

Roteamento Oportunístico em Redes de Sensores Tolerantes a Atrasos e Desconexões

Felipe Domingos da Cunha¹, Heitor S. Ramos^{1,3}, Raquel A. F. Mini^{2*},
Antonio A. F. Loureiro¹

¹Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Belo Horizonte – MG – 31270-901 – Brasil

²Departamento de Ciência da Computação – Pontifícia Universidade Católica de Minas Gerais
Belo Horizonte – MG – 31980-110 – Brasil

³Instituto de Computação – Universidade Federal de Alagoas
Maceió – AL – 57072-970 – Brasil

{fdcunha, hramos, loureiro}@dcc.ufmg.br, raquelmini@pucminas.br

Abstract. *Several MAC-layer protocols for wireless sensor networks have been proposed to schedule the sensors' radio duty-cycle reducing the energy depletion. Sensors' radio are turned off in idle periods and back to operation only when it is necessary. This approach leads to an intermittent connectivity causing large delays and disconnections and imposes new challenges to the routing service. This work proposes a new routing protocol, namely OR-Tree (Opportunistic Routing Tree), which is provided by the awareness of the topology dynamics as a result of a cross-layer design with the MAC protocol. Simulation results show that OR-Tree significantly reduces the latency and the energy consumption without diminishes the delivery rate.*

Resumo. *Protocolos para a camada MAC em Redes de Sensores Sem Fio (RSSFs) são propostos como intuito de fazer o escalonamento do rádio do nó sensor e, assim, reduzir o consumo de energia. Com isso, os rádios devem ser desligados em momentos de ociosidade e permanecerem ativos apenas quando necessário. Essa abordagem torna a conectividade da rede intermitente, gerando atrasos e desconexões na rede, e cria novos desafios para o roteamento de dados. Nesse trabalho é proposto um novo protocolo para coleta de dados, o OR-Tree (Opportunistic Routing Tree), que se adapta ao dinamismo da conectividade da rede através de uma integração com o protocolo da camada MAC. Resultados de simulação mostraram que é possível reduzir a latência e o consumo de energia sem comprometer a taxa de entrega de dados.*

1. Introdução

As Redes de Sensores Sem Fio (RSSFs) são compostas de dispositivos sensores que atuam em conjunto no sentido de monitorar, instrumentar e, eventualmente controlar aspectos do mundo físico [Akyildiz et al. 2002, Estrin et al. 1999, Intanagonwiwat et al. 2002].

*Parcialmente financiada pelo Projeto Universal do CNPq número 478175/2008-6 referente ao Edital MCT/CNPq 14/2008.

Tais redes apresentam diversas restrições como, por exemplo, a quantidade limitada de energia, baixo poder computacional e baixa taxa de transmissão.

Os nós sensores são dispositivos autônomos que apresentam capacidade de sensoriamento, processamento e comunicação e funcionam alimentados por baterias. A vida útil da bateria varia, principalmente, com a quantidade de ações executadas pelo nó sensor. Dessa maneira, em uma determinada rede, alguns sensores podem deixar de funcionar devido ao descarregamento da bateria enquanto outros sensores ainda estão aptos a funcionar. Alguns sensores também podem apresentar falhas por estarem em ambientes inóspitos devido à ação do próprio ambiente. O ambiente monitorado e o fenômeno observado podem mudar constantemente fazendo com que a rede tenha que se adaptar a essas mudanças. Uma RSSF deve coletar informações de interesse da aplicação e enviá-las para um usuário. Esse envio de dados é realizado pelos nós sensores que reportam suas informações coletadas a um elemento especial da rede chamado *sink*, responsável por fazer a interface entre a RSSF e o usuário através, por exemplo, da Internet.

Nesse tipo de rede, a energia é um recurso crítico e diversas pesquisas têm sido desenvolvidas no sentido de propor novas técnicas para criação de protocolos que visam reduzir o consumo de energia e, conseqüentemente, aumentar o tempo de vida da rede [Anastasi et al. 2009]. Uma maneira de economizar de energia é evitar transmissões desnecessárias. Nesse sentido, alguns algoritmos de roteamento tentam minimizar o *overhead* para construção da estrutura de roteamento e também reduzir a distância das transmissões com a utilização de uma comunicação com vários saltos.

Outra maneira de economizar energia é fracionar o tempo em que o rádio do nó sensor permanece ligado. Assim, o funcionamento do rádio é alternado em períodos de escuta e dormência de modo que apenas durante os períodos de escuta o rádio permanece ligado, transmitindo e recebendo dados. Entretanto, poderão ocorrer quebras de rotas e particionamento da rede devido à intermitência da conectividade de seus nós. Assim, um caminho fim-a-fim entre a origem e o destino pode não existir durante um período ou mesmo nunca venha a existir. Nesse cenário, as RSSFs podem ser consideradas como Redes Tolerantes a Atrasos e Desconexões(DTNs) [Oliveira et al. 2007, Warthman 2003].

Tipicamente, os protocolos de camada MAC (*medium access control*) das RSSFs configuram ciclos de trabalho de modo que os nós sensores fiquem acordados apenas durante uma pequena fração de tempo, fazendo com que a topologia seja dinâmica. Dessa maneira, em cenários realistas, os algoritmos de roteamento que não tomem ciência da variação de topologia tendem a produzir atrasos elevados para a entrega dos dados. Por exemplo, em um algoritmo de roteamento baseado em árvore, um nó sensor poderá atrasar a transmissão dos seus dados, caso o seu ancestral se encontre com o rádio desligado no momento do envio de pacotes. Em cenários nos quais as aplicações apresentem requisitos de QoS mais rígidos, esses algoritmos podem não atender às necessidades da aplicação.

Neste contexto, este trabalho propõe um protocolo de roteamento ciente das variações de topologia para RSSFs orientadas a evento, o OR-Tree (*Opportunistic Routing Tree*). Esse protocolo visa interligar os nós fontes de todos os eventos ao *sink*, utilizando uma estrutura de árvore. Diferentemente da maioria dos protocolos encontrados na literatura, o OR-Tree segue um projeto integrado entre as camadas MAC e de roteamento de modo que o roteamento conheça a agenda de ciclos de trabalho dos seus vizinhos e, assim,

possa se adaptar melhor aos requisitos de QoS das aplicações. Dessa maneira, o OR-Tree se mostra capaz de lidar com os atrasos e desconexões originados pelo escalonamento do rádio. Essa técnica é conhecida por roteamento oportunístico, uma vez que aproveita o momento que o nó sensor tem conectividade com o seu vizinho para transmitir a mensagem e, assim, entregá-la ao *sink*.

Esse trabalho está organizado da seguinte maneira: a seção 2 discute alguns trabalhos relacionados, a seção 3 apresenta o protocolo OR-Tree e seus detalhes de funcionamento, a seção 4 apresenta e discute os resultados da avaliação de desempenho do OR-Tree e, finalmente, a seção 5 apresenta as conclusões e direções futuras deste trabalho.

2. Trabalhos Relacionados

Redes Tolerantes a Atrasos e Desconexões (DTNs) são um tipo de rede com conexão intermitente na qual um caminho fim-a-fim entre os nós que estão comunicando pode nunca existir [e Hadi et al. 2007, Mundur and Seligman 2008, Kurhinen and Janatuinen 2007, Abdulla and Simon 2006]. Numa tentativa de contornar os problemas de atrasos e desconexões, as redes DTN implementam técnicas de comutação de mensagens e de armazenamento persistente. Uma mensagem é enviada na rede pelo modo armazena e encaminha (*store-and-forward*), nó a nó, até alcançar o nó destino. Por fazer uso desta técnica, as redes DTNs são conhecidas por redes do tipo armazena-e-encaminha. Assim não há necessidade alguma do nó destino estar ativo no momento que o nó origem envia a mensagem [Oliveira et al. 2007].

Para utilização da técnica de comutação de mensagens e armazenamento persistente de dados, foi necessário a criação de uma sobrecamada (*overlay*) abaixo da camada de aplicação, denominada camada de agregação ou *Bundle Layer*. Nessa camada, todos os nós pertencentes à rede DTN executam um protocolo para agregação, desde a origem até o destino. Além disso, as redes DTN estabelecem uma nomenclatura diferenciada com a inclusão de diversos conceitos. Um deles é conceito de *região*, que consiste em um agrupamento de nós (fixos ou móveis) com características semelhantes. O nó que é responsável pela conexão entre regiões é denominado de nó *mensageiro*. Esse é um tipo especial de nó, tipicamente móvel, que participa de duas ou mais regiões a fim de proporcionar a conexão e a troca de informações entre as regiões. Outro conceito importante definido pelas redes DTN é o *contato*. Um contato é o momento que dois nós se encontram na rede, uma oportunidade para ocorrer uma transmissão de dados [Jain et al. 2004]. Os contatos podem ser classificados de cinco maneiras:

- *Contato persistente*: são contatos que estão sempre disponíveis. O nó pode enviar a mensagem a qualquer momento que houver necessidade.
- *Contato sob demanda*: esse tipo de contato necessita que em algum momento uma ação seja tomada com o objetivo de instanciar o contato. Depois de estabelecido, se comportará como um contato persistente.
- *Contato previsível*: nesse tipo de contato os nós fazem previsões acerca do horário e da duração do contato conforme históricos armazenados de contatos passados.
- *Contato oportunista*: são contatos que ocorrem ao acaso, sem que haja prévia programação.
- *Contato programado*: tipo de contato no qual o momento e a duração do contato são pré-estabelecidos.

Um exemplo de contato programado pode ser encontrado em RSSFs quando os nós sensores “acordam” e “dormem” em horários preestabelecidos. Esse tipo de contato ocorre quando a camada MAC utiliza dessa programação de contatos para desligar o rádio dos nós sensores, com a finalidade de economizar energia. Diversos protocolos são encontrados na literatura que aplicam essa técnica [Ye et al. 2002, Ye et al. 2004, Demirkol et al. 2006, Polastre et al. 2004]. Entretanto, estudos recentes mostraram que, quanto menor for o período em que o rádio do sensor permanecer ligado, maior será o tempo de vida da rede. Em [Ye et al. 2006], é mostrado ser possível efetuar o monitoramento de uma área com períodos de rádio ligado muito pequenos, próximos de 1% do ciclo de funcionamento do nó sensor.

Um dos principais desafios de pesquisa em redes DTN é o roteamento, pois, nesse tipo de rede, é necessário que os protocolos de roteamento sejam capazes de superar os problemas dos atrasos que são muito longos e as frequentes falhas e desconexões. Como tarefa importante, os protocolos de roteamento precisam determinar rotas na rede sem que exista um caminho possível entre o nó fonte e o nó destino da informação. Na literatura são encontrados alguns trabalhos que tratam o roteamento de dados em redes DTN [Vahdat and Becker 2000, Harras et al. 2005, Juang et al. 2002]. Em [Vahdat and Becker 2000], é proposto um protocolo de roteamento epidêmico. Nesse protocolo, quando uma mensagem chega a um nó intermediário, o nó transmite essa mensagem para todos os seus vizinhos. Esse protocolo trabalha com o princípio de que, quando dois nós se encontram na rede, eles trocam entre si as mensagens que cada um tem e ainda não foram recebidas pelo outro nó. Esse processo é repetido até que a mensagem alcance seu destino. Nesse trabalho, é mostrado que o roteamento epidêmico é capaz de entregar praticamente todas as mensagens transmitidas quando a capacidade do *buffer* é suficientemente grande.

Harras em [Harras et al. 2005] apresentou um mecanismo de controle de inundação em redes DTN como um complemento para a arquitetura DTN. O objeto é reduzir a sobrecarga dos pacotes de controle e garantir a alta taxa de entrega das mensagens. Para isso, é proposto um novo conceito: “disponibilidade” que representa o grau de disposição que cada nó tem em participar do encaminhamento de mensagens na rede. Além disso, ela também controla variáveis que definem o intervalo de tempo entre o envio de pacotes de controle para a descoberta de novos vizinhos. Para definição dessas variáveis, fatores como bateria, tamanho do *buffer* e prioridade das mensagens são considerados. Resultados mostraram que foi possível reduzir o número de pacotes de controle necessários para que as mensagens sejam entregues em seu destino.

Uma proposta de protocolo de roteamento para RSSFs tolerantes a atrasos e desconexões é apresentada em [Juang et al. 2002]. Nesse projeto, colares dotados de sensores são anexados às zebras, com o objetivo de monitorar o estilo de vida desses animais. Como os nós da rede estão em constante movimento, estudos de protocolos de roteamento tolerantes a falhas e desconexões são efetuados. Então, duas abordagens foram escolhidas para esse cenário: um protocolo de roteamento baseado em inundação na rede e outro protocolo baseado em histórico. Na abordagem de inundação, os dados são enviados a todos os vizinhos e, em um determinado momento, são entregues ao nó de monitoramento. Na abordagem baseada em histórico, os nós sensores atribuem uma probabilidade para eles transmitirem a mensagem para o nó *sink*, baseado nos últimos contatos com sucesso

com o *sink*. Quanto maior for essa probabilidade, maior também será a chance do nó encontrar o nó *sink* e realizar a entrega das mensagens. Experimentos feitos comprovaram que a abordagem por histórico consegue uma boa redução do consumo de energia quando comparado à abordagem por inundação.

Diferentemente dos trabalhos descritos acima, que consideram a mobilidade dos nós para gerar altos atrasos na rede e desconexões, neste trabalho os altos atrasos são criados pela utilização de um protocolo MAC que divide o funcionamento do rádio em períodos de dormência e escuta. Além disso, devido às restrições de energia, estratégias de roteamento como a epidêmica não parecem promissoras, pois a inundação da rede é uma tarefa bastante dispendiosa. Assim, o desafio é construir a árvore de coleta de dados de forma a reduzir os atrasos encontrados nesse cenário com as restrições de energia presente nas RSSFs.

Este trabalho se propõe encontrar uma árvore de distribuição que conecte todos os nós que possuem informações para enviar ao *sink*. Esta tarefa é um problema conhecido por ser NP-completo [Krishnamachari et al. 2002], e também é conhecida na literatura por construir a árvore de Steiner na rede.

De maneira mais formal, pode-se enunciar o problema da árvore mínima de Steiner da seguinte maneira: dado um grafo $G(\mathbf{N}, \mathbf{E})$, onde \mathbf{N} é o conjunto de vértices, \mathbf{E} é um conjunto de arestas. Associado às arestas existe um custo, que é definido como uma função não negativa, e, seja $\mathbf{T} \subseteq \mathbf{N}$ um subconjunto de vértices de interesse, a árvore mínima de Steiner é qualquer subgrafo que conecta todos os vértices \mathbf{T} e apresenta custo mínimo. O custo do subgrafo é definido pela soma dos pesos atribuídos às arestas. Uma vez que os pesos são positivos, este subgrafo é uma árvore. Nesse caso, G é o grafo que representa a rede, \mathbf{N} é o conjunto de sensores, \mathbf{E} é o conjunto de interconexões que ligam os nós sensores e \mathbf{T} é o conjunto de nós sensores que deseja transmitir informações para o *sink*.

Diversas heurísticas para o problema da árvore mínima de Steiner foram propostas em diferentes contextos resultando em soluções aproximadas para este problema. Algumas heurísticas como a apresentada nos trabalhos de [Robins and Zelikovsky 2000, Hougardy and Prömel 1999], destacam pelo pequeno fator de aproximação para a solução ótima, porém, em suas versões distribuídas, necessitam uma grande quantidade de troca de mensagens, tornando-as inadequadas para RSSF.

Grande parte das heurísticas propostas, mesmo as distribuídas, é proativa, ou seja, constroem periodicamente uma árvore de difusão. Isto envolve uma periodicidade de troca de informações entre os nós e, conseqüentemente, um consumo adicional de energia quando não há eventos de interesse presente na rede. Quando um nó detecta um evento de interesse, a árvore de difusão já está pronta para que ele envie suas informações ao *sink*. Neste trabalho, o interesse é nos algoritmos reativos, pois esses só constroem suas infraestruturas de roteamento no momento em que um evento é detectado na rede.

No contexto de interesse deste trabalho, ou seja, em roteamento reativo nas RSSFs, as seguintes técnicas se destacam: (i) *Shortest Paths Tree* (SPT) [Krishnamachari et al. 2002], (ii) *Center at Nearest Source* (CNS) [Krishnamachari et al. 2002], (iii) *INFRA* [Nakamura et al. 2009] e (iv) *DAARP* [Villas et al. 2009]. A utilização da heurística *Shortest Paths Tree* é uma das formas mais simples de construir uma solução aproximada

para a árvore de Steiner distribuída. Nessa heurística, cada nó N_i que detectou um evento envia seu pacote contendo as informações coletadas para o *sink*, utilizando o menor caminho entre o nó fonte e o *sink*. Quando o caminho entre o nó fonte N_i e N_j ao *sink* se sobrepõe, o pacote é agregado. Na heurística *Center at Nearest Source*, proposta em [Krishnamachari et al. 2002], cada nó sensor N_i que detecta um evento na rede transmite um pacote contendo as informações sensoreadas para algum nó N_j mais próximo do *sink*. O nó N_j agrega todos os pacotes recebidos e envia apenas um pacote para o *sink*.

O InFRA [Nakamura et al. 2009] consiste em uma abordagem reativa que escolhe dinamicamente o próximo salto de roteamento minimizando o impacto de quedas de conectividades. Nesta técnica, os nós se agrupam ao detectarem os mesmos eventos e um dos nós do agrupamento (*cluster head*) fica responsável por agregar os dados fornecidos pelos outros membros do agrupamento. Após os agrupamentos serem formados, o roteamento ocorre seguindo o princípio de que a melhor rota leva ao menor caminho ao *sink* que agrega o maior número de agrupamentos possível.

O DAARP [Villas et al. 2009] utiliza uma estratégia de agrupamentos semelhante à do InFRA. Porém, constrói a árvore de roteamento entre os *cluster heads* de maneira incremental, ou seja, o *cluster head* responsável pelo agrupamento do primeiro evento envia seus dados por uma rota de caminho mínimo. A cada novo evento, o respectivo *cluster head* se agrega à estrutura de roteamento existente através de um caminho mínimo para um dos nós dessa árvore.

Neste trabalho, é proposto o OR-Tree, um protocolo de roteamento que utiliza informações de agendamento dos ciclos e trabalhos da tabela de vizinha do protocolo da camada MAC para a construção da árvore de roteamento. Esse protocolo será descrito na próxima seção e resultados de simulação demonstram que o OR-Tree reduz consideravelmente a latência sem prejudicar a taxa de entrega dos dados.

3. Abordagem Proposta

Nesta seção será apresentado o protocolo *Opportunistic Routing Tree* (OR-Tree). Esse protocolo trata-se de uma heurística para o problema da árvore de Steiner em RSSF tolerantes a atrasos e desconexões. O OR-Tree é um protocolo desenvolvido para a camada de roteamento e tem como principal tarefa a construção de uma árvore de coleta de dados. O OR-Tree trabalha em cooperação com a camada MAC, na qual é utilizado o protocolo *Sensor-MAC* [Ye et al. 2002]. Na seção 3.1, será descrito o funcionamento do protocolo S-MAC e como o mesmo contribui para a geração de um cenário com atrasos na rede. Em seguida, na seção 3.2, será descrito como o protocolo OR-Tree interage com o S-MAC a fim de criar as rotas para escoamento dos dados em direção ao *sink*.

3.1. Sensor-MAC

O protocolo Sensor-MAC [Ye et al. 2002, Ye et al. 2004] é proposto para camada MAC em RSSFs e apresenta como metas a redução do consumo de energia e a redução de colisões. Para alcançar essas metas, o S-MAC utiliza uma combinação de sincronização de agendas e um esquema de contenção.

Uma das fontes de desperdício de energia é conhecida por *idle listening*, que ocorre quando o nó sensor mantém o seu rádio ligado mesmo quando não há eventos na rede, desperdiçando energia. O S-MAC reduz o *idle listening* fazendo com que o ciclo

de funcionamento do nó sensor seja dividido em períodos de escuta e dormência, conforme ilustrado na figura 1. Durante o período de dormência o nó desliga seu rádio e ajusta um temporizador para acordá-lo mais tarde. Nos períodos de escuta, o nó liga o seu rádio e realiza as devidas transmissões e recepções.

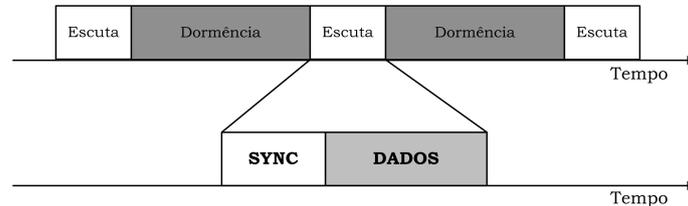


Figura 1. Períodos de escuta e dormência [Ye et al. 2002].

Os períodos de escuta são divididos em duas partes: SYNC e DADOS. Em SYNC, os nós trocam suas agendas de funcionamento com seus vizinhos mais próximos, via difusão. Esse sincronismo de agendas forma na rede agrupamento de nós com a mesma agenda. A comunicação ocorre entre nós de mesmo agrupamento minimizando interferências. Nos períodos de DADOS, os dados são transmitidos. Essa abordagem tem como desvantagem aumentar o atraso na comunicação devido aos períodos de dormência de cada nó.

Esse cenário de constante dormência também acaba prejudicando a conectividade da rede, tornando-a intermitente. Assim, o caminho entre a origem e o destino pode não existir durante um período ou, ainda, pode ser que um caminho entre a origem e o destino nunca chegue a ficar completamente conectado. Nesse contexto, os protocolos MAC que controlam o ciclo de trabalho dos nós geram atrasos e desconexões, típicos das redes DTNs [Oliveira et al. 2007].

3.2. OR-Tree

O OR-Tree é um protocolo de roteamento reativo que trabalha em cooperação com o protocolo da camada MAC. Sua principal característica é o dinamismo na criação de rotas entre o nó fonte e o nó *sink*. Inicialmente o seu funcionamento é determinado pela fase de descoberta de vizinhos do protocolo S-MAC. Nessa fase, todos os nós enviam pacotes de informação via difusão para os seus vizinhos contendo a informação de quando será o seu próximo período de escuta na rede. Assim, com essa informação, o nó conseguirá definir em qual instante cada vizinho estará dormindo e quando estará acordado, determinando o momento do próximo contato com cada vizinho.

Para evitar consumo de energia na construção e atualização da tabela de roteamento, o protocolo S-MAC compartilha a tabela de vizinhos com o protocolo OR-Tree. Desta forma, um projeto integrado de camadas é criado entre as camadas MAC e de rede. Com isso, o protocolo de roteamento saberá o momento exato que ocorrerá contato com cada vizinho. Ao detectar um evento, o nó que estiver na região deverá sensoriar os dados e iniciar o processo de coleta dos dados.

No OR-Tree, a escolha do próximo nó é baseada em dois critérios: (i) escolher dentre seus vizinhos o nó que pertença a um dos caminhos mínimos para o nó *sink* e (ii) qual o contato ocorrerá mais cedo. Como exemplo, na figura 2 o nó *N* possui como vizinhos nos caminhos mínimos para o nó *sink* os nós *N1*, *N2*, *N3* e *N4*. Assim, dentre

os nós escolhidos, o nó fonte escolherá o nó que acordará mais cedo e, por consequência, o nó escolhido será aquele que proporcionará um contato mais rápido com o nó fonte do evento. Ao escolher o nó que proporcionará um contato mais cedo, dentre os contatos programados, obtém-se redução da latência na entrega de dados ao *sink*.

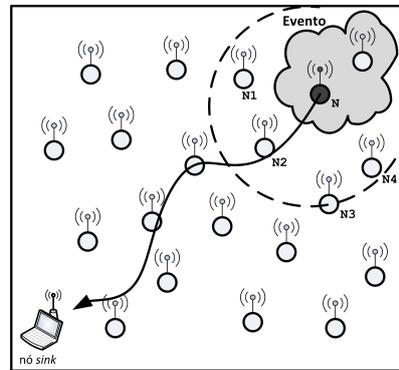


Figura 2. Escolha do próximo salto pelo protocolo OR-Tree.

A figura 3 ilustra o fluxo de funcionamento do OR-Tree. Ao receber um pacote o nó deverá verificar primeiramente se ele é o nó *sink* ou se é um nó intermediário na rota (A). Caso seja o *sink*, o nó apenas armazena as informações recebidas (B). Caso contrário, o nó deverá escolher para qual dos seus vizinhos ele encaminha o pacote. O nó então acessa a tabela de vizinhos (C) para verificar quais vizinhos estão nos caminhos mínimos para o *sink*, formando um conjunto (D). Dentre os vizinhos escolhidos, o nó verifica qual deles proporciona um contato mais cedo, escolhendo-o (E). Finalmente, o pacote é encaminhado ao nó escolhido (F).

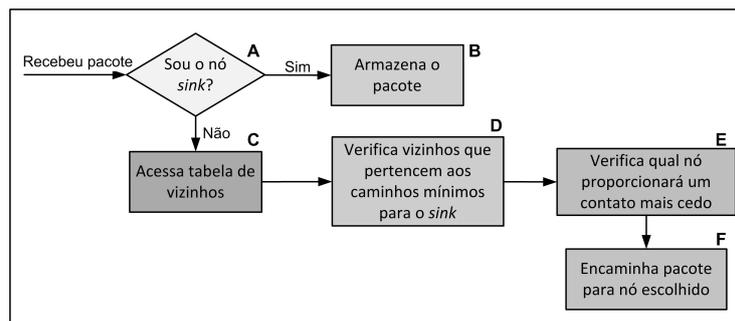


Figura 3. Diagrama de funcionamento do protocolo OR-Tree quando um pacote é recebido.

4. Resultados

Esta seção apresenta os resultados de simulação para o protocolo OR-Tree. A seção 4.1, descreve a configuração padrão do cenário e os parâmetros utilizados nas simulações. Na seção 4.2, são apresentados os protocolos de roteamento que foram avaliados em comparação com o protocolo OR-Tree. Finalmente, a seção 4.3, apresenta os resultados obtidos com as simulações dos protocolos.

4.1. Descrição do Cenário

O cenário para as simulações considera que os sensores são depositados em uma área formando uma topologia plana de $100 \times 100 m^2$ com 200 nós posicionados aleatoriamente, seguindo uma distribuição uniforme, exceto o *sink*, que está posicionado no canto inferior esquerdo e possui posição fixa $(0, 0)$. Não foi considerada nenhuma mobilidade para os nós e todos eles possuem o mesmo *hardware*, caracterizando a rede como homogênea. Cada nó sensor possui uma energia inicial de 25 J e o modelo do consumo de energia utilizado é definido pelo simulador, que decreta o gasto de energia de acordo com o modo de operação que o rádio do nó sensor se encontra. Cada modo de operação tem o seu consumo definido e, neste trabalho, o consumo dos modos foram configurados de acordo com o nó sensor modelo Mica2 [Mica2 2004], conforme definido na tabela 1. O ciclo de trabalho dos nós sensores é de 20 %. Todas as simulações foram feitas no simulador *Network Simulator 2.34* [ns2 2002] e o tempo de simulação é 1200 s. Os resultados obtidos correspondem à média aritmética de r simulações, em que escolheu-se r de tal forma que o intervalo de confiança seja de 95% dos valores [Jain 1991].

Tabela 1. Parâmetros de simulação utilizados.

| Parâmetros | |
|------------------------------|----------------------|
| Topologia da rede | <i>plana</i> |
| Nós (<i>hardware</i>) | homogêneos |
| Dimensão da rede | $100 \times 100 m^2$ |
| Posição do <i>sink</i> | $(0, 0)$ |
| Quantidade de nós | 200 |
| Raio de comunicação | 10 m |
| Energia inicial | 25 J |
| Ciclo de trabalho | 20 % |
| Eventos Aleatórios | |
| Raio do evento mínimo | 2 m |
| Raio do evento máximo | 10 m |
| Valor do parâmetro λ | 0.03 |
| Consumo de Energia do Rádio | |
| Transmissão de dados | 27 mA |
| Recepção | 10 mA |
| Estado de dormência | $1 \mu A$ |
| Potência | +5 dBm |

A geração de eventos na rede ocorre de forma aleatória, dentro das dimensões da rede. O raio de influência de cada evento é um valor dentro do intervalo definido entre *raio do evento mínimo* e *raio do evento máximo* e a duração máxima de cada evento é 50 s. A chegada de eventos à rede é modelada por uma distribuição de Poisson, com parâmetro λ (média) [Ross 1996].

As métricas consideradas nessa avaliação de desempenho são: (i) latência, que captura o tempo entre a transmissão do pacote pelo nó fonte e a chegada do mesmo ao nó *sink*; (ii) consumo de energia, que avalia a energia consumida por todas as operações

realizadas por um nó na rede; (iii) taxa de entrega, ou seja, o número de pacotes que um protocolo consegue entregar com sucesso para o *sink*; (iv) quantidade de contatos, que contabiliza a quantidade de contatos que ocorreram para o encaminhamento da mensagem do nó fonte até o nó *sink*.

4.2. Algoritmos Avaliados

Os dois algoritmos que foram avaliados juntamente com o OR-Tree constroem árvores de escoamento com objetivos diferentes e são ilustrados na figura 4. Ambos os algoritmos são aplicações da heurística *Shortest Path Tree* (SPT) [Krishnamachari et al. 2002]. O primeiro algoritmo, ilustrado na figura 4-(a), é chamado de Energy-Tree. Neste caso, as mensagens são trafegadas por rotas que apresentam o menor custo de energia, baseado em transmissões. O segundo algoritmo, ilustrado na figura 4-(b), é chamado Delay-Tree. Esse algoritmo faz com que as mensagens sejam entregues ao *sink* com o menor atraso possível. Enquanto o algoritmo Energy-Tree faz a mensagem trafegar o maior número de saltos possível, o algoritmo Delay-Tree faz o contrário, fazendo com que os nós sensores utilizem ao máximo as suas potências de transmissão, reduzindo o número de saltos que a mensagem pode trafegar. Ambos os algoritmos também são executados sobre o cenário DTN, utilizando o protocolo *Sensor-MAC*.

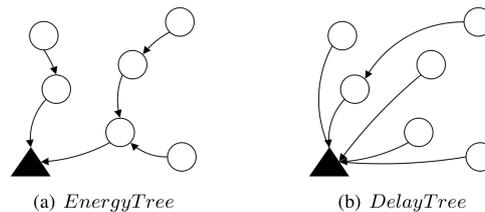


Figura 4. Árvores de escoamento usadas no roteamento.

4.3. Resultados de Simulação

Os resultados seguintes apresentam a avaliação das métricas frente à variação da quantidade de eventos na rede para $\lambda = 0.03$ e $\lambda = 0.09$. Devido a etapa de configuração dos algoritmos, a coleta de dados só é iniciada a partir do instante 150 s. Os gráficos da figura 5 apresentam os resultados da latência média acumulada durante o tempo de simulação para os três algoritmos avaliados. É possível verificar que o protocolo OR-Tree apresenta um desempenho bem superior aos demais protocolos. Seu valor de latência é bem menor que os demais protocolos e apresenta um comportamento constante, ou seja, baixa variância. Essas características são importantes para que padrões rígidos de QoS sejam atingidos. Esse mesmo comportamento não pode ser observado com o protocolo Energy-Tree, que apresenta latência média alta devido à grande quantidade de contatos necessários para chegar ao *sink*.

A avaliação do consumo de energia dos protocolos é apresentada nos gráficos da figura 6. Nota-se que o protocolo Energy-Tree apresentou o maior consumo de energia. Isso se deve ao grande número de mensagens necessárias para configuração da árvore de roteamento. Além disso, não foi realizado controle de potência para tirar proveito das transmissões de curta distância.

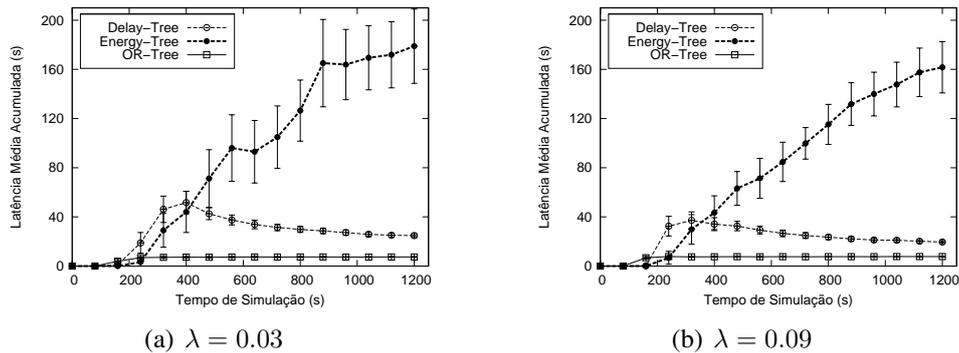


Figura 5. Latência média acumulada variando o valor de λ .

Observa-se também que os protocolos OR-Tree e Delay-Tree apresentam consumo bem semelhante. Porém, o protocolo OR-Tree apresenta o consumo um pouco menor que o Delay-Tree, cerca de 5% a menos. Essa redução se dá pelo fato do protocolo OR-Tree não utilizar pacotes de controle para construção das rotas e sim as informações compartilhadas com a camada MAC. Percebe-se pouca diferença do consumo de energia nos dois cenários de eventos. Em ambos, o tamanho do ciclo de trabalho é o mesmo, 20%, o que justifica a pequena diferença.

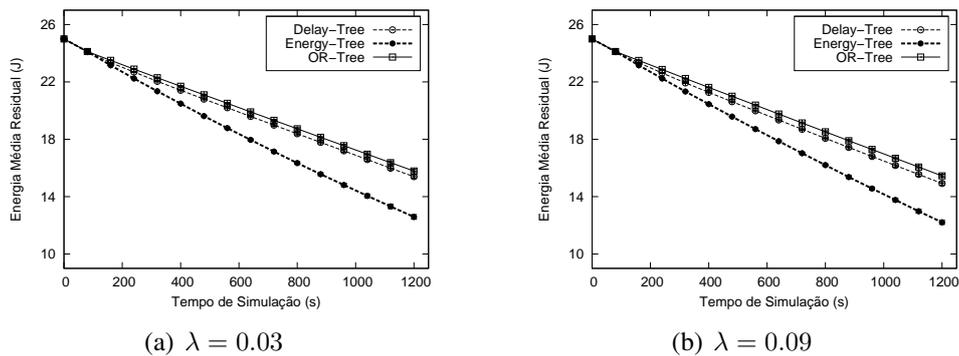


Figura 6. Consumo médio de energia variando o valor de λ .

A figura 7 apresenta a quantidade de dados entregues ao *sink* durante o tempo de duração dos eventos. Por apresentar um alto valor de latência, o Energy-Tree apresenta uma baixa taxa de entrega comparada aos demais protocolos. Também é possível verificar que os protocolos Delay-Tree e OR-Tree apresentaram desempenho bem semelhante. Por escolher sempre a rota com menor atraso, o protocolo Delay-Tree entrega mais dados, mas, com o comportamento bem semelhante ao OR-Tree. A esses resultados foi aplicado o teste da média zero e foi possível concluir que, para esse cenário, os protocolos Delay-Tree e OR-Tree apresentam desempenhos similares. Entretanto, pode-se notar que o aumento na quantidade de eventos também aumentou a taxa de entrega de dados.

Finalmente, os gráficos da figura 8 apresentam os resultados da quantidade de contatos realizados quando a quantidade de eventos é variada. Pode-se observar que dentre os protocolos avaliados, o que apresenta a menor quantidade de contatos é o OR-Tree. Isto indica que esse protocolo apresenta melhor desempenho dentre os três, devido ao fato de manter a taxa de entrega com uma quantidade menor de contatos, com um menor

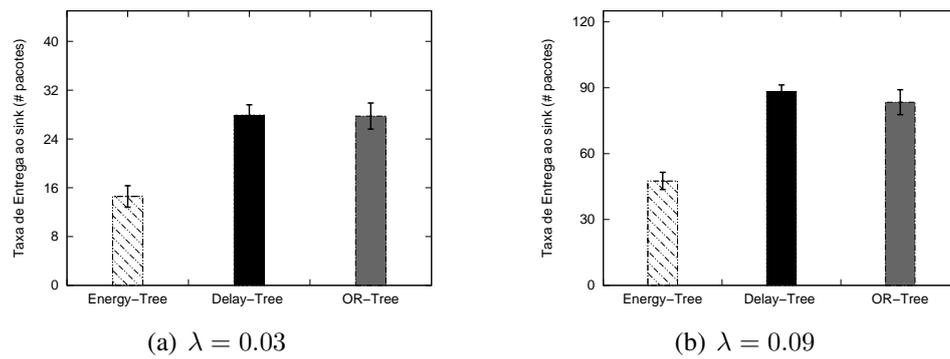


Figura 7. Taxa de entrega ao nó *sink* variando o valor de λ .

consumo de energia, e baixa latência.

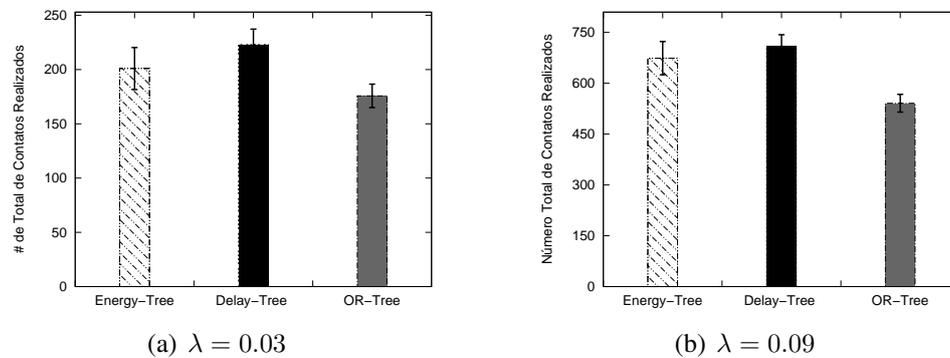


Figura 8. Quantidade de contatos realizados variando o valor de λ .

5. Conclusões e Trabalhos Futuros

Esse trabalho apresentou um novo protocolo para camada de roteamento que faz uso da interação entre camadas com a camada MAC para escolha do próximo salto do pacote. Foi possível concluir que, com o uso OR-Tree pode-se alcançar economia de energia, redução na latência, redução na taxa de contatos sem interferir na taxa de entrega. Este fato indica que o projeto integrado viabiliza a adaptação do roteamento à variação da topologia, e que a modelagem das RSSFs como redes DTNs é útil para um expressivo ganho de desempenho nas métricas citadas.

Como direções futuras pretende-se estudar novas técnicas para contornar problemas no roteamento para RSSFs quando levam em conta atrasos e desconexões. Também é possível estender esse trabalho avaliando o desempenho desses protocolos em cenários com falhas e também em cenários com outros tipos de eventos tais como: tráfego contínuo, híbrido e quando os eventos são iniciados pelo observador. Além disso, pretende-se incluir técnicas de agregação e ajuste do ciclo de trabalho dos nós no protocolo OR-Tree.

Referências

Abdulla, M. and Simon, R. (2006). A simulation analysis of multicasting in delay tolerant networks. In *WSC '06: Proceedings of the 38th conference on Winter simulation*, pages 2234–2241. Winter Simulation Conference.

- Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102–114.
- Anastasi, G., Conti, M., Di Francesco, M., and Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568.
- Demirkol, I., Ersoy, C., and Alagoz, F. (2006). Mac protocols for wireless sensor networks: a survey. *Communications Magazine, IEEE*, 44(4):115–121.
- e Hadi, F., Shah, N., Syed, A., and Yasin, M. (2007). Adaptive anycast: A new anycast protocol for performance improvement in delay tolerant networks. *Integration Technology, 2007. ICIT '07. IEEE International Conference on*, pages 185–189.
- Estrin, D., Govindan, R., Heidemann, J., and Kumar, S. (1999). Next century challenges: scalable coordination in sensor networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 263–270, New York, NY, USA. ACM.
- Harras, K. A., Almeroth, K. C., and Belding-Royer, E. M. (2005). Delay tolerant mobile networks (DTMNs): Controlled flooding in sparse mobile networks. *Lecture Notes in Computer Science*, 3462:1180–1192.
- Hougardy, S. and Prömel, H. J. (1999). A 1.598 approximation algorithm for the steiner problem in graphs. In *SODA '99: Proceedings of the 10th annual ACM-SIAM symposium on Discrete algorithms*, pages 448–453, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- Intanagonwiwat, C., Estrin, D., Govindan, R., and Heidemann, J. (2002). Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *International Conference on Distributed Computing Systems (ICDCS)*, Vienna, Austria.
- Jain, R. (1991). *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation and modeling*. John Wiley.
- Jain, S., Fall, K., and Patra, R. (2004). Routing in a delay tolerant network. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 145–158, New York, NY, USA. ACM.
- Juang, P., Oki, H., Wang, Y., Martonosi, M., shuan Peh, L., and Rubenstein, D. (2002). Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ASPLOS'02: Architectural Support for Programming Languages and Operating*, pages 96–107.
- Krishnamachari, B., Estrin, D., and Wicker, S. B. (2002). The impact of data aggregation in wireless sensor networks. In *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*, pages 575–578, Washington, DC, USA. IEEE Computer Society.
- Kurhinen, J. and Janatuinen, J. (2007). Geographical routing for delay tolerant encounter networks. In *ISCC*, pages 463–467. IEEE.
- Mica2 (2004). Mts/mda sensor and data acquisition boards user's manual. www.xbow.com.

- Mundur, P. and Seligman, M. (2008). Delay tolerant network routing: Beyond epidemic routing. *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, pages 550–553.
- Nakamura, E. F., Ramos, H. S., Villas, L. A., de Oliveira, H. A. B. F., de Aquino, A. L. L., and Loureiro, A. A. F. (2009). A reactive role assignment for data routing in event-based wireless sensor networks. *Computer Networks*, 53(12):1980–1996.
- ns2 (2002). The network simulator. www.isi.edu/nsnam/ns.
- Oliveira, C. T., Moreira, M. D. D., Rubinstein, M. G., Costa, M. G., and Duarte, O. C. M. B. (2007). *Redes Tolerantes a Atrasos e Desconexões*. Minicursos do Simpósio Brasileiro de Redes de Computadores.
- Polastre, J., Hill, J., and Culler, D. (2004). Versatile low power media access for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107, New York, NY, USA. ACM.
- Robins, G. and Zelikovsky, A. (2000). Improved steiner tree approximation in graphs. In *SODA '00: Proceedings of the 11th annual ACM-SIAM symposium on Discrete algorithms*, pages 770–779, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- Ross, S. M. (1996). *Simulation (Statistical Modeling and Decision Science)*. Academic Press, second edition.
- Vahdat, A. and Becker, D. (2000). Epidemic routing for partially connected ad hoc networks.
- Villas, L. A., Ramos, H. S., and Loureiro, A. A. F. (2009). Um protocolo de roteamento eficiente de agregação de dados para redes de sensores sem fio. In *SBRC'09 - Simpósio Brasileiro de Redes de Computadores*, pages 233–246.
- Warthman, F. (2003). Delay-tolerant networks (dtms) - a tutorial.
- Ye, W., Heidemann, J., and Estrin, D. (2002). An energy-efficient mac protocol for wireless sensor networks. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3:1567–1576 vol.3.
- Ye, W., Heidemann, J., and Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(3):493–506.
- Ye, W., Silva, F., and Heidemann, J. (2006). Ultra-low duty cycle mac with scheduled channel polling. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 321–334, New York, NY, USA. ACM.