

Protocolo Adaptativo Híbrido para Disseminação de Dados em Redes de Sensores sem Fio Auto-Organizáveis*

Carlos Maurício S. Figueiredo^{1,2},
Eduardo Freire Nakamura^{1,2}, Antônio Alfredo F. Loureiro¹

¹ Departamento de Ciência da Computação – DCC
Universidade Federal de Minas Gerais – UFMG
Belo Horizonte, MG.

{mauricio, nakamura, loureiro}@dcc.ufmg.br

²Fundação Centro de Análise, Pesquisa e Inovação Tecnológica – FUCAPI
Manaus, AM.

{mauricio.figueiredo, eduardo.nakamura}@fucapi.br

Abstract. *Data dissemination (routing) is a basic function in the wireless sensor networks. Dissemination algorithms for those networks depend on the features of the applications and, consequently, there is no self-contained algorithm appropriate for every case. However, there are scenarios where the behavior of the network may vary a lot, such as an event-driven application, favoring different dissemination algorithms at different instants. Thus, this work proposes a new hybrid and adaptive algorithm for data dissemination, called Multi, that adapts its behavior autonomously in response to the variation of the network conditions. The initial proposal of Multi is based on two algorithms for data dissemination that are also presented and evaluated: SID (Source-Initiated Dissemination), a reactive algorithm where dissemination is started by the source nodes, and EF-Tree (Earliest-First Tree), an algorithm that builds and maintains a tree, in a proactive fashion, to disseminate data towards the sink.*

Resumo. *Disseminação de dados (roteamento) é uma funcionalidade básica de Redes de Sensores Sem Fio. Algoritmos de disseminação a serem utilizados nessas redes dependem das características de suas aplicações e, portanto, não há algoritmo adequado a todos os casos. Contudo, existem cenários em que o comportamento da rede pode variar muito, como em uma aplicação orientada a eventos, favorecendo algoritmos diferentes em instantes diferentes. Dessa forma, este trabalho apresenta a proposta de um algoritmo adaptativo híbrido de disseminação de dados, chamado Multi, que adapta seu funcionamento de forma autônoma conforme variações da rede. A proposta inicial do Multi baseia-se em dois algoritmos de disseminação que também serão apresentados e avaliados: SID (Source-Initiated Dissemination), um algoritmo reativo onde o processo de disseminação é iniciado a partir da origem dos dados, e EF-Tree (Earliest-First Tree), um algoritmo que constrói e mantém pró-ativamente uma árvore para a disseminação de dados de toda a rede.*

*O presente trabalho foi realizado com apoio do CNPq, uma entidade do Governo Brasileiro voltada ao desenvolvimento científico e tecnológico. Processo 55 2111/02-3.

1. Introdução

Em uma rede de sensores sem fio (RSSF) [Loureiro et al., 2003, Pottie and Kaiser, 2000, Akyildiz et al., 2002], um grande número de nós sensores é distribuído em uma área de interesse formando uma rede *ad hoc* cujo objetivo é o monitoramento desta área. Redes de sensores sem fio têm despertado o interesse da comunidade científica devido a sua aplicabilidade que abrange diversas áreas, tais como militar, ambiental, médica e industrial.

Redes de sensores diferem de redes tradicionais em vários aspectos. Algumas dessas particularidades são: Essas redes são formadas por um grande número de nós sensores; sensores possuem fortes limitações de energia, capacidade de processamento e memória; e muitas aplicações de RSSFs demandam características de auto-organização, ou seja, capacidade de se ajustar autonomamente às possíveis mudanças estruturais devido a intervenções externas, tais como mudanças topológicas (causadas por falhas, mobilidade ou inclusão de nós), reação a um evento de sensoriamento ou a uma solicitação feita por uma entidade externa (usuário ou sistema fixo).

O objetivo de uma RSSF é coletar dados de uma região sensoriada e permitir a extração destes por uma entidade externa através de um observador ou gateway (referenciado por nó sorvedouro ou sink). Assim, uma função fundamental dessas redes é a de disseminação dos dados coletados até um sink e, dependendo da aplicação, essa disseminação pode ocorrer de formas distintas, conforme os exemplos citados a seguir [Ruiz, 2002, Tilak et al., 2002]: Contínua, como em uma aplicação de monitoramento ambiental onde pode-se querer saber a variação das condições climáticas durante um período de tempo determinado; Dirigida a eventos, onde uma aplicação de RSSFs para detecção de intrusão pode notificar um observador central caso haja uma invasão da região monitorada; Iniciada pelo observador, onde uma aplicação industrial pode coletar e enviar dados de produção para o observador somente quando este os solicitar explicitamente; ou Híbrida, aplicando mais de uma das formas anteriores, como em uma aplicação aplicação ambiental que pode permitir a coleta de dados somente sob o interesse de um observador, mas em uma situação crítica ou de emergência, pode permitir o envio de dados que notifiquem um evento.

Muitos algoritmos propostos tratam da disseminação (roteamento) dos dados coletados pelos nós sensores a um observador (sink) [Intanagonwiwat et al., 2000, Heinzelman et al., 2000, Ganesan et al., 2001, Kulik et al., 2002], contudo, aplicações e cenários diferentes exigem algoritmos com características diferentes. Isso será mostrado a partir da apresentação e análise de dois algoritmos: o SID (Source-Initiated Dissemination), algoritmo reativo para cenários orientados a eventos, e o EF-Tree (Earlier-First Tree), algoritmo pró-ativo para cenários de comunicação intensa.

Desta forma, diante de uma aplicação ou cenário específico, projetistas de RSSFs podem definir antecipadamente que algoritmo de disseminação é mais adequado para ela. Porém, existem situações em que variações nesse cenário podem ser grandes e/ou imprevisíveis. Por exemplo, um cenário baseado em eventos pode ter intervalos de tempo longos com baixa ou nenhuma incidência de eventos, mas em determinado instante ocorre uma avalanche de eventos, provocando alto tráfego de dados. Nesses casos, podem existir algoritmos mais adequados para determinados instantes da rede, como o SID e o EF-Tree, podendo ser inviável, ou até mesmo impossível, uma entidade externa agir dinamicamente

nessa rede modificando seu comportamento.

O ideal é que a rede tenha capacidade de se auto-organizar perante essas variações, ajustando seu comportamento de forma a manter todas as suas funções (sensoriamento, comunicação, colaboração etc.), economizando recursos (energia, largura de banda etc.), durante todo o seu tempo de existência, sem a necessidade de intervenção externa. O objetivo desse trabalho, então, será apresentar e avaliar um algoritmo híbrido adaptável que ajusta seu comportamento em um cenário com variações, como o exemplificado anteriormente. O resultado é o Multi, que emprega as características do SID e do EF-Tree, alternando seu funcionamento de forma autônoma, representando uma nova abordagem na construção de algoritmos de disseminação de dados para RSSFs.

O restante deste trabalho está organizado conforme descrito a seguir. As Seções 2 e 3, descrevem os algoritmos de disseminação de dados EF-Tree e SID, respectivamente. A Seção 4 apresenta a construção do Multi, algoritmo adaptável híbrido construído a partir dos outros dois. A Seção 5 avalia os algoritmos através de simulações em diferentes cenários. A Seção 6 discute comparativamente o trabalho realizado com outros encontrados na literatura. Por fim, na Seção 7, são colocadas as considerações finais, discussão de trabalhos em andamento e futuros.

2. EF-Tree (Earlier-First Tree)

Em RSSFs, o objetivo principal é coletar dados do ambiente sensoriado extraíndo-os da rede para um sistema computacional mais robusto, de forma a permitir que estes sejam armazenados, analisados e disponibilizados para os seus usuários. Geralmente, em uma RSSF, existe um elemento especial, chamado sorvedouro ou sink, responsável por fazer a ligação entre a rede e o ambiente infra-estruturado. Dessa forma, os dados sensorizados devem ser disseminados em direção ao sink, geralmente em um esquema multi-hop.

Uma estrutura de disseminação simples e eficiente consiste da construção de uma árvore, avaliada em trabalhos anteriores [Sohrabi et al., 2000, Nakamura et al., 2004]. Essa estrutura é criada e mantida pró-ativamente a partir do sink e conecta todos os nós alcançáveis da rede. A implementação aqui apresentada, chamada EF-Tree (Earlier-First Tree), possui a particularidade de reconstrução periódica, para a acomodação de possíveis mudanças topológicas, e seu funcionamento é descrito a seguir:

- O sink, para permitir que os nós da rede disseminem seus dados até ele, deve iniciar a criação de uma árvore. Para isso uma mensagem de controle é enviada em broadcast para toda a rede.
- Cada nó que receber a mensagem de controle de construção de árvore armazena a identificação do seu originador como pai e a reencaminha em broadcast. Dessa forma, cada nó da rede receberá a mensagem de controle enviada por todos os seus vizinhos, porém, só a primeira recebida determinará o seu pai e as demais serão descartadas, por isso o nome Earlier-First Tree (mais cedo primeiro). Outras formas de se determinar o ancestral de um nó poderiam ser empregadas, tais como escolher o nó que levará ao caminho com mais energia ou maior taxa de agregação de dados.
- Quando um nó possuir um dado a ser disseminado, ele o transmitirá para seu pai. Cada nó que receber um dado a ser roteado fará o mesmo, resultando em um esquema multi-hop até chegar à raiz da árvore, ou seja, o sink.

- O processo de construção da árvore é repetido periodicamente para permitir a acomodação de mudanças topológicas possíveis em RSSFs, tais como falhas, migração ou inclusão de novos nós.

3. SID (Source-initiated Dissemination)

O SID apresenta uma proposta de algoritmo reativo de disseminação de dados onde a formação da infra-estrutura de disseminação é iniciada na fonte dos dados quando existir necessidade de envio destes. Essa proposta tende a ser aplicada nos cenários onde a disseminação de informação ocorre orientada a eventos sem a necessidade de interrogação específica para os dados, tendo surgido da observação de que na literatura existem poucos trabalhos com essa característica.

Essa abordagem pode significar uma vantagem em relação a um algoritmo pró-ativo como o EF-Tree, pois não precisa manter continuamente uma infra-estrutura para disseminação de dados em períodos de inatividade da rede, já que em aplicações orientadas a eventos pode-se ter longos períodos de tempo (horas ou dias) sem que dados tenham que ser disseminados.

Nessa solução, assume-se que os nós sensores já são lançados ao ambiente pré-configurados para a detecção dos eventos de interesse, ou ainda, possuem tal definição de fábrica (Por exemplo, um sprinkler já sai de fábrica com a capacidade de detectar incêndios). A rede permanecerá inativa até que um evento seja detectado e o processo de comunicação seja iniciado pela fonte dos dados. A seguir, a descrição do seu funcionamento:

- Quando um evento ocorrer, um nó que o detectar disseminará na rede, em broadcast, os dados coletados identificados com um id. da sua fonte e timestamp de geração (Fig. 1(a)), o que será relevante para o esquema de disseminação.
- Um nó, ao receber um dado gerado por outro nó da rede, armazenará em uma tabela a sua identificação (id. da fonte e timestamp de geração), bem como a identificação do vizinho pelo qual o dado foi recebido. Como o dado é originalmente difundido em broadcast na rede, ele chegará a um nó por todos os seus vizinhos, porém, somente o primeiro que chegar será registrado na tabela e retransmitido em broadcast para a rede, sendo os demais descartados. Da mesma forma que no EF-Tree, outras métricas poderiam ser utilizadas, como disponibilidade de energia no caminho.
- Da mesma forma, os dados passarão a chegar ao sink através de todos os seus vizinhos. O sink então mandará uma mensagem de controle requisitando o envio do dado pelo vizinho pelo qual ele recebeu o dado primeiro, juntamente com a identificação do dado desejado recebido.
- Ao receber essa mensagem de controle, cada nó registrará em sua tabela o identificador do originador da requisição para o respectivo dado recebido, determinando o caminho pelos quais os próximos dados devem seguir. Em seguida, a mensagem de controle será reencaminhada ao vizinho, registrado em sua tabela, pelo qual o dado foi recebido. Esse procedimento se repetirá até a mensagem de controle chegar à fonte do dado (Fig. 1(b)).
- A fonte do dado, por sua vez, ao receber uma mensagem de controle de requisição, atualizará sua tabela e os próximos dados a serem transmitidos serão direcionados para o vizinho originador desta mensagem. Dessa forma, os dados passarão a ser dissemi-

dados pelo caminho inverso percorrido pelas mensagens de requisições do sink, resultando no caminho mais rápido (Fig. 1(c)).

- Para acomodar possíveis alterações topológicas da rede, devido a falhas, mobilidade ou inclusão de nós, as mensagens de controle serão enviadas periodicamente às fontes dos dados enquanto estas estiverem gerando dados. Ao deixar de receber uma mensagem de requisição, devido a uma possível mudança topológica, um nó (fonte ou intermediário) voltará a enviar ou reencaminhar dados em broadcast. Se existir um caminho, portanto, o dado voltará a chegar ao sink que repetirá o processo de requisição descrito anteriormente.
- Quando eventos deixarem de ser registrados, dados deixarão de ser gerados e conseqüentemente, o sink deixará de enviar mensagens de requisição para seus originadores. Sem envio periódico de determinado dado ou mensagem de controle associada, as entradas nas tabelas armazenadas nos nós expirarão e a rede voltará ao estado de inatividade em que estava originalmente.

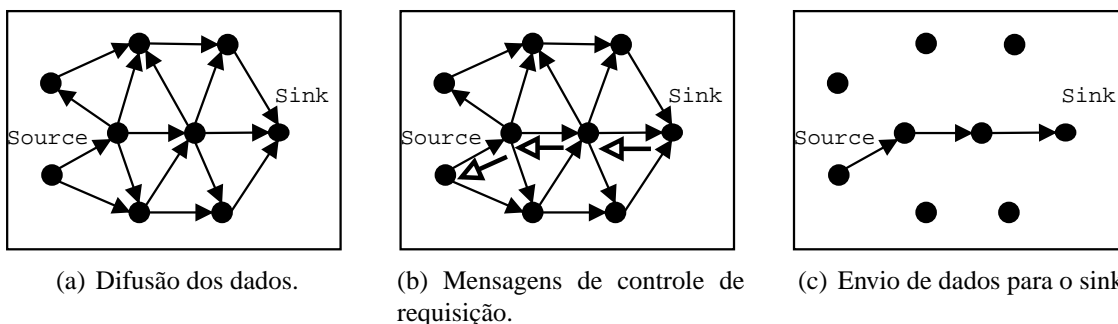


Figura 1: Funcionamento do SID.

A idéia de requisitar dados periodicamente, construindo caminhos entre fonte e sink, é interessante pois espera-se que ao ocorrer um evento este não cesse imediatamente. Essa freqüência de requisição é feita em uma periodicidade maior que a de geração de dados, pois não se espera que RSSFs sejam muito dinâmicas, evitando o overhead de mandar todo dado em broadcast (flooding) ou demandar uma requisição para cada dado.

4. Multi: Protocolo Adaptativo Híbrido para Disseminação de Dados

Pelas características dos algoritmos descritos anteriormente (SID e EF-Tree), podemos observar que eles têm desempenhos diferentes em cenários diferentes. Comparativamente, o SID leva vantagem quando a ocorrência de eventos é imprevisível, ou seja, não se sabe quando há a necessidade de se transmitir dados, pois não necessita manter pró-ativamente uma infra-estrutura para toda a rede como no EF-Tree. Com o SID, essa infra-estrutura é criada e mantida sob demanda, ou seja, somente quando há dados a transmitir.

Quando a necessidade de comunicação é intensa, como em cenários de sensoriamento contínuo ou com uma grande quantidade de fontes de dados, o SID já não é interessante, pois mantém um esquema de disseminação para cada fonte de dados, que é iniciado através de um broadcast, o que causa um consumo grande dos recursos da rede

(energia e largura de banda). Esse cenário passa a ser mais apropriado para o uso do EF-Tree.

Conforme previamente discutido neste artigo, em um cenário baseado em eventos, pode-se ter uma variabilidade grande na ocorrência desses eventos. Por exemplo, a rede pode permanecer dias com pouca ou nenhuma atividade, favorecendo um algoritmo como o SID, mas em determinado momento, começando a surgir uma quantidade elevada de eventos, pode valer a pena a construção e manutenção de uma infra-estrutura como o EF-Tree. A escolha prévia de um dos algoritmos pelo projetista com certeza não trará o melhor desempenho durante todo o tempo de vida da rede, e sua atuação externa de forma dinâmica pode ser inviável ou impossível. Dessa forma, propomos a construção de um protocolo adaptativo híbrido, o Multi, que incorpora características de outros algoritmos e adapta seu funcionamento de forma autônoma de acordo com as condições correntes da rede.

Em sua implementação, o Multi incorpora as características do SID e do EF-Tree. Como o esquema de requisição de dados do SID e da construção da árvore pelo EF-Tree são iniciados pelo sink, o Multi se beneficiará disso para fazer o controle adaptativo do seu funcionamento. Nessa instância, esse controle será feito baseado na quantidade de eventos sendo detectados na rede, e conseqüente tráfego resultante no sink. O funcionamento do Multi é descrito a seguir:

- Inicialmente, a rede funciona exatamente como o SID. Ao detectar um evento, um nó envia em broadcast seus dados para a rede. O sink, ao receber um dado, emite uma mensagem de requisição para a fonte desses definindo o caminho pelo qual os dados passarão a ser entregues.
- O sink contabiliza a quantidade de fontes enviando dados por um intervalo de tempo pré-definido. Em implementação preliminar, definimos esse intervalo de tempo como sendo igual ao período de requisição de dados pelo sink.
- Ao detectar a elevação da quantidade de fontes de dados acima de um limiar fixo e pré-definido, o sink passa a enviar mensagens periódicas de construção de árvore, conforme descrito no EF-Tree. Isso é feito com base na observação de que, a partir de um determinado número de fontes, pode-se esperar uma tendência no crescimento do número destes, e é menos custoso construir e manter uma infra-estrutura de roteamento para toda a rede do que fazê-lo para cada nova fonte individualmente, como será mostrado nas simulações da seção 5.
- Ao receber uma mensagem de construção de árvore, um nó mantém informação de seu pai e uma validade com base no timestamp da mensagem. A partir desse instante, todos os dados gerados ou roteados por este nó passarão a ser enviados pelo caminho estabelecido pela árvore. Isso será feito enquanto a árvore estiver sendo reconstruída, ou seja, a validade da informação do pai do nó é igual ao período de reconstrução da árvore.
- Quando a incidência de eventos voltar a ser reduzida (abaixo do limiar estabelecido), o sink interromperá as reconstruções da árvore e voltará a requisitar dados das fontes individualmente, voltando a funcionar exatamente como o SID. Com isso, os nós terão as validades de seus pais expiradas e voltarão a transmitir seus dados conforme o estabelecido pelas mensagens de requisição.

Estabelecer um limiar de tráfego para determinar quando mudar a característica

do algoritmo depende de vários fatores, entre eles estão o tamanho da rede, a quantidade de fontes de dados (tráfego) e a duração do tráfego, pois esses fatores impactam no custo de execução de cada algoritmo. Assim, espera-se que esse limiar deva ser estabelecido dinamicamente a partir de modelos de ocorrência de eventos nas redes e com base em dados de histórico armazenados no sink. Porém, para fins de simplificação e com o objetivo de avaliar preliminarmente essa abordagem, será definido um valor estático com base nos resultados de simulação do SID e do EF-Tree.

5. Simulação e Avaliação

Nesta seção, os algoritmos abordados neste artigo serão avaliados através de simulações. Inicialmente, serão descritos alguns cenários particulares e será mostrado o desempenho do SID e do EF-Tree comparativamente a outros referenciados na literatura (Flooding e Direct Diffusion). Posteriormente, será mostrada a capacidade adaptativa e a vantagens que podem ser obtidas do Multi em comparação aos algoritmos que o compõe.

Os experimentos foram realizados no NS-2 (Network Simulator 2) [NS-2, 2004]. Os parâmetros de simulação foram ajustados para corresponder a uma rede real baseada no nó sensor Mica2 [Crossbow, 2004] com o protocolo da camada MAC 802.11, conforme Tabela 1. Em todas as simulações, nós consideramos apenas um sink, mensagens de dados de 20 bytes transmitidas a cada 10s e mensagens de controle de 16 bytes transmitidas a cada 100s.

Parâmetro	Valor
Pot. Transmissão	33,3mW
Pot. Recepção	30,0mW
Larg. de Banda	76800 bps
Alcance do Rádio	40m

Tabela 1: Parâmetros de simulação.

5.1. SID × EF-Tree

Para avaliar o funcionamento dos algoritmos SID e EF-Tree foram criados dois cenários de simulação: um baseado em tráfego contínuo e outro baseado na ocorrência de eventos aleatórios. Os aspectos observados nessas simulações foram: Taxa de entrega de dados, para avaliação de perdas de pacotes, e energia total consumida na rede, principal parâmetro avaliado por se tratar de um recurso escasso em RSSFs.

Para se ter base comparativa, foram simulados, também, dois outros algoritmos: o Flooding controlado, onde pacotes duplicados que já foram roteados são descartados, e o Direct Diffusion [Intanagonwiwat et al., 2000], disponibilizado como um módulo do NS-2. Para o segundo, foram configurados os seguintes parâmetros: Período de disseminação de interesses de 100s (igual ao período de construção da árvore no EF-Tree e de requisição de dados do SID), validade de interesse de 200s e opção de reforço negativo habilitada.

A Figura 2 mostra o comportamento dos algoritmos em questão, sob condição de tráfego de dados contínuo, onde dados são gerados no decorrer de toda a simulação. Foi simulada uma rede com 50 nós distribuídos uniformemente em uma área de 100x100m, variando-se a quantidade de fontes (nós gerando dados) de 2 a 50 nós, escolhidos aleatoriamente entre todos os existentes na simulação.

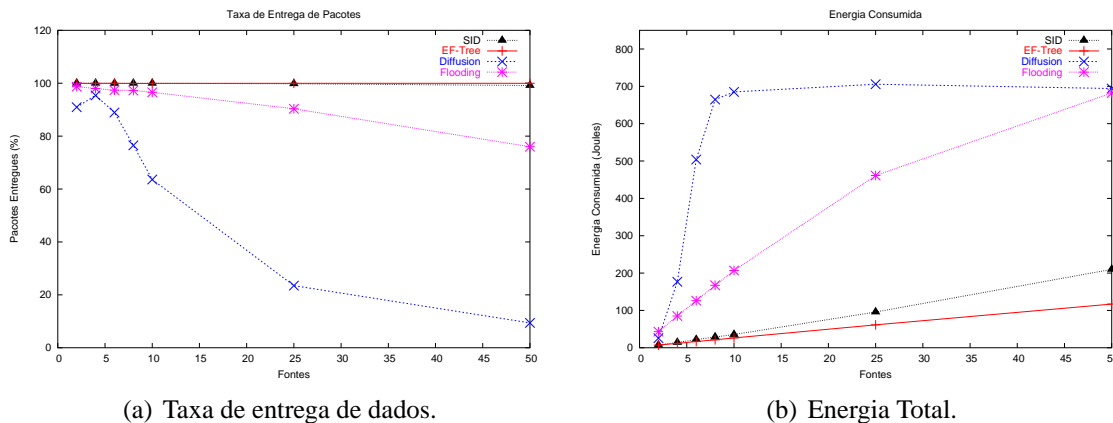


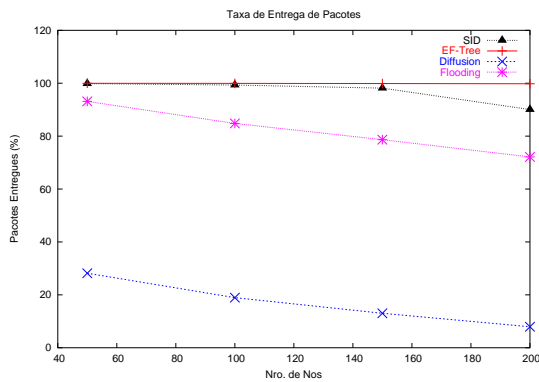
Figura 2: Tráfego de Dados Contínuo.

Como pode-se observar na Figura 2, a difusão de dados em broadcast impacta negativamente na taxa de entrega de pacotes (Fig. 2(a)) em uma rede de largura de banda limitada como a simulada. Esse fato pode ser observado com o Flooding (onde todos os dados são enviados em broadcast), com o decréscimo da taxa de entrega conforme aumento no número de fontes. Com o Direct Diffusion, que faz uso do envio de dados em broadcast para disseminação de interesses e ajuste de gradientes, nota-se um impacto muito maior devido às perdas de mensagens de controle, impedindo o reforço de caminhos e fazendo-o permanecer em um esquema de broadcast. Isso se deve ao fato de a rede não conseguir dar vazão aos dados, enchendo as filas dos nós roteadores, elevando, conseqüentemente, as perdas. O SID também inicia o processo de disseminação de dados com envio de dados em broadcast, mas posteriormente, um caminho é estabelecido para cada fonte de dados, o que evita um inundamento de mensagens mantendo sua taxa de entrega elevada conforme a do EF-Tree.

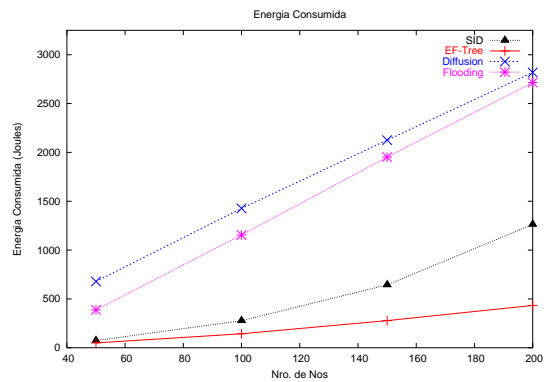
Quanto ao custo de energia, o EF-Tree teve um crescimento menor que os demais algoritmos (Fig. 2(b)), conforme o aumento da quantidade de nós enviando dados, demonstrando sua vantagem em criar pró-ativamente a infra-estrutura de disseminação de dados para toda a rede no cenário simulado, enquanto que o SID teve um crescimento mais acentuado por ter que criar sob-demanda essa infra-estrutura para cada nó individualmente. O Direct Diffusion consome mais energia que o EF-Tree e o SID, superando o Flooding somente quando há duas fontes e funcionando de forma saturada a partir de dez fontes. Esse desempenho é consequência do impacto elevado das perdas de pacotes, conforme descrito anteriormente, tornando-o inviável no cenário simulado (baixa largura de banda e elevado número de fontes).

A avaliação da escalabilidade dos algoritmos em relação ao aumento do tamanho da rede foi baseada no cenário anterior, porém, fixando a quantidade de nós gerando dados (20 nós) e variando o tamanho da rede entre 50 e 200 nós. Os resultados podem ser observados na Figura 3.

O aumento da rede impacta mais nos algoritmos que fazem uso intenso do envio de dados em broadcast, pois causa um maior inundamento de pacotes na rede, conforme descrito anteriormente, elevando as perdas (Fig. 3(a)). Assim, observamos que sob essas



(a) Taxa de entrega de dados.



(b) Energia Total.

Figura 3: Avaliação da Escalabilidade.

condições, o EF-Tree é mais escalável, inclusive mantendo a menor elevação do consumo de energia (Fig. 3(b)). O SID sofreu um impacto ligeiramente maior que o EF-Tree, devido aos broadcasts iniciais, e o Direct Diffusion continuou mostrando-se inviável neste cenário, operando de forma saturada.

Com o objetivo de avaliar a robustez dos algoritmos, foi mantida a quantidade de nós gerando dados (20 nós) e fez-se variar a probabilidade de um nó falhar entre 0 e 50%. Um nó pode falhar em um instante aleatório no tempo de simulação, e se isso ocorrer, ele permanecerá até o final da simulação em estado de total inatividade. Os resultados são apresentados na Figura 4 e mostram que o SID e o EF-Tree mantêm um desempenho superior aos demais com probabilidades menores de falhas, tendo um leve decréscimo e permanecendo com um índice de entrega próximo a 90% conforme aumento no número de falhas. O Flooding tem um comportamento constante, mantendo uma taxa de entrega próxima a 90%, com perdas relativas ao inundamento da rede, e o DirectDiffusion permanece com baixa taxa de entrega por estar operando de forma saturada, conforme observado nas simulações anteriores.

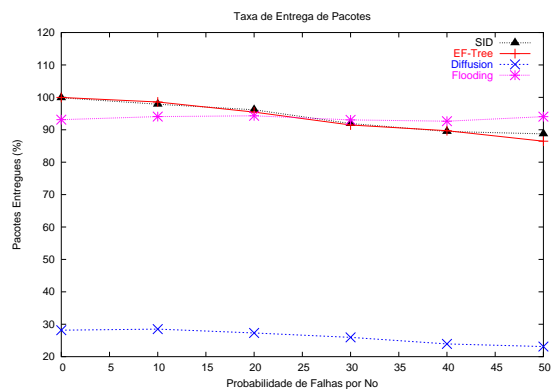


Figura 4: Avaliação da Robustez.

A seguir será mostrada a simulação dos algoritmos em questão em um cenário baseado em eventos, baseado no anterior, onde varia-se a quantidade de nós que detectam a ocorrência de eventos, ou seja, que geram dados. Para representar a ocorrência aleatória de eventos, os nós gerarão dados também aleatoriamente no decorrer do tempo de simulação, com duração variável entre 1 e 50 segundos. Note que dessa forma, diminui-se a probabilidade de geração de dados simultânea entre os nós que detectam eventos.

Nos resultados mostrados na Figura 5, pode-se observar uma vantagem do SID,

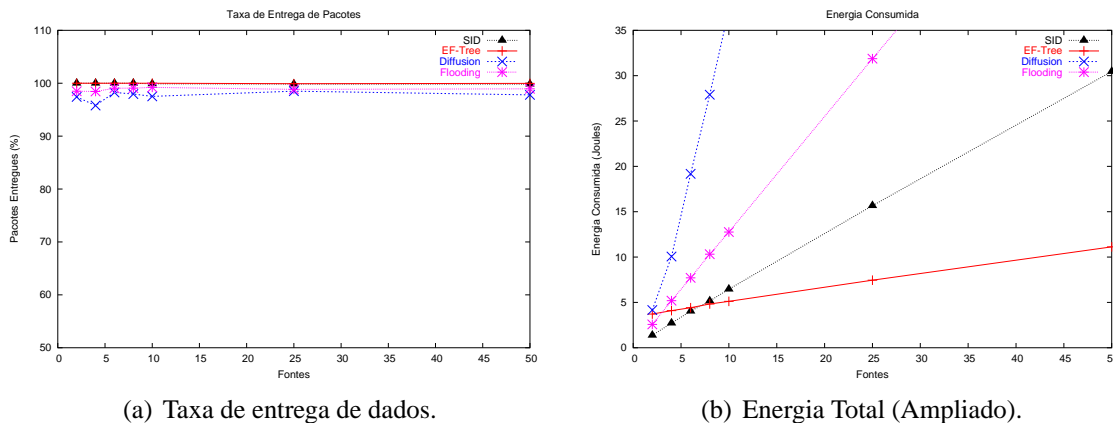


Figura 5: Tráfego Eventual.

em relação ao consumo de energia (Fig. 5(b)), quando a quantidade de nós gerando dados é pequena, pois o algoritmo funciona sob demanda, criando infra-estrutura de disseminação de dados somente quando necessário. Porém, quando a quantidade de tráfego aumenta, o EF-Tree passa a ter vantagem amortizando o custo de manter uma infra-estrutura pela quantidade de nós gerando dados no decorrer de toda a simulação. Nota-se também, que como o tráfego não é tão intenso, todos os algoritmos têm altas taxas de entrega de pacotes (Fig. 5(a)).

5.2. MULTI

Para demonstrar a capacidade adaptativa do Multi, foi criado um cenário onde há uma variação na ocorrência de eventos ao longo do tempo de simulação. Basicamente, foi gerada uma onda de eventos incidindo sobre a rede, o que poderia representar um alvo móvel que entra e sai da área sensoriada pela rede. No instante em que esse alvo começa a entrar na rede, alguns poucos nós detectam a sua presença, quando ele está totalmente inserido na rede, a quantidade de nós detectando-o chega a seu ápice, e quando o alvo começa a deixar a rede, poucos nós o detectam até a rede voltar a sua situação de inatividade. O Multi foi comparado com o SID e com o EF-Tree, com a variação de quantidade máxima de nós que podem detectar o evento em uma rede com 50 nós. O seu limiar de adaptação foi estabelecido empiricamente como sendo 3 (três), observando-se também, das simulações anteriores, que o SID leva vantagem ao EF-Tree para pequenas quantidades de fontes (menor que seis).

Na Figura 6, cada ponto dos gráficos de cada algoritmo representa o consumo de energia acumulado em intervalos de tempo de 10 segundos. Nessa simulação, dados são gerados a partir do instante 500 cessando no instante 900. A Figura 6(a) mostra que quando a quantidade de fontes não atinge o limiar configurado para o Multi, o comportamento do SID e do Multi são idênticos, representando um ganho em relação ao EF-Tree nos intervalos de inatividade da rede (0 a 500s, e 900 a 1400s), já que este último gasta energia nas reconstruções periódicas da árvore e que podem ser observados nos saltos do gráfico a cada 100s. No início do período de atividade, por volta de 500s, o consumo do SID e do Multi tiveram um salto devido ao broadcast inicial dos dados pelos nós fontes. Logo após esses instantes, o crescimento da energia consumida passou a ser semelhante

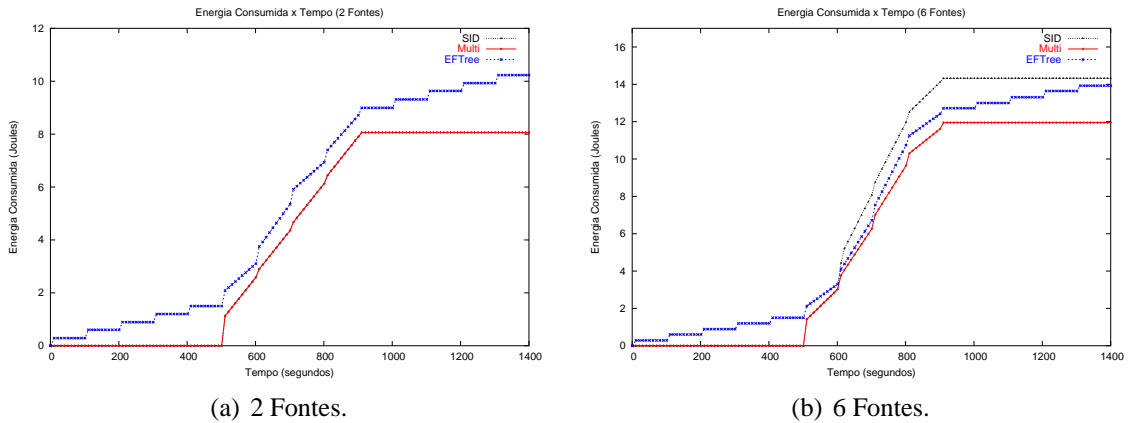


Figura 6: Tráfego com Onda de Eventos.

ao da árvore, pois caminhos entre fontes e sink são definidos através das mensagens de requisição.

Quando a quantidade de nós detectando eventos aumenta, percebemos um custo de energia maior no SID durante o período de tráfego (Fig. 6(b)) em relação ao EF-Tree. Isso é devido à maior quantidade de nós iniciando a disseminação em broadcast (saltos nos instantes 500 e 600s). Neste caso, o Multi adapta seu funcionamento logo após os primeiros nós começarem a gerar dados (entre 500 e 600s), passando a se comportar como o EF-Tree e trazendo o benefício de um consumo menor de energia do que o SID. Quando o tráfego for reduzido novamente (entre 800 e 900s), o Multi adapta-se novamente voltando a funcionar como o SID. Essa adaptação resultou em maior economia de energia em relação ao SID e ao EF-Tree.

Para observar o comportamento do Multi em cenários de simulações exaustivas, repetiu-se os cenários usados para a avaliação do SID e do EF-Tree e comparou-se o consumo de energia dos algoritmos, como pode ser observado na Figura 7.

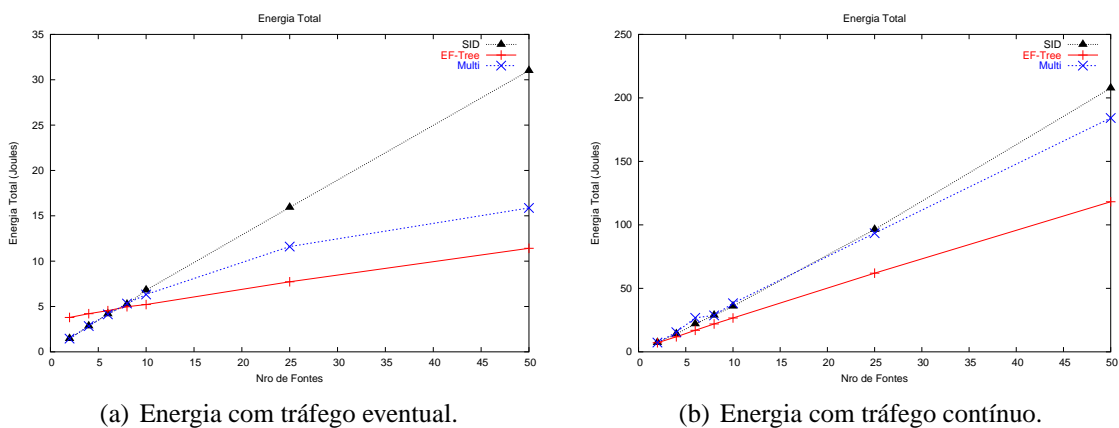


Figura 7: Multi × SID × EF-Tree.

Das simulações realizadas, observa-se a vantagem do Multi em relação ao EF-Tree quando a quantidade de fontes é pequena, pois nesses casos o algoritmo comporta-se

como SID. Quando a quantidade de fontes aumenta, observa-se um comportamento semelhante ao EF-Tree, correspondendo a uma vantagem em relação ao SID (Figura 7(a)). A partir dos pontos onde observa-se o comportamento adaptativo do Multi, os valores de energia observados do Multi não correspondem exatamente ao do EF-Tree devido ao fato dele sempre começar se comportando como SID até atingir o limiar de tráfego definido, representando o overhead do algoritmo para esses casos. A diferença observada nesses instantes depende das condições do cenário simulado (tempo de simulação, taxa de envio de dados e tamanho da rede). Já no cenário de tráfego contínuo (Figura 7(b)), pouca vantagem foi obtida do Multi em relação ao SID, pois todo o tráfego dos fontes foi gerado nos primeiros instantes de simulação, que fez com que o Multi se adaptasse somente quando o maior custo de comunicação já tinha ocorrido. Mesmo assim, nota-se um pequeno ganho do Multi quando a quantidade de fontes é maior que 25, e isso se deve ao fato de que custa menos mandar um broadcast para toda a rede do que mandar mensagens individuais para uma quantidade grande de nós.

Em um cenário orientado a eventos, vale ressaltar que quanto mais esporádicos forem esses eventos, mais vantagens serão obtidas do Multi e do SID em relação ao EF-Tree. Já em um cenário de tráfego intenso constante, talvez seja melhor usar o EF-Tree de forma definitiva.

6. Trabalhos Relacionados

Vários algoritmos de roteamento têm sido propostos para solucionar os problemas de disseminação de dados em RSSFs, como pode-se observar em [Intanagonwiwat et al., 2000, Heinzelman et al., 2000, Ganesan et al., 2001, Kulik et al., 2002]. Porém, nenhum tem a pretensão de ser a solução definitiva de roteamento nessas redes. Eles abordam alguns aspectos ou cenários específicos de aplicação e tentam ser uma boa alternativa para eles. O Direct Diffusion [Intanagonwiwat et al., 2000], é um exemplo clássico de roteamento centrado em dados para RSSFs e bastante referenciado na literatura. Embora esse algoritmo tenha boa capacidade adaptativa em relação a possíveis alterações topológica, ele não se aplica a cenários orientados a eventos, devido à necessidade de disseminação periódica de interesses, que causa um grande consumo dos recursos da rede quando se eleva a quantidade de fontes de dados, conforme mostrado nas simulações feitas. Poucos trabalhos abordam a disseminação de dados em um cenário orientado a eventos, entre eles referenciamos o SPIN [Kulik et al., 2002], que tenta suprir deficiências do Flooding através de negociação local, feita a partir da disseminação da identificação dos dados, para só enviar os dados aos nós que demonstrarem interesse.

Mais recentemente, surgiu um trabalho relacionado a este que trata justamente na necessidade de casar os algoritmos de disseminação de dados com as características das aplicações de RSSFs [Heidemann et al., 2003]. Nesse artigo são propostos dois algoritmos construídos a partir de extensões do Direct Diffusion: O Push Diffusion, para cenários com poucas fontes de dados; e o One-Phase Pull Diffusion, para cenários com elevada quantidade de fontes. Pode-se observar do artigo, que esses algoritmos possuem características muito semelhantes ao SID e ao EF-Tree, inclusive também observa-se a aplicabilidade destes a cenários semelhantes aos aqui apresentados. Porém, em nosso

trabalho, o objetivo principal foi a proposta de uma nova abordagem na construção de algoritmos de disseminação, resultando no Multi. Além disso, foi realizada uma avaliação mais realista dos algoritmos através de simulações comparativas com parâmetros baseados em redes reais (Mica2), enquanto que o artigo em questão se baseou em um cenário ideal contabilizando apenas a quantidade de mensagens trocadas.

Existe na literatura outros trabalhos descrevendo projetos de algoritmos híbridos adaptativos. Um trabalho semelhante ao aqui apresentado descreve o SHARP [Ramasubramanian et al., 2003], protocolo de roteamento para redes *ad hoc* que encontra um ponto de equilíbrio entre protocolos reativos e pró-ativos ajustando o grau com que as informações de roteamento são propagadas na rede. Porém, o trabalho aqui apresentado focaliza as características de disseminação de dados em RSSFs, que pelo nosso conhecimento é um trabalho pioneiro na área.

7. Considerações Finais

Este trabalho mostrou uma nova abordagem de auto-organização de redes de sensores sem fio onde há adaptação autônoma dos seus aspectos estruturais - no caso, no algoritmo de disseminação de dados - para maior eficiência energética, conforme variações do cenário onde a rede se encontra. Isso foi feito através da apresentação e avaliação de dois algoritmos de disseminação de dados, o SID e o EF-Tree, mostrando a variação de comportamento destes em cenários diferentes, e propondo o Multi, que reúne características dos mesmos adaptando seu comportamento autonomamente conforme variação de incidência de eventos na rede.

Verificou-se que o consenso da comunidade científica de que os algoritmos adotados para redes de sensores sejam dependentes da aplicação é verdadeiro. Porém, em um cenário com alta variabilidade, um só algoritmo pode não ter um bom desempenho durante todo o tempo de vida da rede, como foi exemplificado através de uma aplicação orientada a eventos. Como o projetista da rede pode não ter condições de prever ou atuar na rede quando essas mudanças de cenários ocorrerem, mostrou-se vantajosa a adoção de um algoritmo híbrido adaptável como o Multi, proveniente de uma abordagem pioneira em RSSFs, que embora não tenha a pretensão de ser uma solução definitiva para RSSFs, com certeza tem sua aplicabilidade estendida.

Embora a implementação e avaliação inicial do Multi não tenha explorado todo o seu potencial, pode-se perceber que benefícios podem ser obtidos nos casos particulares apresentados no decorrer deste trabalho. Contudo, para torná-lo mais genérico, sua característica adaptativa deve depender de modelos de ocorrência de eventos e predição com base nos dados de histórico. Fazer tal modelagem consiste de nossos trabalhos em andamento, bem como seu aprimoramento e avaliação mais ampla.

Como trabalhos futuros adicionais, além das questões discutidas no decorrer de todo o texto, planeja-se a realização de evoluções do Multi, tais como a introdução de funções de agregação de dados e da adoção de comportamentos diferentes em partições diferentes da rede, conforme a incidência de eventos. Planeja-se, também, realizar a construção de novos algoritmos adaptativos híbridos com a mesma abordagem utilizada, porém, com a utilização ou inclusão de outros trabalhos encontrados na literatura, tais como o Push Diffusion e o One-Phase Pull [Heidemann et al., 2003].

Referências

- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cyirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422.
- Crossbow (2004). Mica2 - wireless measurement system. [Online] Available: http://www.xbow.com/Products/New_product_overview.htm.
- Ganesan, D., Govindan, R., Shenker, S., and Estrin, D. (2001). Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):11–25.
- Heidemann, J., Silva, F., and Estrin, D. (2003). Matching data dissemination algorithms to application requirements. *First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*.
- Heinzelman, W., Chandrakasan, A., and Balakrishnan, H. (2000). Energy-efficient communication protocols for wireless microsensor networks. In *Proceedings of the Hawaiian International Conference on Systems Science (HICSS)*, Maui, Hawaii, USA.
- Intanagonwiwat, C., Govindan, R., and Estrin, D. (2000). Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom'00)*, pages 56–67, Boston, MA, USA. ACM Press.
- Kulik, J., Heinzelman, W., and Balakrishnan, H. (2002). Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless Networks*, 8(2/3):169–185.
- Loureiro, A. A. F., Nogueira, J. M. S., Ruiz, L. B., de Freitas Mini, R. A., Nakamura, E. F., and Figueiredo, C. M. S. (2003). Redes de sensores sem fio. In *21o Simpósio Brasileiro de Redes de Computadores*, pages 179 – 226, Natal, RN, Brazil.
- Nakamura, E. F., Figueiredo, C. M. S., and Loureiro, A. A. F. (2004). Disseminação de dados adaptativa em redes de sensores sem fio auto-organizáveis. In *22o Simpósio Brasileiro de Redes de Computadores*, Gramado, RS, Brazil.
- NS-2 (2004). The network simulator - ns-2. [Online] Available: <http://www.isi.edu/nsnam/ns/>.
- Pottie, G. J. and Kaiser, W. J. (2000). Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58.
- Ramasubramanian, V., Haas, Z., and Sirer, E. (2003). SHARP: A hybrid adaptive routing protocol for mobile ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pages 303–314.
- Ruiz, L. B. (2002). Caracterizacao das rssfs. Technical report, UFMG.
- Sohrabi, K., Gao, J., Ailawadhi, V., and Pottie, G. (2000). Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27.
- Tilak, S., Abu-Ghazaleh, N. B., and Heinzelman, W. (2002). A taxonomy of wireless micro-sensor network models. *ACM Mobile Computing and Communications Review (MC2R)*, 6(2):28–36.