

# Protocolo de encaminhamento para redes móveis usando estruturas binárias eficientes

João Trindade  
Instituto Superior Técnico / INESC-ID  
Av. Rovisco Pais,  
1049-001 Lisboa, Portugal  
Email: jtrindade@tagus.inesc-id.pt

Teresa Vazão  
Instituto Superior Técnico / INESC-ID  
Av. Rovisco Pais,  
1049-001 Lisboa, Portugal  
Email: teresa.vazao@tagus.inesc-id.pt

**Resumo**—Protocolos de encaminhamento não geográficos são pouco eficientes quando aplicados a redes móveis de grande escala compostas por centenas de nós. Por outro lado, protocolos de encaminhamento geográficos possuem a desvantagem de necessitar de um sensor de localização. Este requisito aumenta o custo do equipamento presente em cada nó, bem como o consumo de energia dos dispositivos. Neste artigo propomos um protocolo de encaminhamento para redes móveis, que é escalável a redes compostas por centenas de nós. O protocolo não necessita de qualquer equipamento de localização e é adaptado para dispositivos com poucos recursos de memória e/ou processamento. Este objectivo é conseguido através do uso de filtros *bloom* para armazenar e espalhar informação topológica de uma forma eficiente. Na metodologia seguida, os nós não reencaminham para outros nós mensagens com informação topológica. Para tornar o processo eficiente, cada nó agrega a informação topológica que recebe dos seus vizinhos diretos com a sua própria e somente o resultado desta operação é transmitido para os restantes nós. Várias simulações foram efetuadas no simulador de redes Qualnet de modo a validar o algoritmo proposto pelo HRAN. Os resultados obtidos foram comparados com outros protocolos não-geográficos para redes móveis.

## I. INTRODUÇÃO

Atualmente o uso ubíquo de dispositivos *wireless* por parte de grande parte da população tem enaltecido o papel das redes móveis *Ad hoc*, também denominadas por MANETs. Este tipo de redes tem sido alvo de vários estudos devido às suas características, como o requisito de uma fonte de energia limitada e/ou as frequentes mudanças topológicas [1] [2]. A maior parte dos protocolos de encaminhamento propostos para uso em MANETs são tradicionalmente categorizados em protocolos pró-ativos, reativos ou um modelo híbrido destas duas propriedades. Protocolos de encaminhamento pró-ativos são caracterizados por manter rotas permanentes para todos os possíveis destinos, independente do tráfego existente na rede. Pelo contrário, protocolos reativos somente estabelecem novas rotas quando estas são requeridas por parte de um nó de origem. Entre os protocolos pró-ativos podemos encontrar o Optimized Link State Routing protocol (OLSR), o Destination-Sequence Distance Vector (DSDV) e o Temporally Ordered Routing Algorithm (TORA). Como representantes dos protocolos de encaminhamento reativos, podemos encontrar os protocolos Dynamic source routing (DSR) ou o Ad hoc On-demand Distance Vector (AODV). Existem também protoco-

los, normalmente designados como híbridos, que apresentam características reativas e pro-ativas. Um exemplo de um destes protocolos é o Zone Routing Protocol (ZRP).

Vários estudos foram realizados comparando as várias alternativas. Em [3], os autores simularam protocolos pro-ativos e reativos em redes de média escala e concluíram que este últimos providenciam bons resultados em termos de atraso e percentagem de pacotes entregues. Infelizmente estes protocolos também são os que causam mais sobrecarga na rede quando o pedido de rotas aumenta. Nesse estudo podemos igualmente verificar que os protocolos pró-ativos originam rotas com menos qualidade, bem como apresentam um maior número de pacotes perdidos. Por outro lado, se medirmos o número de mensagens de controlo enviadas, este conjunto de protocolos é considerado mais eficiente.

Os autores em [4] [5] demonstraram que os resultados dos vários protocolos variam conforme os cenários em que são testados. Nomeadamente em [6] e [7] ficou demonstrado que protocolos de encaminhamento não geográficos não escalam para redes compostas por centenas de nós. Nesse tipo de redes o número de mensagens de controlo cresce exponencialmente, penalizando severamente a rede, em termos de recursos, somente com a tarefa de descobrir novas rotas. Protocolos de encaminhamento geográficos têm a desvantagem de requerer um sensor de localização, como um dispositivo GPS, para executar os seus algoritmos de encaminhamento [8]. Em muitos casos o uso deste equipamento é inadequado pois origina um aumento de consumo de energia significativo e encarece o preço do equipamento presente em cada um dos nós. Além disso, o encaminhamento de pacotes é penalizado pois requer um componente adicional: um serviço de localização do destino.

Neste artigo apresentamos e estudamos um novo protocolo de encaminhamento denominado de HRAN (Heat Routing for Ad hoc Networks). O HRAN é um protocolo de encaminhamento escalável para redes de grande escala, no qual os nós possuem recursos limitados (em termos de memória, processamento e largura de banda) e que não requer o uso de dispositivos GPS.

O HRAN usa filtros *bloom* para armazenar e transmitir informação topológica de uma forma eficiente. No HRAN os nós não reenviam mensagens provenientes de outros nós. De forma a poupar recursos, os nós ao receberem men-

sagens com informação topológica, juntam através de uma operação binária de “OU” esta informação com o seu atual conhecimento da rede. Somente o resultado desta operação é partilhado com os outros nós. Esta operação permite ao HRAN espalhar informação topológica de uma forma eficiente, que depois pode ser utilizada na procura de rotas quando estas são requeridas. Este mecanismo permite igualmente criar zonas com informação topológica em torno de uma rota, que podem ser usadas quando for necessário reparar essa rota ou melhorá-la em termos de número de nós que a constituem.

Atualmente já existem algumas soluções propondo o uso de filtros *bloom* em protocolos de encaminhamento para redes *ad hoc* [9]. No protocolo Gradient-ascending routing via footprints (GRASP) [10], os filtros *bloom* são usados para indicar se um nó pertencente a uma rede de sensores é constituinte de uma rota. Devido ao âmbito deste protocolo se restringir somente a redes de sensores, não existe qualquer forma de lidar com mobilidade nos nós. Outro protocolo de encaminhamento que utiliza filtros *bloom* é o Table Attenuation Routing Protocol (TARP) [11]. Neste protocolo cada nó possui um conjunto de filtros *bloom* por vizinho. A posição de um nó nesse conjunto de filtros *bloom* indica a distância a que o nó se encontra. O problema existente no TARP é ser muito penalizado por redes em que a densidade de nós seja elevada ou em redes com bastante mobilidade.

O resto deste artigo encontra-se estruturado da seguinte forma. Na secção 2 o HRAN é descrito, bem como o conceito de filtro *bloom* com informação temporal. Na secção 3 são apresentados os resultados da comparação do HRAN com outros protocolos. Finalmente na secção 4 são explicitadas as conclusões.

## II. DESCRIÇÃO DO HRAN

Esta secção irá apresentar o funcionamento do protocolo de encaminhamento HRAN. Inicialmente irá ser apresentada uma breve descrição do algoritmo, seguida de uma explicação dos filtros *bloom* com informação temporal. Finalmente irá ser realizada uma descrição detalhada do algoritmo.

### A. Conceito

O HRAN tenta imitar um paradigma de “rastros de calor” existente no mundo físico mapeando-o para uma representação de topologia de rede. Cada nó cria uma região de calor à sua volta através de emissão periódica de mensagens de calor enviadas aos seus vizinhos. Nós dentro desta região “sentem” o calor e têm uma pista em como chegar ao destino. Nós mais perto do destino “sentem” o nó mais perto enquanto que nós mais afastados do destino sentem a pista de calor mais fria. Nós que se encontrem fora da região de calor de um nó, sempre que desejarem comunicar com ele terão que procurar lançar uma procura aleatória que prossegue pela rede até atingir uma região de calor do destino. A partir desse momento a procura passa a ser dirigida, passando de zonas de calor mais frias para mais quentes. Finalmente depois de uma rota ser encontrada, existe a criação de um túnel de calor. Este mecanismo é útil para auxiliar futuras procuras pelo mesmo

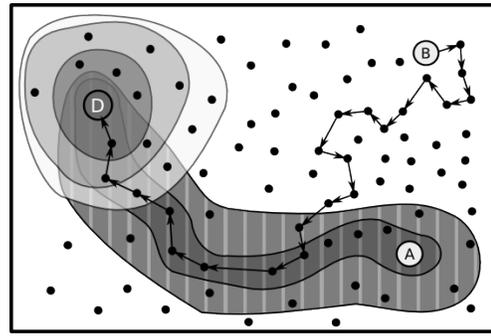


Figura 1. Funcionamento do HRAN: representação dos gradientes de calor do nó D; túnel de calor da rota existente entre o nó A e o nó D e a procura aleatória e procura direcionada entre o nó B e o nó D.

destino, reparar possíveis falhas da rota e melhorar a rota em termos de número de nós que a constituem.

O funcionamento do HRAN pode ser visualizado na figura 1, na qual estão presentes os vários gradientes de calor do nó D (destino). Nós localizados em zonas com cores mais escuras indicam que a presença do nó D encontra-se num gradiente mais quente que em nós localizados em zonas com cores mais claras. A zona com riscas indica o túnel de calor. O processo de comunicação entre o nó B e D encontra-se também representado o conceito de procura aleatória e procura direcionada.

Toda a informação referente ao “calor” dos nós encontra-se representada através de uma adaptação de um filtro *bloom* [12] [13]. Um filtro *bloom* é uma estrutura de dados eficiente, em termos de espaço, que permite testar se um elemento é membro de um conjunto, mas que admite a existência de falsos positivos. Um filtro de *bloom* é instanciado através de um vector de bits de tamanho  $m$  e de  $k$  funções *hash* distintas. Cada uma destas funções de *hash* mapeia um elemento do conjunto para uma das posições da matriz  $m$ . Para adicionar um elemento a um conjunto, basta executar cada uma das funções de *hash* com o seu identificador para obter  $k$  posições. Cada uma destas posições do vector de bits é posta com o valor 1. Para consultar se um elemento pertence ao conjunto, volta-se a inserir o identificador em cada uma das  $k$  funções de *hash* para obter as posições do vector. Se algum dos bits nessas posições for 0, o elemento não pertence ao conjunto.

No protocolo HRAN uma variante destas estruturas binárias de reduzida dimensão é agrupada em gradientes cujas posições indicam a distância, em termos de número de nós intermédios, a que os nós se encontram uns dos outros. Estes filtros *bloom* adaptados são periodicamente trocados entre vizinhos, garantindo desta forma que informação sobre a topologia da rede é disseminada.

### B. Filtro bloom com informação temporal

O protocolo de encaminhamento HRAN requer uma extensão de funcionalidade das estruturas tradicionais de filtros *bloom*, de modo a que se possa verificar se um objecto foi adicionado num certo período temporal. Esta extensão permite

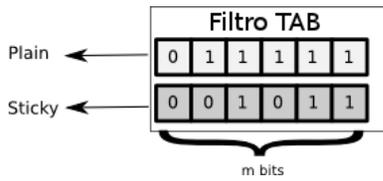


Figura 2. Filtro TAB

aos nós gradualmente perder informação do calor de vizinhos que se tenham afastado, ou que simplesmente tenham falhado. Para obter esta funcionalidade foi criado um novo tipo de filtro *bloom* denominado filtro TAB (*Time-Aware Bloom*).

O filtro TAB é um conjunto de dois filtros *bloom* com o mesmo tamanho  $M$  e o mesmo grupo de  $K$  funções de *hash*. Os dois filtros *bloom* são denominados como o filtro *sticky* e o filtro *plain* como é possível visualizar na Figura 2. O filtro *sticky* é usado para identificar se um bit foi inserido no último intervalo temporal ( $\delta^n$ ). O filtro *plain* contém os bits inseridos durante o último e o anterior intervalo temporal ( $\delta^n \parallel \delta^{n-1}$ ). Os filtros TAB são sempre utilizados em vectores, sendo de seguida explicado o funcionamento das operações de inserção, procura e atualização nestes filtros.

Para adicionar um elemento  $X$  na posição  $i$  de um vector de filtros TAB, o algoritmo realiza a operação de inserção normal nos filtros *bloom plain* e *sticky* presentes nessa posição do vector. Esta operação consiste em realizar as  $K$  funções de *hash* no elemento a inserir de modo a obter as  $k$  posições. Nessas posições é colocado o valor de 1 em ambos os filtros *bloom*.

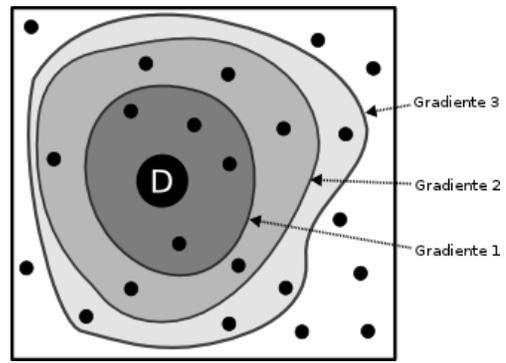
Para procurar um elemento  $Y$  num vector de TAB, a procura é somente realizada nos filtros *plain* presentes no vector. Esta operação consiste em fazer *hash* ao elemento a procurar com as  $K$  funções de *hash* e para cada uma das posições obtidas, verificar em cada filtro *plain* se todos os bits respectivos contêm o valor 1. Se isto for verdade, a procura encontrou o objecto. Se todos os filtros do vector forem percorridos e esta correspondência nunca for encontrada o objecto não foi encontrado.

Finalmente, a operação de atualização, realizada em períodos regulares  $\delta$ , percorre todos os filtros *plain* existentes no vector e reescreve-os com o valor existente no filtro *sticky* presente na mesma posição do vector. Após esta operação, todos os filtros *sticky* são reescritos com o valor de 0 para iniciar um novo intervalo temporal  $\delta$ .

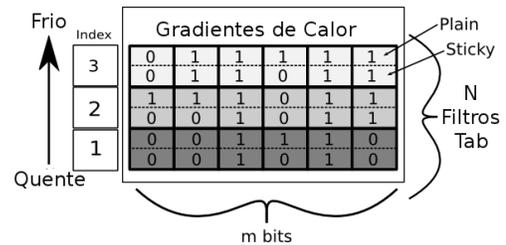
### C. Descrição Funcional

O protocolo de encaminhamento HRAN encontra-se dividido em três fases de execução: construção da camada de calor, procura de calor e manutenção de rota. As próximas secções irão detalhar cada uma destas fases.

1) *Construção da camada de calor*: Na fase de construção da camada de calor cada nó envia a sua informação topológica da rede para os nós vizinhos seguindo um modelo pro-ativo. Cada nó armazena a informação que possui sobre a topologia da rede num vector de filtros TAB com  $N$  posições. Cada uma



(a) Camada de calor de um nó



(b) Gradiente de calor

Figura 3. Camada de calor de um nó

desta posições corresponde a um nível no gradiente de calor. Nós presentes em filtros TAB contidos em posições do vector menores são considerados como sendo mais quentes que nós contidos em posições mais próximas de  $N$ .

A figura 3 representa a camada de calor do nó  $D$  em que os filtros foram configurados para um tamanho de 6 bits. No exemplo apresentado o protocolo HRAN foi configurado com 3 gradientes ( $N = 3$ ). Nós em camadas cinzentas escuras significam que possuem o nó  $D$  em posições do gradiente mais quentes do que nós localizados nas zonas cinzentas claras. Nós que não tem o nó  $D$  nos seus vectores de filtros TAB, não possuem qualquer indicação sobre a localização do nó  $D$ .

O processo de construção da camada de calor é executado por cada nó ao enviar mensagens de *HELLO* em intervalos temporais regulares. Cada mensagem de *HELLO* contém um vector com  $N - 1$  filtros *bloom*, pois o gradiente mais *frio* ( $N$ ) não é transmitido. Os filtros de *bloom* contidos nas mensagens de *HELLO* são uma cópia dos filtros *plain* do nó emissor do gradiente de calor 1 ao  $N - 1$ . Somente filtros *plain* são enviados de forma a reduzir a quantidade de informação enviada para a disseminação de informação topológica.

Quando um nó recebe uma mensagem de *HELLO* uma operação binária de "OU" é realizada entre os filtros *bloom* transmitidos pela mensagem de *HELLO* e o gradiente de calor do próprio nó. Como demonstrado na figura 4 a cópia dos filtros contidos na mensagem começam na posição 2 enquanto a posição na estrutura do nó começa na posição 1. Este mecanismo permite que a informação sobre um só se vá ficando mais fraca quanto mais longe estivermos dele.

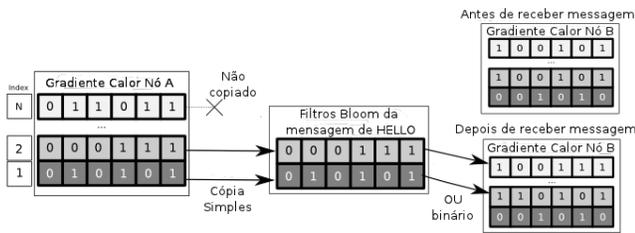
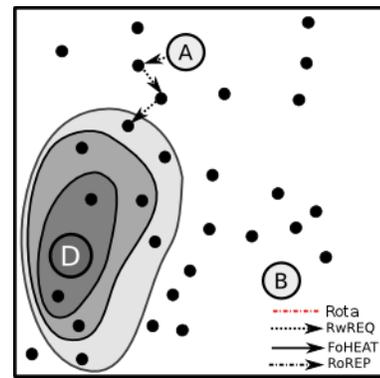


Figura 4. Troca de mensagens de *HELLO* entre nós vizinhos

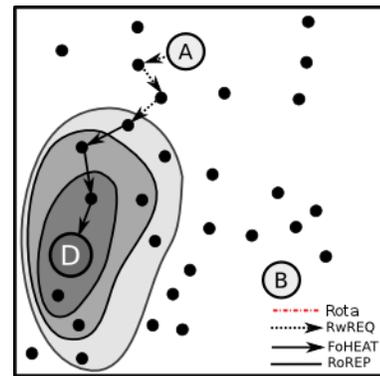
2) *Descoberta de rota*: Quando um nó requer uma nova rota inicia um processo descoberta de rota no qual uma mensagem de procura aleatória denominada *RwREQ* percorre a rede até encontrar o “calor” do nó destino. Quando um nó recebe uma destas mensagens analisa o seu gradiente de calor verificando se possui informação sobre o nó destino. Se a resposta for afirmativa, o nó entra no modo de procura direcionada, no qual uma mensagem de seguir calor (*FoHEAT*) é transmitida contendo o nível de calor encontrado. Se não encontrar informação reenvia a mensagem para um dos seus vizinhos. Nós que uma mensagem de *FoHEAT*, caso possuam um nível de calor superior ao indicado na mensagem, reenviam a mensagem atualizando-a com o seu próprio nível de calor. Somente nós nesta condição reenviam a mensagem. Nós que tenham o nó destino em gradientes mais frios ignoram a mensagem de *FoHEAT*. Esta opção permite reduzir o número de mensagens necessárias para a descoberta de uma nova rota. Quando a mensagem de *FoHEAT* chega ao nó destino, a rota é dada pela lista de nós presente na mesma. Nesta altura o protocolo inicia o modo de retorno, no qual uma mensagem (*RoREP*) é enviado do destino à origem pelo caminho inverso ao descoberto anteriormente.

Devido à natureza aleatória do processo de descoberta de rotas, é possível que informação de calor do nó destino não seja encontrada em tempo útil ou que a existência de falsos positivos nos filtros *bloom* leve a procura do destino por direções erradas. Devido a estas razões tem que existir um mecanismo de redundância que seja executado quando nenhuma rota é descoberta. Caso o nó que pediu o estabelecimento de rota não receba uma mensagem de *RoREP* dentro de um certo período temporal, é executado um protocolo reativo (como o DSR) para o estabelecimento da rota. Esta opção permite a criação de rotas, e de túneis de calor, que irão ser descritos na próxima secção, facilitando assim futuras procuras.

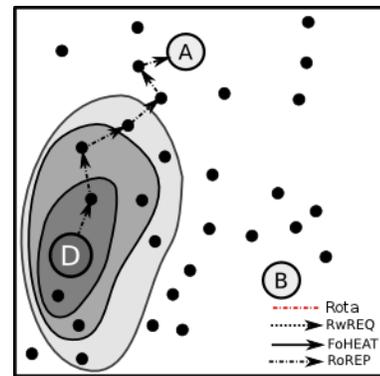
3) *Manutenção de rotas*: O processo de manutenção de rotas encontra-se dividido em três tarefa: (i) a *criação de túneis de calor* cria e mantém os túneis de calor quando uma rota está a ser usada para transferir pacotes de dados; (ii) a *reparação de rotas* recupera a quebra de uma rota devido à falha de um ou mais dos nós que a compõem; (iii) o *processo de melhoria de rotas* iterativamente melhora as rotas de modo a reduzir o impacto da mobilidade e da aleatoriedade existente no processo de descoberta de rota.



(a) Procura aleatória



(b) Procura direcionada

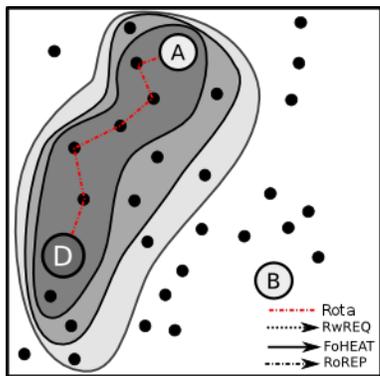


(c) Retorno

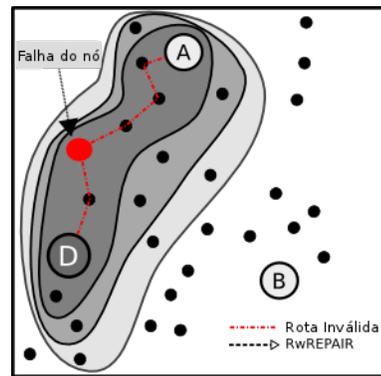
Figura 5. Processo de descoberta de rotas

a) *Criação de túneis de calor*: Quando um pacote dados é encaminhado por um nó intermédio, um túnel de calor é criado através da inserção do identificador no filtro *TAB* presente na posição 1 do gradiente de calor desse nó. Este processo mantém atualizada a informação sobre “calor” das rotas usadas atualizado, enquanto que, para rotas não usadas, essa informação é gradualmente perdida pelo processo normal de atualização dos filtros *TAB*.

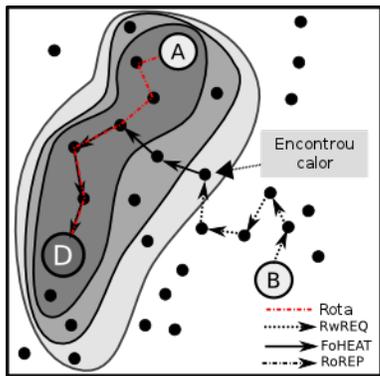
A figura 6a demonstra um túnel de calor do nó A ao nó D. Novamente nós em zonas de cinzento mais escuras possuem



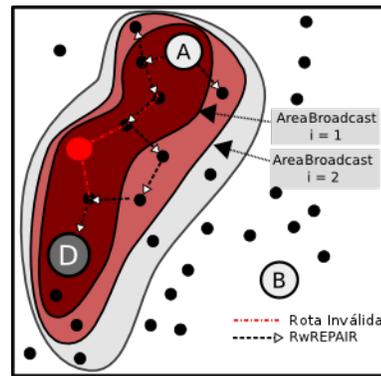
(a) Túnel de calor para o nó  $D$



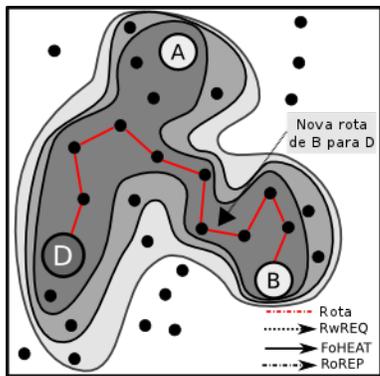
(a) Falha de nó intermédio



(b) Nova rota descoberta



(b) Reparação de rota com  $i=2$ ;



(c) Criação túnel de calor

Figura 6. Descoberta de rota e criação de túnel de calor respectivo

informação sobre o nó  $D$  num gradiente mais baixo (mais quente) que nós que se encontram em zonas mais claras.

Existem dois objetivos para a criação de túneis de calor em rotas ativas. O primeiro objetivo consiste em auxiliar futuras procuras de rotas por parte de outros nós, como se encontra demonstrado na figura 6b. O outro objetivo é possibilitar a reparação de rotas localmente, através do uso de poucas mensagens. A figura 6c demonstra a criação de um novo túnel de calor após a descoberta da nova rota.

Figura 7. Processo de reparação de rota

*b) Reparação de rotas:* Quando é detectada uma falha de rota é desencadeado um processo que utiliza o túnel de calor previamente criado para reparar a rota de uma forma eficiente e rápida. Durante este processo, é feito o *broadcast* iterativo de mensagens de reparação (*REPAIR*) em sucessivas zonas de calor. Este processo termina quando o destino é encontrado ou quando já não existam mais zonas de calor para procurar a nova rota.

Em mais detalhe, para executar o processo de reparação, uma mensagem *REPAIR* é enviada para todos os nós na área de reparação em modo de *broadcast*. Visto que a zona de calor em que os nós possuem o nó destino no gradiente mais quente (gradiente 1) ser a rota que falhou, a primeira área de reparação começa por ser o gradiente 2. A área de reparação é progressivamente incrementada até que atinja o gradiente  $N$ . Nesta fase, a rota é considerada como irreparável e um novo processo de descoberta de rota é iniciado. A figura 7 demonstra o processo de reparação, em que uma nova rota foi descoberta com o gradiente de 2.

*c) Melhoria de rotas:* Devido ao processo aleatório da descoberta de rotas, no protocolo HRAN não é garantido que as rotas descobertas sejam as mais curtas. Deste modo, para melhorar a qualidade das rotas, o protocolo HRAN possui um mecanismo de melhoria de rotas aproximando-as iterativamente da melhor rota possível. Este processo também é útil

para diminuir o efeito da mobilidade dos nós na qualidade nas rotas.

O processo de melhoria de rotas do HRAN usa os túneis de calor previamente criados para descobrir rotas mais curtas. Quando uma rota é usada, o HRAN conta o número de pacotes que são enviados por ela. Quando este valor atinge um limite pre-determinado, um processo de melhoria de rota é iniciado. Este processo consiste no envio em *broadcast* de mensagens de melhoria de rota (*RoIMP*) contendo a identificação do nó de destino. Esta mensagem é reenviada por todos os nós que contenham a identificação do nó destino em qualquer um dos seus níveis no gradiente de calor. Este processo acontece recursivamente até que uma destas mensagens chegue ao nó de destino. Quando isto acontece, é desencadeado um processo de retorno com mensagens *RoREP* similar à descoberta de novas rotas.

Ao usar este mecanismo, as mensagens de melhoria de rota só são enviadas para nós que pertençam ao túnel de calor do nó destino. Isto reduz o número de mensagens transmitidas e permite que, caso exista, seja descoberta uma rota com menos nós.

### III. AVALIAÇÃO DE DESEMPENHO

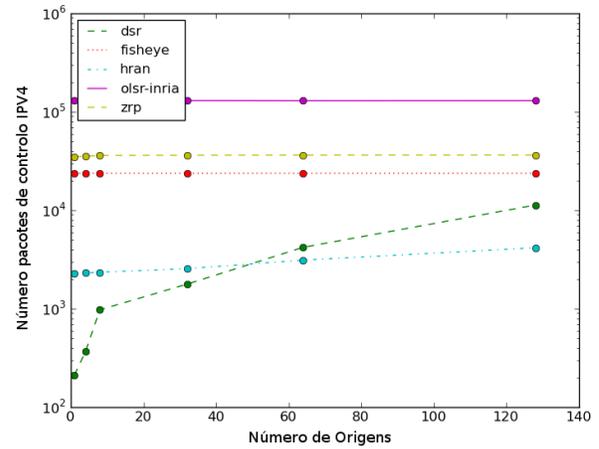
As simulações foram executadas no Qualnet versão 5.0.1. Foi escolhido este simulador pois suporta redes de larga escala com centenas de nós e possui uma biblioteca com vários protocolos para MANETs já implementados, possibilitando assim uma comparação fácil e fidedigna. Para todos os testes as condições de teste foram as seguintes:

- tamanho do mapa - 1200m por 1200m
- distância de comunicação máxima -150m
- modelo de propagação - *Two-ray*
- tempo simulado - 110s
- velocidade dos nós - 15m/s
- direção dos nós - *aleatória*

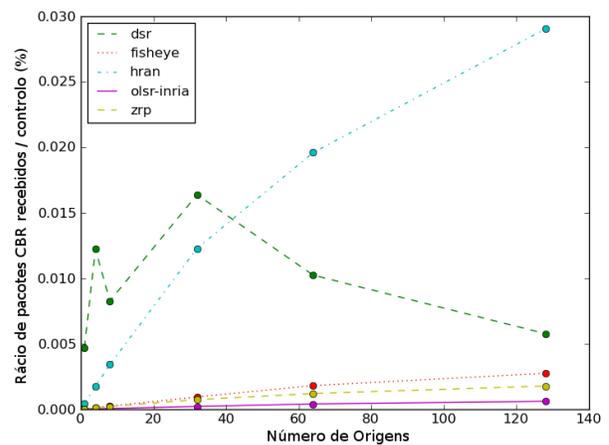
Os parâmetros configurados para o HRAN foram os seguintes. Intervalo entre mensagens de *HELLO* de 7s, uso de 4 níveis nos gradientes de calor, tamanho de filtros *bloom* de 256 bits, um tempo de atualização dos filtros TAB de 21 segundos e um intervalo de melhoria de rotas de 20 segundos.

#### A. Aumento do número de pedidos de rota

O primeiro teste de desempenho consiste em medir o número de pacotes enviados pelo o protocolo de encaminhamento para descobrir novas rotas em uma rede composta por 200 nós. Neste cenário a variável independente escolhida foi o número de nós de origem. O nó de destino mantém-se igual para todos os testes. Como é possível verificar na figura 8a os protocolos ZRP e OLSR (implementação INRIA) mantêm um valor perto de constante independente do número de rotas pedidas. Para valores de nós de origem reduzidos, o protocolo DSR necessita de enviar menos mensagens que o HRAN, mas à medida que o número de pedidos de rotas aumenta, o mecanismo de camadas e túneis de calor presentes no HRAN, melhora a eficiência do processo de procura de rotas. Na figura 8b é possível observar a percentagem de pacotes de



(a) Número de mensagens de controlo



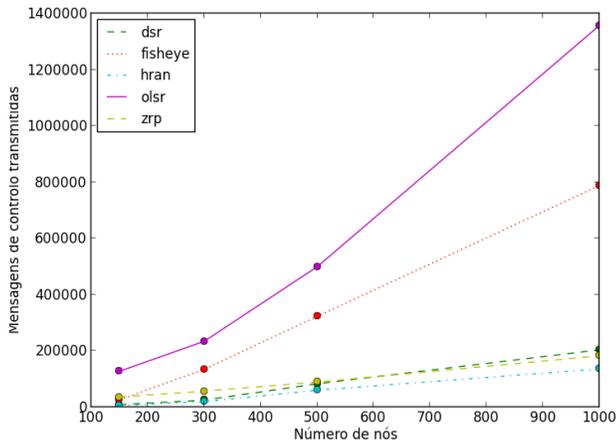
(b) Rácio de mensagens entregues para mensagens de controlo

Figura 8. Desempenho protocolos com o aumento de nós de origem

controlo enviados por cada pacote de dados recebido pelo destino, novamente o protocolo de encaminhamento HRAN apresenta os melhores valores para números de pedidos de rota superiores.

#### B. Aumento do número de nós

A principal funcionalidade do protocolo de encaminhamento HRAN é a capacidade de suportar redes compostas por centenas de nós. O segundo teste consiste em analisar o número de mensagens transmitidas pelo protocolo de encaminhamento à medida que o número de nós na rede aumenta. Na figura 9a é possível observar o que para redes compostas por 150 nós tanto o DSR como o protocolo HRAN requerem menos pacotes de controlo que os restantes protocolos. Apesar de o DSR possuir um valor ligeiramente mais reduzido para a rede menor composta por 150 nós, para redes maiores o HRAN apresenta valores mais reduzidos seguido do protocolo ZRP. Em termos de do atraso referente à descoberta da nova rota podemos verificar na figura 9b que os dois protocolos



(a) Mensagens de controlo

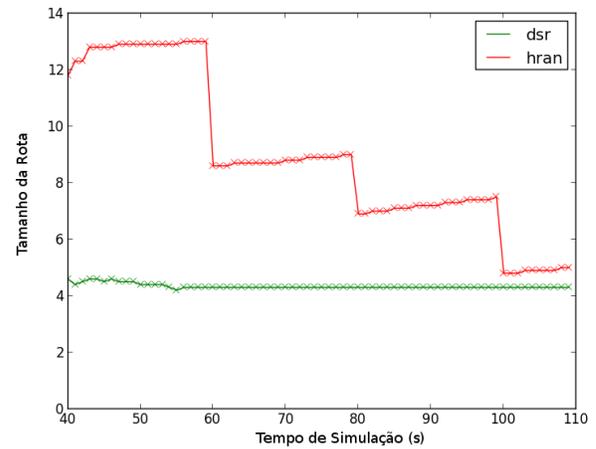
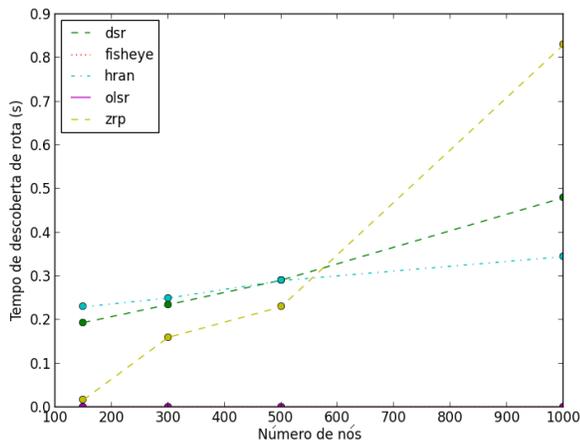


Figura 10. Tamanho médio das rotas utilizadas



(b) Tempo de descoberta de rota

Figura 9. Desempenho com aumento de número de nós

pró-ativos (OLSR e fisheye) têm sempre rotas estabelecidas, logo não possuem atrasos quando novas rotas são pedidas. Em relação aos restantes protocolos de encaminhamento é possível verificar que para redes de grande dimensão, o HRAN possui valores mais reduzidos devido ao uso das suas estruturas de calor para guiar procura de novas rotas.

### C. Tamanho das rotas

Finalmente, o último teste centra-se na análise do tamanho das rotas usadas pelo protocolos HRAN e DSR à medida que a simulação decorre. A figura 10 apresenta os resultados onde é possível verificar que apesar de o protocolo HRAN apresentar rotas mais longas, quando comparado com o DSR, o mecanismo de melhoria de rota ativado a cada 20 segundos gradualmente diminui o tamanho de rota para valores semelhantes ao DSR. Estes resultados validam a utilidade deste mecanismo para mitigar um dos problemas associados ao uso do protocolo HRAN devido à procura de rota parcialmente aleatória.

## IV. CONCLUSÕES

Neste artigo foi apresentado o protocolo HRAN desenhado para redes MANET compostas por centenas de nós. O algoritmo desenvolvido assenta no uso de filtros *bloom* para armazenar informação topológica sendo esta usada para descobrir e manter rotas. Este tipo de estrutura de dados permite armazenar uma grande quantidade de informação binária de uma forma eficiente, reduzindo os recursos requeridos pelo protocolo de encaminhamento.

Os resultados das simulações demonstram que o HRAN requer menos mensagens de controlo que outros protocolos de encaminhamento e que o seu processo de reparação de rotas permite uma superior percentagem de pacotes entregues em cenários móveis. Como trabalho futuro será interessante estudar um mecanismo de adaptação dinâmica dos parâmetros de configuração do protocolo HRAN para as condições da rede.

## AGRADECIMENTOS

Este trabalho foi financiado por fundos nacionais através da FCT - Fundação para a Ciência e a Tecnologia, no âmbito do projecto PEst-OE/EEI/LA0021/2011.

## REFERÊNCIAS

- [1] Sarkar, N.I., Lol, W.G.: A study of manet routing protocols: Joint node density, packet length and mobility. In: Computers and Communications (ISCC), 2010 IEEE Symposium on. (2010) 515–520
- [2] Kumar, J., Rajesh, R.: Performance analysis of manet routing protocols in different mobility models. IJCSNS International Journal of Computer Science and Network Security **9** (2009) 22–29
- [3] Das, S., Castaneda, R., Yan, J., Sengupta, R.: Comparative performance evaluation of routing protocols for mobile, ad hoc networks. In: Computer Communications and Networks, 1998. Proceedings. 7th International Conference on. (1998) 153–161
- [4] Wu Chin, K., Judge, J., Williams, A., Kermod, R.: Implementation experience with manet routing protocols. ACM SIGCOMM Computer Communications Review **32** (2002) 49–59
- [5] Mohseni, S., Hassan, R., Patel, A., Razali, R.: Comparative review study of reactive and proactive routing protocols in manets. In: Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on. (2010) 304–309

- [6] Wang, Y., Dong, L., Liang, T., Yang, X., Zhang, D.: Cluster based location-aided routing protocol for large scale mobile ad hoc networks. *IEICE Transactions* **92-D** (2009) 1103–1124
- [7] Boukerche, A.: Performance evaluation of routing protocols for ad hoc wireless networks. *Mobile Network Applications* **9** (2004) 333–342
- [8] Karp, B., Kung, H.: Gpsr: Greedy perimeter stateless routing for wireless networks. In: *Annual International Conference on Mobile Computing and Networking (Mobicom)*, ACM (2000) 243–254
- [9] Broder, A., Mitzenmacher, M.: Network applications of bloom filters: A survey. *Internet Mathematics* **1** (2005) 485–509
- [10] Lim, J.J., Shin, K.G.: Gradient-ascending routing via footprints in wireless sensor networks. *Real-Time Systems Symposium, IEEE International* **0** (2005) 298–307
- [11] Gilbert, R., Johnson, K., Wu, S., Zhao, B.Y., Zheng, H.: Location independent compact routing for wireless networks. In: *Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking, MobiShare '06, New York, NY, USA, ACM* (2006) 57–59
- [12] Tarkoma, S., Rothenberg, C., Lagerspetz, E.: Theory and practice of bloom filters for distributed systems. *Communications Surveys Tutorials, IEEE* (2011) 1–25
- [13] Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM* **13** (1970) 422–426