

CRAL: um algoritmos de roteamento baseado em centralidade e energia para Redes de Sensores Sem Fio

Bruno P. Santos¹, Luiz F. M. Vieira¹, Marcos A. M. Vieira¹

¹ Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
31270-010 – Belo Horizonte – MG – Brasil

{bruno.ps, lfvieira, mmvieira}@dcc.ufmg.br

Abstract. *High throughput and energy are two important constraints of the Wireless Sensor Networks (WSNs). In this paper, we present the Centrality-based Routing Aware for L2Ns (CRAL)-Low Drop and Fast Delivery to mitigate these problems. The routing protocols are centrality-based and employ link quality estimators Expected Transmission Count (ETX) and Expected Transmission Time (ETT) to find best paths in wireless links. The suitable combinations of these techniques lead the algorithms to improve the literature results in delivery rate, energy consumption, time to delivery data packets. CRAL does this by building routing trees with high throughput and maintains low energy consumption. The simulation results show that CRAL is more reliable, efficient in energy consumption, robust, and favoring data fusion than Centrality Tree (CT) and Shortest Paths Tree (SPT).*

Resumo. *Vazão dos dados e energia disponível são algumas das principais limitações das Redes de Sensores Sem Fio (RSSF). Este artigo descreve os algoritmos Centrality-based Routing Aware for L2Ns (CRAL)-Low Drop e Fast Delivery para mitigar esses problemas. Os algoritmos são baseados em centralidade Sink Betweenness e métricas Expected Transmission Count (ETX) e Expected Transmission Time (ETT). A combinação apropriada dessas técnicas leva os algoritmos a melhorarem as soluções existentes em termos de taxa de entrega, consumo de energia e tempo de entrega das mensagens de dados. CRAL faz isso através a árvores de roteamento de alta vazão, ao passo que mantém um baixo consumo energético. Os resultados de simulação mostram que os algoritmos CRALs são mais eficientes energeticamente, robustos e favorecem a fusão de dados do que Centrality Tree (CT) e Shortest Paths Tree (SPT).*

1. Introdução

Eventos em sistemas computacionais são entidades de programação que representam um acontecimento do mundo real ou não. Como exemplo de sistema de processamento de eventos destacam-se as Redes de Sensores Sem Fios (RSSFs), as quais tem por finalidade coletar, analisar e reagir adequadamente aos eventos. . As fases de coleta e análise dos eventos são os mais interessantes desafios em RSSF. Na primeira fase, questões pertinentes são: como criar uma estrutura de roteamento? Quais são os modelos mais adequados para as características da rede? Na segunda fase, quais vantagens são possíveis tirar das diferentes técnicas de fusão de dados, mineração de dados, aprendizado de máquinas, dentre outras para que as RSSF sejam eficientes e poupem recursos?

RSSF geralmente são dirigidas a eventos, isto é, toda a rede coopera para que eventos coletados pelos nós sensores cheguem até os nós sorvedouros. Esta tarefa é desafiadora em face das severas limitações computacionais, de comunicação, e, em especial, do consumo de energia dos dispositivos empregados nessas redes. Portanto, modelos regulados especialmente para estes sistemas devem ser propostos.

Este trabalho trata do problema de roteamento de mensagens em redes de sensores sem fio. Neste sentido, são mostradas técnicas para ajustar métricas adequadas para o ambiente das RSSF com uso de centralidade para construção da estrutura de coleta dos dados. Os resultados de simulação mostram que a combinação apropriada dessas técnicas conduz a árvores de roteamento com alta vazão, ao passo que economiza energia, duas das principais limitações dos nós da rede. Além disso, as estratégias apresentadas se mostraram mais favoráveis a fusão de dados e se mostram mais eficientes ao entregar os pacotes de dados, quando comparadas com outras soluções.

Em resumo, as contribuições deste artigo incluem: 1) proposta do Centrality-based Routing Aware for L2Ns (CRAL)-*Low Drop* e *Fast Delivery* como algoritmos para roteamento baseado em centralidade e qualidade do enlace para RSSF; 2) uma avaliação da confiabilidade, eficiência e robustez dos algoritmos CRAL-*Low Drop* (CRAL-LD) e CRAL-*Fast Delivery* (CRAL-FD) em comparação com Shortest Paths Tree (SPT) e Centrality Tree (CT) (o estado da arte em roteamento baseado em centralidade); 3) avaliação de diferentes funções de fusão de dados em conjunto com CRAL-LD e CRAL-FD.

Os algoritmos propostos baseiam-se na importância topológica dos nós (centralidade) para escolher nós intermediários apropriados que irão compor a árvore de roteamento, além disso, CRAL-LD e CRAL-FD usam métricas que encontram caminhos de alta vazão em rede sem fio. As árvores criadas pelos algoritmos CRALs são aproximações da árvore de Steiner que é a solução ótima para o problema considerado neste trabalho.

Na próxima seção são discutidos os principais trabalhos relacionados. Na Seção 3, são discutidos os detalhes do problema abordado. Após isso na Seção 4, são exibidos os detalhes da arquitetura e algoritmos CRAL-LD e CRAL-FD. Na Seção 5, uma avaliação é realizada para mostrar o desempenho dos algoritmos CRALs comparados entre si e em relação ao estado da arte. Na Seção 6 são apresentadas as considerações finais.

2. Trabalhos Relacionados

Tabela 1. Comparação dos algoritmos de roteamento

	SPBC	TLC	FBC	RBC	CT	CRAL-LD	CRAL-FD	SPT	CNS	InFRA	DAARP	CTP
Centralidade	BC	BC	BC	BC	SBC	SBC	SBC	—	—	—	—	—
Distribuído	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
LQE	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓

*BC = Betweenness Centrality

*SBC = Sink Betweenness Centrality

Na Tabela 1 são classificados os algoritmos de roteamento de acordo com as características a seguir: i) aplicação de centralidade na construção da árvore de roteamento; ii) uso de Link Quality Estimation (LQE) para escolher enlaces de melhor qualidade. iii) implementação distribuída ou centralizada. Neste sentido, este artigo apresenta as bases de como construir os algoritmos CRAL-*Low Drop* e *Fast Delivery*, os quais são distribuídos e usam combinações de técnicas de centralidade e LQEs.

Shortest Path Betweenness-Centrality (SPBC) e Traffic Load Centrality (TLC) são os algoritmos de roteamento baseado na centralidade *Betweenness* mais tradicionais. Nestas abordagens, somente os menores caminhos são usados para transportar os dados. TLC difere de SPBC pois utiliza o tráfego para ponderar as rotas. Um dos problemas dessas estratégias de roteamento reside na utilização os menores caminhos, tornando-as estratégias inflexíveis para balanceamento de carga, tolerância a falhas ou acordo de níveis de serviço [Dolev et al. 2010]. Flow Betweenness-Centrality (FBC), ao contrário de SPBC e TLC, assume que as rotas são simples (não possuem ciclos) e considera igualmente caminhos de todos os comprimentos. FBC é razoável ao assumir rotas simples, porém FBC falha ao considerar todas as rotas igualmente, pois em redes é preferível caminhos mais curtos em detrimento dos longos. Em [Dolev et al. 2010] é apresentado o Routing Betweenness Centrality (RBC) como uma generalização das estratégias SPBC, TLC, e FBC. Contudo, estes algoritmos não foram projetados de modo distribuído como CRAL-LD e CRAL-FD. Além disso e não consideram as restrições impostas pelas RSSF como, por exemplo, taxa de perda nos enlaces da rede ou energia residual dos nós.

Dos algoritmos não baseados em centralidade e que não usam LQE apresentados na Tabela 1, tem-se SPT e Center at Nearest Source (CNS) como representantes tradicionais. Para SPT, os menores caminhos são utilizados para encaminhar todo o fluxo da rede. Já para CNS, um nó mais próximo do sorvedouro recebe mensagens dos descendentes e encaminha os dados para o sorvedouro.

Information Fusion-based Role Assignment (InFRA) e Data-Aggregation Aware Routing Protocol (DAARP) são discutidos em [Villas et al. 2010], estas abordagens são aproximações da árvore de Steiner [Fasolo et al. 2007]. InFRA e DAARP são otimizados para eventos que ocorrem de forma aglomerada (*clustering*) na área monitorada, isto é, os eventos ocorrem em poucos pontos da região monitorada e somente os nós a uma distância S do local do evento reportam informações. Ao contrário dessa estratégia que restringe as aplicações das RSSF, os algoritmos CRALs são otimizados para a estratégia que generaliza a anterior. Desta forma, permite-se que os eventos ocorram em locais aleatórios da região monitorada. Além disso, CRAL utiliza centralidade para escolher os nós intermediários da árvore de roteamento, enquanto InFRA e DAARP fazem o roteamento para cada *cluster* de evento detectado.

Em ambientes sem fio, a qualidade do enlace é um quesito que gera grande impacto no desempenho da RSSF [Baccour et al. 2009]. O consumo energético, tempo de entrega dos pacotes e efeito *overhearing* sofrem quando a qualidade dos enlaces é baixa. Collection Tree Protocol (CTP) usa o estimador de qualidade de enlace denominado Expected Transmission Count (ETX), específico para redes sem fio. Algoritmos como RBC e CT [Oliveira et al. 2010] usam centralidade para construir suas rotas, porém não consideram a qualidade dos enlaces. CRAL-LD, ao contrário dos anteriores, além de utilizar centralidade *Betweenness* para escolher os melhores nós repetidores no meio da rede, emprega ETX para encontrar caminhos de qualidade, minimizar o consumo de energia e efeito *overhearing*. CRAL-FD utiliza Expected Transmission Time (ETT), uma variação do ETX que ajusta a taxa de transmissão, como estimador do enlace, deste modo rotas são configuradas para obter alta vazão e baixa perda de pacotes.

3. Problema do roteamento em RSSF

Roteamento em RSSF apresenta diversos compromissos, muitos deles impostos pelas severas restrições energéticas, computacionais e de comunicação. Geralmente quatro requisitos são tidos como fundamentais e todo algoritmo de roteamento em RSSF deve buscar o equilíbrio entre eles: 1) **confiabilidade**, o protocolo deve entregar a maior quantidade de mensagens de dados, quando existe uma rota entre os participantes da comunicação; 2) **robustez**, o protocolo deve operar em diferentes topologias, cargas, e quantidade de nós sensores e na presença de falhas. 3) **eficiência**, em que o protocolo deve entregar as mensagens de dados com a menor quantidade de transmissões e tempo possível, economizando energia e mantendo a menor quantidade de estados possível. 4) **flexível**, o protocolo deve permitir técnicas adicionais sejam anexadas com facilidade tais como fusão de dados, mineração de dados, aprendizado de máquinas.

Uma alternativa para coletar dados de uma RSSF de modo confiável, robusto, eficiente e flexível é o emprego dos algoritmos CRALs. Estas abordagens balanceiam os compromissos impostos pelas quatro metas fundamentais das RSSF. Para realizar estas tarefas, os algoritmos CRALs combinam de modo inteligente centralidade com modelos adequados para ambientes sem fio, os quais serão discutidos nas próximas seções.

3.1. Centralidade

A posição topológica de um nó influencia o potencial de monitoramento e controle sobre os dados que fluem na rede. A centralidade *Betweenness* é a medida que pode ser aplicada para quantificar a importância topológica de um nó. Deste modo, em uma RSSF os nós sensores mais centrais são aqueles em que a maior parte das mensagens de dados fluem através deles.

A árvore de Steiner é conhecida em redes de computadores por suas diferentes aplicações. Para RSSF, esta árvore apresenta pontos ótimos para a fusão de dados, o que minimiza o consumo energético [Fasolo et al. 2007]. Entretanto, o problema de encontrar a árvore de Steiner é NP-Difícil, assim soluções aproximadas devem ser propostas.

A informação da centralidade dos nós é um dos objetos chave deste trabalho. Baseado na informação de centralidade é possível construir uma árvore de roteamento especial, em que os nós intermediários são os mais centrais. Estes nós sensores possuem a responsabilidade de encaminhar as mensagens dos nós em níveis mais baixos da árvore. Usando os nós mais importantes, segundo a sua centralidade, a quantidade de nós intermediários é reduzida e pontos de fusão de dados são favorecidos. Isto coloca esta estratégia como uma aproximação do problema da árvore de Steiner.

Sink Betweenness Centrality (SBC) é uma métrica de centralidade apropriada para coleta de dados em RSSF. O cálculo do SBC de um nó t depende do número de caminhos possíveis partindo dele até o sorvedouro s , e do número de suas participações, como nó intermediário, nos menores caminhos dos nós descendentes até o sorvedouro. Formalmente, seja $G = (V, E)$ uma árvore, onde V é um conjunto de vértices e E é um conjunto de arestas, além disso, seja $s \in V$ o sorvedouro, então o SBC de um nó t é definido como $SBC_t = \sum_{i \in \psi_t} \frac{\sigma_{ts}}{\sigma_{is}}$, onde σ_{ts} é o número de menores caminhos de t até s , σ_{is} é o número de menores caminhos de i até s , uma vez que $\psi_t = \{i \in V | t \in SP_{i \rightarrow s}\}$ é o conjunto de todos os menores caminhos de i até s ($SP_{i \rightarrow s}$) que contém t como um nó intermediário em, ao menos, um de seus menores caminhos.

A Figura 1 mostra o SBC para cada vértice de um grafo. Para calcular SBC_A é necessário saber que a quantidade de rotas partindo de A até S é 1 (numerador da equação) e o número de participações de A nas menores rotas dos vértices B, C, D que são respectivamente 1, 1, 2 (denominadores da equação). Portanto, $SBC_A = \frac{1}{1} + \frac{1}{1} + \frac{1}{2}$. O raciocínio é o mesmo para encontrar o SBC dos demais vértices. Note que S possui 0 rotas e D não faz parte do menor caminho de nenhum outro vértice, portanto o $SBC_{S,D} = 0$.

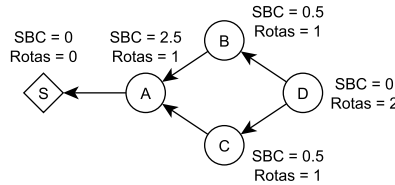


Figura 1. Sink Betweenness Centrality para cada vértice do grafo.

3.2. Qualidade do enlace sem fio

O estudo de estimadores de qualidade de enlace está sendo objeto de estudo dos últimos anos [Baccour et al. 2009, Baccour et al. 2012, Javaid et al. 2009]. Todo este esforço é explicado pelo fato de que os estimadores podem ter grande impacto no desempenho da rede, além de afetar o projeto dos protocolos das camadas superiores.

Em [De Couto et al. 2005] é apresentada a métrica ETX. Esta métrica é uma aproximação mais apropriada para estimar a qualidade dos enlaces sem fio da rede. ETX tem como principal característica minimizar o número esperado de transmissões e retransmissões requeridas para entregar os pacotes. Expected Transmission Time (ETT) [Draves et al. 2004] mantém todas as características do ETX e vai além possibilitando o ajuste da taxa de transmissão e do tamanho das mensagens de dados. ETT considera que os enlaces possuem tempos de transmissão diferentes.

4. Centrality-based Routing Aware for L2Ns

CRAL é um algoritmo de roteamento que seleciona nós intermediários da rede usando o critério de centralidade SBC e considera que os enlaces entre os nós sensores possuem taxa intermediária de perda, além de possuírem diferentes taxas de transmissão. São apresentadas duas versões do algoritmo CRAL: 1) CRAL-Low Drop utiliza como LQE a métrica ETX que reduz o número de perdas de mensagens ao escolher enlaces que necessitam de menor quantidade de transmissões para entregar os dados; 2) CRAL-Fast Delivery emprega ETT como métrica para estimar os enlaces, desta forma enlaces com menor tempo de entrega são escolhidos. Vale ressaltar que ETT deve ser usado sobre rádios que permitam diferentes taxas de transmissão. Deste modo, será assumido que CRAL-FD irá operar sobre o padrão 802.11.

Similar a maioria dos algoritmos de coleta de dados, CRAL começa pela requisição do sorvedouro. Esta requisição agenda o estabelecimento da árvore de roteamento usando nos estimadores de enlace ETX e ETT. Ao receber a requisição os nós enviam uma resposta de volta para seus pais na árvore. Deste modo, cada nó pode deduzir quantos filhos possui e seus caminhos, o que permite o cálculo da centralidade SBC.

4.1. Arquitetura

A arquitetura dos algoritmos CRALs são semelhantes exceto pelo estimador de enlaces utilizado. A Figura 2 exibe os principais módulos e seus relacionamentos. A arquitetura é dividida em plano de dados e plano de controle. No plano de controle, tem-se o Módulo de Roteamento e Centralidade (MRC) que é responsável pelo preenchimento da Tabela de Rotas, as rotas são escolhidas baseando-se nas informações do Estimador de Enlace (EE) e no Sink Betweenness Centrality dos nós na rede. O EE obtém informações dos enlaces por consultar a camada de enlace de dados. No plano de dados, o Módulo de Encaminhamento (ME) que é responsável pela recepção e encaminhamento das mensagens. Caso seja interceptada uma mensagem de controle, o ME sinaliza ao MRC para que as decisões apropriadas sejam tomadas. Caso a mensagem recebida seja de dados, ME consulta a Tabela de Rotas para verificar qual nó é o próximo salto da rota até o sorvedouro. Na Figura 2 também é mostrado um módulo provê a interface entre a camada de rede e as superiores.

CRAL usa quatro tipos de mensagens de controle: 1) Mensagem de Descoberta (MD), o qual contém os campos *ID* do nó, *P* número de caminhos, *H* número de saltos, e *RQ* qualidade da rota; 2) Mensagem de Resposta (MR) apresenta os mesmos campos da MD, porém adiciona o campo *SBC* para representar a centralidade; 3) Mensagem de Requisição de Rota (MRR) e Mensagem de Requisição de Desligamento (MRD) necessitam somente do campo *ID* do nó;

4.2. Algoritmo

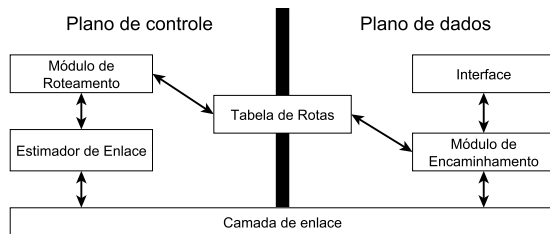


Figura 2. Arquitetura dos algoritmos CRALs.

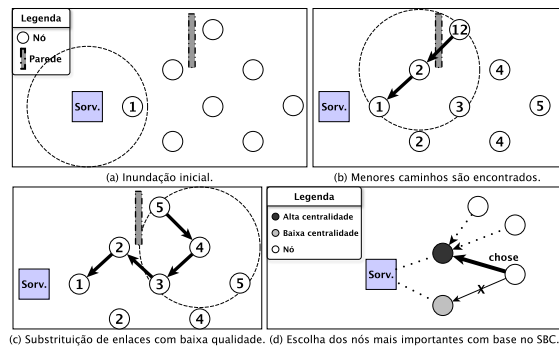


Figura 3. Operação dos algoritmos CRALs.

Descoberta e Resposta são as duas principais fases do algoritmo CRAL. Na primeira fase, o intuito é encontrar os caminhos com melhores enlaces segundo os estimadores ETX e ETT, além de encontrar o número de caminhos entre um nó e o sorvedouro. O Algoritmo 1 e a Figura 3 (a)(b)(c) ilustram essa fase, que tem seu início quando o sorvedouro inunda a rede com a MD (linha 2 Algoritmo 1 e Figura 3(a)). Os nós ao receberem a mensagem de descoberta, anotam as informações do enlace de melhor qualidade, número de saltos e o número de rotas de melhor qualidade até o sorvedouro (linhas 4-9 Algoritmo 1 e Figura 3(b)). O número de saltos é o numerador da equação SBC (vide Seção 3.1). Com o uso de estimadores como ETX e ETT, o caminho destacado na Figura 3(c) seria escolhido, o qual possui melhor qualidade.

Na fase de Resposta, os nós enviam, através dos caminhos encontrados, as informações computadas na fase inicial e seu valor de centralidade. O cálculo do SBC e escolha dos nós mais centrais são os objetivos dessa fase (veja o Algoritmo 2). O início dessa fase se dá quando os nós escalonam o envio da MR no final da fase de Descoberta. No momento que uma mensagem MR é interceptada (linha 1), o nó intermediário possui as informações necessárias para o cálculo do SBC, isto é, número de participação nas rotas dos descendentes e número de rotas (linhas 4 e 5 do Algoritmo 2), após o cálculo do SBC o nó intermediário atualiza a mensagem MR com suas informações de centralidade, número de rotas e qualidade da rota, após isso efetua o broadcast de MR. Um nó t ao receber uma mensagem com informações atualizadas verifica se a mensagem recebida é de um nó x mais próximo do sorvedouro (linha 6) e se x é o mais central (linha 7). Caso isso seja confirmado, o nó t irá escolher x como próximo salto (linha 9 do Algoritmo 2), a Figura 3(d) também ilustra essa operação de escolha.

A entrada e saída dos nós na rede são rotinas simples. Na primeira, o nó t envia para seus vizinhos diretos uma MRR. Os nós ao receberem MRRs, enviam uma MD com suas informações atualizadas, isto irá disparar o ciclo de atualização de rotas abordadas nos Algoritmos 1 e 2, como consequência o recálculo do SBC e escolha do nó mais central. Ao se desligar da rede, um nó envia uma MRD. Os nós filhos diretos ao receberem MRD dispara uma mensagem MRR para o estabelecimento de sua nova rota.

Algoritmo 1. Fase Descoberta.

```

1: if Sorvedouro then
2:   Inicializa inundação com MD.
3: else
4:   if Recebeu MD then
5:     if  $MD_{RQ} < RQ_t$  then
6:       Nova rota foi encontrada.
7:     else if  $MD_{RQ} = RQ_t$  then
8:       Outro caminho foi encontrado.
9:     end if
10:    Escalona broadcast MD.
11:    Escalona broadcast MR.
12:  end if
13: end if

```

Algoritmo 2. Nó t faz a escolha do nó mais central.

```

1: if Recebeu MR then
2:   if  $MR_{RQ} > RQ_t$  then
3:     //MR veio de um nó filho
4:     Efetua o cálculo do SBC.
5:     Broadcast MR. //atualiza os dados da mensagem antes do envio
6:   else if  $MR_{RQ} + RQ_{t \rightarrow MR_{ID}} \leq RQ_t$  then
7:     if  $MR_{SBC} > Vizinho_{maxSBC}$  then
8:        $Vizinho_{maxSBC} \leftarrow MR_{SBC}$ 
9:        $ProxSalto \leftarrow MR_{ID}$ 
10:    end if
11:  end if
12: end if

```

4.3. Funções de fusão de dados

A principal ideia da fusão de dados em RSSF é enviar mensagens agregadas, ao invés de enviar individualmente as mensagens dos nós para o sorvedouro. Isto faz da fusão de dados um poderoso mecanismo para reduzir o número de transmissões e consequentemente o consumo de energia da rede. Este trabalho foca em fusão de dados para RSSF no cenário em que o sorvedouro coleta os dados que são produzidos periodicamente pelos nós sensores. Neste sentido, três tipos de fusão de dados são as mais empregadas na literatura [Solis and Obraczka 2003], sendo elas: 1) *Simple*, em que um nó funde todas as men-

sagens que chegam durante um tempo determinado t . *Simple* tem fácil implementação, porém mensagens oriundas dos nós folha levam $t \times \text{saltos}$ para serem entregues ao sorvedouro. 2) *Periodic per hop simple (PerHop)*, os nós esperam uma pré-definida quantidade de tempo para fundir as mensagens, este tempo depende da posição do nó na árvore de roteamento. A vantagem desta abordagem sobre *simple* é a redução do tempo de pior caso. 3) *Periodic per hop adjusted (PerHopAd)*, os nós esperam uma quantidade de tempo t , que depende da sua distância (em tempo) até o sorvedouro. Além disso, a função dispara a mensagem agregada assim que uma mensagem de cada nó descendente é recebida. O tempo de espera faz com que aconteça um efeito de cascata, assim é possível reduzir ainda mais o tempo de entrega das mensagens agregadas, quando comparado com *PerHop*, sempre que seja frequente o envio de dados dos nós sensores.

4.4. Discussão

Manutenção da escolha dos enlaces é uma questão pertinente do algoritmo CRAL. Os estimadores de enlace ETX e ETT são baseados em software [Baccour et al. 2012] e necessitam que mensagens de sonda sejam envidadas. Esta característica implica num compromisso entre estabelecimento/manutenção das rotas e o consumo energético devido ao envio das mensagens de sonda. Para equilibrar esse compromisso, o protocolo CTP utiliza o algoritmo Trickle [Levis et al. 2003], o qual é adaptável no sentido de enviar poucas sondas quando a rede é estável, e aumentar o número de sondas quando a rede é instável reagindo adequadamente a mudanças topológicas. Este algoritmo também pode operar em conjunto com o algoritmo CRAL, visto que a sua arquitetura permite que seja anexada diferentes algoritmos para controlar o Estimador de Enlace (vide Figura 2).

Como tratar a questão dos nós mais centrais e próximos do sorvedouro tenderem a consumir mais energia? Uma maneira de tratar este problema seria efetuar o balanceamento do tráfego entre os nós com maior centralidade e os com baixa centralidade. Outra alternativa seria rotear baseado-se na energia residual dos nós. Uma terceira abordagem seria usar técnicas de *energy hole* [Li and Mohapatra 2005] para mitigar o problema.

5. Avaliação

Nesta seção serão analisados os algoritmos CRAL-LD e CRAL-FD em comparação com duas outras abordagens da literatura: CT e SPT¹ (como *baseline*). Dos algoritmos apresentados na Tabela 1, RBC e seus derivados não foram projetados para RSSF, portanto não podem ser comparados com CRAL-LD e CRAL-FD. InFRA e DAARP são de classes distintas de algoritmos de coleta de dados, eles não apresentam resultados satisfatórios quando eventos ocorrem de forma não aglomerada. CTP não utiliza centralidade e não foi projetado para operar com fusão de dados. CNS é um algoritmo clássico e apresenta resultados inferiores aos apresentados pelos algoritmos CT e SPT.

As análises aqui realizadas têm como objetivos principais validar os algoritmos propostos e avaliar o desempenho dos algoritmos CRAL-LD e CRAL-FD em relação ao estado da arte. Foram utilizadas as seguintes métricas para comparação: 1) Consumo de energia; 2) Número de Steiner nodes; 3) Eficiência (número transmissões para entregar as mensagens e taxa de entrega); 4) *Overhearing*; 5) Latência; 6) Qualidade da rota; 7) *Freshness* (velocidade de entrega das mensagens agregadas).

¹SPT não é exibido em alguns resultados, pois apresenta valores não competitivos com as abordagens apresentadas.

Figura 3. Parâmetros de Simulação

Parâmetro	Valor
Sorvedouro	1 borda
Número de nós sensores	1024
Rádio Padrão	802.11b
Taxa de perda no enlace (valor/aresta)	$0 < p \leq 1$
Densidade (nós/ m^2)	20
Mensagens de dados (mensagens/nó)	20
Duração do envio das mensagens de dados	20
Número padrão de retransmissões	10

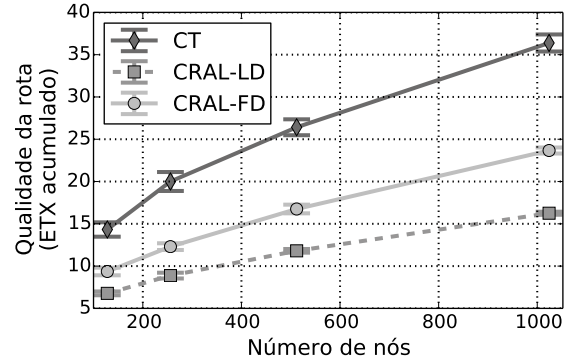


Figura 4. ETX médio das rotas variando número de nós sensores.

5.1. Cenários de Simulação e Métricas Utilizadas

O cenário padrão usado nos resultados é exibido na Tabela 3, para alguns resultados, os parâmetros podem ser variados e isso é descrito no conforme necessário. Para avaliar a escalabilidade, foram usadas redes de diferentes portes: pequeno (128 nós), intermediário (256 e 512 nós), e grande (1024 nós). Foram executados experimentos em 33 topologias diferentes (em cada porte da rede), em cada uma das topologias realizou-se 33 simulações, totalizando 1089 simulações por porte. Em cada gráfico, a curva representa os valores médios e as barras de erro representa o intervalo de confiança de 95%. Todos os algoritmos foram implementadas no simulador Sinalgo *v.0.75.3* [Sinalgo 2014]. Para cada simulação em que o número de nós sensores foi variado a área monitorada foi ajustada para manter a densidade de aproximadamente $20 \text{ nós}/m^2$.

5.2. Resultados de simulação

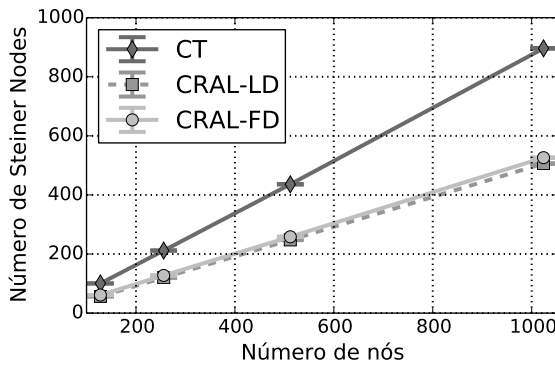


Figura 5. Steiner Nodes variando o número de nós sensores.

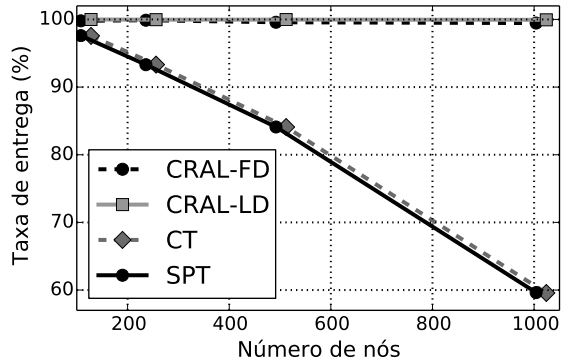


Figura 6. Taxa de entrega das mensagens de dados.

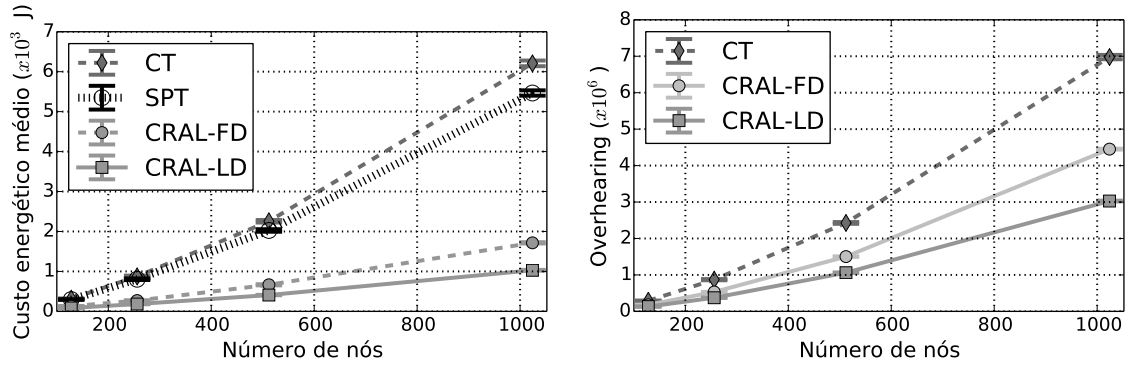
Inicialmente para mostrar a melhor qualidade das rotas encontradas pelos algoritmos CRAL-LD e CRAL-FD foi calculado o ETX acumulado médio das rotas. A Figura 4 mostra a qualidade das rotas encontradas pelos algoritmos CRAL-LD, CRAL-FD, CT e SPT. Nota-se que CRAL-LD encontra rotas com menor número de retransmissões para entregar os dados, CRAL-FD apresenta maior ETX acumulado que CRAL-LD, pois prioriza rotas com alta vazão além da qualidade, isto geralmente conduz a maior quantidade

de saltos. CT em todos os casos apresenta rotas de baixa qualidade, isto se deve ao uso do *hop count* como métrica. Baixo número de Steiner Nodes representa melhor árvore para fusão de dados e economia de energia. A Figura 5 mostra o número de Steiner Nodes para cada um dos algoritmos de roteamento. CT apresenta maior número de Steiner Nodes quando comparado com CRAL-LD e CRAL-FD para todos os cenários considerados. A maior diferença é atingida com redes de grandes porte, em que CRAL-LD e CRAL-FD apresentam aproximadamente metade dos Steiner Nodes de CT.

Os demais resultados dos experimentos foram divididos em duas partes: 1) *Reenvio limitado*, em que um número máximo de retransmissões é realizado, a mensagem é descartada quando esse número é atingido; 2) *Reenvio ilimitado*, em que as mensagens não são descartadas e são retransmitidas até que sejam entregues.

5.2.1. Reenvio limitado

Para mostrar que as estratégias aqui apresentadas são confiáveis, foi verificada a taxa de entrega dos dados variando o porte da rede. A Figura 6 exibe os resultados para cada algoritmo avaliado. Percebe-se que CRAL-LD e CRAL-FD apresentam alta taxa média de entrega 99.6% e 99.1% respectivamente. CT e SPT mostram queda na taxa de entrega quase linear quando o crescimento da rede é exponencial, isto é explicado pelos enlaces de baixa qualidade escolhidos e descarte de pacotes. Portanto, é de grande importância a consideração dos enlaces com taxa de perda intermediários na escolha das rotas.

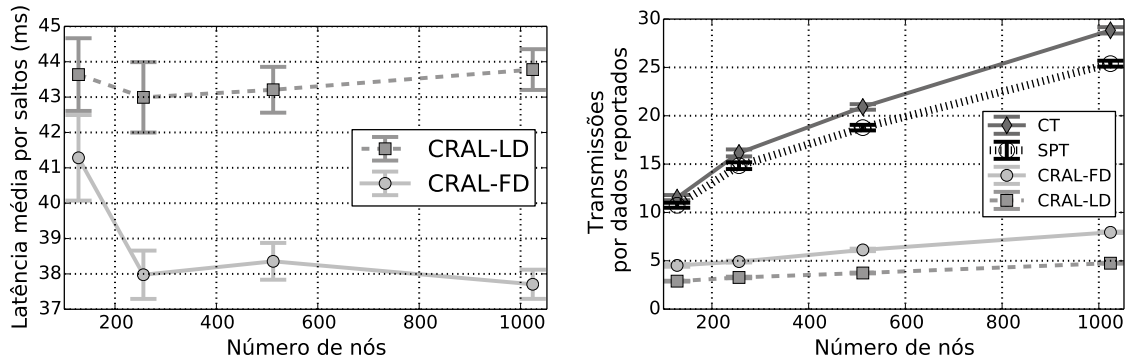


(a) Consumo de energia em Joules variando o número de nós sensores. (b) Efeito *overhearing* variando o número de nós sensores.

Figura 7. Avaliação do consumo energético e efeito *overhearing* dos algoritmos.

5.2.2. Reenvio ilimitado

Foram feitas análises sobre o consumo energéticos das abordagens CRAL-LD, CRAL-FD e CT em termos de bateria e efeito *overhearing*. Na Figura 7(a) é apresentado o consumo em Joules para cada algoritmo, percebe-se que CRAL-LD e CRAL-FD apresentam menor consumo energético, porém CRAL-FD supera o consumo de CRAL-LD, isto se deve ao fato de CRAL-FD apresentar rotas com ETX acumulado ligeiramente superiores ao CRAL-LD, além de um número de saltos superior. SPT e CT são similares para redes



(a) Latência média por saltos variando o número de nós sensores.

(b) Número de transmissões por dados reportados variando o número de nós sensores.

Figura 8. Avaliação eficiência dos algoritmos de roteamento em termos de latência por saltos e número de transmissões por dados reportados.

de tamanho pequeno e intermediário, porém CT apresenta maior consumo energético em redes de grande porte.

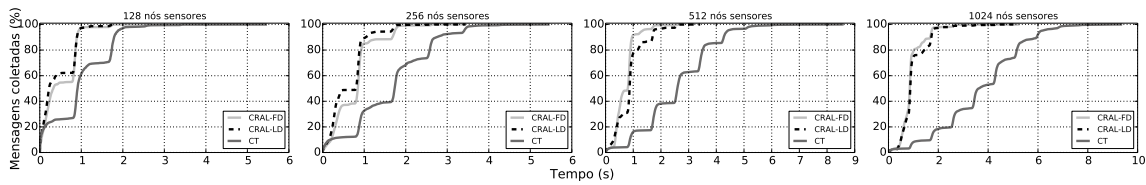
Neste trabalho, o efeito *overhearing* é definido como o número de mensagens escutadas por um nó que não são destinadas ao mesmo, o que implica em consumo de energia. A Figura 7(b) exibe o *overhearing* para diferentes tamanhos de rede. CRAL-LD é o algoritmo que apresenta menores valores, seguido por CRAL-FD e, por fim, CT.

A Figura 8(a) apresenta a latência para diferentes portes da rede. Neste experimento, a latência é o tempo médio por saltos das mensagens de dados. Percebemos que a estratégia que utiliza estimador de enlace que considera o tempo de transmissão (CRAL-FD) obtém melhores resultados. CT não é exibido, visto que seus resultados apresentam valores superiores a 30% aos aferidos por CRAL-LD ou CRAL-FD. Um resultado complementar é o número de transmissões por dados reportados. Este deve ser entendido como eficiência, isto é, a entrega dos dados com menor quantidade de transmissões possível. Na Figura 8(b), são exibidos os números de transmissões por dados, nota-se que CRAL-LD e CRAL-FD, novamente, apresentam os melhores resultados que não superam 8 transmissões por mensagem de dados, ao passo que SPT e CT superam 25 transmissões por mensagem de dados. Um alto número de transmissões para entregar os dados acarreta em desperdício de energia, aumento do efeito *overhearing* e alta latência.

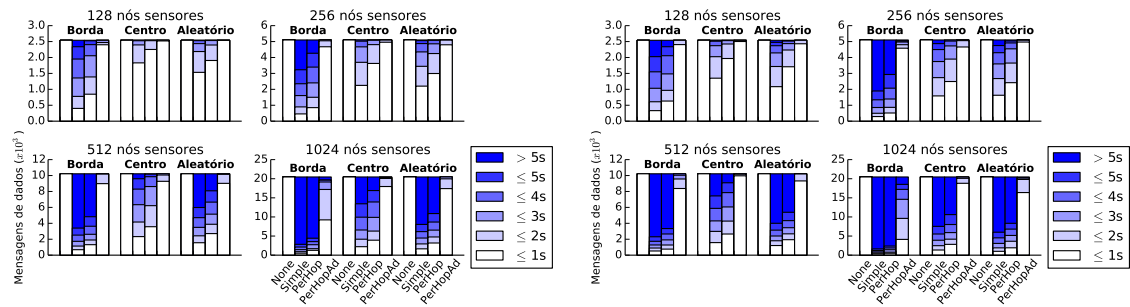
Para avaliar a flexibilidade das abordagens aqui apresentadas as diferentes técnicas de fusão de dados, discutidas na Seção 4.3, foram incorporadas aos algoritmos. O desempenho dos algoritmos sobre o *freshness* dos dados coletados são exibidos na Figura 9. *Freshness* é o tempo que a mensagem foi gerada subtraído do tempo que foi coletada no sorvedouro. Na Figura 9(a), tem-se a porcentagem de coletas e o tempo de entrega, usando *PerHopAd* e com o sorvedouro no centro da rede (cenário que apresenta melhores resultados para todos os algoritmos). É fácil perceber que os algoritmos CRAL entregam rapidamente as mensagens. Em redes grandes, CRAL-LD e CRAL-FD entrega mais que 90% das mensagens com apenas 2s, enquanto CT entrega somente 20% com o mesmo tempo. Esta característica é similar em todos os cenários avaliados.

Nas Figura 9(b)(c) são apresentados os resultados do *freshness*² para os algoritmos CRALs. Ao comparar as duas abordagens percebe-se pouca mudança. Assim, pode-se escolher CRAL-FD se o tempo de entrega de cada mensagem é um quesito mais importante ou CRAL-LD caso a taxa de entrega seja o fator mais relevante.

Em relação a idade dos dados com e sem uso de funções de fusão de dados, percebe-se que as funções *Simple* e *PerHop* não são tão ágeis quanto *PerHopAd*, isto se deve ao efeito cascata da função *PerHopAd*, o qual favorece a fusão de dados ao passo que mantém o tempo baixo de entrega dos dados fundidos. Assim, a combinação do *PerHopAd* com CRAL torna a fusão de dados ágil em redes de pequeno, intermediário e grande portes. O emprego de técnicas de fusão de dados com o algoritmo CT diminui consideravelmente a velocidade da entrega dos dados, sendo pouco ágil e escalável.



(a) Porcentagem de mensagens coletadas no tempo.



(b) *Freshness* para o algoritmo CRAL-LD.

(c) *Freshness* para o algoritmo CRAL-FD.

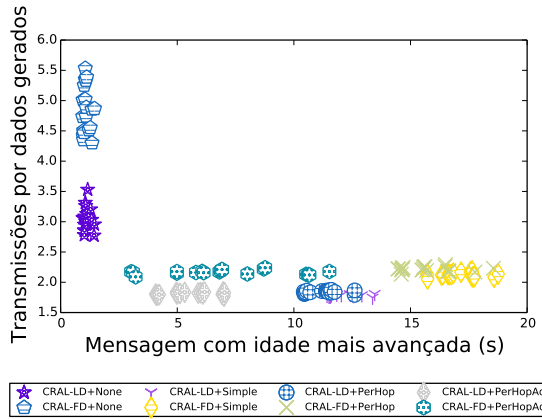
Figura 9. Avaliação do comportamento dos algoritmos de roteamento sobre a idade dos dados coletados em face às funções de fusão de dados.

Um dos maiores benefícios do uso de funções de fusão de dados é a redução do número de transmissões na rede. A redução do número de transmissões acarreta, por exemplo, em menor consumo energético, menor probabilidade de colisões na rede e atenuação do efeito overhearding. Na Figura 10 mostra o número de transmissões algoritmos de roteamento em conjunto com as funções de fusão de dados. O compromisso entre o número de transmissões e o *freshness* dos dados é analisado na Figura 10(a). Os algoritmos CRAL com o uso de fusão de dados reduzem o número de transmissões em detrimento do tempo de entrega das mensagens. É possível perceber que CRAL-FD em algumas simulações balanceia este compromisso obtendo baixo tempo de entrega e número de transmissões. Já CRAL-LD, reduz ao máximo o número de transmissões e mantém um tempo de entrega moderado entre 4s e 7s. CT apresenta número de transmissões e tempo superiores aos dos CRAL-LD e CRAL-FD em todos os cenários avaliados, o que é comprovado nos demais gráficos da Figura 10.

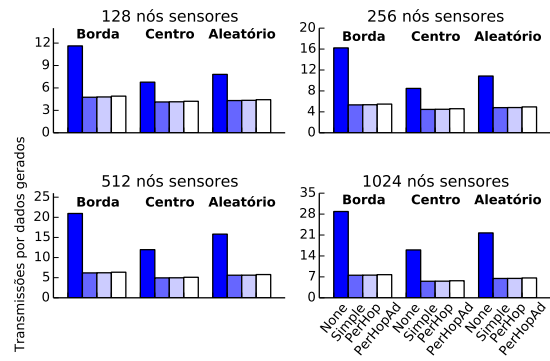
²O *Freshness* foi discretizado como 1s, 2s, ..., deste modo tem-se para cada função as mensagens recebidas com até 1s, 2s, ... de atraso, o qual chamaremos de idade da mensagem.

O número de transmissões por dados gerados para o da árvore CT é exibido na Figura 10(b). Percebe-se que CT precisa de alto número de transmissões para entregar os dados, quando não são usadas funções de agregação. Isto representa um alto consumo energético. Com o uso de funções de fusão de dados esse número é reduzido drasticamente, a maior diferença acontece quando a rede é de grande porte e o sorvedouro está na borda. Neste cenário, o uso das funções chega a transmitir 3 vezes menos do que o roteamento sem funções de fusão de dados.

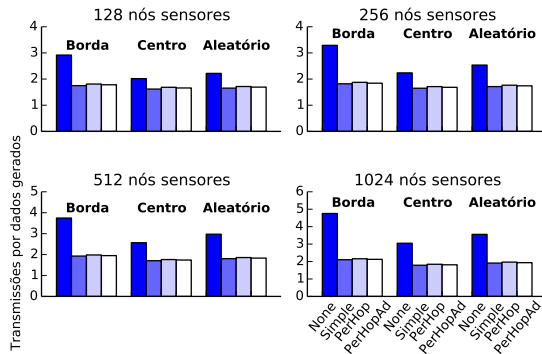
Os algoritmos CRALs se mostram mais econômicos do que CT em termos de transmissões por dados gerados. As Figuras 10(b)(c) mostram os resultados para respectivamente CRAL-LD e CRAL-FD, nota-se que o número de transmissões dessas abordagens sem o uso de funções de fusão de dados são aproximadamente o número médio de saltos da rede. Isto mostra que as abordagens são eficientes mesmo sem o uso de fusão de dados quando comparado com CT. Quando são empregadas as funções otimizadoras, o número de transmissões cai pela metade na maioria dos casos, chegando a ser aproximadamente 3 vezes menor quando em redes grandes e sorvedouro posicionado na borda da rede com o uso do CRAL-FD. Os algoritmos CRAL, quando comparado com CT, mostram uma substancial redução de transmissões e consequentemente menor consumo de energia, tanto com e sem o uso de fusão de dados.



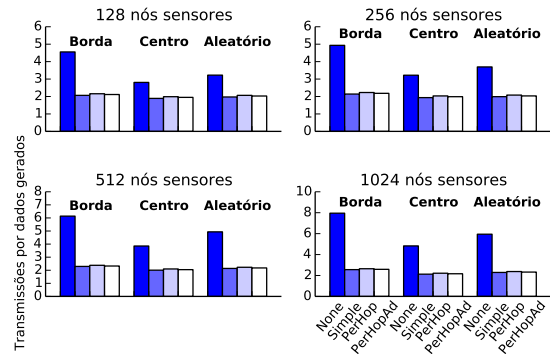
(a) Compromisso entre transmissões e *freshness*.



(b) Algoritmo CT.



(c) Algoritmo CRAL-LD.



(d) Algoritmo CRAL-FD.

Figura 10. Avaliação do número de transmissões por dados gerados para cada algoritmo de roteamento em face às funções de fusão de dados.

6. Conclusão

Neste artigo foi apresentado o CRAL, um algoritmo que roteia mensagens baseado na importância topológica dos nós intermediários em RSSF. CRAL utiliza-se das métricas de centralidade e estimadores de enlace para prover rotas de alta vazão, ao passo que economiza energia. CRAL é uma aproximação da árvore de Steiner, que é a solução ótima para o problema considerado neste trabalho. Os resultados apresentados mostram que CRAL apresenta alta taxa de entrega chegando a 99.6%, reduz o número de transmissão para entregar os dados em até 3 vezes. A entrega de dados atuais e fusão de dados apresenta um compromisso em que os algoritmos CRAL equilibram. CRAL-LD e CRAL-FD operando em conjunto com técnicas de fusão de dados, em especial *Periodic per hop adjusted*, se mostram flexíveis e adaptativos, ainda permanecendo eficientes, pois mais de 90% das mensagens levam até 2s para serem entregues, enquanto seu concorrente mais próximo leva triplo do tempo. Além disso, os algoritmos CRAL reduzem drasticamente o número de transmissões por dados gerados e consumo de energia na presença ou não de técnicas de fusão de dados, isto quando comparado com seu concorrente. Estes resultados colocam CRAL como uma alternativa para roteamento de baixo consumo energético e para fusão de dados em RSSF.

Referências

- Baccour, N., Koubaa, A., Jamaa, M., Youssef, H., and Zuniga, M. (2009). A comparative simulation study of link quality estimators in wsn. In *MASCOTS '09*. IEEE.
- Baccour, N., Koubaa, A., Mottola, L., Zúñiga, M. A., Youssef, H., Boano, C. A., and Alves, M. (2012). Radio link quality estimation in wireless sensor networks: A survey. *ACM Trans. Sen. Netw.*
- De Couto, D. S. J., Aguayo, D., Bicket, J., and Morris, R. (2005). A high-throughput path metric for multi-hop wireless routing. *Wirel. Netw.*
- Dolev, S., Elovici, Y., and Puzis, R. (2010). Routing betweenness centrality. *JACM*.
- Draves, R., Padhye, J., and Zill, B. (2004). Routing in multi-radio, multi-hop wireless mesh networks. MobiCom, New York, USA.
- Fasolo, E., Rossi, M., Widmer, J., and Zorzi, M. (2007). In-network aggregation techniques for wireless sensor networks: a survey. *Wireless Communications, IEEE*.
- Javaid, N., Javaid, A., Khan, I., and Djouani, K. (2009). Performance study of etx based wireless routing metrics. In *2nd IC4 2009*.
- Levis, P. A., Patel, N., Culler, D., and Shenker, S. (2003). *Trickle: A self regulating algorithm for code propagation and maintenance in wireless sensor networks*. Computer Science Division, University of California.
- Li, J. and Mohapatra, P. (2005). An analytical model for the energy hole problem in many-to-one sensor networks. In *VTC-2005-Fall IEEE 62nd*.
- Oliveira, E., Ramos, H., and Loureiro, A. (2010). Centrality-based routing for wireless sensor networks. In *Wireless Days, 2010 IFIP*.
- Sinalgo, D. C. G. (2014). Simulator for network algorithms (sinalgo).

Solis, I. and Obraczka, K. (2003). In-Network Aggregation Trade-offs for Data Collection in Wireless Sensor Networks.

Villas, L., Boukerche, A., de Araujo, R. B., and Loureiro, A. A. F. (2010). Highly dynamic routing protocol for data aggregation in sensor networks. In *ISCC 2010*. IEEE.