

Um Novo Protocolo de Roteamento para Redes Móveis Ad hoc

Daniel Câmara* Antônio A.F. Loureiro

Departamento de Ciência da Computação

Universidade Federal de Minas Gerais

Caixa Postal 702

30123-970 Belo Horizonte, MG

Email: {danielc,loureiro}@dcc.ufmg.br

Resumo

A mobile ad hoc network (MANET) is comprised of mobile hosts that can communicate with each other using wireless links. In this paper we present a novel routing algorithm called GPSAL which is based on GPS (Global Positioning System) and mobile software agents modeled on ants for routing in ad hoc networks. We compare our algorithm to the Location-Aided Routing (LAR) [16] algorithm for MANET which is also based on GPS. Simulation results show that our algorithm perform the routing with less overhead than the LAR.

Resumo

Uma rede móvel ad hoc é formada por computadores móveis que podem comunicar entre si através de enlaces de comunicação sem fio. Neste trabalho é proposto um novo algoritmo de roteamento chamado GPSAL, que é baseado em GPS (*Global Positioning System*) e agentes de software móveis modelados como formigas para roteamento em redes móveis ad hoc. O algoritmo proposto é comparado com o algoritmo *Location-Aided Routing* (LAR) [16] para redes móveis que também é baseado em GPS. Resultados de simulação mostram que o algoritmo apresentado neste trabalho realiza o roteamento com um *overhead* menor que o LAR.

1 Introdução

Este trabalho tem como objetivo apresentar um novo algoritmo de roteamento para redes móveis ad hoc chamado GPSAL (*GPS Ant Like Routing Algorithm*). Nos últimos anos tem havido um crescimento extraordinário nas áreas de comunicação celular, redes locais sem fio e serviços via satélite que já permitem que informações e recursos possam ser acessados e utilizados em qualquer lugar e a qualquer momento. Esse novo paradigma,

*Aluno de mestrado em Ciência da Computação da Universidade Federal de Minas Gerais.

formigas para roteamento em redes móveis ad hoc. O algoritmo proposto foi comparado com o algoritmo *Location-Aided Routing* (LAR) [16] para redes móveis que também é baseado em GPS. Resultados de simulação mostraram que o algoritmo apresentado neste trabalho realiza o roteamento com um *overhead* menor que o LAR.

Em todos os experimentos o GPSAL apresentou excelentes resultados e, na comparação com o LAR 1 e 2, um comportamento mais homogêneo. Este tipo de comportamento, mais que desejável, é essencial no momento de se prever e projetar o crescimento ou mudança de equipamentos da rede. É importante salientar que os gráficos estão em escala logarítmica, o que diminui "visualmente" o impacto da diferença entre os números do GPSAL e do LAR. Outro ponto a ressaltar é que as informações nas trocas de tabelas são atualizações que o GPSAL herda dos algoritmos do tipo *link state*, sendo pacotes pequenos, normalmente menores que pacotes típicos de dados.

Referências

- [1] Phil Belanger and Wim Diepstraten. Mac entity: Mac basic access mechanism privacy and access control. Technical report, IEEE 802.11, March 1996.
- [2] Mark Billinghamurst and Thad Starner. Wearable devices: New ways to manage information. *Computer*, 1(1):57-64, January 1999.
- [3] M.S. Corson and A. Ephremides. A distributed routing algorithm for mobile wireless networks. *ACM Journal on Wireless Networks*, 1(1):61-81, 1995.
- [4] S. Corson, S. Batsell, and J. Macker. Architectural considerations for mobile mesh networking. Internet draft RPC, version 2, May 1996.
- [5] S. Corson and J. Macker. Mobile ad hoc networking (*manet*): Routing protocol performance issues and evaluation considerations. IETF RFC 2501, January 1999.
- [6] B. Das, E. Sivakumar, and V. Bhargavan. Routing in ad-hoc networks using a spine. In *IEEE International Conference on Computer Communications and Networks*, 1997.
- [7] R. Dube, D.D. Rais, K. Wang, and S.K. Tripathi. Signal stability based adaptive routing (SSA) for ad hoc mobile networks. *IEEE Personal Communications*, February 1997.
- [8] Greg Ennis. 802.11 architecture. Technical report, IEEE 802.11, March 1996. Tutorial of draft Standard IEEE 802.11.
- [9] Z.J. Haas and M.R. Pearlman. The zone routing protocol (ZRP) for ad hoc networks. Internet-draft, August 1998.
- [10] Zygmunt J. Haas. A new routing protocol for the reconfigurable networks. *ICUPC'97*, October 1997.
- [11] Hewlett-Packard. Gps and precision timing applications. Application Note 1272 Publication Number:5965-2791E, Hewlett-Packard, May 1996.
- [12] R. Hinden and S. Deering. Ip version 6 addressing architecture. IETF RFC 1884, Dec. 1995.
- [13] M. Jiang, J. Li, and Y.-C. Tay. Cluster based routing protocol (CBRP) functional specification. Internet Draft, August 1998.

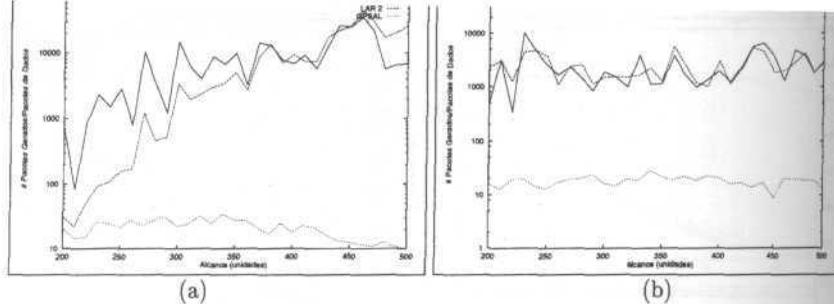


Figura 8: Comparação entre o número de pacotes gerados por pacotes de dados com a variação do alcance, sendo a velocidade média dos nodos em (a) é de 4,5 e em (b) 25 unidades/iteração

O gráfico da figura 9 (a) e da figura 9 (b) apresentam uma comparação entre o LAR e o GPSAL com relação ao número de pacotes gerados por pacotes transmitidos e o número de nodos participando da rede. No LAR, com o aumento do número de nodos, há um aumento do número de comunicações recebidas por nodos, participantes ou não da zona de requisição. O problema se agrava com a velocidade média sendo de 25 unidades/segundo pois a zona de requisição fica maior, envolvendo mais nodos. O GPSAL também apresentou um aumento no tráfego gerado, mas este foi referente principalmente às trocas de tabelas inerentes aos nodos e não à retransmissão dos pacotes de dados iniciais, como no LAR.

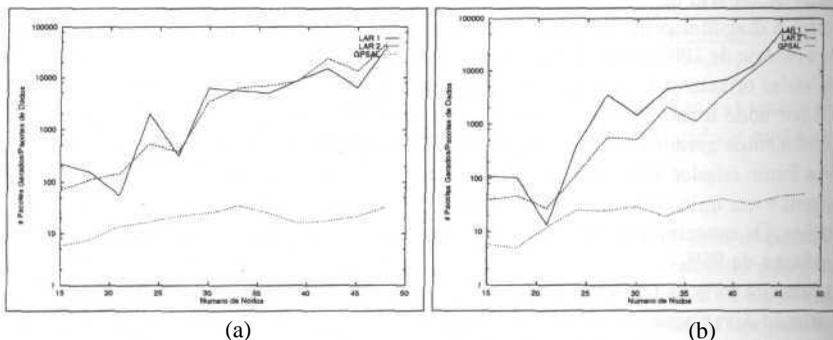


Figura 9: Comparação entre o número de pacotes gerados por pacotes de dados e a variação do número de nodos, para a velocidade média de, em (a), 4,5 unidades/segundo e em (b) de 25 unidades/segundo

5 Conclusões

Este trabalho apresentou um novo algoritmo de roteamento chamado GPSAL, que é baseado em GPS (*Global Positioning System*) e agentes de software móveis modelados como

um aumento significativo da perda de pacotes, como pode ser observado na figura 7 (b)

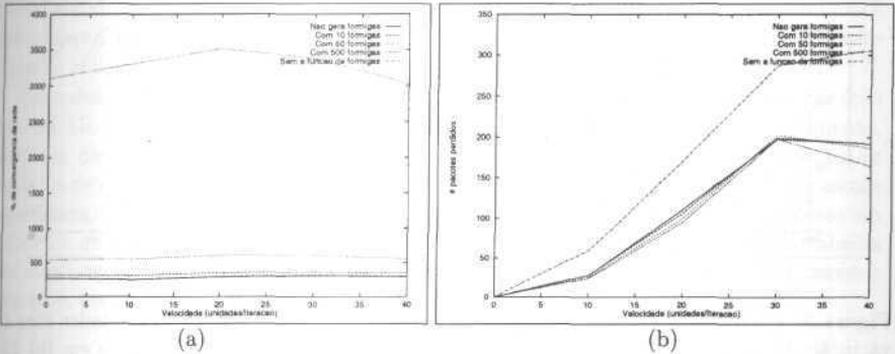


Figura 7: Comparação com relação à velocidade dos nodos. Em (a) tem-se o número de pacotes na rede e em (b) o número de pacotes de dados perdidos

4.4 Comparação com o LAR

O objetivo desta seção é comparar o desempenho do GPSAL com outro algoritmo de roteamento, da mesma classe, para redes móveis ad hoc. O algoritmo escolhido foi o LAR (*Location-Aided Routing*), apresentado na seção 2. O motivo principal desta escolha se deve ao fato do LAR ser um algoritmo geográfico e ter sido publicado recentemente [16].

O modelo de simulação usado foi: alcance de 300 unidades, número de nodos de 30, tamanho do cenário de 1000 x 1000 unidades, e *timeout* para o LAR ser o suficiente para percorrer a diagonal principal mais duas unidades. Por exemplo, com o alcance de 300 sendo o cenário de 1000x 1000, o *timeout* é de 7. A quantidade de pacotes gerados é sempre 10, de várias origens para um mesmo destino e a probabilidade de se gerar uma formiga é de 0,5 por nodo a cada iteração. Estes parâmetros foram escolhidos tomando como base os experimentos gerados para o algoritmo LAR em [16]. Para cada ponto mostrado nos gráficos foram criados 4 cenários de movimentação, onde são escolhidas de forma aleatória 10 origens e um único destino. Cada cenário é executado 32 vezes, sendo um total de 128 execuções. Os experimentos foram analisados com um corte de 5 percentil, e um intervalo de confiança de 95%.

A primeira série de testes tem por objetivo caracterizar o comportamento do algoritmo, com relação à variação de alcance dos nodos. Na figura 8 (a) observamos que o LAR apresentou uma maior quantidade de pacotes na rede, mas os dois algoritmos tiveram um comportamento uniforme. Isto se deve a velocidade baixa dos nodos nestes experimentos. Na figura 8 (b) temos o que parece uma inconsistência com o gráfico anterior, pois a curva do LAR apresenta um aumento e do GPSAL uma diminuição de valores. Deve-se prestar atenção a escala dos gráficos e lembrar que quanto maior a velocidade dos nodos, maior será a *area de flooding* no LAR, o que faz com mais nodos recebam a comunicação. O GPSAL não apresenta este problema pois quanto maior o alcance, mais rápido o pacote irá chegar ao destino, e menos pacotes estarão trafegando na rede. Por este motivo a curva do LAR é ascendente e do GPSAL é descendente no gráfico.

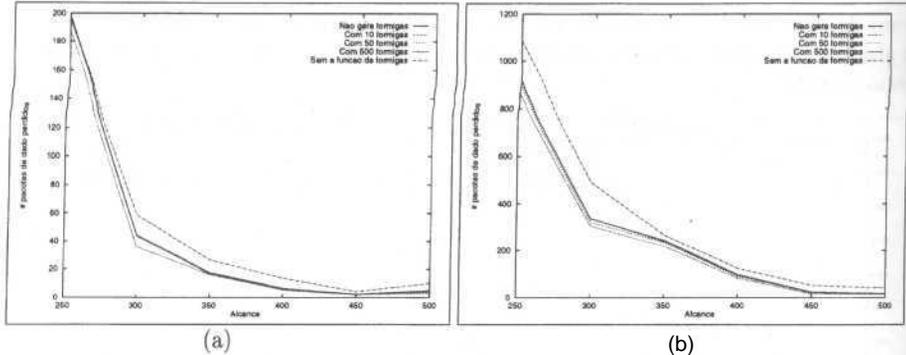


Figura 5: Comparação entre a relação do número de pacotes de dados perdidos e o alcance dos nós. Em (a) tem-se cenários com velocidade média de 5 unidades/iteração e em (b) com velocidade média de 25 unidades/iteração

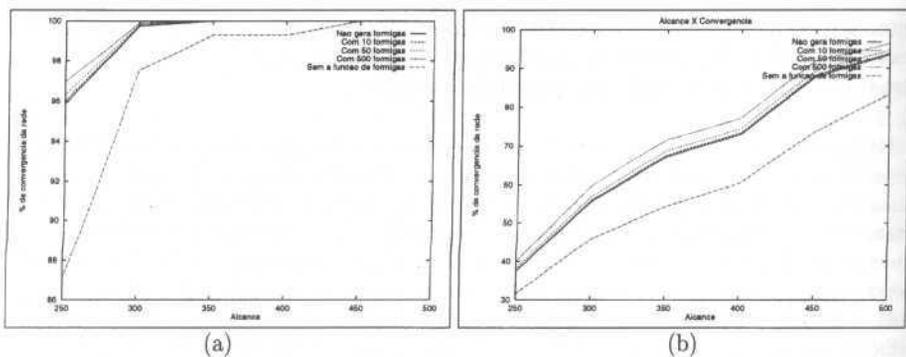


Figura 6: Comparação entre a convergência e o alcance dos nós. Em (a) tem-se cenários com velocidade média de 5 unidades/iteração e em (b) com velocidade média de 25 unidades/iteração

rede. A não ser no experimento onde não havia pacotes de formiga e nem os pacotes de dados agindo como formigas. O comportamento da curva pode ser explicado pelo grande número e tipo de mensagens trafegando na rede. As formigas são pacotes do algoritmo de roteamento, portanto não são consideradas informações úteis, ao contrário dos pacotes de dados. Sendo assim, as mensagens de formiga podem ser descartadas mais facilmente que pacotes de dados. Quando a velocidade é baixa, o número de perda de pacotes é baixo pois os pacotes formiga vão e voltam do seu destino rapidamente, não ficam "vagando" atrás do destino, como é o caso quando se aumenta a velocidade dos nós. A baixa da curva quando há um aumento de mais de 25 unidades/iteração se deve ao descarte de pacotes de formiga. Quanto maior a velocidade, mais difícil é manter as tabelas de roteamento atualizadas, e isto pode levar a rotas incorretas. Para pacotes de dados todos os dispositivos são tentados, mas se o pacote for de formiga, caso o roteamento seja muito caro, esta formiga deve ser consumida pelo nó. Mas mesmo para pacotes de dados há

ponto de equilíbrio. O número de pacotes começa a diminuir porque, com o aumento do alcance, os pacotes são entregues mais rapidamente, diminuindo o volume de pacotes médio na rede.

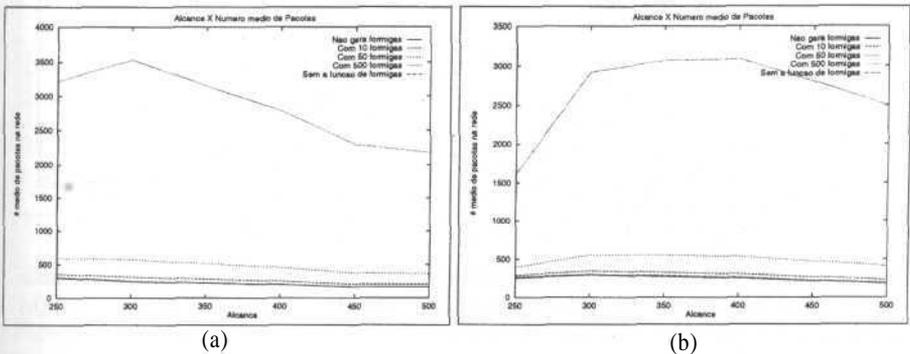


Figura 4: Comparação entre a relação do número médio de pacotes trafegando na rede e o alcance dos nodos. Em (a) tem-se cenários com velocidade média de 5 unidades/iteração e em (b) com velocidade média de 25 unidades/iteração

Na figura 5 os gráficos mostram o número de pacotes de dados perdidos pelo GPSAL com relação a variação do alcance médio dos nodos. O comportamento dos experimentos variando o número de formigas tiveram neste conjunto de testes um comportamento mais homogêneo. Há uma clara diminuição no número de pacotes de dados perdidos com o aumento do alcance. Quando se aumenta o alcance da comunicação, menos nodos são necessários para entregar as mensagens, e mais rápida e precisamente a informação de roteamento caminha pela rede. Melhorando a qualidade das informações de roteamento, melhoram-se as rotas criadas pelo GPSAL e diminui-se a probabilidade do pacote se perder na rede. Pode-se observar ainda que o gráfico da figura 5 (a), com menor velocidade, teve um comportamento melhor que o gráfico da figura 5 (b), com maior velocidade. Novamente a explicação para este comportamento é a qualidade das informações de roteamento. Quanto maior a velocidade, mais complicada e difícil é a atualização dos dados dos nodos.

Os gráficos da figura 6 apresentam experimentos do GPSAL comparando a convergência média dos nodos da rede e o alcance. Como se pode observar o alcance teve uma grande influência no comportamento do algoritmo. Quanto maior o alcance, maior a convergência média da rede. Isto ocorre pelo fato do algoritmo obter informações mais rapidamente e, por consequência, mais precisas. Outro fator determinante na convergência é a velocidade. Como se pode ver no gráfico da figura 6 (a), onde a velocidade é menor, tem-se uma maior convergência que no gráfico da figura 6 (b). Isto é esperado pois com o aumento da velocidade o nodo se desloca mais, piorando assim o conhecimento dos nodos sobre a rede. O comportamento das curvas foi também o esperado: os nodos que tinham mais formigas e por conseguinte mais informações sobre a rede, tiveram uma convergência melhor.

Os gráficos da figura 7 mostram o número médio de pacotes trafegando na rede e o número de pacotes de dados perdidos pelo GPSAL de acordo com a variação da velocidade dos nodos da rede. No gráfico da figura 7 (a), que apresenta o número de pacotes trafegando na rede com o aumento da velocidade dos nodos, pode-se observar que a velocidade dos nodos não apresenta um grande impacto no número de pacotes trafegando na

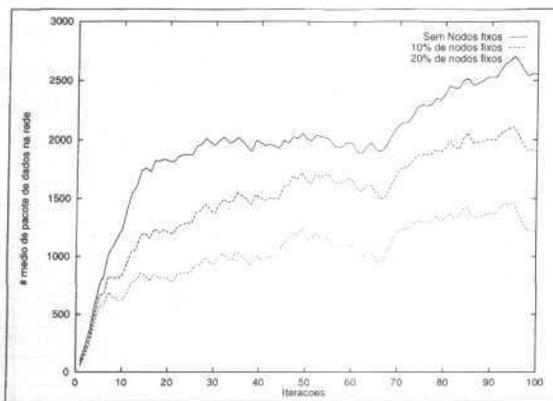


Figura 3: Comparação do *overhead* de pacotes gerados na rede ad hoc com o uso ou não da rede fixa

4.3 Caracterização do GPSAL

Esta seqüência de testes tem como objetivo avaliar o comportamento do GPSAL nas mais diferentes situações. Foram variados aqui o número de formigas na rede, o número de nodos, o alcance e a velocidade. A comparação foi feita com relação ao número médio de pacotes na rede, pacotes de dados perdidos e convergência. O modelo de simulação para os testes, caso não seja explicitamente dito o contrário, são: alcance de 300 *unidades*, número de nodos de 30, tamanho do cenário de 1000 x 1000 *unidades*, o número de pacotes de dados gerados é de 100, sempre de várias origens para vários destinos.

O objetivo destes testes é avaliar o comportamento do GPSAL com relação à variação do alcance dos nodos que participam da rede. Serão analisados o número de pacotes na rede e o número de pacotes de dados perdidos. Além disto serão consideradas duas velocidades médias diferentes, de 5 e 25 unidades/iteração sem nodos fixos na rede. Nos gráficos serão apresentadas cinco cenários: "Não gera formigas" que significa que os cenários *não criam pacotes de formiga utilizando apenas os cabeçalhos dos pacotes de dados como formigas*. Deste modo não é gerado *overhead* algum, pois os pacotes de dados devem ser enviados de qualquer maneira. "Com 10/50/500 formigas", que são os cenários onde são gerados 10/50/500 pacotes de formigas, respectivamente, durante toda a simulação em pontos aleatórios da rede. Finalmente, "sem a função de formiga" onde formigas não são geradas nem as informações dos cabeçalhos dos pacotes de dados são utilizadas, ou seja, nem os pacotes de dados agem como formiga.

Os gráficos da figura 4 mostram o número médio de pacotes na rede durante os experimentos de variação do alcance da comunicação. O comportamento dos experimentos é o esperado, na medida em que os experimentos que geram mais formigas têm um tráfego maior de pacotes. O comportamento da curva da figura 4 (a) se deve ao alcance do nodo, pois quanto maior o alcance, mais rápido o pacote vai ser entregue e menos pacotes vão coexistir na rede. O comportamento da curva do GPSAL sem formigas e sem os pacotes de dados agindo como formigas, na figura 4 (b), pode ser explicado pela perda de pacotes e pela entrega mais rápida de pacotes. Quando o alcance é pequeno, muitos pacotes de formiga são perdidos, diminuindo assim o número de pacotes na rede, até atingir um

52,5%, enquanto que na primeira convergência o ganho foi de 54,63%.

Probabilidade de cada nodo gerar uma formiga por iteração	# de formigas	# de iterações para convergência completa		# de iterações para primeira convergência	
		GPSAL sem formigas	GPSAL com formigas	GPSAL sem formigas	GPSAL com formigas
0,0	0	43,8	29,4	21,6	13,4
0,1	197,4	43,8	27,0	21,6	12,6
0,2	365,6	43,8	25,8	21,6	11,6
0,3	531,2	43,8	25,2	21,6	11,8
0,4	685,0	43,8	24,0	21,6	10,4
0,5	843,8	43,8	24,0	21,6	11,2
0,6	967,6	43,8	22,8	21,6	10,2
0,7	1151,8	43,8	23,4	21,6	10,0
0,8	1255,0	43,8	22,2	21,6	10,6
0,9	1391,0	43,8	21,6	21,6	10,0
1,0	1486,0	43,8	21,0	21,6	9,8

Tabela 1: Convergência da MANET usando o GPSAL com e sem formigas

4.2 O uso da rede fixa

Esta seção tem como objetivo verificar a validade do uso da rede fixa, visando a diminuição do tráfego dentro das redes ad hoc. Com o uso da rede fixa pode-se desviar o tráfego de dentro da rede móvel para a rede fixa, tentando desta forma aumentar a autonomia das unidades móveis e, ao mesmo tempo, melhorar a qualidade do serviço de entrega de mensagens. Nesta simulação levamos em consideração que a rede fixa não tem preocupação com gastos de energia, sendo portanto o custo de transmissão de mensagens na rede fixa inferior ao custo na rede móvel. Além disto, assume-se que a rede fixa é mais confiável e rápida que a rede móvel.

O modelo simulado possui as seguintes características: 150 nodos em uma região quadrada de $N \times N$, alcance de transmissão de $0,3N$ unidades, velocidade média de $0,04N/\text{iteração}$ e máxima de $0,08N/\text{iteração}$. Em cada iteração o nodo tem a probabilidade de $0,5$ de gerar e enviar um pacote a outro nodo qualquer da rede, sendo analisadas 100 iterações. Podemos observar que no período inicial, antes da rede atingir uma certa estabilidade, os nodos da rede fixa não representam uma grande vantagem, mas à medida que o número de pacotes trafegando na rede aumenta, há uma sensível diminuição no número de pacotes trafegando na rede, quando a rede fixa é usada. Sendo assim há uma economia de recursos na rede, de uma forma geral. Outro ponto que pode ser avaliado neste experimento é o número de nodos intermediários. No experimento da figura 3 o tamanho médio dos caminhos é de 3,7275 para 20% de nodos fixos, de 4,6801 para 10% de nodos fixos e de 7,4481 para somente unidades móveis na rede. Esta diminuição no tamanho dos caminhos dos pacotes representa, além de uma economia de recursos da rede, uma melhora significativa na qualidade dos serviços oferecida. Quanto menos *links* móveis houverem na comunicação, menor é a chance de perda de pacotes, de erros de transmissão e mesmo de atraso dos pacotes.

4.1 Influência das Formigas na Convergência da Rede

O tempo de convergência é o tempo necessário para que um nodo tome conhecimento que houve alguma alteração no estado da rede. Quanto menor for este tempo melhor, pois menos pacotes vão ser enviados por um caminho pior, ou mesmo inexistente. Esta métrica se torna mais importante ainda quando falamos de redes móveis, pois além da dinamicidade da rede, existe um alto custo, em termos de gasto de energia, associado a rotas erradas e à *retransmissão* de mensagens.

O modelo usado para simular os algoritmos nesta fase é constituído de um grid de $N \times N$ unidades contendo 70 computadores entre móveis e fixos, sendo que a probabilidade do nodo ser criado como móvel é 90%. A distribuição e movimentação dos computadores é feita de forma aleatória. O alcance máximo de comunicação de uma estação é $O, 2N$ a partir de sua posição. Assume-se que cada nodo tem uma velocidade máxima que é dada pela metade do seu alcance. Esta é uma restrição que tem como objetivo apenas controlar o grau de mobilidade.

A tabela 1 mostra os tempos de convergência total e do primeiro nodo para o GPSAL com formigas e para o GPSAL sem formigas. Neste experimento, criou-se um cenário aleatório de movimentação e envio de pacotes. Em um determinado instante da simulação, a rede é parada e nenhum nodo se movimenta. A partir deste momento conta-se o tempo, em iterações, gasto para que os nodos tenham o conhecimento da posição real de todos os outros nodos da rede. O tempo de convergência do primeiro nodo é o tempo gasto até que o primeiro nodo conheça a posição real de todos os outros da rede. O tempo de convergência total, é o tempo gasto para que todos os nodos tenham o conhecimento da posição real de todos os outros nodos da rede. Para os dois casos o GPSAL com formigas teve um tempo de convergência menor. Pode-se observar na tabela 1 o ganho do GPSAL com formigas em número de iterações necessárias a convergência.

Na tabela 1, o número de iterações para o GPSAL sem formiga não varia, pois é o resultado da execução do algoritmo nos mesmos cenários que o GPSAL com formigas, só que, como não apresenta nenhum componente aleatório, e nem utiliza formigas, o seu valor não se altera com o aumento da probabilidade de gerar formigas. A primeira linha, onde se observa que a probabilidade do nodo gerar formiga é zero para o GPSAL com formigas, significa que apenas os pacotes de dados estão trabalhando como formigas. Mesmo assim, pode-se observar que o ganho foi considerável. Para a convergência total o ganho foi de 32,88% e para a convergência do primeiro nodo o ganho foi de 37,96%. Deve se observar que este ganho foi conseguido sem nenhum custo adicional, pois foram utilizados somente os pacotes de dados que de qualquer forma já teriam que ser transmitidos. A primeira linha que insere pacotes de formiga é a de probabilidade $O, 1$ do nodo gerar uma formiga na iteração. Com apenas aproximadamente 200 formigas geradas, num universo de 6 a 8 mil pacotes de dados gerados em média por cenário, conseguiu-se um ganho de 38,36% para a convergência total e de 41,67% para a primeira convergência. Pode-se observar na tabela 1 que quanto mais formigas se insere na rede, menor é o tempo convergência. Esta taxa de ganho em termos de convergência diminui gradualmente, mas ainda assim apresenta uma melhora. O extremo testado neste experimento foi a probabilidade de 100% do nodo gerar um pacote de formiga a cada iteração. Isto significa que cada nodo, a cada iteração, gerou uma formiga para um destino qualquer. Neste experimento, em que foram gerados 1486 formigas em média, a taxa de convergência geral do GPSAL sem formiga foi de 43,8 e do GPSAL com formiga foi de 21,0. Isto representa um ganho de

4 Experimentos

Nesta seção serão descritos os diversos experimentos realizados com o intuito de caracterizar e avaliar a eficiência do GPSAL. Os dados utilizados nestas comparações foram obtidos através da simulação dos algoritmos GPSAL, LAR1 e LAR2.

Os algoritmos LAR1, LAR2 e o GPSAL foram simulados usando a linguagem C padrão. Os experimentos foram executados em máquinas Sun Sparc Ultra-1, com 128MB de memória RAM, e Sun Sparc Ultra-Enterprise com 512MB de memória RAM, com o sistema operacional SunOS 5.5.1. O simulador foi projetado para tratar os pacotes da mesma forma que o protocolo IEEE 802.11 [1, 8], trabalhando em iterações. A cada iteração, cada nodo trata os pacotes enviados a ele na iteração anterior, e envia os pacotes que têm armazenado no buffer de envio. Desta forma, a camada MAC (*Medium Access Control*) também é simulada. O simulador é todo parametrizado, ao contrário dos simuladores padrão, como por exemplo o NS *Network Simulator* que até a última versão, de dezembro de 1999, não era possível inserir nodos fixos e móveis, ou mesmo criar nodos com alcances variados. No simulador desenvolvido para estes experimentos, além destes dois, diversos outros valores podem ser informados, dando uma maior liberdade e uma maior flexibilidade para a realização dos testes.

O simulador pode gerar cenários ou executar simulações sobre cenários já gerados. Nos arquivos de cenários estão a movimentação dos nodos, todos os pacotes enviados, possíveis variações de alcance e perda de pacotes. Isto garante a fidelidade das informações adquiridas, pois todas as simulações podem ser repetidas fielmente, tanto no LAR como no GPSAL.

Diversos modelos diferentes foram utilizados nas simulações. Isto foi necessário para que se pudesse submeter o algoritmo ao maior conjunto de situações possível. O intuito deste conjunto de variações é testar a eficiência do protocolo nas mais diversas situações. De forma geral, os parâmetros que foram variados nos testes foram a velocidade dos nodos, o alcance da comunicação, a quantidade de nodos da rede, a quantidade de formigas e a percentagem de nodos móveis e fixos. Todos os pontos dos gráficos são o resultado da simulação dos protocolos em de mais de 30 cenários diferentes.

Em todas as simulações as trocas de tabelas no GPSAL ocorrem a cada três iterações. Este espaço curto entre as trocas favorece o LAR, já que as trocas de tabelas são frequentes e têm um alto custo. Com o objetivo de normalizar as comparações, já que o LAR suporta apenas comunicação através de datagramas, tanto a comunicação do GPSAL com formigas quanto a do GPSAL sem formigas serão feitas somente através de datagramas.

Não foi utilizada nenhuma distribuição para definir a posição dos nodos ou a forma como seria o fluxo de dados entre nodos da rede. Optou-se por não se usar nenhum padrão de mobilidade conforme recomendação da RFC 2501 [5], que trata de características de MANETS, e diz que os nodos são livres para se moverem livremente pela rede. Mesmo que nas simulações tenha-se atribuído uma certa "inércia" ao nodo, que é uma tendência de continuar no mesmo sentido de movimentação, nenhum padrão de movimentação foi adotado.

As seções a seguir serão organizadas de acordo com a variação de cenários e objetivos que motivaram aquele determinado conjunto de experimentos.

net. Como o serviço é não confiável, o nodo destino não envia necessariamente nenhuma confirmação de recebimento do datagrama.

O nodo origem, ao enviar um pacote, sugere uma rota baseado nas informações obtidas na sua tabela de roteamento e envia a mensagem para o primeiro nodo do caminho. Caso não consiga enviar o pacote, tenta uma nova rota. O nodo intermediário, ao receber um datagrama, atualiza tanto as informações da sua tabela, quanto as informações no pacote. O nodo verifica a rota para o destino e, se necessário, a altera no datagrama passando a nova sugestão de rota para o próximo nodo. Na verdade, para o datagrama existe apenas a preocupação com a informação local sendo que o custo do redirecionamento só se justifica se o pacote não estiver se deslocando em direção ao destino.

3.4 Formigas

As formigas são agentes enviados para nodos aleatórios da rede. O objetivo principal desses agentes é coletar e disseminar informações sobre localização dos nodos. As formigas armazenam informações sobre os *hops* da rota que estão percorrendo. Isto possibilita aos nodos por onde a formiga passa obter informações novas sobre nodos distantes na rede. Esta informação é particularmente interessante no momento da troca de tabelas, pois os nodos vizinhos estarão aptos a trocar informações sobre nodos distantes, melhorando desta forma a qualidade das trocas de tabelas de roteamento. Pacotes datagrama trabalham de certo modo como formiga, à medida que os nodos intermediários aprendem com a informação que está no seu cabeçalho. Naturalmente, existe um *overhead* associado a este processo.

3.5 Métricas para redirecionamento de rotas

Uma das principais características do GPSAL é o redirecionamento de rotas por nodos intermediários que fornece um bom desempenho do algoritmo e, assim, deve ocorrer com *cuidado*. *Os principais pontos a serem observados e* considerados no momento de se optar por uma nova rota são: a existência de um caminho "provavelmente" melhor que o atual, a possibilidade de uso da rede fixa e se o nodo possui informações sobre movimentação do nodo destino.

O caminho melhor que o atual é certamente o ponto mais genérico e abstrato, sendo que vários parâmetros podem, e devem, ser observados quando consideramos um novo caminho. Por exemplo caso o host móvel tenha informações de que existe um caminho menor, ele deve alterar o atual.

O uso da rede fixa é sempre interessante, pois o que é levado em consideração aqui, é que o custo para a transmissão de dados através da rede fixa é normalmente menor que na rede móvel, pois a velocidade é maior, a quantidade de erros bem menor [23] e não se tem uma preocupação tão grande com o gasto de energia. Sempre que possível os nodos devem tentar desviar o tráfego da rede *ad hoc* para a rede fixa. Assim, diminui-se o gasto de recursos dos nodos intermediários, aumentando a autonomia da rede, pois mais recursos estarão disponíveis. Outra consideração que é feita é que na rede fixa estarão sendo executados protocolos como o IP-móvel [21] ou o IPv6 [22, 12] que sabem encontrar o nodo fixo na rede.

A tabela de rotas contém o endereço de cada nodo, a posição (X,Y), a posição anterior conhecida, velocidade, o *timestamp* da informação, *timestamp* da posição anterior, e se o computador é móvel ou fixo. A cada troca de tabelas, todas estas informações são enviadas entre os nodos.

O algoritmo de roteamento é independente dos níveis inferiores. O endereçamento pode ser tanto IP quanto IPX ou outro qualquer. A única restrição quanto à forma como é feito o endereçamento, é que cada nodo da rede deve possuir um identificador único. Nos cabeçalhos dos pacotes iremos considerar os endereços como sendo de 32 bits, o mesmo tamanho do endereço no protocolo IPv4.

Cada vez que o nodo repassa informações a seu respeito, junto com estas informações é colocado um *timestamp*. Isto garante, aos outros nodos, que é sempre possível identificar se uma informação, sobre um determinado nodo, é mais nova ou mais antiga que outra.

Normalmente a velocidade da comunicação via rádio, e de processamento das mensagens, é superior à velocidade de movimentação dos hosts móveis.

3.2 Funcionamento Básico

Caso um computador móvel deseje enviar uma informação para outro nodo, ele verifica em sua tabela a última posição conhecida do destino. Se esta informação estiver na tabela, determina o melhor caminho baseado nas informações de posicionamento que tem. O caminho é colocado na mensagem juntamente com as informações referentes à posição dos nodos intermediários e seus respectivos *timestamps*. Em seguida a mensagem é enviada para o primeiro nodo da lista. Se a tabela de rotas não tiver informações sobre o nodo destino, o computador móvel envia uma mensagem para o computador fixo mais próximo. Este, caso exista, fica encarregado de encontrar o destino, caso contrário o destino é considerado não alcançável.

As informações sobre os nodos são um retrato de um tempo anterior e podem não retratar a situação atual. Devido a isto cada nodo deve atualizar a sua tabela de acordo com as informações contidas nos pacotes, se estas forem mais atuais que as de sua tabela de roteamento. Cada nodo, ao receber uma mensagem deve sempre atualizar as informações referentes a ele mesmo garantindo que os nodos posteriores terão informações recentes sobre os nodos por onde o pacote já passou. Feito isto, o nodo verifica se é capaz de encontrar alguma rota melhor que a atual e, se conseguir, faz o redirecionamento do pacote para esta nova rota. O nodo intermediário pode refazer a rota a partir dele, mas não deve alterar o caminho por onde o pacote já passou. Este processo é repetido em todos os nodos até que o pacote alcance o destino.

Um novo host móvel, ao conectar-se à rede ad hoc, ouve o meio até que encontre um nodo vizinho. Assim que encontrar, envia um "pedido de tabela" e recebe em troca a tabela completa do vizinho. Desta forma, o HM (*host móvel*) tem acesso à posição de todos os nodos da rede móvel. Esta é a única vez que toda a tabela é transmitida. A partir deste momento, o nodo só receberá, e enviará, periodicamente informações de atualização relativas à esta tabela.

3.3 Datagramas

O GPSAL prove a entrega de dados através de datagrama não confiável. Este serviço é semelhante ao oferecido pelo protocolo UDP (*User Datagram Protocol*) do modelo Inter-

o nó ou quanto o nó D se repassa a comunicação estão mais próximos do destino que o nó anterior. A distância entre C e D ($d(CD)$) é menor que a distância entre O e D, e a distância entre I e D ($d(ID)$) por sua vez é menor que $d(CB)$. Isso não ocorre com o nó B, por exemplo, onde a distância entre B e D ($d(BD)$) é maior que $d(CB)$. Deste modo o nó B não repassa a informação.

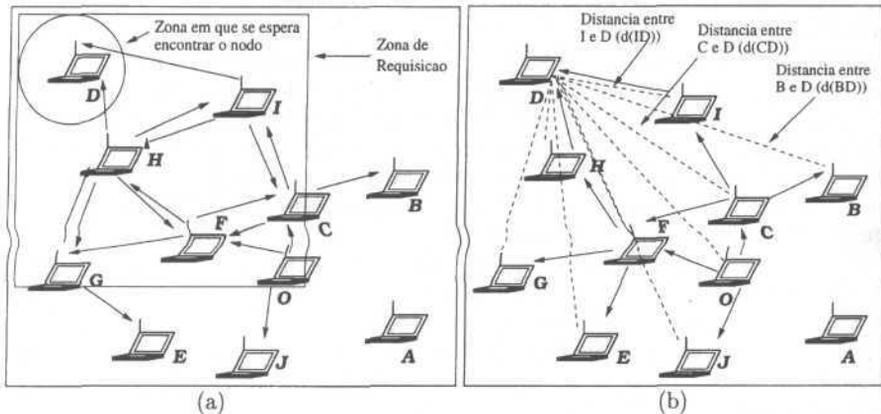


Figura 2: A forma de funcionamento do LAR1 na entrega de pacotes do nó O para o nó D

3 O Algoritmo GPSAL

O algoritmo GPSAL (*GPS Ant Like Routing Algorithm*), proposto neste Trabalho, também utiliza informações de localização dos nós para fazer o roteamento. Estas informações podem ser adquiridas, a qualquer momento, através de um dispositivo de GPS. Outra característica marcante do protocolo GPSAL é a utilização de agentes modelados como formiga para coletar e disseminar as informações sobre localização dos nós. Isso permite que os nós da rede possuam informações mais precisas sobre os outros nós.

3.1 Algumas Considerações

Como considerações iniciais, necessárias ao funcionamento do algoritmo, devemos observar que cada computador móvel deve ser dotado de uma unidade de GPS capaz de fornecer a posição geográfica do nó sempre que solicitado. O GPS fornece informações tridimensionais [11], mas por questão de simplicidade consideraremos somente as informações de latitude e longitude (X, Y), ignorando a altitude do nó (Z). Esta simplificação tem o objetivo de simplificar a explicação e o entendimento do algoritmo, mas não há perda de generalidade, pois a consideração da altitude tende a melhorar ainda mais a acurácia do GPSAL.

A comunicação entre dois nós vizinhos é feita pelo protocolo da camada MAC (*Medium Access Control*), que garante a entrega e a integridade dos pacotes entre os nós intermediários.

2 Trabalhos Relacionados

Esta seção apresenta as características e o funcionamento básico de alguns algoritmos de roteamento para redes ad hoc.

O *Dynamic Source Routing* (DSR) [14] é um algoritmo do tipo *source routing*, ou seja o nodo origem determina toda a seqüência dos nodos por onde passará o pacote. O DSR não envia mensagens periódicas para trocar informações de roteamento. Isto reflete em economia de banda de rede e, conseqüentemente, de energia da bateria. O DSR faz duas considerações básicas: a primeira é que a velocidade dos nodos é moderada se comparada com a velocidade de transmissão. A segunda consideração é que o diâmetro da rede é pequeno, estando entre cinco e dez nodos [15].

O *Zone Routing Protocol* (ZRP) [9, 10] é baseado em zonas de roteamento definidas a partir de cada nodo. Para cada *host* é exigido que ele conheça apenas as informações referentes à sua zona de roteamento. Assim as informações de roteamento podem ser propagadas apenas localmente. Para o roteamento dentro da zona do nodo, qualquer algoritmo de roteamento pode ser utilizado e no roteamento inter-zonas é utilizado o *on-demand*.

O *Location-Aided Routing* (LAR) [16] usa informações de localização obtidas através de GPS (*Global Positioning System*) para descobrir uma rota para um dado destino, ou seja, o roteamento é baseado na posição geográfica. Usando essa informação é possível restringir a área a ser pesquisada ao se determinar uma nova rota. Essa área é definida em função da provável localização do nodo destino no momento da descoberta da rota. No LAR a requisição de informações sobre a localização do nodo destino é feita através de um *flooding* puro pela rede. O destino ao receber esta requisição envia de volta ao nodo origem a sua informação de localização. Em [16] é proposto duas variantes do LAR. A primeira chamada de LAR1 e uma segunda chamada de LAR2.

O LAR1 se utiliza de uma zona de requisição de formato retangular que deve conter o nodo destino. Esta zona de requisição deve ter um tamanho tal que inclua o nodo origem e toda a área onde se espera que esteja o nodo destino. O tamanho dessa área é proporcional a velocidade de movimento do nodo destino e ao tempo decorrido desde o registro da última localização do destino. No LAR1 cada nodo, ao receber um pacote, verifica se está dentro da zona de requisição ou não. Se estiver, reenvia novamente a mensagem para seus vizinhos, caso contrário ignora a mensagem. O pacote é repassado de nodo a nodo até que chegue ao destino ou até que seu TTL expire.

No LAR2, a mensagem contém duas informações que são utilizadas na escolha de uma zona: as coordenadas (x, y) do nodo destino e uma estimativa d de quão longe esse nodo pode estar dessas coordenadas. Essas informações são usadas pelos nodos intermediários para determinar a zona mais provável de se encontrar o destino. Cada nodo ao receber uma mensagem, nesta abordagem, verifica se está mais próximo do destino que o nodo de onde recebeu a mensagem. Caso esteja, faz um *broadcast* deste pacote para seus vizinhos, caso contrário simplesmente ignora o pacote.

A figura 2 (a) mostra o funcionamento do LAR1 e a forma como as mensagens são entregues. Os nodos que estão dentro da zona de requisição fazem *flooding* dos pacotes de dados, até que o TTL dos pacotes expire. Os nodos de fora da região simplesmente ignoram os pacotes. A figura 2 (b) mostra como uma comunicação se procede no LAR2. Os nodos só repassam o pacote caso estejam mais próximos do destino que o nodo que lhe enviou o pacote. No caso da comunicação entre O e D pelo caminho (O, C, I, D) tanto

chamado de computação móvel, tem como objetivo permitir que os usuários tenham acesso a serviços independente de onde estão localizados e, o mais importante, de mudanças de localização, ou seja, proporcionando aos usuários mobilidade. Dessa forma, a computação móvel amplia o conceito tradicional de computação distribuída. Isso é possível graças à comunicação sem fio que elimina a necessidade do usuário manter-se conectado à uma infra-estrutura fixa e, em geral, estática. Duas são as estruturas propostas de redes móveis para prover este acesso [1, 8]: redes móveis estruturadas e não estruturadas. Nas redes móveis estruturadas as estações móveis trocam mensagens com uma estação rádio base (ERB), que se encarrega de entregar as mensagens aos destinos. Essa é basicamente a arquitetura utilizada na telefonia celular. O segundo tipo de rede móvel é a não estruturada, também chamada de rede móvel ad hoc (*MANET-Mobile Ad hoc NETWORK*), onde os dispositivos computacionais são capazes de trocar informações diretamente entre si, conforme mostrado na figura 1.

Cenários onde redes móveis ad hoc podem ser utilizadas são por exemplo, em situações de desastre, como furacões, terremotos ou inundações, onde equipes de resgate precisam se coordenar e não se tem uma rede fixa disponível. Soldados num campo de batalha trocando informações táticas, empresários compartilhando informações numa reunião ou estudantes usando *laptops* para participar de uma aula interativa são também exemplos de situações onde redes ad hoc podem ser necessárias. Outra aplicação onde MANETs deverão ter no futuro um papel importante é na interconexão de *wearable computers* [2].

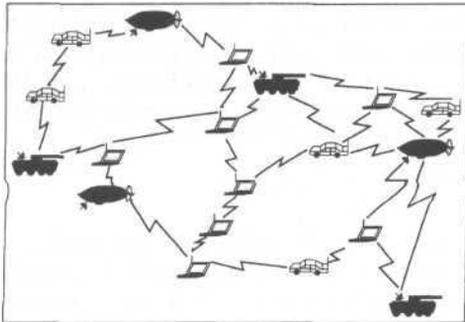


Figura 1: Modelo de comunicação em redes móveis ad hoc.

Numa MANET uma rota entre dois computadores pode ser formada por vários *hops* através de um ou mais computadores na rede, como visto na figura 1. Um dos problemas fundamentais numa rede ad hoc é determinar e manter as rotas, já que a mobilidade de um computador pode causar mudanças na topologia. Vários algoritmos de roteamento para redes ad hoc foram propostos na literatura [3, 4, 6, 7, 9, 13, 15, 16, 17, 18, 19, 20, 24]. Estes algoritmos diferem na forma em que novas rotas são determinadas e como as existentes são modificadas, quando necessário.

Este trabalho está organizado da seguinte forma. A seção 2 descreve alguns dos principais algoritmos de roteamento para redes móveis ad hoc. A seção 3 descreve o algoritmo GPSAL. A seção 4 descreve os experimentos feitos para avaliação do protocolo de roteamento GPSAL e, finalmente, a seção 5 apresenta as conclusões do trabalho.

- [14] D. Johnson, D.A. Maltz, and J. Broch. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, 1996.
- [15] D. Johnson, D.A. Maltz, and J. Broch. The dynamic source routing protocol for mobile ad hoc networks. Internet Draft, March 1998.
- [16] Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 66-75, Dallas, Texas, USA, October 25-30 1998.
- [17] P. Krishna, M. Chatterjee, N.H. Vaidya, and D.K. Pradhan. A cluster-based approach for routing in ad hoc networks. In *USENIX Symposium on Location Independent and Mobile Computing*, April 1995.
- [18] V.D. Park and M.S. Corson. Temporally-ordered routing algorithm (TORA) version 1 functional specification. Internet Draft, August 1998.
- [19] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM Symposium on Communication, Architectures and Protocols*, 1994.
- [20] C.E. Perkins and E.M. Royer. Ad hoc on demand distance vector (AODV) routing. Internet Draft, August 1998.
- [21] Charles E. Perkins. Ip mobility support. IETF RFC 2002, Oct. 1996.
- [22] Charles E. Perkins and David B. Johnson. Mobility support in ipv6. In *Proceedings of the Second Annual International Conference on Mobile Computing and Networking (MobiCom'96)*. ACM/IEEE, November 1996.
- [23] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, third edition edition, 1996.
- [24] C.-K. Toh. A novel distributed routing protocol to support ad-hoc mobile computing. In *IEEE 15th Annual Int. Phoenix Conf. Comp. and Commun.*, March 1996.