

Algoritmo de Roteamento para Redes Móveis Ad hoc Baseado em GPS*

Daniel Câmara Antonio A. F. Loureiro Rodrigo Barro Almeida

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Caixa Postal 702
30123-970 Belo Horizonte, MG
{danielc,loureiro,barra}@dcc.ufmg.br

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) [14] para redes móveis que também é baseado em GPS. Resultados de simulação mostram que o algoritmo apresentado neste trabalho tem uma taxa de convergência melhor que o LAR e um *overhead* menor.

Palavras chave: Redes de computadores

Abstract

A mobile ad hoc network 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) [14] algorithm for MANET which is also based on GPS. Simulation results show that our algorithm has a better performance than LAR with less overhead.

Keyword: Computer Network

1 Introdução

O crescimento extraordinário que tem ocorrido nesta década nas áreas de comunicação celular, redes locais sem fio e serviços via satélite permitirá que informações e recursos possam ser acessados e utilizados em qualquer lugar e em qualquer momento. Dado o atual crescimento do segmento de computadores pessoais portáteis e PDAs (*Personal Digital Assistants*), estima-se que em poucos anos, dezenas de milhões de pessoas terão um *laptop*, *palmtop* ou algum tipo de PDA. Independente do tipo de dispositivo portátil, a maior parte desses equipamentos deverá ter capacidade de se comunicar com a parte fixa da rede e, possivelmente, com outros computadores móveis. A esse ambiente de computação se dá o nome de computação móvel ou computação nômade. Nesse ambiente, o dispositivo computacional não precisa ter uma posição fixa na rede. Neste trabalho, o termo computador móvel será usado para referenciar genericamente um computador portátil ou um PDA.

*Este trabalho foi parcialmente financiado pelo projeto MCT/FINEP/PRONEX número 76.97.1016.00.

Esse novo paradigma permite que usuários desse ambiente tenham acesso a serviços independente de onde estão localizados e, o mais importante, de mudanças de localização, ou seja, 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.

Um tipo importante de rede móvel é a rede ad hoc (MANET—*Mobile Ad hoc NETwork*), onde os dispositivos computacionais são capazes de trocar informações diretamente entre si, conforme mostrado na figura 1. Nessa rede também é possível ter acesso a computadores na rede fixa, dependendo do tipo de infra-estrutura disponível. Se existir um canal de comunicação entre as duas redes pode-se assumir que a rede fixa será capaz de rotear mensagens para a rede móvel usando um protocolo como o IP móvel [19].

Alguns cenários onde uma rede móvel ad hoc pode ser usada são por exemplo, equipes de resgate que precisam se coordenar numa situação de desastre como furacão, terremoto ou inundação, soldados num campo de batalha trocando informações táticas, empresários compartilhando informações numa reunião e estudantes usando *laptops* para participar numa aula interativa. Outra aplicação onde uma MANET terá um papel importante é a interconexão de *wearable computers* [1].

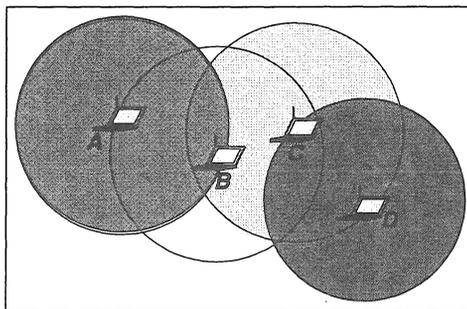


Figura 1: Exemplo de uma comunicação entre os computadores A e D em uma rede ad hoc. Os círculos demonstram o alcance da comunicação das unidades móveis, deste modo as mensagens devem passar, neste caso, pelos nodos B e C para chegar em D

Numa MANET uma rota entre dois computadores pode ser formada por vários *hops* através de um ou mais computadores na rede. Um problema fundamental numa rede ad hoc é determinar e manter 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, dentre eles destacam-se [3, 4, 5, 7, 8, 12, 13, 14, 15, 18, 20, 21, 26]. Estes algoritmos diferem na forma em que novas rotas são determinadas e como as existentes são modificadas.

A grande maioria desses algoritmos possuem duas características em comum. A primeira é o uso de *flooding* para enviar uma mensagem quando um computador não consegue descobrir uma rota para o nodo destino. Isso diminui drasticamente a eficiência e escalabilidade desses protocolos. A segunda característica é que não levam em consideração a localização física dos nodos.

Neste trabalho é proposto um novo algoritmo de roteamento baseado em GPS (*Global Positioning System*) [6, 11, 16] e agentes de software formiga [2]. Este algoritmo é chamado GPSAL. O algoritmo assume que é possível ter acesso à rede fixa para, eventualmente, enviar mensagens para um computador destino e evitar, sempre que possível, o uso da rede móvel. Isto é uma melhoria e não um requisito, já que no caso de não se ter acesso à rede fixa podemos usar apenas a rede móvel. Agentes formiga têm o papel de coletar e disseminar informações sobre localização dos nodos. Este é um ponto importante no desempenho do algoritmo. Todos os mecanismos aqui descritos têm a função de evitar o uso de *flooding*,

já que este degrada o desempenho dos algoritmos e reduz sua escalabilidade como será visto na seção 5. A idéia central é que se um nodo não for encontrado através do GPSAL não o será também com *flooding*, pois deve estar desligado ou inacessível.

Este algoritmo é comparado através de simulações com o algoritmo LAR (*Location-Aided Routing*) [14]. O LAR foi escolhido por ser um algoritmo de roteamento para redes ad hoc que também é baseado em GPS. Os resultados obtidos mostram que o GPSAL tem um *overhead* bem menor que o LAR.

Este trabalho está organizado como descrito a seguir. A seção 2 discute os trabalhos relacionados. São descritos vários algoritmos de roteamento para MANETs e, em particular, o algoritmo LAR que é usado para comparação neste trabalho. A seção 3 apresenta o funcionamento do algoritmo GPSAL. Como será visto, o algoritmo é baseado em conceitos básicos como GPS, agente formiga, propagação e atualização de informações. A seção 4 discute diferentes métricas para redirecionamento de rotas que podem ser aplicadas para melhoria da comunicação entre os nodos. A seção 5 apresenta os resultados de simulações feitos para o GPSAL. Finalmente a seção 6 apresenta as conclusões do trabalho.

2 Trabalhos Relacionados

O projeto de algoritmos de roteamento é um problema fundamental em redes móveis ad hoc [4, 24]. Vários são os algoritmos propostos na literatura, entre eles destacam-se [3, 4, 5, 7, 8, 12, 13, 14, 15, 18, 20, 21, 26]. Uma propriedade desejável que um algoritmo de roteamento deve ter é a capacidade de se adaptar a diferentes padrões de tráfego.

Johnson et al. [13] argumentam que os protocolos de roteamento existentes são inadequados para redes ad hoc, já que o tráfego relacionado com o roteamento pode ser uma fração significativa da largura de banda disponível, principalmente em protocolos que atualizam suas tabelas periodicamente. Os autores propõem o algoritmo DSR (*Dynamic Source Routing*) que é baseado em descoberta de rota sob demanda. Várias otimizações foram propostas para reduzir o *overhead* associado à descoberta da rota. Perkins e Royer [21] apresentam o protocolo AODV (*Ad hoc On Demand Distance vector routing*) que também usa um procedimento de estabelecimento de rota sob demanda.

Park e Corson [18] propõem o algoritmo TORA (*Temporally-Ordered Routing Algorithm*) que tem como objetivo minimizar o impacto das mudanças topológicas da rede nas atualizações das tabelas de rotas. Isso é feito restringindo as mensagens relacionadas com o roteamento a um pequeno conjunto de nodos que esteja perto de onde houve a mudança.

Hass e Pearlman [8] tentam combinar as abordagens pró-ativa e reativa no algoritmo ZRP (*Zone Routing Protocol*). A descoberta da rota é feita sob demanda, mas limitada a vizinhança do nodo inicial. Informações sobre mudanças na topologia ficam restritas à vizinhança de onde houve a mudança.

Ko e Vaidya [14] propõem o algoritmo LAR (*Location-Aided Routing*) que usa informações de localização para descobrir uma rota para um dado destino. 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 de descobrir a rota.

O algoritmo GPSAL proposto neste trabalho também usa informações de localização dos nodos para fazer o roteamento. Além disso utiliza agentes formiga para coletar e disseminar informações sobre localização dos nodos. Isso permite que os nodos da rede tenham informações mais precisas sobre os outros nodos. Na seção 5.2 será discutido mais profundamente o funcionamento do LAR. A seguir, será descrito o algoritmo GPSAL.

3 O Algoritmo GPSAL

Nesta seção é descrita a forma básica de funcionamento e as principais características do GPSAL.

3.1 Algumas Considerações

Como considerações iniciais necessárias ao funcionamento do algoritmo devemos observar que cada computador móvel deve ter uma unidade de GPS, capaz de fornecer a posição geográfica do nodo sempre que solicitado. O GPS fornece um posicionamento tridimensional [9], mas por questão de simplicidade consideraremos somente as informações de latitude e longitude (X, Y), ignorando a altitude do nodo (Z). A comunicação entre dois nodos vizinhos é feita pelo protocolo da camada MAC (*Medium Access Control*).

A tabela de rotas contém o endereço de cada nodo, posição (X,Y), a posição anterior conhecida, velocidade, período de validade (TTL-*Time To Live*) da informação, TTL da posição anterior, e se o computador é móvel ou fixo.

3.2 Funcionamento Básico

Quando um computador móvel deseja 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 a posição dos nodos intermediários e seus respectivos TTL. Em seguida a mensagem é enviada para o primeiro nodo da lista. Caso a tabela de rotas não tenha informações sobre o nodo destino, o computador móvel envia uma mensagem para o computador fixo mais próximo que fica encarregado de encontrar o destino.

As informações sobre os nodos são um retrato de um tempo anterior e podem não retratar a situação atual. Por esta razão cada nodo atualiza a sua tabela de acordo com as informações contidas no pacote, caso as informações recebidas sejam mais recentes que as suas. No pacote são sempre atualizadas as informações referentes ao nodo atual. Feito isso o nodo verifica se existe uma rota melhor e, se for o caso, faz o redirecionamento dos nodos. O nodo intermediário pode refazer a rota a partir dele, mas nunca antes. Este processo é repetido até o pacote alcançar o destino.

A figura 2 ilustra dois possíveis momentos de uma rede onde se deseja enviar uma mensagem do computador A para o G. O algoritmo de roteamento deve sempre escolher a melhor rota no momento do envio da mensagem.

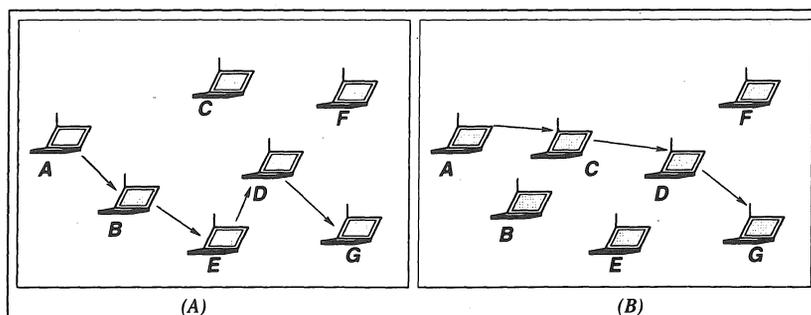


Figura 2: O computador A deseja enviar uma mensagem para o computador G e a rota escolhida é mostrada no cenário da esquerda. Com a mudança de localização do computador C, conforme mostrado no cenário da direita, deve ser escolhida uma nova rota para G, agora passando por C já que este novo caminho é melhor que o antigo.

3.3 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 à medida em que vão passando pelos diversos *hops* de uma rota. Isso permite que os computadores da rede tenham informações mais precisas sobre os outros nodos. Naturalmente existe um *overhead* associado a este processo, assim como podemos ver no gráfico da figura 4. Este *overhead* pode ser sentido através do número de formigas trafegando na rede.

Um ponto interessante é como determinar um destino para o qual deve-se enviar uma formiga a fim de coletar e disseminar informações sobre as localizações dos nodos. O destino não precisa ser um computador com quem o nodo mantém contato ou mesmo com quem estabelecerá uma comunicação num curto espaço de tempo. Uma possibilidade é escolher o nodo da tabela de rotas que está com a informação mais antiga ou o nodo cuja informação indique que esteja mais distante geograficamente.

Um nodo, ao enviar uma formiga para um destino, cria uma rota que não precisa ser necessariamente ótima. Quando um computador móvel recebe uma formiga, atualiza sua tabela de rotas, modifica na mensagem recebida as informações que estejam desatualizadas a seu respeito e faz um redirecionamento da rota caso a formiga esteja indo para o caminho errado. O funcionamento de um agente formiga é parecido com o envio de um datagrama a não ser pelo fato de que a formiga deve voltar para o nodo origem, preferencialmente por um caminho diferente. Na figura 3, pode-se observar o comportamento típico de uma formiga através da rede. O principal objetivo da formiga é a disseminação de informações através da rede ad hoc. Ela normalmente escolhe um caminho qualquer e não o melhor, que no caso da figura 3 seria através da rede fixa pelos nodos F' e F''.

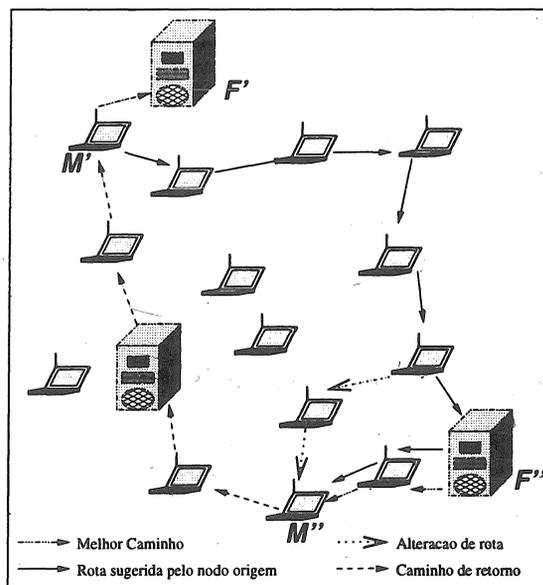


Figura 3: É mostrado aqui o comportamento típico de um agente formiga.

Outra vantagem do uso de formigas é que elas tornam as trocas de tabelas entre vizinhos mais confiáveis e ricas. Agora os nodos não têm acesso apenas a informações dos vizinhos, mas possivelmente também a informações recentes de nodos relativamente distantes na rede. Estas informações são então difundidas para nodos vizinhos. Um caso que exemplifica este cenário muito bem é a integração de um novo nodo com a rede. No caso de não haver um agente formiga, ou seja, apenas trocas de tabelas, o tempo necessário para nodos nos extremos da rede tomarem conhecimento da existência de um novo computador é proporcional

ao tempo de propagação de tabelas entre vizinhos. No caso do GPSAL, o nodo pode enviar uma ou mais formigas pela rede, o que irá disseminar a informação tanto da existência do nodo quanto de sua localização de forma mais rápida e eficiente.

3.4 Datagramas

O GPSAL provê a entrega de dados através de datagrama não confiável, da mesma forma que ocorre em redes fixas. Descreveremos a seguir, detalhadamente, a forma de operação do datagrama no GPSAL.

O nodo origem, ao enviar um pacote, *sugere* uma rota usando as informações que tem, e envia a mensagem para o primeiro nodo. Se não conseguir, tenta outra 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, no datagrama existe apenas a preocupação com a informação local. Não existe uma preocupação fim-a-fim ou com o estado da rede de uma forma “global”.

O datagrama no GPSAL desempenha também o papel de formiga, no sentido que também dissemina informações sobre o estado da rede. Sendo assim, mesmo que não seja gerada nenhuma formiga, a própria comunicação entre os nodos da rede se encarrega de disseminar a informação sobre a posição dos nodos por onde está passando.

3.5 Troca de tabelas

O Host Móvel (HM), ao conectar-se à rede ad hoc, “ouve o meio” até que descubra um nodo vizinho. Ao encontrar, envia um “pedido de tabela” e recebe em troca a tabela completa do vizinho. Desta forma, o HM tem acesso à posição de todos os nodos da rede móvel. A partir deste ponto, ele só receberá e enviará periodicamente informações de atualização relativos à esta tabela, incrementando sempre o tempo de vida das informações.

O TTL das informações é na verdade mais um contador de repasse. Sempre que um nodo repassa as atualizações sobre a sua tabela, incrementa o tempo de vida de todas as informações.

4 Métricas para redirecionamento de rotas

Uma das principais características do GPSAL é o redirecionamento de rotas por nodos intermediários, sendo este um dos maiores responsáveis pelo bom desempenho do algoritmo. Tendo em vista este fato, os redirecionamentos devem ocorrer com o maior cuidado possível. Os pontos principais que devem ser 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 de uma rede fixa
- Informações sobre movimentação do nodo destino

4.1 Caminho melhor que o atual

Este é certamente o ponto mais genérico e abstrato. Por isso vários parâmetros podem, e devem, ser observados quando consideramos um novo caminho. Caso o host móvel tenha informações que existe um caminho melhor, ele deve alterar o atual. Este “caminho melhor” depende da métrica utilizada, pode ser menor, menos congestionado ou com maior capacidade.

Pode acontecer de simplesmente não ser possível o contato com o próximo nodo. Sendo assim, outro caminho tem que ser encontrado. Um ponto que sempre é levado em consideração em qualquer tentativa de encontrar um melhor caminho, é a possibilidade do uso da rede fixa.

4.2 Uso da rede fixa

Sempre que um HM tentar encontrar um caminho melhor deve considerar a possibilidade de usar a rede fixa, caso exista alguma disponível. 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. A velocidade é maior e a quantidade de erros bem menor [25]. O HM quando cria um caminho considera o custo de comunicação dentro da rede fixa como nulo. Sendo assim, havendo uma rota que passa pela rede fixa, que tem o mesmo custo ou é menor que a rota que passa pela rede móvel, esta é automaticamente escolhida. Sempre que possível os nodos tentam desviar o tráfego para a rede fixa, diminuindo assim o gasto de recursos dos nodos intermediários. Pode se observar no gráfico da figura 6 o impacto que o uso da rede fixa representa no tráfego médio de uma rede de computadores.

A rede fixa é considerada “inteligente” o suficiente para encontrar o melhor Host Fixo (HF) em contato com o HM destino. Esta colocação não é uma restrição forte se considerarmos que estejam atuando na rede fixa protocolos como o IP-móvel [22] ou o IPv6 [23, 10]. Sempre que uma comunicação tiver como destino um host fixo, ou algum host que não esteja na tabela, esta comunicação deve seguir para o host fixo mais próximo, devido aos motivos descritos acima. O direcionamento acontecerá normalmente, no caso da comunicação com um HF qualquer.

4.3 Movimentações do nodo destino

O nodo intermediário pode observar a movimentação do destino e, principalmente, a sua direção e velocidade. Sendo necessário, pode redirecionar a comunicação de forma a “interceptar” o host móvel. Isto é possível levando em consideração a mudança de posição entre duas amostragens de tempo. O destino pode mudar sua posição e direção entre o momento que informou pela última vez e o momento atual, mas uma certa “inércia” é atribuída aos hosts móveis. Mesmo que o HM tenha mudado de direção, à medida que nos aproximamos da antiga posição, as informações sobre o nodo ficam mais recentes, favorecendo a convergência. Normalmente a velocidade da comunicação via rádio, e de processamento das mensagens é superior à velocidade de movimentação dos hosts móveis.

5 Resultados de Simulação

Nesta seção estão descritos vários experimentos comparando o desempenho do GPSAL com e sem formigas e com o LAR.

5.1 Modelo

O modelo usado para simular os algoritmos 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 $0,2N$ 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.

As trocas de tabelas no GPSAL ocorrem a cada três iterações. Este espaço curto entre as trocas favorece o LAR e o GPSAL sem formigas, já que as trocas de tabelas são frequentes e têm um alto custo, mas para o GPSAL sem formigas é a única forma que tem de conhecer o estado da rede.

5.2 Algoritmos Simulados

Em [14] Ko e Vaidya propõem duas variantes do LAR. A primeira chamada de LAR1 usa uma zona de requisição de formato retangular que pode conter o nodo destino. O tamanho dessa área é proporcional à velocidade de movimento do nodo destino e ao tempo decorrido desde o registro da última localização do destino. A segunda variante é o LAR2, onde uma mensagem possui duas informações usadas 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 onde o computador destino se encontra. Nas duas versões é feito um *flooding* inicial para se descobrir a posição do nodo destino. Em nossos experimentos desconsideramos este *flooding* inicial, pois só ele já teria um custo maior que a transmissão do pacote no LAR. O simulador sempre fornece a posição real do nodo destino ao LAR.

Foi simulado também uma versão do LAR sem formigas, onde os datagramas também não funcionam como formigas. O que se quer demonstrar com isto é que as formigas realmente exercem uma influencia na convergência da rede.

5.3 Experimentos

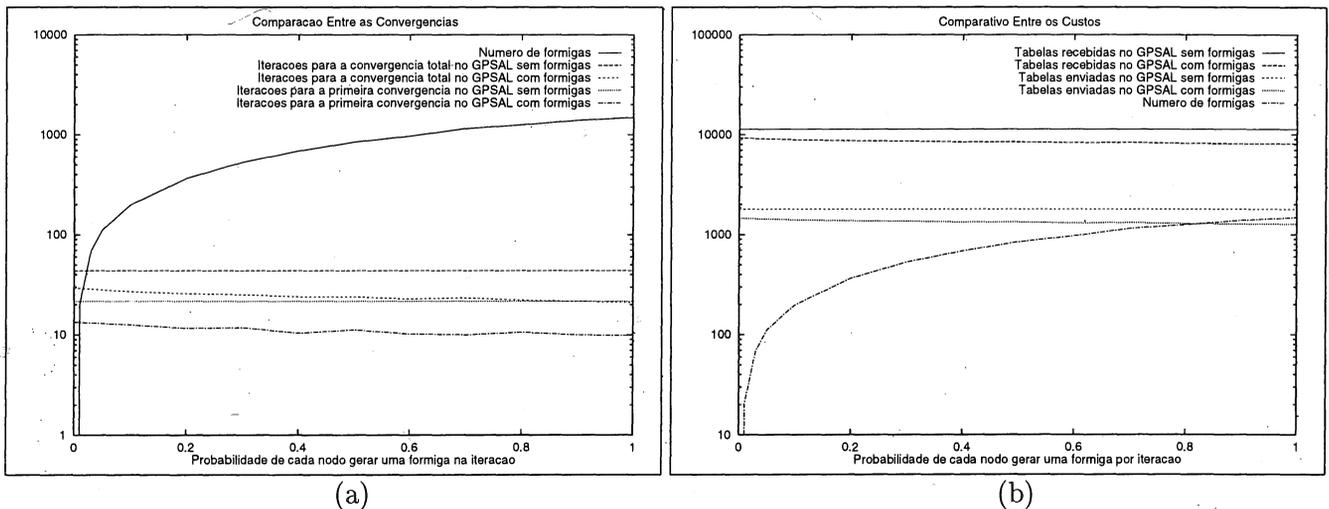


Figura 4: O gráfico da figura (a) apresenta uma comparação entre os tempos de convergência do GPSAL com e sem formigas e o da figura (b) apresenta a quantidade de tabelas enviadas e recebidas no GPSAL com e sem formigas

A figura 4 (a) mostra os tempos de convergência total e do primeiro nodo para o GPSAL com e sem formigas. O tempo de convergência é o número de iterações gasto pelo nodo para conhecer a posição real de todos os outros nodos da rede. Para os dois casos o GPSAL com formigas teve um tempo de convergência menor. O tempo de convergência é uma métrica importante em algoritmos de roteamento pois quanto melhor for o conhecimento do estado real da rede pelo nodo, melhor será o desempenho da rede.

A figura 4 (b) mostra a quantidade de trocas de tabelas nos algoritmos GPSAL com e sem formigas. As tabelas são enviadas por difusão e são recebidas por todos os nodos que estão na área de alcance. O valor "Tabelas Enviadas" indica a quantidade de tabelas enviadas por todos os nodos e o valor "Tabelas Recebidas" a quantidade de tabelas que os nodos receberam. Esse valor é maior devido à difusão. Como já era de se esperar, a quantidade de tabelas enviadas e recebidas no GPSAL é menor que no GPSAL sem formigas. Novamente, isso se deve ao uso de formigas. A diminuição do número de troca de tabelas é um fator importante pois causa uma economia de recursos dos computadores móveis, normalmente uma mensagem de troca de tabelas é maior que um pacote formiga.

Na figura 5 os gráficos (a) e (b) fazem uma comparação entre o GPSAL, o LAR1 e o LAR2, com relação ao tráfego médio gerado na rede por cada um. É considerado como tráfego, para o GPSAL, todas as trocas de tabelas e formigas enviadas na rede até a entrega de todos os pacotes. Para o LAR apenas os pacotes de dados gerados a partir do *flooding*. É passado para as duas versões do LAR a posição real do destino no momento do envio do pacote. Estamos assim desconsiderando um *overhead*, maior que a entrega da mensagem propriamente dita, que é o *flooding* puro inicial. Este *flooding* é necessário no LAR para o descobrimento da posição do destino.

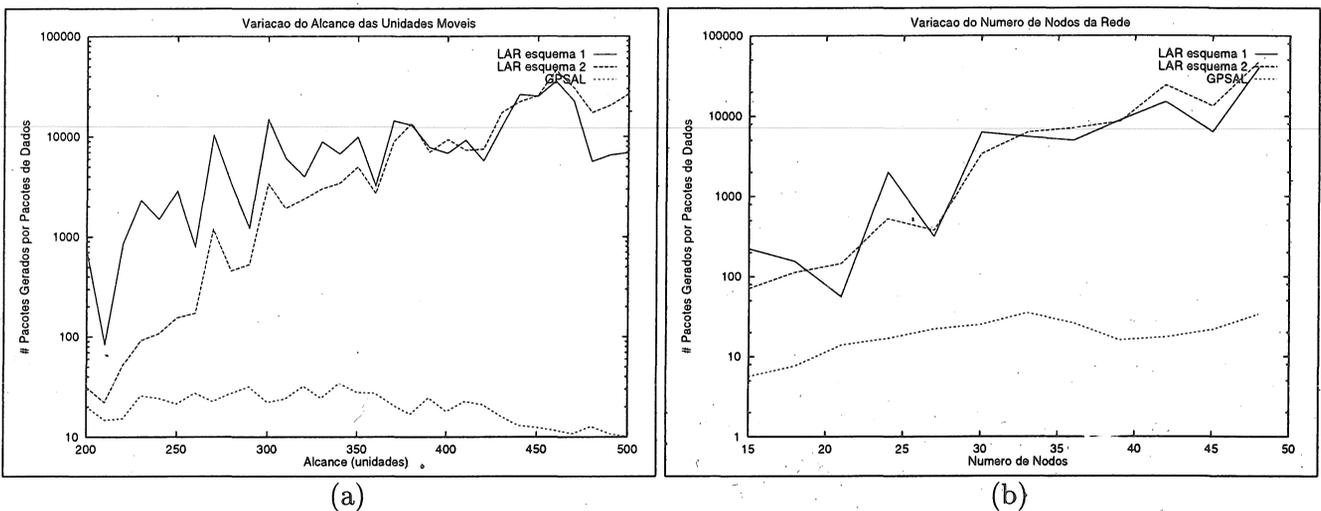


Figura 5: São apresentados nestes gráficos comparações entre o número de pacotes gerados pelo LAR1, LAR2 e GPSAL. Na figura (a) a comparação é em relação a variação do alcance e na figura (b) com relação a variação do número de nodos

Pode-se observar que o GPSAL, mesmo desconsiderando o *flooding* inicial no LAR e considerando no GPSAL a comunicação da rede como um todo, gera menos tráfego que o LAR. Na figura 5 (a) a comparação é feita com relação à variação do alcance. Nota-se que quanto maior o alcance, no GPSAL a tendência é que exista uma diminuição da comunicação já que os pacotes são entregues em um tempo menor. No LAR ao contrário, temos um aumento do número de pacotes, pois quanto maior o alcance maior é o número de nodos que participam do *flooding*. Na figura 5 (b) é apresentada uma comparação em termos da variação do número de nodos da rede. Pode-se notar que o GPSAL tem um comportamento mais uniforme que as duas versões do LAR. O LAR apresenta um grande crescimento no número de pacotes devido ao *flooding*. O crescimento no GPSAL do número médio de pacotes na rede se deve basicamente as trocas de tabelas exigidas pelos novos nodos. Estas trocas são necessárias para que se possa manter as tabelas atualizadas, não é um tráfego referente aos pacotes de dados. Outra vantagem do GPSAL é a escalabilidade: quanto maior for o número de pacotes, maior tende a ser esta diferença, já que os pacotes de dados, para o GPSAL,

não são o maior *overhead*.

O cenário utilizado no experimento da figura 6 é composto de um grid de $N \times N$ unidades contendo 150 computadores entre móveis e fixos. A distribuição e movimentação dos computadores é feita de forma aleatória. O alcance máximo de comunicação de uma estação é $0,17N$ a partir de sua posição, sendo o *timeoût* de 10 iterações e o experimento tendo rodado 50 iterações.

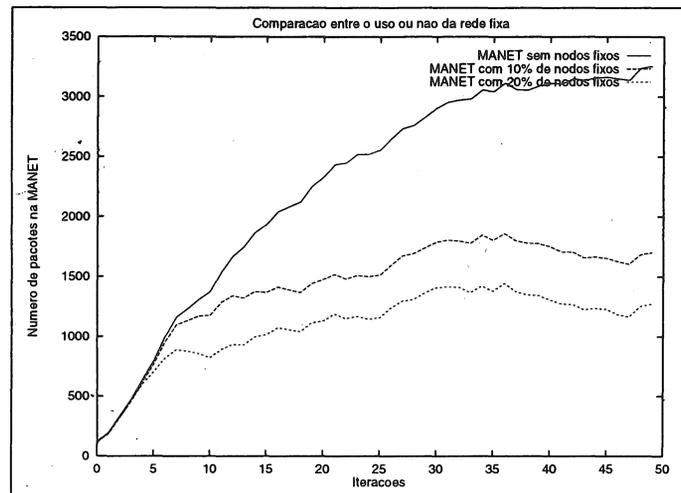


Figura 6: O gráfico compara o tráfego dentro da rede móvel em relação ao uso da rede fixa

Pode-se observar que com o uso da rede fixa consegue-se uma diminuição do tráfego dentro da rede móvel. No experimento, o tamanho médio dos caminhos ficou em 3.7275 nodos para 20% de nodos fixos na rede, em 4.6801 nodos para 10% e em 7.4481 nodos para a rede composta apenas de unidades móveis. Esta sensível diferença entre o número de nodos intermediários é um indicativo do aumento de precisão nas comunicações e também fornece uma noção da economia de energia que o uso da rede fixa proporciona. O tráfego na rede móvel, quando temos uma rede fixa disponível, é composto apenas do estritamente necessário, ampliando-se assim a capacidade da rede.

6 Conclusões

A integração de computadores com comunicações e outras formas de tecnologias de informação estão criando novas formas de sistemas e serviços de informação distribuída. A tendência é haver computadores mais poderosos, conectados a uma rede mundial de serviços e recursos através de uma infra-estrutura de alto desempenho. É o surgimento dos ambientes de computação ubíquos que deverá ser a nova forma de trabalho da próxima década. Este é o cenário altamente desafiador e excitante que motiva a computação móvel. Nesse cenário as redes móveis ad hoc deverão ter uma importância cada vez maior, sendo o roteamento de mensagens um problema fundamental nesse tipo de rede [17].

Neste trabalho é proposto um novo algoritmo de roteamento baseado em informações de localização e agentes formiga. A combinação do uso da localização de um computador com a coleta e disseminação de informações sobre a localização dos nodos da rede fazem com que esse algoritmo tenha um comportamento muito bom.

Os resultados de simulação mostraram também que o GPSAL tem um desempenho bem melhor que o LAR. Isso se deve fundamentalmente ao uso de formigas que disseminam informações pela rede fazendo com melhores rotas sejam sempre escolhidas e a ausência de *flooding*.

Referências

- [1] Mark Billinghurst and Thad Starner. Wearable devices: New ways to manage information. *Computer*, 1(1):57–64, January 1999.
- [2] G. Di Caro and M. Dorigo. Mobile agents for adaptive routing. In *31st Hawaii International Conference on Systems*, Big Island of Hawaii, USA, January 6–9 1998.
- [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] 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.
- [6] G. Dommety and R. Jain. Potential networking applications of global positioning systems GPS. Technical Report TR-24, Department of Computer Science, The Ohio State University, April 1996.
- [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] Z.J. Haas and M.R. Pearlman. The zone routing protocol (ZRP) for ad hoc networks. Internet-draft, August 1998.
- [9] Hewlett-Packard. Gps and precision timing applications. Application Note 1272 Publication Number:5965-2791E, Hewlett-Packard, May 1996.
- [10] R. Hinden and S. Deering. Ip version 6 addressing architecture. IETF RFC 1884, Dec. 1995.
- [11] Iowa State University GPS Page. <http://www.cnde.iastate.edu/gps.html>.
- [12] M. Jiang, J. Li, and Y.-C. Tay. Cluster based routing protocol (CBRP) functional specification. Internet Draft, August 1998.
- [13] D. Johnson, D.A. Maltz, and J. Broch. The dynamic source routing protocol for mobile ad hoc networks. Internet Draft, March 1998.
- [14] 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.
- [15] 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.
- [16] NAVSTAR GPS Operations. <http://tycho.usno.navy.mil/gpsinfo.html>.
- [17] Robert Castañeda and Samir R. Das. Query localization techniques for on-demand routing protocols in ad hoc networks. Technical Report CS-99-1, Division of Computer Science, The University of Texas at San Antonio, San Antonio, January 1999.

- [18] V.D. Park and M.S. Corson. Temporally-ordered routing algorithm (TORA) version 1 functional specification. Internet Draft, August 1998.
- [19] C. Perkins. IP Mobility Support. Request For Comments 2002, October 1996.
- [20] 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.
- [21] C.E. Perkins and E.M. Royer. Ad hoc on demand distance vector (AODV) routing. Internet Draft, August 1998.
- [22] Charles E. Perkins. Ip mobility support. IETF RFC 2002, Oct. 1996.
- [23] 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.
- [24] S. Ramanathan and M. Steenstrup. A survey of routing techniques for mobile communication networks. *Mobile Networks and Applications*, pages 89–104, 1996.
- [25] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, third edition edition, 1996.
- [26] C.-K. Toh. A novel distributed routing protocol to support ad-hoc mobile computing. *Wireless Personal Communication*, January 1997.