

BiO4SeL: Um Algoritmo Autônomo Biologicamente Inspirado para Roteamento em Redes de Sensores Sem Fio

Levi Bayde Ribeiro¹, Miguel Franklin de Castro¹

¹Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)
Departamento de Computação – Universidade Federal do Ceará – Fortaleza-CE

{levi,miguel}@great.ufc.br

Abstract. *One of the main challenges in a Wireless Sensor Network (WSN) is to make its organization and routing energetically efficient. It is required to perform these task in an autonomic and self-sustainable way. This paper presents BiO4SeL (Bio-inspired Optimization for Sensor network Lifetime), a self-organization and routing algorithm for WSN in which the main objectives are autonomy and network lifetime optimization. BiO4SeL is based on Swarm Intelligence, more specifically in Ant Colonies. Results show that BiO4SeL accomplishes its objectives when compared to other algorithms used for WSN (and ad-hoc) routing: AODV and ARAMA.*

Resumo. *Um dos principais desafios das Redes de Sensores Sem Fio (RSSF) é realizar de forma energeticamente eficiente sua organização e o seu roteamento. É necessário que estas tarefas sejam executadas de forma autônoma e auto-sustentável. Este artigo propõe o BiO4SeL (Bio-inspired Optimization for Sensor network Lifetime), um algoritmo de auto-organização e roteamento para RSSF, cujos principais objetivos são autonomia e otimização do tempo de vida da rede. O BiO4SeL é baseado em Inteligência de Enxames, mais especificamente em Colônias de Formigas. Os resultados mostram que o BiO4SeL atinge seus objetivos quando comparados a outros algoritmos utilizados para roteamento em RSSF: AODV e ARAMA.*

1. Introdução

No contexto de Computação Ubíqua e Pervasiva, sensores são dispositivos computacionais importantes e potencialmente bastante numerosos. Eles coletam informação do ambiente, como temperatura, movimentação, humidade, dentre outros, e comunica-se de forma direta ou através de seus vizinhos para enviar seus dados até a Estação Base (EB, ou BS, *Base Station*), ou sorvedouro (*sink*). Eles podem ser utilizados em uma grande variedade de aplicações. Por exemplo, podem ser distribuídos em uma floresta para monitoramento de incêndios.

Uma das mais abordadas questões em redes de sensores é seu tempo de vida, ou seja, o tempo de utilização da rede até o primeiro nó ficar sem energia. Durante a operação, alguns nós podem ser mais requisitados que outros para encaminhar a informação de outros nós. Isto poderia gerar locais sem cobertura na área de abrangência da rede. Além disso, os nós devem fazer balanceamento de carga para maximizar seu tempo de vida.

Manter a cobertura da rede completa pelo máximo de tempo possível é o objetivo de diversos algoritmos de roteamento para redes de sensores. Este problema é particularmente importante em aplicações críticas, como monitoramento de radioatividade, pois nelas as áreas “cegas” são inaceitáveis. Soluções para este problema, ou para parte dele, podem ser encontradas na literatura. A maioria delas é baseada no nível de roteamento. Muitas delas, entretanto, têm fortes restrições, como conhecimento de localização, processamento centralizado, etc.

Este artigo apresenta o BiO4SeL (*Biologically-inspired Optimization for Sensor network Lifetime*, ou Otimização Biologicamente Inspirada para o Tempo de Vida de Redes de Sensores), um protocolo de roteamento baseado em colônias de formigas distribuído e autônomo. Seu objetivo é maximizar o tempo de vida em redes de sensores. Para isto, aplica o conceito Inteligência de Enxame, formando inteligência a partir de comportamento rudimentar distribuído pelos sensores.

BiO4SeL, assim como outras abordagens, inspira-se na idéia de criar caminhos na rede para enviar os dados dos nós para a EB, distribuindo o consumo de energia pelos nós ao longo deste caminho. BiO4SeL, entretanto, diferencia-se de outros algoritmos por utilizar informação de nível energético de bateria não somente para construir a tabela de roteamento inicialmente, mas também para atualizá-la a medida que a energia é consumida. Em consequência disto, a responsabilidade pelo encaminhamento de pacotes tende a ser igualmente distribuída. Assim, mantém-se a variância do mapa energético da rede o menor possível, aumentando o tempo de vida da rede. Experimentos de simulação comparando BiO4SeL ao ARAMA (*Ant-based Routing Algorithm for Manets*, ou Algoritmo de Roteamento baseado em Formigas para RSSF) [Hussein et al. 2005] e AODV (*Ad-hoc On-demand Distance Vector*, ou Vetor de Distância Ad-hoc sob demanda) [Perkins et al. 2003] mostram como o BiO4SeL alcança seus objetivos, apresentando melhor organização de inicialização da rede e cenários mais densos (testados até 320 nós).

O restante deste artigo está organizado como segue. A Seção 2 aborda alguns trabalhos relacionados em roteamento para redes de sensores. Na Seção 3, a abordagem BiO4SeL é introduzida e sua especificação, mostrada. A Seção 4 traz a descrição dos experimentos e a discussão sobre seus resultados, enquanto a Seção 5 contém as conclusões e propostas para trabalhos futuros.

2. Trabalhos Relacionados

Inicialmente, Inteligência de Enxames foi aplicada a grafos como uma abordagem de Inteligência Artificial para a solução do problema do Caixeiro Viajante utilizando-se formigas [Dorigo et al. 1991]. Depois disto, este paradigma tornou-se cada vez mais popular. No domínio de redes de comunicações, ele foi inicialmente utilizado para o roteamento de pacotes, com o algoritmo AntNet [Caro and Dorigo 1997]. Fazendo-se um paralelo entre a Natureza e as redes, para algoritmos de roteamento, cada formiga é um pacote de controle. Quando um caminho deve ser criado de um nó para outro, algumas formigas são enviadas. Elas movem-se e, probabilisticamente, selecionam o próximo salto, baseadas na distribuição feromônica ao longo do caminho ou em alguma outra heurística do caminho. Quando a formiga alcança seu destino, ela calcula o feromônio a ser depositado, baseado nesta heurística. Depois disso, a formiga é enviada de volta. Para seguir os mesmos saltos, elas mantêm uma lista dos nós visitados durante sua ida até o destino. Durante

seu retorno, elas aprimoram os feromônios em cada salto, aumentando a probabilidade de aquele caminho ser o escolhido para as próximas formigas. Este algoritmo é baseado em tabelas de roteamento probabilísticas.

O ARAMA (*Ant-based Routing Algorithm for Manets*, ou Algoritmo de Roteamento baseado em Formigas para RSSF) [Hussein et al. 2005] utiliza estas tabelas de roteamento probabilísticas para enviar pacotes em redes Ad-hoc. A principal diferença entre esta estratégia e o AntNet é a influência da quantidade energética das baterias dos nós influenciarem no cálculo da deposição feromônica.

Uma variante do ARAMA, o AOER (*Ant-Based On-Demand Energy Routing*, ou Roteamento baseado em Formigas por Demanda Energética) [Shuang et al. 2007], almeja ajustar o roteamento para aumentar o tempo de redes Ad-hoc. Nesta abordagem, as formigas coletam informação sobre a potência de bateria remanescente em cada nó, assim como a capacidade de reenvio e contagem de saltos. Esta informação é utilizada para construir uma tabela de roteamento probabilística inversa. Ela é inversa, pois quanto mais formigas seguem pelo caminho, menos poder de energia ele tem. Além disso, uma formiga não guarda o caminho visitado, reduzindo seu tamanho e, conseqüentemente, a energia consumida para enviá-la, recebê-la e processá-la. Infelizmente, o artigo não provê detalhes de implementação suficientes que possam permitir uma comparação com o BiO4SeL.

Além de formigas, outras estratégias podem ser encontradas na literatura para economia energética em redes de sensores. Uma destas estratégias, também relacionada ao BiO4SeL, é o balanceamento de carga, uma solução herdada do domínio de telecomunicações. Utilizar alguma infraestrutura torna este processo mais simples. Por exemplo, abordagens encontradas em [Akyildiz et al. 2002, Al-Karaki and Al-Mashaqbeh 2007, Iqbal et al. 2006] empregam agrupamento (*clustering*) para conseguir agregação de dados e revezamentos de nó cabeça, conseqüentemente economizando gasto energético.

Após um profundo estudo e comparação de vantagens e desvantagens destas abordagens, este artigo propõe um novo protocolo de roteamento baseado em formigas, chamado BiO4SeL (*Biologically-inspired Optimization for Sensor network Lifetime*, ou Otimização Biologicamente Inspirada para o Tempo de Vida de Redes de Sensores). Seu objetivo é aumentar o tempo de vida de uma Rede de Sensores, como descrito na próxima sessão.

3. O Protocolo BiO4SeL

Esta seção descreve o protocolo de roteamento BiO4SeL (*Biologically-inspired Optimization for Sensor network Lifetime*, ou Otimização Biologicamente Inspirada para o Tempo de Vida de Redes de Sensores), assim como sua implementação na ferramenta Network Simulator (NS-2) [NS-2 Webpage 2010], versão 2.32 (também testado na versão 2.33). Assume-se cada nó ter uma identificação única e utilizar o IEEE 802.15.4 [IEEE 2003] nas camadas PHY e MAC. Nas simulações, os identificadores dos nós são números começando por 0.

Em resumo, o protocolo funciona como segue. Compreende-se por três estágios: Reconhecimento (*Bootstrap*), Descoberta Inicial de Rotas e Troca de Dados. No primeiro estágio, cada nó reconhece seus vizinhos utilizando uma sinalização específica. Após

isto, na fase de Descoberta Inicial de Rotas, formigas são utilizadas para criar as rotas iniciais na rede, de cada nó até a EB. Assim, na fase de Troca de Dados, dados de aplicação começam a fluir, levando formigas de carona (*piggyback*) para aprimorar a probabilidade das rotas. A medida que os dados são encaminhados, a energia das baterias dos sensores ao longo dos melhores caminhos diminui mais rápido que em outros nós. Uma das diferenças entre o BiO4SeL e outras abordagens baseadas em formigas é o fato de levar esta energia em consideração para a evaporação feromônica. Quanto menos poder remanescente houver na bateria, mais rápido o feromônio no nó decrementa, tornando este nó menos provável de ser escolhido. Isto causa a seleção de rotas menos escolhidas, mas com mais energia, mesmo formando um caminho mais longo (mais saltos). O pseudocódigo deste protocolo é mostrado no Algoritmo 1.

Algorithm 1 Algoritmo do Protocolo BiO4SeL

```

1: {Fase de Reconhecimento}
2: nós enviam ihello
3: if nó recebe ihello then
4:   guarda informação de bateria do vizinho
5: end if
6:
7: {Fase de Descoberta Inicial de Rotas}
8: nós criam suas tabelas de roteamento
9: EB envia pacotes iant
10: if nó recebe iant then
11:   diminui a estimativa energética dos vizinhos
12:   if já recebida (mesmo identificador) then
13:     descarta iant
14:   else
15:     if saltos do pacote < distância atual à EB then
16:       atualiza distância à EB
17:     else
18:       decrease feromônio
19:     end if
20:   end if
21: end if
22:
23: {Fase de Troca de Dados}
24: eventualmente, cada nó envia hello
25: eventualmente, cada nó envia dados
26: while nó recebe pacote do
27:   decrementa estimativa energética dos vizinhos
28:   if pacote = hello then
29:     if previamente na tabela de roteamento then
30:       conserta estimativa energética dos vizinhos
31:     else
32:       envia requestHello
33:     end if
34:   end if
35:   if pacote = requestHello then
36:     guarda informação de bateria
37:     envia respRqHello
38:   end if
39:   if pacote = respRqHello then
40:     guarda informação de bateria
41:   end if
42:   if pacote = dados then
43:     if previamente recebido then
44:       descarta
45:       envia formiga negativa no caminho inverso
46:     else
47:       calcula próximo salto
48:       ajusta feromônio
49:       encaminha dados
50:     end if
51:     if EB then
52:       descarta
53:     end if
54:   end if
55:   if pacote = formiga negativa then
56:     decrementa feromônio
57:     if primeiro nó do laço then
58:       descarta
59:     else
60:       encaminha
61:     end if
62:   end if
63: end while

```

3.1. Fase de Reconhecimento

Esta fase inicia-se quando o sensor é ligado. Neste momento, ele envia uma mensagem `ihello` para todos os seus vizinhos, ou seja, uma mensagem de inundação com Tempo de vida (TTL) igual a 1. Ela é composta pelo tipo do pacote, `hlTp`, e sua própria informação de bateria. Esta informação muda de um tipo de sensor para outro, e é composta de energia inicial (`InitEn`) e taxa de consumo específicas do modelo nos possíveis estados do sensor: ocioso (`IdEnCte`), envio (`SndEnCte`) e recepção (`RcvEnCte`).

Assim como o BiO4SeL leva em consideração a energia do vizinho para calcular o feromônio relacionado a ele, a informação de bateria de cada vizinho é mantida e atualizada em cada nó.

3.2. Fase de Descoberta Inicial de Rotas

Esta fase consiste em inundar a rede com uma quantidade configurável *iant_cnt* de formigas (*iant* packets) de cada EB para a rede. Elças são compostas pelo tipo de pacote (*iaTp*) e identificador de inundação (*broadId*). Este identificador, juntamente com o Id do nó fonte, é armazenado em cada salto ao longo do caminho. Se um nó receber a mesma *iant* novamente, ela não será reencaminhada (detecção de *loops*).

Estas formigas agem diferentemente das de outros protocolos. Na primeira chegada de uma *iant* a cada nó, ela salva a distância do nó até a EB, com o próximo nó sendo o nó do qual ela veio. Além disso, ela mantém, na tabela de roteamento, o mínimo de saltos para alcançar a EB. A partir da segunda chegada, elas podem modificar o feromônio em certos saltos. Ainda que incrementar feromônios seja impossível nesta fase, eles podem ser decrementados se um nó receber uma *iant* de um vizinho com distância (saltos) a EB maior que a guardada no nó. Isto representa uma formiga andando na direção da EB e não procurando por nós remanescentes. Como consequência, a probabilidade de escolha desse salto seria reduzida, por pertencer a um caminho reverso ou maior que o necessário. Depois de algumas rodadas de *iants*, o feromônio é concentrado no sentido dos nós para a EB.

Quando ocorre a situação citada acima, a taxa de decréscimo feromônico em cada nó é calculada de acordo com a Equação 1, como segue:

$$F_d^v = F_d^v - \text{Min}\{K \times (2 - S_d^v); 0.9\} \times (F_d^v - F_{min}) \quad (1)$$

Sendo F_d^v o feromônio relacionado ao destino (sorvedouro) d pelo vizinho v , K o coeficiente arbitrário de decréscimo feromônico, S_d^v a relação de saltos do vizinho v para o destino d , e F_{min} é o mínimo feromônico para um salto, escolhido arbitrariamente.

A relação de saltos referida na Equação 1 é dada por $S_d^v = S_d^{min} / S^a$, sendo S_d^{min} o mínimo de saltos, dentre todos os vizinhos, para o destino d , dado por $S_d^{min} = \text{Min}\{S_d^v\}, \forall$ vizinho v , e S^a a quantidade de saltos da formiga. S_d^v representa o quão longe o caminho da formiga está do melhor caminho previamente alcançado. Assim, o feromônio em cada nó é diretamente influenciado por sua distância até a EB. Esta equação também é útil para evitar ciclos, pois caminhos mais longos são desencorajados.

Além disto, cada nó pode estimar a quantidade de potência de bateria disponível em cada vizinho ao longo do tempo. Isto é possível pelo fato de o nó saber como se processa o consumo energético deles pelas informações do *ihello*. Isto é feito para cada mensagem enviada ou recebida. Esta estimativa é útil para fazer uma redução feromônica mais aproximada nos saltos. Periodicamente, esta estimativa é atualizada com dados reais provenientes de formigas de carona.

Quando uma mensagem é enviada a um vizinho, a estimativa energética para ele é atualizada de acordo com a fórmula $E^v = E^v - (C_{rcp}^v \times FC_{rcp}^n) \times T$, sendo E^v a estimativa energética atual do vizinho v , C_{rcp}^v o coeficiente de consumo energético para o estado de recepção do vizinho v (informado no pacote *ihello*), FC_{rcp}^n o fator de correção de recepção do vizinho v , calculado na próxima fase e inicialmente 1, e T o tamanho total do pacote, em bits. Este fator de correção é utilizado para tornar a estimativa tão aproximada quanto possível, pois compara estimativas prévias com valores reais para

calcular as futuras.

De maneira similar, quando uma mensagem é recebida por um vizinho, a fórmula utilizada é $E^v = E^v - (C_{env}^v \times FC_{env}^v) \times T$, sendo C_{env}^v o coeficiente de consumo energético para o estado de envio do vizinho v (também contido no `ihello`), e FC_{env}^v o fator de correção de envio para o vizinho v , calculado na próxima fase e inicialmente 1.

Além disso, o nó mantém o *timestamp* (marcação de tempo) para a última mensagem recebida deste vizinho. Assim, ele pode estimar o consumo energético durante o tempo ocioso, com a fórmula $E^v = E^v - C_{oci}^n \times (TS_0 - TS_{-1})$, sendo C_{oci}^n o coeficiente de consumo energético no estado ocioso para o nó n , e TS_0 e TS_{-1} os *timestamps* para o atual e último pacotes recebidos, respectivamente.

BiO4SeL inclui uma estratégia de retransmissão de pacotes. Em um ambiente muito populoso, um reencaminhamento de `iant` poderia colidir e se perder. Desta forma, se, dentro de um tempo pré-definido, o nó não receber `iants` de todos os seus vizinhos, ele reenvia a formiga, com um *delay* aleatório. Isto acontece somente uma vez para cada nó e cada formiga.

3.3. Fase de Troca de Dados

Nesta fase, o foco é o envio de informação dos nós para a EB. Neste ponto, o caminho inicial e a tabela de roteamento já estão calculadas, somente esperando para os dados serem enviados. A cada envio de informação, a tabela é consultado e algum vizinho é escolhido probabilisticamente, baseado nos feromônios dos saltos. Quando o pacote é encaminhado, as energias e os feromônios são atualizados e ao nó continua esperando pelo próximo pacote.

A cada intervalo de tempo $\Delta\tau_a$, formigas de anúncio, ou `hellos`, são enviados de cada nó. Seu tempo de vida é um salto, e seu objetivo é informar aos vizinhos sobre o nó estar vivo e seu nível energético atual. Com estas informações, os vizinhos podem atualizar suas tabelas de roteamento. Se um nó receber esta mensagem de um vizinho já em sua tabela de roteamento, o nó atualiza-a com um novo tempo de expiração para o remetente e sua potência de bateria. Além disso, atualiza o fator de erro na estimativa de energia gasta. Para isto, utiliza as seguintes fórmulas:

$E_{rcp}^n = C_{rcp}^v \times Cnt_{rcp}^v$, sendo E_{rcp}^n o consumo energético de recepção de pacotes desde o último `hello`, e Cnt_{rcp}^v o contador de pacotes recebidos do vizinho v neste mesmo tempo.

De maneira semelhante, $E_{env}^n = C_{env}^v \times Cnt_{env}^v$, sendo E_{env}^n o consumo energético de envio de pacotes desde o último `hello`, e Cnt_{env}^v o contador de pacotes enviados ao vizinho v neste mesmo tempo.

Por fim, $E_{tot}^v = E_{rcp}^v + E_{env}^v$, sendo E_{tot}^v a energia total gasta desde o último `hello`.

Por fim, os coeficientes de correção podem ser calculados, como segue:

- $FC_{rcp} = FC_{rcp} + (((E_{rea}^v - E^v) \times (E_{rcp}^v / E_{tot}^v)) / Cnt_{rcp}^v)$
- $FC_{env} = FC_{env} + (((E_{rea}^v - E^v) \times (E_{env}^v / E_{tot}^v)) / Cnt_{env}^v)$

Sendo E_{rea}^v a energia real do vizinho v (informado no pacote `ihello`).

Se o nó receptor não tiver o remetente em sua tabela de roteamento, ele requisita, via pacote `requestHello`, ou pedido de anúncio de inicialização, o pacote `InitHello` contendo as informações do novo vizinho. Se o vizinho moveu-se, ele deve limpar sua vizinhança e substituí-la com uma nova. Para apagar os vizinhos antigos, há um temporizador para expirá-los.

O `InitHello` carrega a informação de bateria, de um modo similar ao *ihello*. Como o nó receptor não reconhece o nó enviando o `hello`, ele deve ter se movido para aquela localização. Recebendo o pacote, ele salva o perfil de seu novo vizinho, atualizando sua tabela de roteamento. Esta é uma mensagem em *unicast*, e de somente um salto. Após isto, ele responde com o `respRqHello`, resposta a requisição de anúncio de inicialização, um pacote de inundação e de somente um salto. Como todos ou, ao menos, algum vizinho será novo para ele, o novo poderá salvar suas informações. Além disso, todos receberam seus `hello`, e todos enviarão seus próprios `requestHello`. Desnecessário responder mais de uma vez. Assim, somente uma mensagem será enviada.

Esta fase, entretanto, é caracterizada pela transmissão de dados. Cada pacote de dados enviado vai juntamente com uma formiga de carona, `pbAnt`. Esta formiga é somente composta por um campo, o `pb_hop`. Ao enviar um pacote de dados, este campo leva a distância, em saltos, até a EB. Esta distância é utilizada na deposição feromônica, a cada nó.

As fórmulas para ela estão a seguir, nas quais todas as constantes arbitrárias e os pesos citadas foram obtidas durante os experimentos, com o objetivo de otimizar o protocolo.

$$F_d^v = F_d^v + ((maxF - F_d^v) \times (c + e) \times fc)$$

Sendo F_d^v o feromônio do vizinho v com relação ao destino d , $maxF$ o máximo feromônio possível em um salto, c a constante de caminho no cálculo feromônico, e a constante energética e fc a constante arbitrária associada ao cálculo.

Assim, o feromônio é adicionado de uma porcentagem do restante a ser colocado, ou seja, da subtração entre o máximo de feromônio e o atual. Esta porcentagem é calculada a partir de um balanço entre o quão bom o caminho é, ou seja, o menor possível (em saltos), e quanta energia o próximo salto possui. Utiliza-se, ainda, uma constante arbitrária para diminuir ou aumentar a mudança no feromônio.

A constante de caminho associada ao cálculo feromônico é calculada de acordo com as fórmulas $c = cc \times pc$ e $cc = minH / (cH + cnH)$, sendo cc a variável de caminho, pc o peso do caminho no cálculo, $minH$ a mínima quantidade de saltos até o destino a partir da fonte, calculado na fase de reconhecimento e contido na formiga de carona em cada pacote de dados, cH a quantidade de saltos contada pelo pacote, cnH o mínimo de saltos até o destino a partir do nó corrente, calculado na fase de reconhecimento e salvo no próprio nó.

A constante energética, por sua vez, utiliza as fórmulas $e = ce \times pe$, $pe = 1 - pc$ e $ce = E^v / EI^v$. Para elas, ce é a variável energética, pe o peso da energia no cálculo, E^v a estimativa atual da energia do vizinho v , EI^v a energia inicial do vizinho v . Assim, a constante arbitrária da energia (ce) é dada pelo restante de energia do vizinho, e o peso da energia (pe) é sempre o complemento do peso do caminho (pc).

Como consequência da atualização do feromônio em um salto, o feromônio total até aquele destino deve ser recalculado. Descreve este cálculo a equação $F_d = F_d - (F_d^v - F_d^v)$, sendo F_d o feromônio total para o destino d , F_d^v o feromônio ao destino d passando pelo vizinho v , e F_d^v o feromônio anterior deste salto, antes da atualização.

Ainda durante o reencaminhamento, a cada certa quantidade (EvC , variável obtida durante os testes de otimização do protocolo) de pacotes de dados recebidos de um vizinho v e indo para uma EB d , o feromônio somente naquele salto é evaporado. Para isto, utiliza-se a seguinte equação.

- $F_d^v = \min F + (Ev^{EvI} \times (F_d^v - \min F))$
- $Ev = E^v / EI^v$

sendo F_d^v o feromônio ao destino d passando pelo vizinho v , $\min F$ o mínimo feromônio possível, Ev a porcentagem de evaporação, EvI o índice de evaporação, E^v a estimativa atual de energia do vizinho v , EI^v a energia inicial do vizinho v .

Nesta fórmula, a porcentagem de evaporação feromônica é diretamente proporcional à energia restante no vizinho. Esta decisão foi tomada com o objetivo de fazer as rotas variarem, e diminuir a probabilidade de uma rota com pouca energia ser utilizada. Deste modo, espera-se aumentar o tempo de vida da rede como um todo, dividindo-se a carga pela rede inteira. Esta porcentagem é aplicada à diferença entre o feromônio atual e o mínimo. Além disto, é modificada pelo índice de evaporação, uma variável de configuração do protocolo, obtida por meio de testes. Tem o intuito de calibrar esta porcentagem para uma evaporação ideal. Este tipo de variável é utilizada em protocolos com este fim.

Novamente, como consequência da atualização (evaporação) do feromônio em um salto, o feromônio total até aquele destino deve ser recalculado, como na equação descrita.

Há uma chance, pE , de o salto não ser escolhido por esta probabilidade, mas pela maior energia restante nos vizinhos. Esta chance é dada pela fórmula $pE = 1 - (0,8 \times ce + 0,2 \times cf)$, sendo $cf = (F_d^v - \min F) / (\max F - \min F)$. Assim, quanto mais próximo estiver do máximo de energia ou do máximo feromônico, menor a probabilidade de haver uma escolha aleatória, pois é menor a sua necessidade. Isto somente ocorre quando o melhor caminho já está abaixo de dois terços de sua energia original. Esta é uma das técnicas utilizadas para garantir uma melhor distribuição das rotas pela rede.

Ao receber um pacote de dados, o nó salva uma entrada para o pacote, em uma fila. Esta entrada é similar a aquelas utilizadas pelas *ants*, consistindo no id do nó fonte e da mensagem e tempo de expiração. Também há um temporizador para eliminar entradas antigas. Ao receber um pacote de dados, o nó testa a presença da entrada. Se não estiver lá, é adicionada. Se ela estiver, a *negAnt* (*negative ant*, formiga negativa) é enviada pelo caminho inverso da mensagem. Esta formiga evapora o feromônio no ciclo, desencorajando futuras formigas a ir por este caminho.

4. Resultados e Discussão

Para verificar a efetividade do BiO4SeL, alguns experimentos foram feitos utilizando o simulador de redes NS-2. Nestes experimentos, foi utilizado o modelo de nós móveis existente, e ele foi estendido para adicionar um novo algoritmo de roteamento. A maioria dos parâmetros IEEE 802.15.4 foram deixados em seus parâmetros padrão.

Os outros parâmetros utilizados foram os seguintes.

Tabela 1. Parâmetros do protocolo

| Parâmetro | Valor |
|--|----------------------|
| Tempo de envio de cada <i>ihello</i> | $t = 0,01 \times id$ |
| Quantidade de formigas de inicialização (<i>iant</i>) enviadas | $iant_{cnt} = 5$ |
| Intervalo entre cada envio de <i>iant</i> | 0,5s |
| Intervalo entre cada envio de <i>hello</i> | 7,5s a 12,5s |
| Tempo de expiração do contador de <i>hello</i> | 13s |
| Tempo de expiração das entradas de <i>unicast</i> | 0,5s |
| Tempo de expiração do contador de entradas de <i>unicast</i> | 1s |
| Mínimo de feromônio (<i>minF</i>) | 0.000001 |
| Máximo de feromônio (<i>maxF</i>) | 0.01 |
| Feromônio inicial (<i>iF</i>) | 0.0001 |
| Intervalo de envio de <i>hello</i> (<i>hlInt</i>) | 10s |
| Expiração do contador de pacotes (<i>EvC</i>) | 2 pacotes |
| Índice de evaporação (<i>EvI</i>) | 3 |
| Peso do caminho (<i>pc</i>) | 0,1 |
| Constante arbitrária associada ao cálculo feromônico (<i>fc</i>) | 0,1 |
| Coefficiente de diminuição feromônica (<i>K</i>) | 0,6 |

Legenda: *id* - identificador único de cada nó.

Foram testados cenários com diferentes quantidades de nós. Para cada uma delas, foi experimentalmente definido o tamanho do cenário para ter, ao menos, 90% dos cenários aleatoriamente gerados plenamente conectados, ou seja, sem nós inalcançáveis. As configurações resultantes são apresentadas na Tabela 2.

Tabela 2. Cenários de simulação para os experimentos.

| Quantidade de nós | Tamanho do Cenário |
|-------------------|--------------------|
| 10 | 20x20 |
| 20 | 30x30 |
| 40 | 50x50 |
| 80 | 70x70 |
| 160 | 100x100 |
| 320 | 140x140 |

Nos experimentos, o BiO4SeL foi comparado a duas outras abordagens relacionadas. A primeira é o AODV (*Ad-Hoc On-Demand Distance Vector*) [Perkins et al. 2003], um padrão *de-facto* para protocolos de roteamento em MANETs, geralmente utilizado em comparações em outros trabalhos da mesma área encontrados na literatura. A outra é o ARAMA (*Ant-based Routing Algorithm for Manets*) [Hussein et al. 2005], formalmente apresentado na Seção 2, utilizado pela facilidade de reprodução, devido a quantidade de detalhes de descrição do protocolo no artigo.

Para cada configuração mostrada na Tabela 2, 100 cenários diferentes foram gerados, aleatoriamente. Para cada cenário, a simulação foi repetida 50 vezes, mudando-se as sementes de aleatoriedade. Apesar disso, para o AODV não houve repetição, pois ele é um algoritmo determinístico. Os resultados apresentados nesta seção incluem um intervalo de confiança considerando $\alpha = 0,95$.

Em cada cenário, há somente um transmissor (nó captando do ambiente) e somente um receptor (Estação Base ou sorvedouro), ambos localizados em cantos opostos

da área. Este tipo de operação é possível para aplicações nas quais eventos são monitorados baseados em *threshold*, ou seja, quando nenhum evento de exceção é sensoriado, nenhum dado é transmitido pelo nó. Por exemplo, este é o caso do monitoramento de presença para um único ator em uma área (um animal monitorado dentro de uma grande jaula). Neste caso, todos os outros nós que não estão produzindo dados são reencaminhadores (roteadores).

Para observar o comportamento do consumo energético em um curto período de simulação, a geração de dados do nó transmissor foi deixada com uma vazão relativamente alta, não necessariamente o padrão para este tipo de aplicação (um pacote de 76 *KB* a cada 0,4 *s*). Pacotes de dado começam a ser enviados a cada 28 *s* de simulação, quando as fases iniciais (Reconhecimento e Descoberta Inicial de Rotas) estão completas. Utilizou-se 200 *s* como tempo total de simulação. Com a vazão escolhida, este tempo é suficiente para observar os eventos requeridos (morte do primeiro nó e da rede). Os parâmetros para cada algoritmo (BiO4SeL, ARAMA and AODV) foram escolhidos para haver comportamentos comparáveis entre eles.

4.1. Experimento 1: Conectividade dos Cenários

Neste experimento, foi validada a efetividade dos algoritmos na construção de redes plenamente conectas, sem conhecimento prévio de vizinhos, destinos ou posicionamento geográfico. Os resultados para estes experimentos são mostrados na Tabela 3, contendo a porcentagem destes cenários gerados para cada quantidade de nós testada.

Tabela 3. Percentual de Cenários com Rede Completamente Conexa (Experimento 1).

| Algoritmo | Quantidade de Nós no Cenário | | | | | |
|-----------|------------------------------|------|------|------|-----|-----|
| | 10 | 20 | 40 | 80 | 160 | 320 |
| BiO4SeL | 100% | 100% | 100% | 100% | 93% | 97% |
| AODV | 100% | 99% | 100% | 98% | 98% | 95% |
| ARAMA | 97% | 70% | 8% | 1% | 0% | 0% |

Os resultados mostram que o BiO4SeL fica mais eficiente em construir tabelas de roteamento com o aumento do cenário. A chave para isto é a combinação de dois detalhes do protocolo. O primeiro é a descoberta de rotas ser inicializada pela EB ao invés de pelos nós, deixando-o escalável. O outro é o mecanismo de desencorajamento de formigas de retorno implementado, evitando ciclos e fazendo as formigas moverem-se para frente.

Um fato interessante a se notar é o ARAMA não ser efetivo em sua descoberta de rotas com grandes cenários como o é nos pequenos. De fato, os resultados mostrados no artigo que o descreve se baseiam em pequenos cenários, os melhor adaptados para ele.

4.2. Experimento 2: Tempo de Vida da Rede de Sensores

Estes experimentos validam o tempo de vida da rede de sensores, ou seja, a quantidade de tempo desde a disposição da rede até o momento da morte do primeiro nó, o término de sua energia. Com isto, há a possibilidade de criar-se uma falha de cobertura na rede. A Figura 1 mostra a o tempo de vida da rede para cada tamanho de cenário (10, 20, 40, 80, 160 e 320 nós).

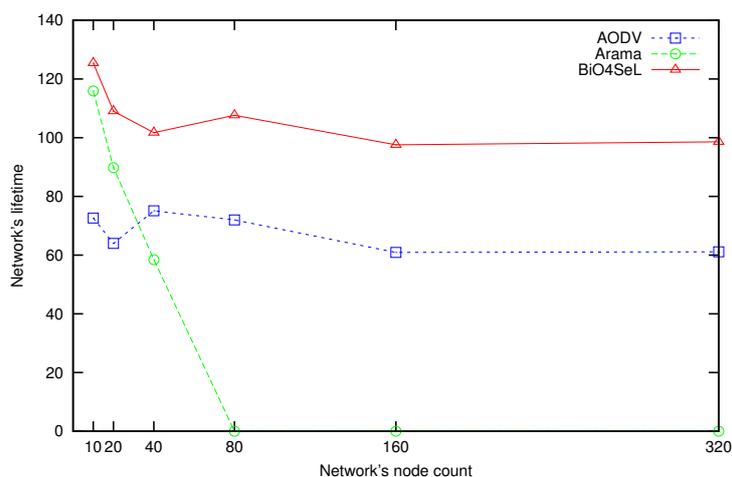


Figura 1. Tempo de Vida da Rede de Sensores (Experimento 2).

Como o ARAMA não conseguiu construir cenários conexos a partir de 80 nós, considera-se seu tempo de vida como zero nestes casos. A Figura 1 mostra como o BiO4SeL alcança melhores valores de tempo de vida em todos os cenários, mesmo quando o ARAMA é eficiente em construir seus cenários. Além disso, o comportamento relativamente linear do BiO4SeL e AODV sugerem sua escalabilidade no crescimento dos cenários.

4.3. Experimento 3: Potência de Bateria dos Nós no Momento do Tempo de Vida

Este experimento tem por objetivo validar como a potência da bateria é economizada na rede. Isto é possível quando a rede reduz a quantidade de dados produzida e encaminhada. Considerando os dados de aplicação constantes para cada algoritmo testado, a economia energética é alcançada quando a quantidade de mensagens de sinalização é reduzida. A Figura 2 mostra o nível médio de potência da bateria disponível nos nós da rede no momento de falha do primeiro nó. Quatro cenários são mostrados: 40 nós (Figura 2(a)), 80 nós (Figura 2(b)), 160 nós (Figure 2(c)) e 320 nós (Figure 2(d)).

A Figura 2(a) mostra como o BiO4SeL mantém a média energética mais alta se comparado aos outros protocolos, para 40 nós. No cenário de 160 nós, ARAMA mantém a mais alta. Isto acontece por sua bateria não ser consumida depois de certo tempo, dado ele não encontrar rotas na rede. Assim, somente consome a bateria próximo a fonte, até a morte dos nós adjacentes, depois disso não conseguindo mais enviar dados. O BiO4SeL reduz a necessidade de sinalização enviando formigas por carona nos pacotes de dados e otimiza as mensagens. Assim, reduz a quantidade destes pacotes trafegados na rede, mantendo uma boa taxa de entrega, como comprovado pelo próximo experimento.

4.4. Experimento 4: Taxa de Entrega dos Pacotes

Este experimento mostra quão efetiva foi a entrega dos pacotes nos algoritmos. A Tabela 4 mostra a porcentagem média de pacotes enviados por um nó e recebidos pela Estação Base, para cada tamanho de cenário.

Como pode ser visto na Tabela 4, BiO4SeL mantém a melhor taxa de entrega de pacotes, não importa qual o tamanho do cenário testado. Como o ARAMA não alcança conectividade em cenários com 80 nós, ele tem taxa de entrega nula.

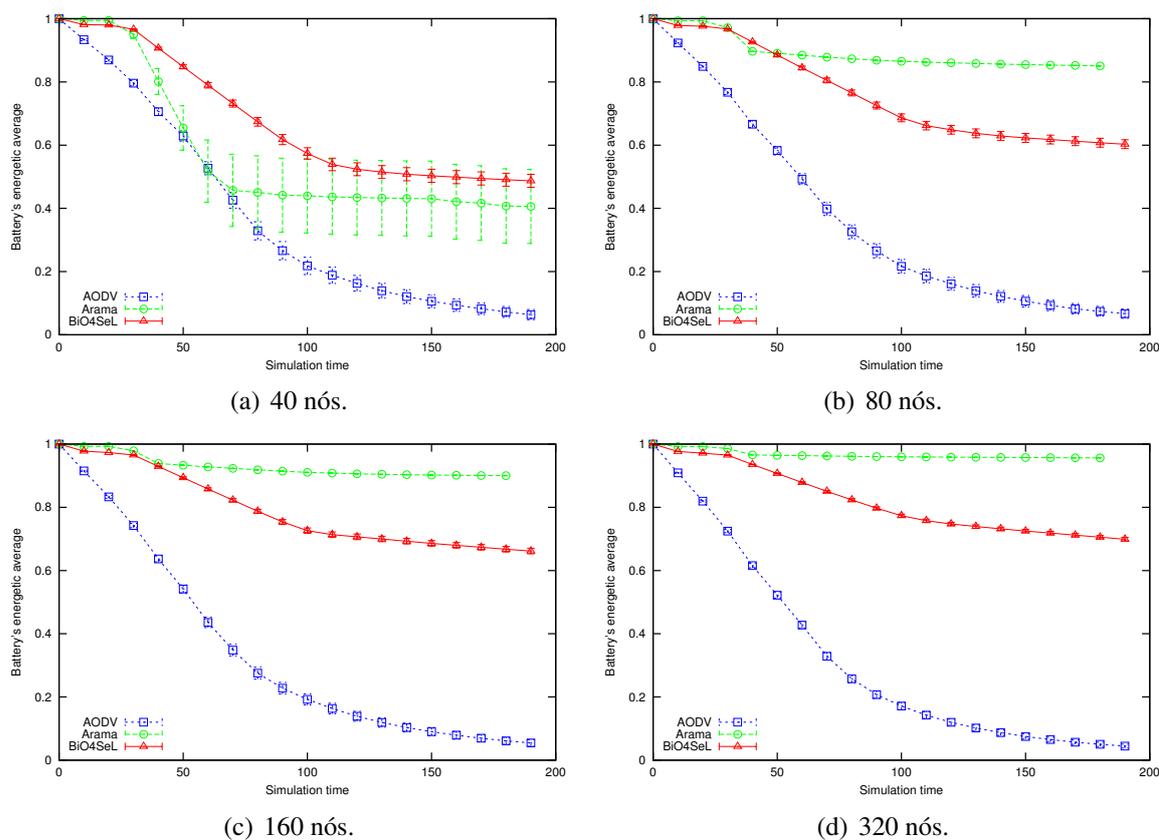


Figura 2. Média de Potência de Bateria dos Nós no Momento do Tempo de Vida (Experimento 3).

Tabela 4. Taxa de Entrega dos Pacotes (Experimento 4).

| Algoritmo | Quantidade de Nós no Cenário | | | | | |
|-----------|------------------------------|-----|-----|-----|-----|-----|
| | 10 | 20 | 40 | 80 | 160 | 320 |
| BiO4SeL | 98% | 96% | 95% | 93% | 87% | 75% |
| AODV | 79% | 58% | 57% | 52% | 44% | 39% |
| ARAMA | 79% | 77% | 92% | 0 | 0% | 0% |

4.5. Experimento 5: Distribuição de Carga de Roteamento

Este experimento mostra os efeitos do algoritmo BiO4SeL na distribuição da responsabilidade de encaminhamento de pacotes entre os nós da rede com relação aos protocolos AODV e Arama. Como o BiO4SeL desencoraja a passagem de pacotes através de nós com baixos níveis de bateria, através de seu esquema de aceleração da evaporação de feromônios, a consequência esperada é que nós com mais níveis de bateria encaminhem mais pacotes, enquanto os pacotes com pouca bateria seriam poupados dessa tarefa. Após um certo tempo, é esperado que os nós apresentem uma distribuição de níveis de bateria mais uniformes.

Para verificar esse experimento, foi calculado o coeficiente de variação dos níveis de bateria entre todos os nós, nos diferentes cenários. O resultado é apresentado na Figura 3.

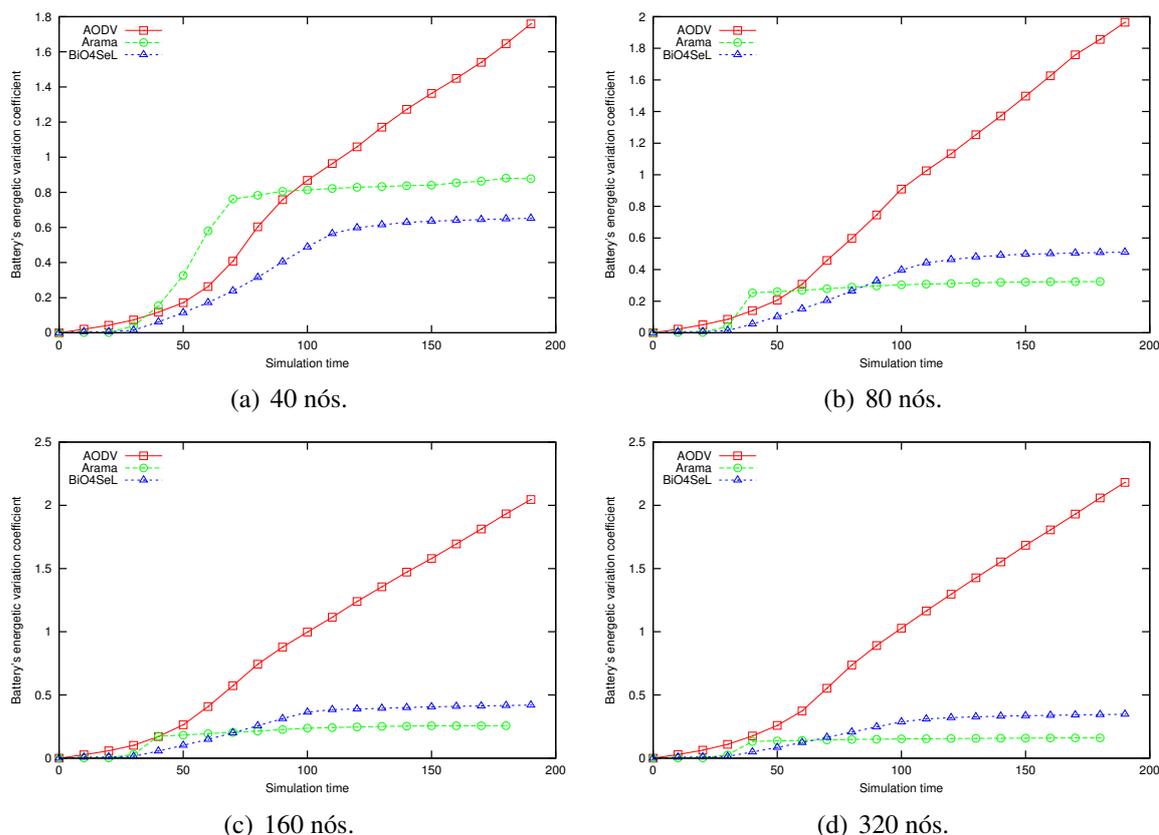


Figura 3. Coeficiente de Variação dos Níveis de Bateria dos Nós da Rede (Experimento 5).

Os resultados mostraram que o BiO4SeL efetivamente reduziu o coeficiente de variação no cenário com 40 nós (Figura 3(a)). Nos demais cenários (Figuras 3(b)–3(d)), o BiO4SeL só não obteve menor coeficiente de variação do que o Arama. No entanto, a razão pela qual o Arama tem, nestes cenários, baixo coeficiente de variação é o fato deste protocolo não ter conseguido configurar rotas. Desta forma, a maioria dos nós, sem ter demanda de encaminhamento de pacotes, apresentam um comportamento de gasto de bateria uniforme, apenas através do envio periódico de *hellos*. Apenas os nós próximos ao sorvedouro e os nós vizinhos àquele que produz dados apresentam gasto energético diferenciado. No entanto, tanto as formigas que tentam estabelecer rotas próximo ao sorvedouro quanto os pacotes de dados produzidos que não podem ser encaminhados ao sorvedouro por falta de rotas acabam por se extinguir sem percorrer outros nós, por colisão, por exemplo.

5. Conclusões e Trabalhos Futuros

Este artigo apresenta o projeto, implementação e experimentação de um novo protocolo de roteamento biologicamente inspirado, chamado BiO4SeL, cujo objetivo principal é maximizar o tempo de vida de uma rede de sensores sem fio. O protocolo inspira-se na inteligência de enxame encontrada em colônias de formigas com o objetivo de agregar autonomia, distribuição e eficiência ao algoritmo.

O algoritmo foi modelado utilizando o simulador de redes NS-2. Os resultados

obtidos a partir da comparação do BiOSeL com as abordagens de roteamento *Ad-hoc* AODV e ARAMA mostraram que o BiO4SeL foi eficiente para atingir seus objetivos. O algoritmo mostrou melhor fator de escala com relação ao aumento de tamanho dos cenários (em área geográfica e quantidade de nós) do que os outros dois. O tempo de vida da rede de sensores foi efetivamente aumentado em todos os cenários comparados, ao mesmo tempo em que manteve uma melhor taxa de entrega de pacotes e melhor conectividade. Os experimentos mostraram também que o BiO4SeL conseguiu melhorar o consumo de energia da rede como um todo através da redução da sinalização e do emprego de *piggybacking* sempre que possível.

O BiO4SeL apresentou seus melhores resultados em cenários onde apenas alguns nós produzem dados, o que pode ser considerado seu cenário ideal de aplicação. Outros testes devem ser realizados com o intuito de avaliar como o algoritmo se comporta quando uma quantidade maior de nós produzem dados ao mesmo tempo.

Além disso, apesar do algoritmo ter sido projetado para suportar mobilidade de nós e oferecer suporte a mais de um sorvedouro, novos experimentos precisam ser realizados com o objetivo de verificar se o bom desempenho é mantido.

Referências

- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38:393–422.
- Al-Karaki, J. N. and Al-Mashaqbeh, G. A. (2007). Energy-centric routing in wireless sensor networks. *Microprocess. Microsyst.*, 31(4):252–262.
- Caro, G. D. and Dorigo, M. (1997). AntNet: a mobile agents approach to adaptive routing. Technical Report IRIDIA/97-12, Université Libre de Bruxelles, Belgium.
- Dorigo, M., Maniezzo, V., and Coloni, A. (1991). The ant system: An autocatalytic optimizing process. Technical Report 91-016, Politecnico di Milano, Italy. (Revised version Sub' IEEE Transactions on Systems, Man, and Cybernetics).
- Hussein, O., Saadawi, T., and Lee, M. J. (2005). Probability routing algorithm for mobile ad hoc networks' resources management. *IEEE Journal on Selected Areas in Communications*, 23(12):2248–2259.
- IEEE (2003). *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. IEEE 802.15.4.
- Iqbal, M., Gondal, I., and Dooley, L. (2006). Online load balancing for energy-aware anycast routing. In *APCC '06: Asia-Pacific Conference on Communications*, pages 771–775, Washington, DC, USA. IEEE Computer Society.
- NS-2 Webpage (2010). <http://isi.edu/nsnam/ns/>. Last access on December, 06, 2010.
- Perkins, C., Belding-Royer, E., and Das, S. (2003). Ad hoc on-demand distance vector (AODV) routing. IETF RFC 3561.
- Shuang, B., Li, Y., Li, Z., and Chen, J. (2007). An ant-based on-demand energy routing protocol for ad hoc wireless networks. In *WiCom '07: Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1516–1519, Washington, DC, USA. IEEE Computer Society.