

Encaminhamento Multi-Rota de Baixo Acoplamento para Redes Sem-Fios em Malha

Cristina Fonseca
INESC-ID Lisboa/IST
Email: cfonseca@gsd.inesc-id.pt

José Mocito
INESC-ID Lisboa/IST
Email: jmocito@gsd.inesc-id.pt

Luís Rodrigues
INESC-ID Lisboa/IST
Email: ler@ist.utl.pt

Resumo—Este artigo aborda o problema do encaminhamento multi-rota em redes sem fios em malha. Neste contexto, propomos um algoritmo de *clustering* que facilita a escolha de múltiplas rotas que não interfiram mutuamente ao nível rádio, assim como um algoritmo que combina facetas pró-activas e reactivas para determinar e manter estas rotas. O protocolo resultante é avaliado através de simulações no NS-2, mostrando que a nossa solução obtém um equilíbrio interessante entre o custo de sinalização necessário para descobrir as rotas e a eficácia das rotas encontradas.

I. INTRODUÇÃO

As Redes Sem-Fios em Malha (WMN, do Inglês, *Wireless Mesh Networks*) têm sofrido uma rápida evolução e sido instaladas num número crescente de ambientes. Em consequência, espera-se que este tipo de redes venha no futuro a servir de base para o fornecimento de rede sem fios num conjunto vasto de cenários, de âmbito pessoal, local, ou metropolitano [1], [2].

Como em qualquer rede, quando usadas na prática, é normal que algumas rotas sejam usadas mais intensamente que outras. No entanto, nestas redes, devido ao número limitado de frequências disponíveis, não é fácil sobredimensionar o equipamento que suporta as rotas mais utilizadas. Desta forma, técnicas que permitam usar cabalmente todas as rotas possíveis tornam-se particularmente relevantes. O encaminhamento multi-rota (do Inglês, *Multipath Routing*), é uma técnica promissora para atingir este objectivo. Ao estabelecer múltiplas rotas entre a fonte e o destino, abre-se a possibilidade de usar técnicas de distribuição de carga e, desta forma, aumentar a largura de banda oferecida às aplicações.

Existem no entanto vários desafios que se colocam quando se tenta utilizar encaminhamento multi-rota em redes sem fios em malha. Um dos desafios é que as diversas rotas não devem interferir ao nível rádio. Caso contrário, a utilização de uma rota pode afectar a comunicação noutras rotas, resultando numa degradação (em vez de um aumento) da largura de banda efectiva. De facto, numa rede sem fios, a interferência rádio é um dos factores limitativo da capacidade da rede [3], [4].

Neste trabalho propomos um novo protocolo de encaminhamento multi-rota que permite encontrar múltiplos caminhos que não interferem entre si. Para facilitar a descoberta e manutenção destes caminhos recorreremos a um algoritmo de *clustering*, capaz de auto-organizar os nós em grupos e eleger um líder para cada grupo, de forma a que dois líderes nunca interfiram mutuamente. Este líderes definem uma rede sobreposta que é usada para encontrar rotas independentes.

Apresentamos resultados obtidos através da simulação do nosso protocolo usando o NS-2. Os resultados mostram que os caminhos encontrados pela nossa solução permitem obter melhorias na largura de banda recorrendo a um equilíbrio interessante entre o custo de sinalização necessário para descobrir as rotas e a qualidade das rotas encontradas.

O remanescente do artigo está organizado da seguinte forma. Na Secção II é apresentado o trabalho relacionado. A Secção III descreve o nosso protocolo e a Secção IV apresenta a sua avaliação. Finalmente, na Secção V apresentam-se as conclusões e direcções de trabalho futuro.

II. TRABALHO RELACIONADO

A maioria dos protocolos que foram propostos para suportar o encaminhamento em redes sem fios, ad hoc ou em malha, consideram o caso em que uma única rota é criada entre a fonte e o destino de um fluxo de pacotes. O objectivo do encaminhamento multi-rota é o de utilizar múltiplas rotas até ao destino, e não apenas a melhor rota.

No entanto, para que as múltiplas rotas sejam úteis, estas não devem interferir mutuamente. É um facto conhecido que na comunicação sem fios, a qualidade da transmissão pode ser severamente degradada se múltiplas transmissões ocorrem em regiões adjacentes; este fenómeno é designado por *acoplamento de rotas* [5]. É óbvio que duas rotas que possuam nós ou elos em comum (sem considerar a própria origem ou destino) estarão acopladas. Infelizmente, o acoplamento das rotas pode acontecer mesmo em cenários em que as rotas não possuem nós ou elos em comum: basta que estes nós estejam ao alcance rádio uns dos outros. De forma a minimizar a interferência, é necessário recorrer a rotas de baixo acoplamento (também designadas rotas por zonas disjuntas).

Tanto quanto sabemos, o AODV-DM (*Ad hoc On-Demand Distance Vector - Decoupled Multipath*) [4] é o único protocolo de encaminhamento reactivo multi-rota que procura explicitamente o desacoplamento entre rotas. Para este efeito o protocolo define uma região de isolamento à volta da rota primária. A rota principal é descoberta através da inundação de pacotes de RREQ (*route request*) pela rede. Múltiplos RREQ podem ser recebidos pelo destino através de rotas distintas. O nó destino responde ao primeiro RREQ (portanto, o RREQ que usou a rota mais curta) enviando um PRREP (*primary route reply*) pelo caminho inverso, criando a rota primária. Todos os nós que escutam o PRREP no seu percurso, marcam-se como

membros da região de isolamento. Depois de um tempo de espera, que pretende dar oportunidade da região de isolamento ficar definida, o nó de destino responde a outros RREQ usando pacotes SRREP (*secondary route reply*). Um pacote SRREP que necessite de ser encaminhado por um nó que faz parte da região de isolamento gera o envio de um pacote de RREJ (*route reject*), cancelando o estabelecimento dessa rota e despoletando a procura de rotas alternativas. Este protocolo possui a desvantagem de ser lento no estabelecimento das rotas e de apenas garantir a formação de duas rotas não interferentes.

Existem alguns protocolos pro-ativos multi-rota baseados no OLSR. O MOLSR [6] calcula múltiplas rotas, apesar de usar apenas uma em cada momento (pelo que o problema do acoplamento não se coloca). No QOLSR [7] procuram-se rotas que consigam satisfazer um conjunto de requisitos de largura de banda e latência. Estes caminhos possuem um acoplamento fraco mas não usam garantidamente zonas disjuntas. O protocolo MP-OLSR [8] calcula múltiplas rotas, que podem ser disjuntas em termos de nós ou elos, de acordo com várias funções de custo. Todos estes protocolos são baseados no OLSR, possuindo um núcleo de funcionamento comum que permite descobrir a topologia da rede. Em primeiro lugar, cada nó descobre os seus vizinhos através da troca periódica de pacotes HELLO. Posteriormente, cada nó difunde pacotes TC (*topology control*) pela rede. Estes anúncios permitem que todos os nós obtenham uma visão global da topologia. Finalmente, as rotas são calculadas localmente usando algoritmos de caminhos mais curtos adaptados para descartar rotas acopladas. Mesmo usando algumas optimizações para mitigar o custo de distribuição dos anúncios (por exemplo, recorrendo a *multipoint relays*), uma vez que todos os nós necessitam de difundir anúncios, estes protocolos incorrem em elevados custos de sinalização, em particular em redes densas.

Recentemente, em [9] foi sugerida a utilização de algoritmos de *clustering* para suportar o encaminhamento multi-rota. Nesse artigo, os autores sugerem a utilização destes algoritmos para agrupar os nós em conjunto que poderiam posteriormente ser usados para encontrar múltiplas rotas desacopladas. No entanto, o artigo não identifica quais os algoritmos de agrupamentos adequados ao efeito nem especifica nenhum protocolo em concreto para descobrir e procurar as rotas.

Os algoritmos de *clustering* têm sido utilizados para melhorar o desempenho das redes sem fios de acordo com vários critérios, incluindo a latência, consumo de largura de banda, consumo de energia, débito, e capacidade de escala [10]. Estes algoritmos dividem a rede em grupos disjuntos à volta de um nó eleito como líder do grupo. Os algoritmos de *clustering* podem ser muito diversos, dependendo dos cenários de aplicação. Por exemplo, os algoritmos LCA [11] e CLUBS [12] não possuem capacidade de escala pois exibem um tempo de convergência de $O(n)$. O algoritmo de Lowest ID [13] garante que os líderes não interferem entre si, o ACE [14] privilegia a formação de grupos uniformes, outros enviam a escolhas dos líderes de grupo de acordo com outros critério como a mobilidade, potência, energia disponível, entre outros [15]–[19].

Este artigo estende o trabalho de [9] em diversas dimensões. Em primeiro lugar propomos um algoritmo de *clustering* adaptado aos nossos objectivos, em que: i) todos os nós de um grupo estão no máximo a 2 saltos de qualquer outro nó do mesmo grupo (isto é suficiente para assegurar o desacoplamento dos líderes); ii) a sobreposição entre os agrupamentos é minimizada e; iii) o algoritmo é descentralizado, com baixa sinalização e rápida convergência. Para além disso propomos algoritmos concretos para descobrir e manter rotas que exploram os agrupamentos resultantes do passo anterior.

III. ENCAMINHAMENTO MULTI-ROTA

Nesta secção passamos a descrever um novo protocolo para Encaminhamento Multi-Rota com Baixo Acoplamento baseado em Clustering, que futuramente designamos apenas por EMR-BAC. O protocolo possui um módulo de agrupamento e um módulo de encaminhamento, ambos com componentes de criação e manutenção.

Como referimos na secção anterior, o protocolo EMR-BAC é baseado nas ideias sugeridas em [9], de forma a evitar as limitações do AODV-DM, nomeadamente a grande latência na descoberta de rotas e o número limitado de rotas descobertas. O EMR-BAC usa um módulo de *clustering*, encarregue de agrupar os nós da rede e eleger os líderes de cada grupo. Este módulo evita que os líderes interfiram mutuamente, de forma a que a rede sobreposta definida pelos líderes possa ser usada para identificar rotas desacopladas. Um segundo módulo é usado para descobrir as rotas. Este módulo combina facetas pró-activas e reactivas. Cada um destes módulos será descrito de forma pormenorizada nas próximas sub-secções.

A. Agrupamento de Nós

De forma a poder ser usado para o encaminhamento multi-rota de forma eficiente, o algoritmo de *clustering* deve atender aos seguinte requisitos: i) os agrupamentos devem ser uniformes, de forma a que a sua distribuição se aproxima de uma grelha hexagonal com baixa sobreposição; ii) os líderes de cada grupo não devem interferir mutuamente; iii) quando o algoritmo termina todos os nós pertencem a um grupo. Para este efeito, desenvolvemos um algoritmo com base no ACE (*Algorithm for Cluster Establishment*) [14], uma vez que este último oferece uma boa distribuição dos grupos.

O algoritmo usa duas fases: uma fase de criação de grupos e uma fase de manutenção. A primeira fase, cria os agrupamentos e elege os líderes de cada grupo. A segunda fase tem por objectivo manter os agrupamentos na presença de falhas ou mobilidade.

1) *Formação dos Grupos*: O algoritmo de formação de grupos executa-se em dois passos. O primeiro controla a eleição dos líderes de cada grupo e a formação de grupos temporários. O segundo passo promove a migração de nós entre agrupamentos para reduzir a sobreposição entre grupos.

O algoritmo não pressupõe a execução sincronizada dos diversos nós, pelo que cada nó pode começar a executar o algoritmo em instantes distintos. Durante a execução do

protocolo, os nós respondem de imediato a mensagens enviadas por outros nós mas só iniciam interações em intervalos aleatórios de forma a evitar colisões. Uma sequência de mensagem-respostas iniciada por um nó é designada uma iteração. Em consequência da execução de várias iterações, os nós podem transitar entre os seguintes estados: *líder*, *isolado*, ou *agrupado*. No estado inicial, todos os nós se encontram isolados. Um agrupamento é definido por um líder e por um conjunto de nós associados a esse líder no estado agrupado. Um nó pode encontrar-se associado a mais que um líder.

Durante a execução do algoritmo, os grupos são criados deixando que os nós se auto-proponham como líderes de novos grupos. O algoritmo que regula esta eleição é aplicado em cada iteração e como referimos, cada nó inicia uma iteração depois de esperar um tempo aleatório após a última iteração. As suas acções na nova interação dependem do seu estado, conforme descrito abaixo.

- **isolado:** um nó neste estado sonda os seus vizinhos para determinar quais são os seus *súbditos leais*. Um súbdito leal é um nó que está associado a um único líder. Quando o número dos seus súbditos leais excede um patamar pré-definido, um nó declara-se líder, e gera um identificador para o seu grupo. Posteriormente, o nó difunde uma mensagem de RECRUTAMENTO para os seus vizinhos. Os nós que recebem esta mensagem tornam-se súbditos do novo líder.
- **líder:** neste estado, o nó tenta aferir se a migração da liderança do grupo para um nó seu vizinho poderia melhorar a distribuição dos agrupamentos. Uma mensagem de SONDAGEM é enviada pelo líder para todos os seus vizinhos, de forma detectar vizinhos com um maior número de *súbditos leais*. A existência de um vizinho nessa situação leva o nó a abdicar de ser líder. Seja a o nó que abdica e l o nó que permanece líder. O nó l irá ficar com mais súbditos (nomeadamente, com o nó a e todos os outros nós que estejam ao alcance do novo líder l). A migração é iniciada por a que envia uma mensagem de PROMOÇÃO para l o qual, por sua vez, irá enviar uma mensagem de RECRUTAMENTO. Quando a recebe esta última mensagem, envia uma última mensagem de ABDICAÇÃO, de forma que os seus súbditos que não estejam ao alcance de l possam escolher novos líderes.
- **agrupado:** neste estado, o nó abstém-se de iniciar uma iteração.

As trocas de mensagens descritas acima repetem-se um número pré-determinado de iterações i . Este número é determinado experimentalmente, executando o algoritmo num conjunto representativo de cenários. No nosso caso, usamos 3 iterações, um valor que é coerente com os resultados reportados para o algoritmo ACE original [14]. Após a última iteração, todos os nós executam a seguinte rotina de término:

- Se um nó líder detecta a existência de outros líderes na sua vizinhança com maior número de *súbditos leais*, abdica (de forma a apenas permanecer líder aquele com um maior número de súbditos).

- Se o nó está agrupado, espera até que todos os líderes tenham terminado e torna-se súbdito do líder mais próximo. A distância ao líder é estimada a partir da força do sinal das mensagens enviadas por cada líder.
- Se o nó está isolado, o que significa que não existem líderes na sua vizinhança, o nó promove-se a líder.

Quando um nó termina a sua execução, envia uma mensagem de TÉRMINO, indicando o identificador do grupo a que pertence e o seu estado. Esta mensagem é difundida a 2 saltos e desta forma, no final do algoritmo, todos os nós conhecem não só os vários líderes ao seu alcance mas também os grupos com os quais podem interferir e quais os nós que os ligam a esses grupos. Ou seja, quando o algoritmo termina, todos os nós possuem informação sobre as potenciais fontes de interferência na sua vizinhança.

O nosso algoritmo difere do ACE em dois aspectos fundamentais: i) o ACE não garante que os líderes não interferem mutuamente e; ii) nós ao alcance de múltiplos líderes escolhem o seu agrupamento de forma aleatória, aumentando a sobreposição dos agrupamentos. Por exemplo, considere-se uma rede de 16 nós dispostos numa grelha 4×4 em que os nós estão ao alcance apenas dos seus vizinhos directos na grelha; o ACE elege os 4 nós centrais e os 4 nós colocados nos cantos como líderes, uma vez que esta é a configuração que maximiza o número de súbditos leais. Infelizmente, esta disposição não permite criar rotas não acopladas passando pelos nós líder.

2) *Manutenção dos Agrupamentos:* Nas redes sem fios a topologia pode ser modificada devido a nós falharem, abandonarem a rede, entrarem, ou simplesmente deslocarem-se. Desta forma, a configuração dos agrupamentos pode alterar-se, obrigando à execução de um procedimento de manutenção. Uma vez que estas alterações ocorrem com muito menos frequência nas redes sem fios em malha do que em redes ad hoc arbitrárias, optámos por simplesmente re-executar o algoritmo de agrupamento quando uma alteração à topologia é detectada.

Para detectar alterações à topologia, usamos uma estratégia reactiva que consiste em detectar a falha de elos durante a propagação de dados através de uma dada rota, evitando gerar tráfego de controlo exclusivamente para este fim. O nó que detecta a alteração à topologia, devido ao facto de não conseguir encaminhar um pacote para o próximo nó na rota escolhida, difunde por inundação, e para toda a rede, uma mensagem de REAGRUPAMENTO. Esta mensagem, re-inicia os nós no estado isolado e despoleta a re-execução do algoritmo de agrupamento.

B. Descoberta e Manutenção de Rotas

1) *Descoberta de Rotas:* A nossa solução combina facetas dos algoritmos pró-activos baseados no estado dos elos (*link-state*) com aspectos dos protocolo reactivos com encaminhamento na origem, de forma a combinar as vantagens das duas abordagens.

A faceta pró-activa consiste em manter em cada nó, uma base de dados do estado dos elos que caracterizam a rede sobreposta multi-salto que liga todos os líderes de grupos

adjacentes. Desta forma, tal como nos protocolos baseados no estado dos elos, os nós mantêm informação global sobre a topologia. No entanto, ao contrário desses algoritmos, a nossa solução apenas mantém estado sobre o a rede que liga os vários nós líder. Isto diminui de forma significativa os requisitos de memória e, sobretudo, o custo de sinalização necessário para manter esta informação actualizada. Note-se que apenas um ou dois nós são necessários para assegurar a ligação entre dois líderes de regiões adjacentes.

Para suportar a componente pró-activa do protocolo, quando o algoritmo de agrupamento termina, os nós líder (e apenas estes nós) difundem um anúncio com informação sobre as suas adjacências (ou seja, nós líder de grupos adjacentes), assim como quais os nós intermédios que permitem estabelecer a comunicação entre dois líderes (e que definem um elo virtual entre os dois vizinhos). Estas mensagens são propagadas por toda a rede e mantidas nas bases de dados de todos os nós. Como resultado, todos os nós ficam a conhecer a rede sobreposta que liga os líderes de grupo.

Usando a rede sobreposta referida acima, é possível encontrar os caminhos disjuntos entre um grupo de origem e um grupo de destino. Por construção, estes caminhos não são acoplados.

A componente reactiva está associada à descoberta do grupo em que se localiza o nó de destino e do subsequente cálculo das rotas entre o esse grupo e o grupo da fonte. Uma vez que o componente pró-activo só mantém informação sobre os nós líder, sempre que se pretende encaminhar para um nó súbdito, é necessário primeiro descobrir qual a sua filiação. Isto é feito recorrendo a uma variante simples de um protocolo RREQ/RREP em que os RREQ apenas são difundidos na rede sobreposta criada pelos nós líder. A resposta ao pedido de rota inclui o identificador do nó destino, o conjunto de agrupamentos na vizinhança desse nó, assim como os nós intermédios que permitem ao nó de destino comunicar com os líderes dos agrupamentos adjacentes. Após recolher esta informação, o nó de origem calcula de forma pró-activa as rotas para o destino.

O algoritmo de cálculo das rotas, utiliza a seguinte informação para descobrir rotas desacopladas:

- O conjunto de grupos na vizinhança da fonte. Este conjunto é designado por *GruposFonte*.
- O conjunto de grupos que permitem o acesso ao nó destino, obtido através da mensagem de RREP. Este conjunto é designado por *GruposDestino*.
- O conjunto de grupos que interferem com rotas que já tenham sido adoptadas, designado por *GruposExcluidos*.

Estas variáveis são usadas no cálculo das rotas da seguinte forma. Uma rota é aceite se o primeiro e último passo (excluindo a origem e o destino) pertencem aos conjuntos *GruposFonte* e *GruposDestino*, respectivamente, e nenhum dos restantes grupos na rota pertence ao conjunto de *GruposExcluidos*.

Quando uma rota é seleccionada, as variáveis anteriores são actualizadas da seguinte forma: o grupo de origem da rota é removido do conjunto *GruposFonte*, o grupo de destino é

removido do conjunto *GruposDestino*, os os restantes grupos da rota são adicionados ao conjunto *GruposExcluidos*.

Note-se que seria possível adaptar o protocolo para usar uma abordagem puramente pró-activa, fazendo com que os nós líder, nos anúncios de estado, enviassem também a filiação completa dos seus grupos. Isto evitaria a necessidade de executar o protocolo de descoberta do agrupamento onde se encontra o nó de destino. No entanto, não só as mensagens de estado seriam muito maiores, como teriam que ser trocadas de forma mais frequente, pois qualquer alteração a filiação dos grupo teria que ser anunciada, mesmo que os os líderes permanecessem inalterados.

2) *Encaminhamento Multi-Rota*: O algoritmo descrito anteriormente é capaz de descobrir as várias rotas desacopladas que podem ser estabelecidas com recurso aos nós líder. Dado que apenas o nó fonte calcula estas rotas, de forma reactiva, o nosso algoritmo utiliza encaminhamento na origem para encaminhar os pacotes pelas diversas rotas. Desta forma nenhuma informação adicional necessita de ser mantida nos restantes nós do sistema.

O modo como a aplicação usa as múltiplas rotas é, de certa forma, ortogonal à nossa solução. Em todas as experiências reportadas neste artigo, a fonte usa as várias rotas disponíveis usando um algoritmo de *round-robin*, distribuindo os pacotes por todas as rotas de forma uniforme.

3) *Manutenção das Rotas*: Consideramos três eventos distintos que podem obrigar as rotas a ser recalculadas, nomeadamente: a fonte dos pacotes move-se; o nó de destino move-se; a topologia da rede sobreposta definida pelos nós líder é alterada.

Se a fonte se move mas a rede sobreposta se mantém inalterada, apenas o nó fonte necessita de recalculer novas roas para os pacotes. Mais nenhuma acção é necessária.

Se o nó de destino se move e se torna súbdito de outro líder, necessita de enviar para a fonte uma mensagem de MOBILIDADE, equivalente à resposta a um pedido de rota. Mais uma vez, a fonte recalcula as rotas, baseada na nova informação sobre os grupos adjacentes ao nó de destino.

Finalmente, caso existam alterações ao nível dos nós líder, é necessário re-executar o algoritmo de agrupamento, actualizar as bases de dados de estado dos elos e recalculer novas rotas sobre a topologia actualizada. Como referimos, assumimos que estes eventos serão relativamente raros numa rede sem fios em malha.

IV. AVALIAÇÃO

Para avaliar o desempenho do protocolo EMR-BAC recorreu-se ao simulador NS-2, tendo-se desenvolvido uma concretização do EMR-BAC para este simulador. De forma a comparar a nossa solução com as existentes, desenvolveu-se também uma versão do AODV-DM [4] assim como uma versão do protocolo OLSR [20] modificada para descobrir vários caminhos não interferentes entre um nó fonte e um nó destino (designada OLSR+). Uma vez que o OLSR+ consegue construir o grafo representativo de toda a rede, os caminhos descobertos são óptimos do ponto de vista de

número de saltos (considerando que o caminho mais curto é sempre usado) e o número de caminhos descobertos é o máximo possível para a rede em causa.

A avaliação centra-se sobre três aspectos: a qualidade das rotas descobertas, o custo de sinalização necessário para descobrir estas rotas e a latência na descoberta das rotas.

Para avaliar estas métricas, criámos vários cenários distintos, colocando 120 nós aleatoriamente num espaço com dimensão $1500m \times 1500m$. A rede é estática e o tempo de simulação em cada caso foi de 100s. Neste cenários é sempre possível encontrar pelo menos duas rotas não acopladas. Para cada um destes cenários executaram-se os três algoritmos. De seguida apresentam-se os resultados registados em cada um dos casos.

A. Número de Saltos dos Caminhos Descobertos

A qualidade das rotas é aferida usando como métrica o número de saltos dos caminhos descobertos, dado que a largura de banda e fiabilidade de um caminho são inversamente proporcionais ao seu comprimento. Em particular, estamos interessados em mostrar que as rotas descobertas pelo EMR-BAC não induzem caminhos significativamente mais longos que o OLSR+ ou o AODV-DM.

Nesta análise consideramos apenas os dois melhores caminhos uma vez que, como discutido em [21], na maioria das redes, a probabilidade de encontrar um terceiro caminho não interferente de comprimento útil é relativamente baixa. É também com este número de caminhos que o mesmo estudo reporta vantagens mais significativas na largura de banda oferecida à aplicação.

Como se pode observar pelos resultados apresentados na Tabela I, não existem diferenças significativas entre o comprimento das duas rotas encontradas pelo EMR-BAC e os outros protocolos. Note-se que o OLSR+, ao possuir informação total sobre a topologia, encontra sempre em primeiro lugar o caminho mais curto (sem restrições) e, de seguida, o caminho mais curto que não interfere com o anterior. Por sua vez, o AODV-DM encontra esses mesmos caminhos uma vez que o caminho mais curto é usado como base para eliminar nós que interferem com esse mesmo caminho, sendo que a segunda rota encontrada tem o número mínimo de hops possível. No caso do protocolo por nós proposto nem sempre isto se verifica já que a restrição que obriga a que os caminhos passem pelos líderes dos grupos que atravessam pode aumentar ligeiramente (1 ou 2 saltos) o tamanho dos caminhos descobertos.

B. Tráfego de Sinalização

Para calcular os custos de sinalização, contabiliza-se todo o tráfego de controlo necessário para a descoberta de rotas, somando as componentes pró-activas e reactivas de cada protocolo. Note-se que o OLSR+ só possui uma componente pró-activa, o AODV-DM só possui uma componente reactiva, e o EMR-BAC possui ambas as componentes. Note-se também que o custo da componente reactiva depende do comprimento da rota, pois parte da execução de alguns protocolos só envolve os nós que potencialmente possam fazer parte de uma rota.

| Caminho | OLSR+ | | EMR-BAC | | AODV-DM | |
|-----------|-------|----|---------|----|---------|----|
| | 1 | 2 | 1 | 2 | 1 | 2 |
| Cenário 1 | 3 | 5 | 3 | 6 | 3 | 5 |
| Cenário 2 | 4 | 10 | 5 | 11 | 4 | 10 |
| Cenário 3 | 5 | 7 | 5 | 7 | 5 | 7 |
| Cenário 4 | 6 | 8 | 6 | 8 | 6 | 8 |
| Cenário 5 | 7 | 9 | 7 | 9 | 7 | 9 |

Tabela I
COMPARAÇÃO DO NÚMERO DE SALTOS DE CADA CAMINHO NOS DIFERENTES PROTOCOLOS.

| Custos Fixos | | | Custos Variáveis | | | |
|--------------|---------|---------|------------------|----|---------|------|
| OLSR+ | EMR-BAC | AODV-DM | EMR-BAC | | AODV-DM | |
| | | | 4 | 6 | 4 | 6 |
| 934591 | 63682 | 0 | 59 | 64 | 1403 | 1662 |

Tabela II
CUSTOS DE SINALIZAÇÃO (EM NÚMERO DE MENSAGENS).

Os resultados apresentados na Tabela II dizem respeito a um cenário em que se realizaram duas descobertas de rota entre nós que distam 4 e 6 saltos e se descobriram 2 caminhos para cada uma das rotas. Nas colunas da esquerda são apresentados os custos fixos de cada protocolo e nas colunas da direita os custos variáveis, dependendo do tamanho dos caminhos encontrados. Como se pode observar, o EMR-BAC, possui um custo fixo muito menor que o OLSR+ e um custo variável muito inferior ao AODV-DM.

Qualquer protocolo pró-activo só se justifica se o custo fixo puder ser amortizado através de várias descobertas de rotas. A Figura 1 apresenta o custo total aproximado de sinalização dos protocolos, quando se estabelece um número crescente de rotas. Estes pedidos de rota devem acontecer no período entre o refrescamento da informação recolhida pró-activamente, 5 segundos neste caso. Isto permite identificar, qual o padrão de utilização da rede para a qual cada protocolo é mais favorável. Como se pode ver, a partir da descoberta de 40 rotas (ou 8 rotas por segundo), o EMR-BAC já é mais vantajoso que o AODV-DM, sendo que o OLSR+, devido ao seu elevado custo fixo, só é vantajoso se o número de pedidos de rota exceder os 13600 no intervalo considerado.

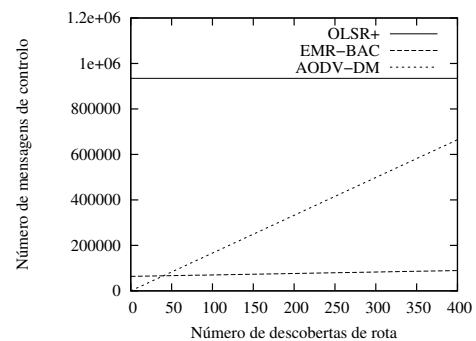


Figura 1. Sinalização vs número de rotas.

C. Tempo para a Descoberta de Duas Rotas

Finalmente, medimos o tempo necessário para descobrir duas rotas não interferentes, em função do comprimento dessas rotas. Obviamente, o protocolo OLSR+, uma vez que armazena toda a informação necessária para calcular as rotas localmente, só necessita de tempo de computação, que consideramos desprezável face aos tempos gastos na troca de mensagens necessárias nos protocolos reactivos. Desta forma, na Figura 2 só apresentamos resultados para o EMR-BAC e o AODV-DM. Como se pode ver o EMR-BAC é significativamente mais rápido a calcular as rotas que o AODV-DM (mais de 10 vezes mais rápido). Estes resultados devem-se ao facto do número de mensagens associadas à componente reactiva do EMR-BAC ser muito inferior ao AODV-DM (os RREQs são difundidos apenas pelos líderes e apenas é enviada uma mensagem de resposta - RREP). No caso do AODV-DM o número de RREQs difundidos é muito superior e são necessárias mais mensagens para formar a região de isolamento e garantir que as rotas formadas não incluem nós dessa região.

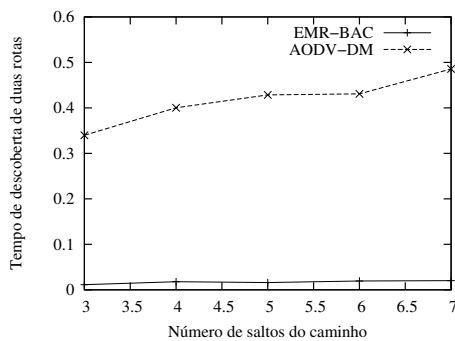


Figura 2. Tempo que cada protocolo demora a obter duas rotas.

V. CONCLUSÕES

Neste artigo apresentamos o EMR-BAC, um novo protocolo para suportar encaminhamento multi-rota em redes sem fios em malha. Estas rotas possuem baixa interferência, sendo calculadas com recurso a uma rede sobreposta de nós líder que resulta de um algoritmo de *clustering* desenhado para efeito. O protocolo combina facetas pró-activas (associadas com manutenção da informação sobre a rede sobreposta) e reactivas (relacionadas com a descoberta de rotas após a localização do destino). Os resultados experimentais mostram que o nosso algoritmo possui um custo de sinalização baixo e encontra um número de rotas desacopladas semelhante à abordagem óptima.

A actual solução assume que as redes em malha são pouco dinâmicas, e que a topologia definida pelo algoritmo de agrupamento é relativamente estável. Para suportar cenários com maior dinamismo, os algoritmos de manutenção poderiam ser melhorados. Seria também interessante explorar a nossa solução para otimizar o funcionamento das redes em malha, mesmo nas situações em que o encaminhamento multi-rota

não é necessário (por exemplo, para minimizar a interferência entre rotas independentes).

Agradecimentos: Este trabalho foi parcialmente suportado pela FCT (financiamento plurianual do INESC-ID) através do PIDDAC e através do projecto “Redico” (PTDC/EIA/71752/2006).

REFERÊNCIAS

- [1] N. S. Nandiraju, D. S. Nandiraju, and D. P. Agrawal, “Multipath routing in wireless mesh networks,” in *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, 2006, pp. 741–746.
- [2] I. Akyildiz and X. Wang, “A survey on wireless mesh networks,” *Communications Magazine, IEEE*, vol. 43, no. 9, pp. S23–S30, Sept. 2005.
- [3] S.-J. Lee and M. Gerla, “Split multipath routing with maximally disjoint paths in ad hoc networks,” in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 10, 2001, pp. 3201–3205 vol.10.
- [4] X. Hu and M. J. Lee, “An efficient multipath structure for concurrent data transport in wireless mesh networks,” *Comput. Commun.*, vol. 30, no. 17, pp. 3358–3367, 2007.
- [5] M. Pearlman, Z. Haas, P. Sholander, and S. Tabrizi, “On the impact of alternate path routing for load balancing in mobile ad hoc networks,” in *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*, 2000, pp. 3–10.
- [6] S. Xuekang, G. Wanyi, X. Xingquan, X. Baocheng, and G. Zhigang, “Node discovery algorithm based multipath olsr routing protocol,” *Information Engineering, International Conference on*, vol. 2, pp. 139–142, 2009.
- [7] H. Badis and K. Al Agha, “QOLSR, QoS routing for ad hoc wireless networks using OLSR,” *European Transactions on Telecommunications*, vol. 16, no. 5, pp. 427–442, 2005.
- [8] J. Yi, A. Adnane, S. David, and B. Parrein, “Multipath Optimized Link State Routing for Mobile ad hoc Networks,” *Ad Hoc Networks*, 2010.
- [9] G. Y. L. H. J. K. Jie Zhang, Choong Kyo Jeong, “Cluster-based multipath routing algorithm for multi-hop wireless network,” *International Journal of Future Generation Communication and Networking*, 2008.
- [10] S. Chinara and S. Rath, “A Survey on One-Hop Clustering Algorithms in Mobile Ad Hoc Networks,” *Journal of Network and Systems Management*, vol. 17, no. 1, pp. 183–207, 2009.
- [11] D. Baker and A. Ephremides, “The architectural organization of a mobile radio network via a distributed algorithm,” *Communications, IEEE Transactions on [legacy, pre - 1988]*, vol. 29, no. 11, pp. 1694–1701, 1981.
- [12] R. Nagpal and D. Coore, “An algorithm for group formation in an amorphous computer,” in *Proc. 10th International Conference on Parallel and Distributed Computing Systems*, 1998.
- [13] C. Lin and M. Gerla, “Adaptive clustering for mobile wireless networks,” *IEEE Journal on Selected areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.
- [14] H. Chan and A. Perrig, “Ace: An emergent algorithm for highly uniform cluster formation,” in *Proceedings of the First European Workshop on Sensor Networks (EWSN)*, 2004, pp. 154–171.
- [15] A. Parekh, “Selecting routers in ad-hoc wireless networks,” in *Proceedings SBT/IEEE Intl Telecommunications Symposium*, 1994, pp. 420–424.
- [16] P. Basu, N. Khan, and T. Little, “A mobility based metric for clustering in mobile ad hoc networks,” *icdcs*, p. 0413, 2001.
- [17] S. Basagni, “Distributed clustering for ad hoc networks,” in *ispan*. Published by the IEEE Computer Society, 1999, p. 310.
- [18] S. Basagni, I. Chlamtac, and A. Farago, “A generalized clustering algorithm for peer-to-peer networks,” in *Workshop on Algorithmic Aspects of Communication*, 1997.
- [19] M. Chatterjee, S. Das, and D. Turgut, “WCA: A weighted clustering algorithm for mobile ad hoc networks,” *Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.
- [20] P. Jacquet, P. Mühlthaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, “Optimized link state routing protocol for ad hoc networks,” in *Proceedings of the 5th IEEE Multi Topic Conference (INMIC 2001)*, 2001.
- [21] Y. Liaw, A. Dadej, and A. Jayasuriya, “Throughput performance of multiple independent paths in wireless multihop network,” in *IEEE International Conference on Communications*, vol. 7, jun. 2004, pp. 4157 – 4161.