospedeiro para algum vizinho, isto é feito em um passo do relógio, com 100% de probabilidade de sucesso.

Os cenários de mobilidade foram criados com o aplicativo *setdest* que acompanha o *ns-2*. Para uma área, um número

de nós e um tempo de duração pré-definidos, esta aplicação gera aleatoriamente uma coordenada inicial (passo 1), uma coordenada de destino (passo 2) e uma velocidade de deslocamento entre estas duas coordenadas (passo 3). Ao chegar à coordenada de destino, é gerado aleatoriamente um intervalo de repouso para o nó (passo 4). Em seguida, o processo se repete dos passos 2 a 4, até o término do período de duração definido.

Cada vez que um AM atinge a região alvo, é contado um **sucesso**. Então, a região alvo muda da região destino para a região de retorno (ou vice versa). O simulador conta ainda o número de migrações dos AM e registra o instante de todos os eventos de interesse.

V. ANÁLISE DOS ALGORITMOS

Para avaliação, comparação e classificação dos algoritmos, selecionamos três parâmetros principais de análise: velocidade, alcance de comunicação de rádio e o esforço computacional envolvido. Como métricas, adotamos os sucessos, que ocorrem quando o AM atinge a região alvo, e as migrações, relacionadas sobretudo com o consumo de energia. Ao final de cada simulação, calculamos a média destes dois valores, para cada AM.

A. O impacto da velocidade dos nós

O aumento da velocidade de deslocamento dos nós tem impacto positivo sobre os dois algoritmos que se baseiam na distância até a região alvo: **somente-distância** e **distância-e-coeficiente-angular**. O maior dinamismo dos nós faz com que eles passem mais vezes pelas regiões alvo. É interessante observar que um alto grau de mobilidade pode ser prejudicial a muitos algoritmos de roteamento, na medida em que rotas construídas se tornarão inválidas nos instantes seguintes. Na presente proposta, um maior grau de mobilidade melhora o desempenho ou ao menos não o reduz.

O algoritmo que considera somente o coeficiente angular não desfruta dos benefícios do maior dinamismo e chega até a ser prejudicado por ele. Como a migração ocorre para o vizinho se movendo na direção mais próxima do coeficiente angular ótimo, concluímos que as opções descartadas, para o cenário simulado, eram mais interessantes do que a escolhida.

Existe, entretanto, um efeito colateral da velocidade sobre o algoritmo **somente-distância**. Este algoritmo desconsidera a direção de deslocamento do nó destino, quando da decisão do AM. Assim, o AM pode decidir migrar para um vizinho mais próximo da região alvo, mesmo que ele esteja se afastando dela e o seu hospedeiro atual esteja indo em direção a ela. Esta decisão, combinada com a granularidade de tempo de execução (ver precisão do relógio na seção IV) pode representar uma migração errada e irrecuperável. Isto seria devido ao fato da migração de volta ao hospedeiro original não ser mais possível. Quanto menor o alcance do rádio e quanto maior a velocidade dos nós, maior a chance disso acontecer.

Para esta análise, as simulações foram feitas com os seguintes parâmetros: 300 nós distribuídos e movimentados aleatoriamente; velocidade dos nós: distribuição normal entre 1 e os

valores do eixo X das figuras 1 e 2; intervalo de repouso: distribuição uniforme entre 0 and 20s; área: 500m x 500m (delimitada pelos pontos (0, 0) e (500, 500)); tempo de simulação: 1.000s; precisão do relógio (passo): zero casas decimais (passo = 1s); alcance do rádio: 20m; erro do algoritmo de localização: 0%; região de destino: $A_D(350, 450)$, $B_D(450, 350)$, onde A_D é o vértice superior esquerdo e B_D é o vértice inferior direito; região de retorno: $A_R(50, 150)$, $B_R(150, 50)$ (mesmos vértices descritos acima); três AM são injetados em cada nó (um de cada tipo de algoritmo) e começam a processar no instante 5s, executando até o fim da simulação.

A figura 1 mostra o ganho dos dois algoritmos baseados em distância no número médio de sucessos. O aumento mais significativo ocorre até a velocidade de 30m/s. Após este valor, vemos uma leve queda do algoritmo **somente-distância**, como mencionado acima. Da mesma forma, o algoritmo **somente-coeficiente-angular** apresenta um leve ganho de desempenho. O algoritmo **distância-e-coeficiente-angular**, por ser híbrido, combina a perda e o ganho mencionados e se mantém estável a partir dos 30m/s. Não há perda significativa de desempenho em nenhum dos casos.

A figura 2 mostra o número de migrações para cada um dos algoritmos. Este parâmetro refere-se ao consumo de energia, já que o gasto com comunicação é o mais relevante em RSSF. O algoritmo **somente-distância**, por ser “guloso”, efetua muito mais migrações, devido às correções que tem que fazer quando o AM migra equivocadamente. O **somente-coeficiente-angular** faz menos migrações, mas presta menos serviço (menos sucessos). O melhor custo benefício fica com o algoritmo híbrido, já que o número médio de sucessos é próximo ao do algoritmo **somente-distância** e o consumo de energia por migrações é bem inferior.

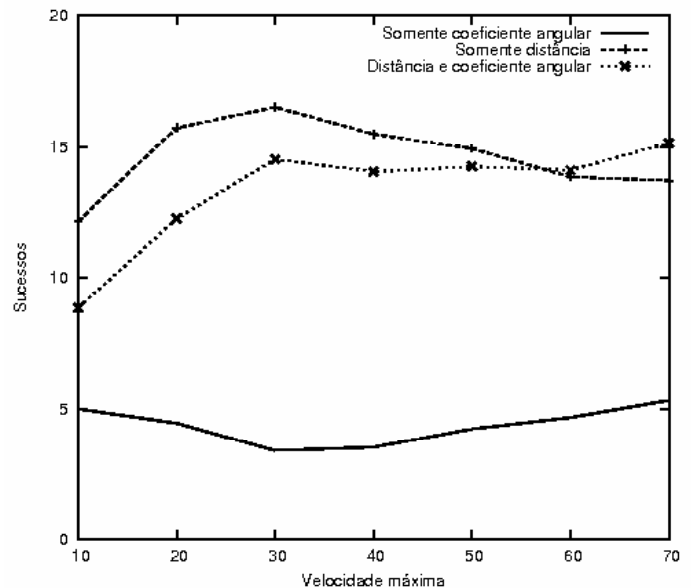


Fig. 1. Impacto da velocidade - Média de sucessos por AM

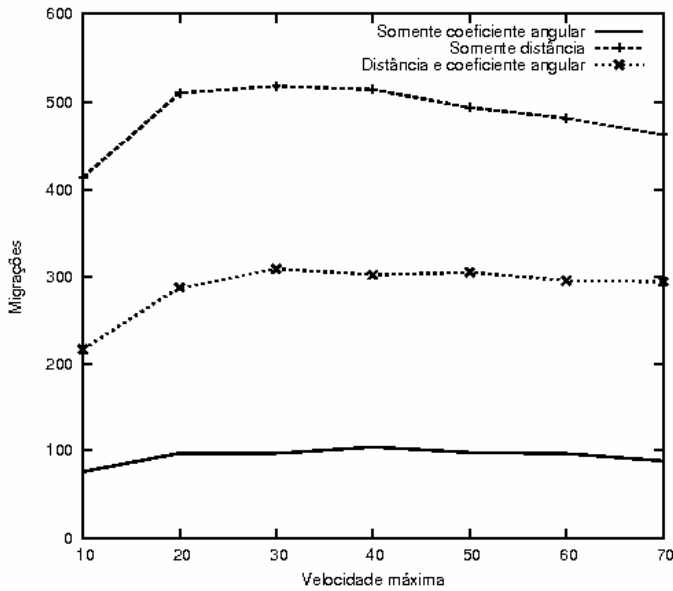


Fig. 2. Impacto da velocidade - Média de migrações por AM

B. Análise do alcance de comunicação

O alcance de comunicação dos sensores se reflete diretamente no desempenho da rede. Como vimos na seção V-A, o algoritmo **somente-distância** depende da execução de mais de uma iteração (mais de um passo) para um mesmo conjunto de vizinhos (ou um conjunto não muito diferente do anterior) para poder se recuperar a tempo das migrações equivocadas. Deste modo, quanto maior o raio de alcance do rádio de comunicação, maiores as chances de recuperação.

De um modo geral, quanto maior o raio de comunicação, maior o número de vizinhos no conjunto V_i e mais acuradas as decisões dos AM. Em contrapartida, mais mensagens trocadas e maior esforço computacional nos laços de decisões dos algoritmos.

Para esta análise, usaremos o mesmo cenário da seção V-A, com a velocidade fixa em $30m/s$ e a faixa de comunicação variando de 10 a $90m$ de raio.

Na figura 3 vemos uma grande superioridade dos algoritmos que consideram a distância ao destino na tomada de decisão de migração, sendo o **somente-distância** o de melhor desempenho.

O algoritmo **somente-coeficiente-angular** não atende a este cenário. A taxa de sucesso não se beneficiou do aumento do raio de cobertura do rádio. Os dois algoritmos que consideram a distância se valem da possibilidade de migrações que cubram distâncias próximas a 100% do alcance do rádio. Isto permitiu a escalabilidade observada.

É interessante observar na figura 4 que o algoritmo **somente-coeficiente-angular** aumentou razoavelmente o número de migrações, mesmo sem conseguir uma boa taxa de sucessos. Já os algoritmos **somente-distância** e **distância-e-coeficiente-angular** aumentaram rapidamente o número de migrações, se valendo do maior alcance do rádio para obter mais sucessos. Entretanto, por volta de 50 e $80m$, respec-

tivamente, os algoritmos estabilizaram as migrações e, conseqüentemente, parte do consumo de energia. Isto se deve ao fato da maior densidade de nós enxergados já não se refletir em ganho representativo de distância percorrida. Este é um bom comportamento destas soluções.

Devido à paridade de resultados entre os dois algoritmos de distância, o puro e o híbrido. Extraímos das simulações um fator de migrações por sucesso. Ele incorpora o custo energético, de tempo e computacional para se obter um sucesso na tarefa do AM. Este fator está na figura 5. Observamos que os poucos sucessos do algoritmo **somente-coeficiente-angular** custaram mais caro que os muitos dos outros dois algoritmos, sobretudo a partir de $20m$ de raio de alcance. Notamos ainda que o algoritmo só de distância começa com um custo mais elevado que o híbrido, só se equiparando a ele a partir de $40m$ de raio de cobertura de rádio.

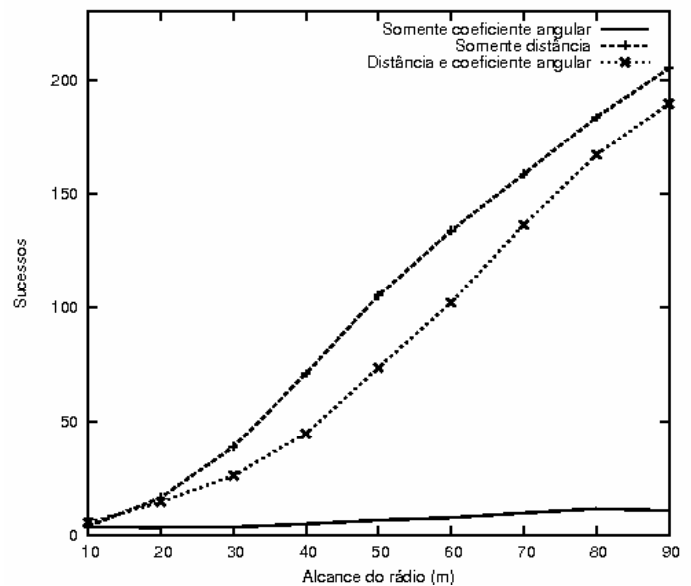


Fig. 3. Análise do alcance da comunicação - Média de sucessos por AM

C. Passo do relógio: esforço computacional

Para que os AM possam executar seus respectivos algoritmos de decisão com uma granularidade de tempo menor (mais passos por segundo), é necessário que todo o esquema seja executado na mesma granularidade. Nos referimos ao algoritmo de localização subjacente, às previsões de localização e direção executadas nos nós e à descoberta e manutenção de vizinhos.

As simulações foram feitas em um cenário com 700 nós distribuídos e movimentados aleatoriamente; velocidade dos nós: distribuição normal entre 1 e $30m/s$; intervalo de repouso: distribuição uniforme entre 0 and $20s$; área de $1.000m \times 1.000m$; tempo de simulação de $1.000s$; precisão do relógio (passo): 0 , 1 e 2 casas decimais (passos $1s$, $0,1s$ e $0,01s$, respectivamente); alcance do rádio de $10m$; erro do algoritmo de localização: 0% ; região de

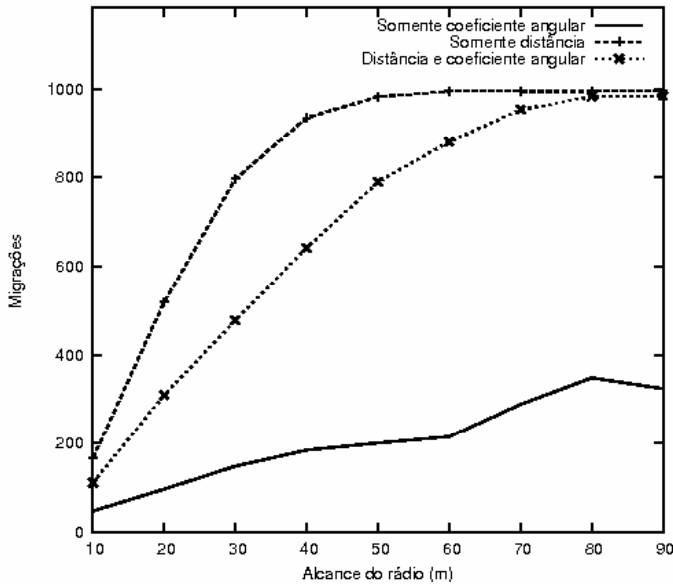


Fig. 4. Análise do alcance da comunicação - Média de migrações por AM

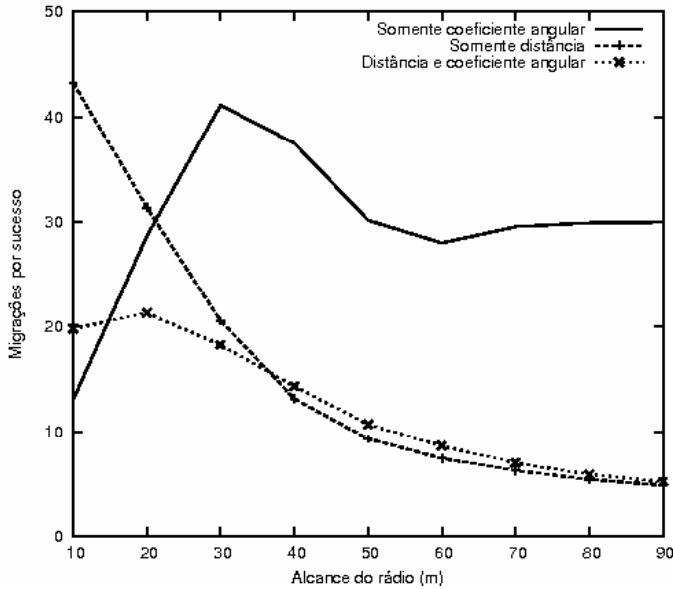


Fig. 5. Análise do alcance da comunicação - Número de migrações para cada sucesso

destino: $A_D(700, 900)$, $B_D(900, 700)$. A_D é o vértice superior esquerdo e B_D é o vértice inferior direito; região de retorno: $A_R(100, 300)$, $B_R(300, 100)$ (mesmos vértices descritos acima); 300 AM são injetados nos 300 primeiros nós, começam a processar no instante 5s e executam até o fim da simulação.

A figura 6 mostra que a redução do passo de 1s para 0,1s gerou um ganho considerável de rendimento para os três algoritmos, sobretudo para o **somente-distância**. Vale lembrar, entretanto que este aumento significa multiplicar por 10 todas as tarefas descritas no primeiro parágrafo desta seção. Já uma nova redução de uma casa decimal, para 0,01s, produziu um

ganho menos significativo.

Na primeira redução, o aumento do número de migrações foi moderado para o algoritmo **somente-coeficiente-angular**, médio para o algoritmo híbrido e elevado para o **somente-distância** (figura 7). Este último se valeu das iterações adicionais para corrigir erros de migração. Na segunda redução, os algoritmos de distância pouco aumentaram ou até diminuíram as migrações, apesar do aumento de sucessos. Já o algoritmo do coeficiente angular puro, teve um aumento exagerado no número de migrações. Esta anomalia já foi isolada. Trata-se de uma oscilação no algoritmo. O AM fica migrando sucessivamente entre dois nós vizinhos, até que eles saiam do alcance um do outro. Esta oscilação ocorre quando a trajetória ótima está entre as trajetórias de dois nós móveis. Estamos estudando uma solução para esta situação.

Na figura 8, confirmamos, através do custo de cada sucesso, o bom desempenho do algoritmo **somente-coeficiente-angular** nestes cenários, para os passos 1s e 0,1s. Para o passo 0,01, a oscilação mencionada se refletiu nas migrações deste algoritmo, mas ainda assim ele permaneceu mais econômico que o **somente-distância**.

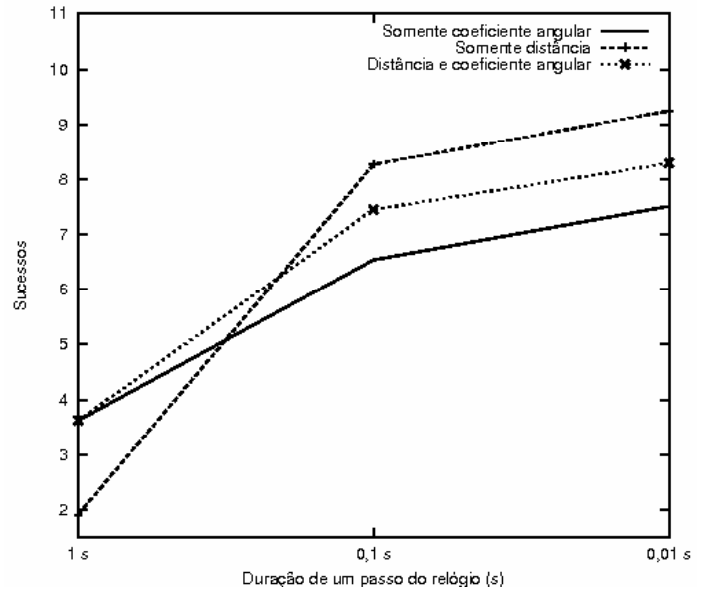


Fig. 6. Granularidade de tempo - Média de sucessos por AM

D. Considerações finais da seção

O algoritmo **somente-distância** tende a ser mais agressivo que os demais e na maioria das vezes atinge uma maior taxa de sucesso. Entretanto, tem maior tendência a efetuar um maior número de migrações. O **somente-coeficiente-angular** se comporta de maneira mais econômica em termos de consumo de energia (migrações), mas não escala em algumas situações. O algoritmo híbrido consegue incorporar as vantagens dos outros dois, mas carrega os problemas de ambos também.

Em ambientes sem mobilidade, apenas a localização (distância) será eficaz, não tendo sentido o conceito de direção do movimento.

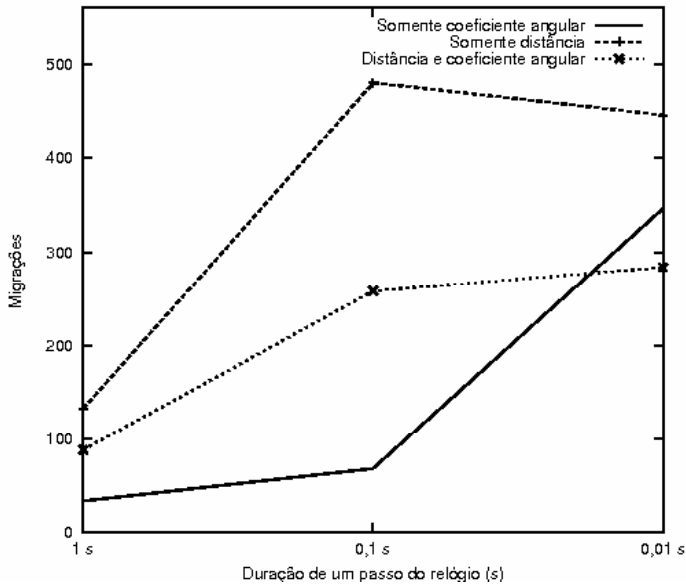


Fig. 7. Granularidade de tempo - Média de migrações por AM

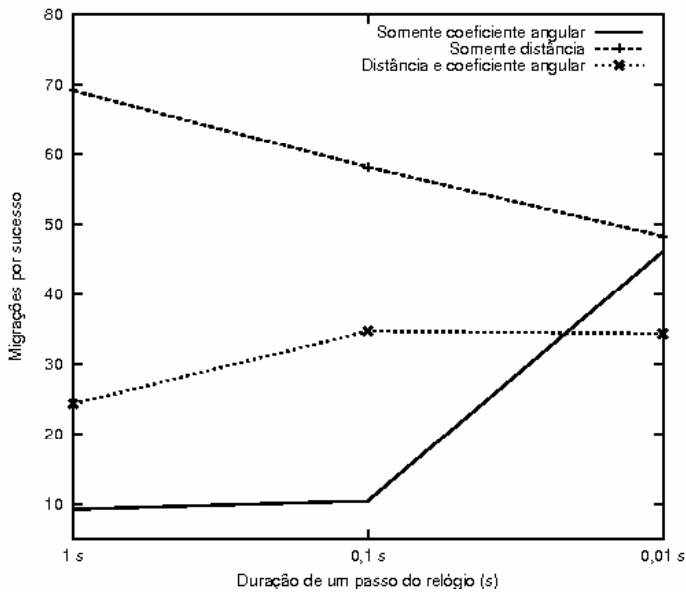


Fig. 8. Granularidade de tempo - Número de migrações para cada sucesso

Não há como generalizar um único algoritmo para todas combinações dos parâmetros aqui analisados. Cada um deles terá melhor desempenho em determinadas situações.

As figuras 9 e 10 ilustram os comentários acima. Elas resumem simulações feitas com os mesmos cenários da seção V-C, com o número de nós variando de 10 a 800. Em linhas gerais, temos os algoritmos baseados em distância e as simulações com passo 0,1s com mais sucessos e mais migrações.

Ainda na figura 9, vemos a execução do algoritmo **somente-distância** sem nenhuma reação positiva ao aumento do número de nós. A combinação de velocidades variando entre 1 e

30m/s com o alcance de rádio em 10m tornou este algoritmo inviável.

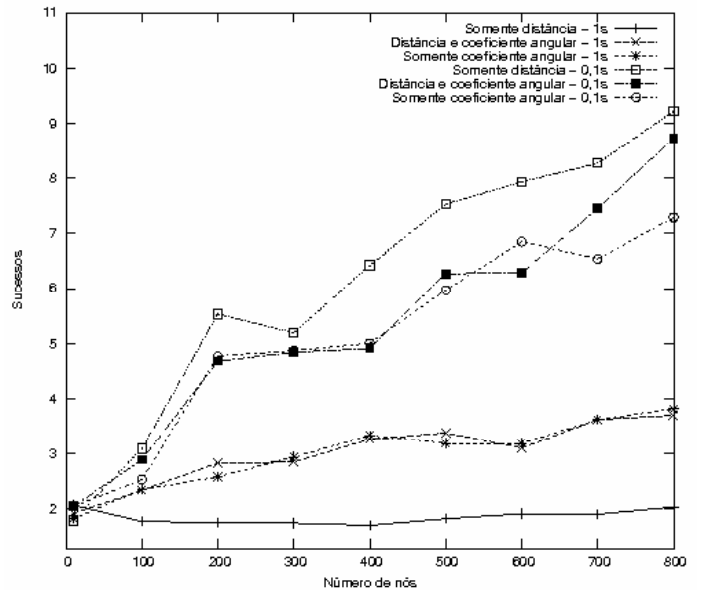


Fig. 9. Média de sucessos por AM

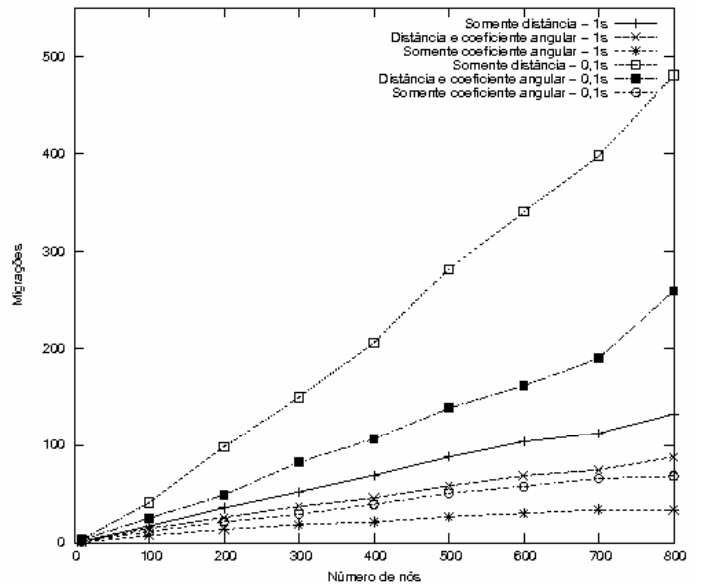


Fig. 10. Média de migrações por AM

VI. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalhos nós analisamos diversos cenários de mobilidade, raio de cobertura de comunicação e número de iterações dos algoritmos. Em conjunto, três algoritmos distintos foram avaliados.

As condições de simulação foram severas, com alto grau de mobilidade e as mudanças bruscas de direção que o *setdest* implementa.

Concluimos que todos os algoritmos são de uso viável e podem prestar serviço de monitoramento em diversos ambientes onde o fenômeno, os nós e os sorvedouros podem ter mobilidade aleatória.

As particularidades que podem surgir em ambientes com mobilidade aleatória, tornam difícil a adoção de um único algoritmo de decisão para os AM. Na verdade, cada algoritmo se adequará a um conjunto de cenários. Observa-se ainda que pode-se optar por um algoritmo para obter um melhor serviço (mais sucessos no cumprimento das tarefas), mas haverá um custo energético adicional, na forma de mais migrações dos AM.

Como trabalhos futuros, pretendemos introduzir na rede formas simplificadas de algoritmos genéticos, que permitam uma adaptação dinâmica dos AM aos perfis da rede, aos moldes das classificações analíticas que mostramos neste trabalho. Será uma escolha ou mudança automática para um dos três algoritmos, de acordo com um melhor rendimento que o mesmo esteja apresentando.

Consideramos importante avaliar o impacto do erro no algoritmo de localização e variações nos parâmetros α e γ , todos já previstos no projeto do simulador.

Por fim, temos a intenção de introduzir um modelo de erro no canal de comunicação do simulador.

REFERENCES

- [1] K. Römer and F. Mattern, "The design space of wireless sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54–61, Dec 2004.
- [2] T. Haenselmann, "Wireless sensor network textbook," http://www.informatik.uni-mannheim.de/~haensel/sn_book, April 2006, retrieved on 2008-08-26.
- [3] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28–36, April 2002.
- [4] L. Hu and D. Evans, "Localization for mobile sensor networks," *MobiCom '04*, pp. 45–57, 2004, aCM.
- [5] J. Antol, P. Calhoun, J. Flick, G. A. Hajos, R. Kolacinski, D. Minton, R. Owens, and J. Parker, "Low cost mars surface exploration: The mars tumbleweed," August 2003, NASA Langley Research Center. NASA/TM-2003-212411.
- [6] F. C. Jabour, E. Giancoli, and A. C. P. Pedroza, "Um esquema em duas camadas para suporte a mobilidade em redes de sensores sem fio," *XXV SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES - SBrT*, 2007.
- [7] F. C. Jabour, E. Giancoli, and A. C. P. Pedroza, "Mobility support for wireless sensor networks," *3rd IEEE European Conference on Smart Sensing and Context (EuroSSC)*, Oct 2008, to appear as poster.
- [8] —, "Mobility support for wireless sensor networks," *INTERNATIONAL CONFERENCE ON COMPUTER AND ELECTRICAL ENGINEERING (ICCEE)*, Dec 2008, to appear.
- [9] M. S. Corson, S. Papademetriou, P. Papadopoulos, V. Park, and A. Qayyum, "An Internet MANET encapsulation protocol (IMEP) specification," *INTERNET-DRAFT*, August 1999, IETF MANET Working Group.
- [10] C. Chevally, R. E. Van, and D. T. A. Hall, "Self-organization protocols for wireless sensor networks," in *In Thirty Sixth Conference on Information Sciences and Systems*, 2002.
- [11] H. Qi, S. S. Iyengar, and K. Chakrabarty, "Multi-resolution data integration using mobile agents in distributed sensor networks," in *IEEE-Systems Man Cybernetics*, vol. 31, n. 3, Ago 2001, pp. 383–390.
- [12] M. Chen, T. Kwon, Y. Yuan, Y. Choi, and V. C. M. Leung, "Mobile agent-based directed diffusion in wireless sensor networks," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, pp. 219–219, 2007.
- [13] C.-L. Fok, G.-C. Roman, and C. Lu, "Rapid development and flexible deployment of adaptive wireless sensor network applications," *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pp. 653–662, 2005.
- [14] "Java," <http://java.sun.com>.
- [15] "ns-2 network simulator," <http://www.isi.edu/nsnam/ns>, 1998.