

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

Bruno P. Santos<sup>1</sup>, Luiz F. M. Vieira<sup>1</sup>, Marcos A. M. Vieira<sup>1</sup>

<sup>1</sup> 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)<sup>1</sup>-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, and 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 de energia. Os resultados de simulação mostram que os algoritmos CRALs são mais robustos, eficientes no consumo de energia 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 [Boukerche et al. 2014]. Como exemplo de sistema de processamento de eventos destacam-se as Redes de Sensores Sem Fios (RSSFs), as quais têm por finalidade coletar, analisar e reagir adequadamente aos eventos. As fases de coleta e análise dos eventos apresentam vários desafios como criar rotas, reduzir o tempo de transmissão e agregar dados, entre outros [Ramos et al. 2014, Boukerche et al. 2014]. Na fase de coleta, questões pertinentes são: como criar uma estrutura de roteamento?

---

<sup>1</sup>Low power and Lossy Networks.

Quais são os modelos mais apropriados para as características da rede sem fio? Já na fase de análise, como aplicar 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 e toda a rede coopera para que eventos sensoreado pelos nós alcancem seu destino. Esta tarefa é desafiadora em face das severas limitações computacionais, de comunicação, e, em especial, de energia dos dispositivos empregados nessas redes. Portanto, modelos regulados especialmente para estes sistemas devem ser propostos. O roteamento influi diretamente no consumo dos recursos dos dispositivos e é fundamental para realizar análise de dados em tempo de roteamento. Por isso, este trabalho trata do problema de roteamento de mensagens em redes de sensores sem fio. Neste sentido, são apresentadas técnicas para ajustar métricas adequadas para o ambiente das RSSF combinado com a importância topológica dos nós na rede para construção da estrutura de coleta dos dados. Os resultados de simulação mostram que a combinação apropriada dessas técnicas conduz a árvores de roteamento eficientes em dois quesitos: alta vazão e consumo de energia. Além disso, as estratégias apresentadas se mostraram mais favoráveis à fusão de dados e apresentam baixo número de transmissões para entregar os 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 protocolos de roteamento baseado em centralidade e qualidade do enlace para RSSF; 2) proposta do uso de diferentes funções de fusão de dados em conjunto com *CRAL-Low Drop (CRAL-LD)* e *CRAL-Fast Delivery (CRAL-FD)*.

Os algoritmos aqui propostos se diferem dos apresentados na literatura por um ou dois fatores. *Collection Tree Protocol (CTP)*, por exemplo, não faz uso da importância topológica dos nós para criar rotas, enquanto *Centrality Tree (CT)* não emprega *Link Quality Estimation (LQE)* para escolher as melhores rotas sem fio. Deste modo, *CRAL* leva em conta esses fatores para prover rotas de alta qualidade ao passo que cria rotas com pontos intermediários comuns, o que facilita a análise de dados. *CRAL-LD* e *CRAL-FD* se diferenciam no método utilizado para qualificar as rotas, em que empregam as métricas *Expected Transmission Count (ETX)* e *Expected Transmission Time (ETT)* respectivamente.

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

## 2. Trabalhos Relacionados

**Tabela 1. Comparação dos algoritmos de roteamento**

	SPBC	TLC	FBC	RBC	CT	CRAL-LD	CRAL-FD	CTP	SPT	CNS	InFRA	DAARP
Centralidade	BC	BC	BC	BC	SBC	SBC	SBC	-	-	-	-	-
Cod. 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 LQE para escolher enlaces de melhor qualidade. iii) código implementado de forma 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. Nessas 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, os autores de dos algoritmos não foram projetados de modo distribuído como CRAL-LD e CRAL-FD, portanto o cálculo da centralidade deve ser realizado antes da execução do roteamento. Além disso, não são consideradas as restrições impostas pelas RSSF como, por exemplo, taxa de perda nos enlaces da rede e não é realizada uma avaliação do consumo de energia dos nós.

Dos algoritmos não são baseados em centralidade e que não usam LQE apresentados na Tabela 1, tem-se *Shortest Paths Tree (SPT)* e *Center at Nearest Source (CNS)* como representantes tradicionais. Para SPT, os menores caminhos são utilizados para encaminhar todo o fluxo da rede. CNS escolhe 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]. InFRA e DAARP são aproximações da árvore de Steiner [Fasolo et al. 2007] e foram otimizados para eventos que ocorrem de forma aglomerada (*clustering*), isto é, os eventos ocorrem em poucos pontos da região monitorada e somente os nós com uma distância  $S$  do evento reportam os dados. O CRAL é otimizado para sensoramento periódico que generaliza a anterior. Desta forma, é permitido que eventos ocorram em locais aleatórios da região monitorada. Além disso, CRAL usa centralidade para escolher nós intermediários na árvore de roteamento, enquanto InFRA e DAARP fazem o roteamento baseado em cada grupo (*cluster*) de evento detectado.

A qualidade do enlace é um quesito que gera grande impacto no desempenho da RSSF [Baccour et al. 2009]. O consumo de energia, tempo e taxa de entrega dos dados sofrem quando a rota é de baixa qualidade. CTP usa um LQE denominado *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 *ETT*, que considera não só o *ETX* como também a taxa de transmissão do rádio, assim rotas são configuradas para obter alta vazão e baixa perda de pacotes.

### 3. Problema do roteamento em RSSF

Roteamento em RSSF enfrenta compromissos impostos por restrições energéticas, computacionais e de comunicação. Geralmente quatro princípios são tidos como fundamentais e os algoritmo de roteamento devem buscar o equilíbrio entre eles: 1) **confiabilidade**, o protocolo deve entregar a maior quantidade dos pacotes possível, quando existe rota entre as entidades da comunicação; 2) **robustez**, o protocolo deve operar em diferentes topologias, cargas, quantidade de nós e na presença de falhas. 3) **eficiência**, em que deve-se entregar os pacotes com a menor número de transmissões e tempo possível, economizar energia e manter a menor quantidade de estados possível.

Coleta dados de em RSSF de modo confiável, robusto e eficiente pode ser realizada com o protocolo CRAL, o qual balanceia os compromissos impostos pelos princípios das RSSF. Para realizar esta tarefa, CRAL combina inteligentemente centralidade com modelos regulados para ambientes sem fio, os quais serão discutidos nas próximas seções.

#### 3.1. Centralidade

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 posição topológica do nó influencia seu potencial controle sobre os dados que fluem na rede [Dolev et al. 2010]. A centralidade *Betweenness* é uma medida que quantifica a importância topológica de um nó [Almiron et al. 2010]. Um alto valor de *Betweenness* de um nó, em uma RSSF, indica que o nó roteia grande parte dos pacotes da rede. A informação da centralidade dos nós é um dos objetos chave deste trabalho, pois baseado nesta informação é possível construir uma árvore de roteamento especial, em que os nós intermediários (mais centrais) são pontos comuns às rotas da maioria dos nós. Usando os nós centrais a quantidade de intermediários é reduzida e pontos de fusão de dados são favorecidos. Isto coloca eleva esta estratégia a uma aproximação da árvore de Steiner.

*Sink Betweenness Centrality (SBC)* é uma métrica baseada no *Betweenness* para coleta de dados em RSSF [Almiron et al. 2010, Oliveira et al. 2010]. O computar o SBC de um nó  $t$  depende do número de caminhos partindo dele até o sorvedouro  $s$ , do número de suas participações, como intermediário, nos menores caminhos dos seus descendentes até o sorvedouro. Formalmente, seja  $T = (V, E)$  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  $\sigma_{ts}$  é o número de menores caminhos de  $t$  até  $s$ ,  $\sigma_{is}$  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  $SBC_{S,D} = 0$ .

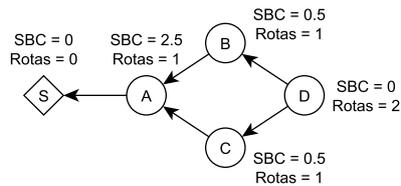


Figura 1. Sink Betweenness Centrality para cada vértice do grafo.

### 3.2. Qualidade do enlace sem fio

LQEs 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 grande impacto desta informação no desempenho da rede sem fio, além de afetar o projeto dos protocolos das camadas superiores. Em [De Couto et al. 2005] é apresentada a métrica ETX, a qual é uma aproximação mais apropriada para escolher enlaces sem fio. ETX minimiza o número esperado de transmissões e retransmissões para entregar os pacotes. *Expected Transmission Time (ETT)* [Draves et al. 2008] é uma variação mantém todas as características do ETX e vai além possibilitando escolher enlaces baseado-se na taxa de transmissão dados, isto é, ETT considera enlaces com tempos de transmissão diferentes.

## 4. Centrality-based Routing Aware for L2Ns

CRAL é um algoritmo de roteamento que seleciona nós intermediários usando o SBC e considera que os enlaces possuem taxa intermediária de perda, além de diferentes taxas de transmissão. Neste sentido, são apresentadas duas versões do protocolo: 1) *CRAL-Low Drop* que emprega ETX como LQE para minimizar a quantidade de perda de pacotes; 2) *CRAL-Fast Delivery* que usa ETT para estimar os enlaces, desta forma são escolhidos enlaces para reduzir o tempo de entrega e perda de pacotes. É assumido que ETT irá operar sobre rádios com multiplas taxas de transmissão, por exemplo, rádios operando com o padrão IEEE 802.11. Similar à maioria dos protocolos de coleta de dados, CRAL começa pela requisição do sorvedouro. Esta requisição agendará o estabelecimento da árvore de rotas usando estimadores ETX e ETT. Ao receber a requisição os nós enviam uma resposta de volta para o sorvedouro. Deste modo, cada nó pode deduzir quantos filhos possui e seus caminhos, o que permite calcular do SBC.

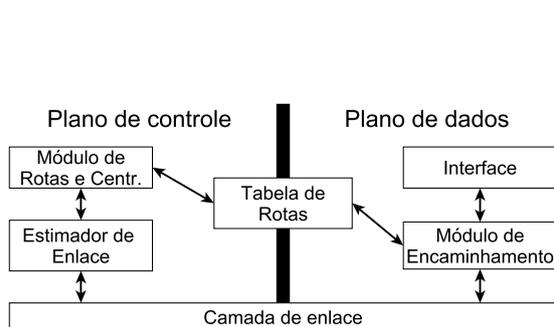
### 4.1. Arquitetura

A arquitetura do CRAL-LD e CRAL-FD é semelhante 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 Rotas e Centralidade (MRC) responsável pelo preenchimento da Tabela de Rotas, as rotas são escolhidas conforme as informações do Estimador de Enlace (EE) e da *Sink Betweenness Centrality* dos nós. 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. Em caso de mensagens de dados, o ME consulta a Tabela de Rotas para encaminhar o pacote ao próximo salto da rota até o sorvedouro. Na Figura 2 também é mostrado

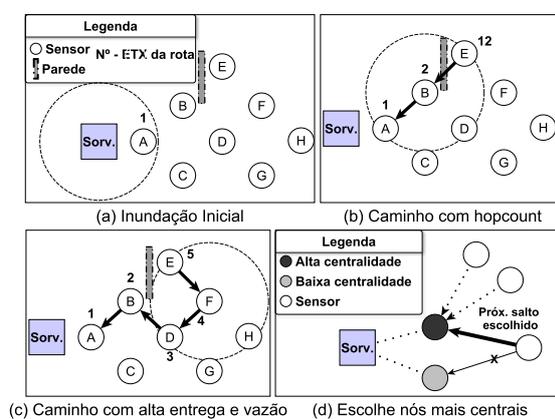
um módulo de interface que permite a comunicação inter-camadas. E através desta interface, foram implementadas as funções de fusão de dados descritas na Seção 4.3).

CRAL usa quatro tipos de mensagens de controle: 1) Mensagem de Descoberta (MD), o qual que contém os campos  $ID$  do nó,  $P$  número de caminhos,  $H$  número de saltos, e  $RQ$  qualidade da rota (ETX 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 8 Algoritmo 1 e Figura 3(a)). Os nós ao receberem a mensagem de descoberta, anotam as informações do melhor enlace, número de saltos e o número de rotas de melhor qualidade até o sorvedouro (linhas 10 a 15 e Figura 3(b)). No final desta fase obtemos o numerador da equação do SBC (Seção 3.1). E ao usar  $RQ$  baseado no ETX ou ETT o caminho da Figura 3(c) é favorecido.

Na fase de Resposta, através dos caminhos encontrados, os nós enviam 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 (Algoritmo 2). Esta fase inicia na linha 17 do Algoritmo 1. No momento que uma MR é interceptada (linha 1 Algoritmo 2) por um nó intermediário  $t$ , este possui as informações necessárias para o cálculo do SBC, isto é, o número de rotas de  $t$  e o número de participações de  $t$  nas rotas dos seus descendentes. Após o cálculo do SBC (linhas 4) o nó intermediário atualiza a mensagem MR com suas informações de centralidade, número de rotas e qualidade da rota e encaminha a MR para o sorvedouro (linha 6). Um nó  $t$  ao receber uma MR verifica se a mensagem recebida veio de um nó  $x$  mais próximo do sorvedouro do que  $t$  (linha 8) e se  $x$  é o mais central (linha 9). Em caso afirmativo, o nó  $t$  escolhe  $x$  como próximo salto (linha 9 do Algoritmo 2 e Figura 3(d)).

---

<b>Algoritmo 1: Fase Descoberta</b>	
[1]	$Sorv \leftarrow 1; Caminhos_t \leftarrow 1;$
[2]	$SBC_t \leftarrow 0; MAX_{SBC} \leftarrow \emptyset;$
[3]	$Saltos_t \leftarrow \infty; RQ_t \leftarrow \infty;$
[4]	$PróxSalto_t \leftarrow \emptyset;$
[5]	<b>se</b> $Sorv$ <b>então</b>
[6]	$MD_H \leftarrow Saltos;$
[7]	$MD_P \leftarrow Caminhos;$
[8]	Broadcast MD
[9]	<b>senão se</b> Recebeu MD <b>então</b>
[10]	<b>se</b> $MD_{RQ} < RQ_t$ <b>então</b>
[11]	//Melhor rota
[12]	<b>fim</b>
[13]	<b>se</b> $MD_{RQ} = RQ_t$ <b>então</b>
[14]	/*Rotas de mesma
[15]	qualidade*/
[15]	<b>fim</b>
[16]	Escalona Broadcast MD;
[17]	Escalona Broadcast MR;
[18]	<b>fim</b>

---



---

<b>Algoritmo 2: Fase Resposta</b>	
[1]	<b>se</b> Recebeu MR <b>então</b>
[2]	<b>se</b> $MR_{RQ} > RQ_t$ <b>então</b>
[3]	//MR de um nó filho
[4]	Efetua cálculo do SBC;
[5]	//Atualiza MR
[6]	Broadcast MR;
[7]	<b>fim</b>
[8]	<b>se</b>
[9]	$MR_{RQ} + RQ_{t \rightarrow MR_{ID}} \leq RQ_t$
[9]	<b>então</b>
[10]	<b>se</b> $MR_{SBC} > Max_{SBC}$
[11]	<b>então</b>
[11]	$MAX_{SBC} \leftarrow MR_{SBC};$
[12]	$PróxSalto_t \leftarrow MR_{ID};$
[12]	<b>fim</b>
[13]	<b>fim</b>
[14]	<b>fim</b>

---

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.

### 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 mecanismo poderoso 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 os dados que são produzidos periodicamente. 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 agrega as mensagens que chegam durante um tempo  $t$ . *Simple* tem fácil implementação, porém mensagens oriundas dos nós folha levam  $t \times saltos$  para serem entregues ao sorvedouro. 2) *Periodic per hop simple (PerHop)*, os nós também esperam uma pré-definida quantidade de tempo para fundir as mensagens, porém o tempo depende da posição do nó na árvore de rotas. A vantagem desta técnica sobre *simple* é a redução do tempo de pior caso. 3) *Periodic per hop adjusted (PerHopAd)*, os nós esperam um 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 ocorra um efeito de cascata, em que os nós próximos ao sorvedouro levam mais tempo agregando mensagens ao passo

que nós próximos as folhas da árvore levam menos tempo. Assim é possível reduzir ainda mais o tempo de entrega das mensagens agregadas.

#### 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, por isso o protocolo CRAL assume a existência do módulo EE para cálculo do LQE. O EE enfrenta um compromisso entre estabelecimento/manutenção das rotas e o consumo energético por enviar sonda. Para equilibrar o compromisso, CTP emprega o algoritmo *Trickle* [Levis et al. 2003], o qual envia poucas sondas quando a rede é estável e aumenta o número de sondas quando a rede é instável reagindo às mudanças topológicas. CRAL também pode operar em conjunto com o *Trickle*, visto que a arquitetura permite anexar diferentes algoritmos para controlar o EE (vide Figura 2).

Como mitigar o problema dos nós mais centrais e próximos ao sorvedouro consumirem mais energia? Uma maneira seria efetuar o balanceamento do tráfego entre os nós com maior centralidade e os com baixa centralidade. Alternativamente rotear baseando-se na energia residual dos nós é possível. Uma terceira abordagem é o uso de técnicas *energy hole* [Li and Mohapatra 2005].

### 5. Avaliação

Nesta seção serão analisados os algoritmos CRAL-LD e CRAL-FD em comparação com duas outras abordagens da literatura: CT e SPT<sup>2</sup> (como referência). 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).

#### 5.1. Cenários de Simulação e Métricas Utilizadas

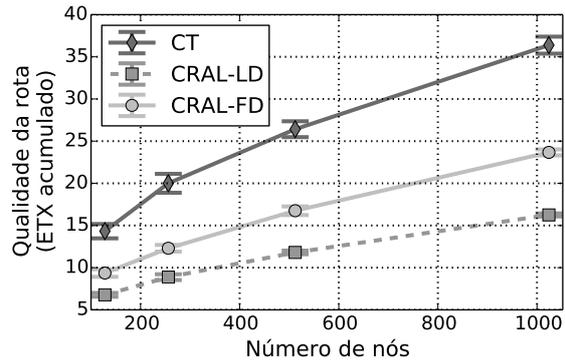
O cenário padrão usado nos resultados é exibido na Tabela 4, para alguns resultados, os parâmetros podem ser variados e isso é descrito no conforme necessário. Para avaliar a escalabilidade, foram usadas redes de diferentes portes: pequeno (128 nós), médio (256 e 512 nós), e grande (1024 nós) dispostos na região de interesse aleatoriamente. 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

---

<sup>2</sup>SPT não é exibido em alguns resultados, pois apresenta valores equivalentes ou inferiores aos do CT, o qual calcula o SPT internamente.

**Figura 4. Parâmetros de Simulação**

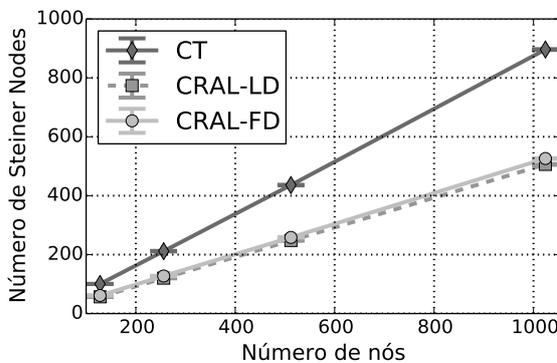
Parâmetro	Valor
Sorvedouro	1 borda
Nº de nós sensores	1024
Rádio Padrão	802.11b
Taxa de perda no enlace (valor/aresta)	$0 \leq p \leq 1$
Densidade (nós/m <sup>2</sup> )	20
Mensagens de dados (mensagens/nós)	20
Duração do envio das mensagens de dados (s)	20
Nº padrão de retransmissões	10



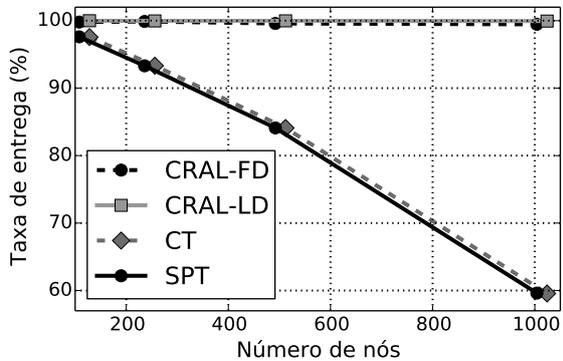
**Figura 5. ETX médio das rotas variando número de nós sensores.**

porte. Vale ressaltar que os parâmetros referentes ao tráfego de dados foram customizados para o envio periódico de mensagens. O modelo multi-taxa utilizado neste trabalho é baseado no padrão [IEEE 802.11b 1999]. Os parâmetros de alcance são baseados nas especificações do rádio Lucent ORiNOCO PC Card [Orinoco wireless networks 2014], assim é possível avaliar também a métrica ETT que considera diferentes taxas de transmissão. 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 implementados no simulador Sinalgo *v.0.75.3* [Sinalgo 2014].

## 5.2. Resultados de simulação



**Figura 6. Steiner Nodes variando o número de nós sensores.**

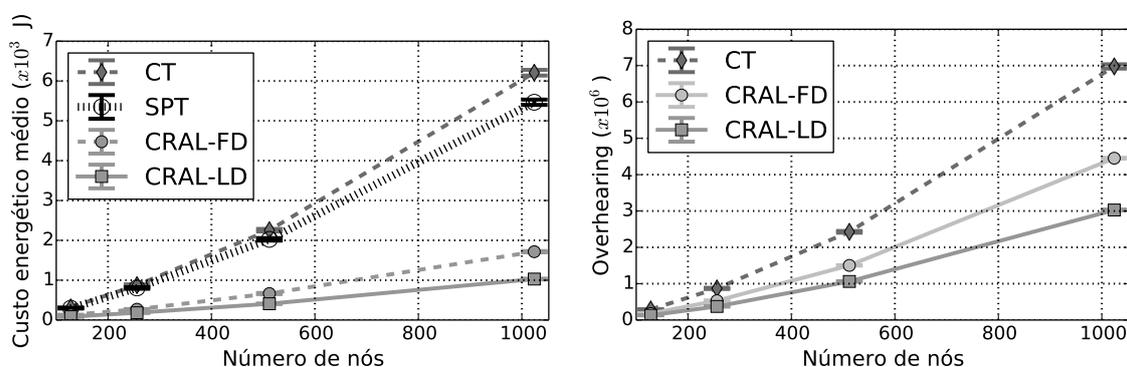


**Figura 7. Taxa de entrega das mensagens de dados.**

Inicialmente o ETX médio foi calculado para mostrar a qualidade das rotas, este valor indica o número médio requerido de transmissões (incluindo retransmissões) para rotear os dados até o sorvedouro. A Figura 5 mostra o ETX médio das rotas encontradas por CRAL-LD, CRAL-FD, CT e SPT. O protocolo CRAL-LD encontra rotas com menor número de transmissões para entregar os dados, CRAL-FD valores maiores 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 por usar o *hop count* como métrica de rota. Um baixo número de nós Steiner implica em uma árvore para fusão de dados melhores e que economia de energia. A Figura 6 mostra o número de nós Steiner para cada protocolo. CT apresenta maior quantidade de nós

Steiner do que CRAL-LD e CRAL-FD em todos os cenários considerados. A diferença é intensificada em redes de grande porte, em que CRAL-LD e CRAL-FD apresentam aproximadamente metade dos nós Steiner do que CT.

A taxa de entrega de dados foi verificada variando o porte da rede. Neste experimento foi habilitada uma quantidade máxima de retransmissões do pacote por saltos (Tabela 4). A Figura 7 exibe os resultados para este experimento. As abordagens CRAL-LD e CRAL-FD apresentaram taxa média de entrega superior a 99%, este alto valor pode ser explicada pela escolha de rotas de alta qualidade (Figura 5). CT e SPT mostram queda na taxa de entrega quando o crescimento da rede é exponencial, isto é esclarecido pela opção de enlaces de baixa qualidade. O aprendizado deste resultado é a importância de se considerar enlaces com taxa de perda intermediária ao escolher rotas.



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

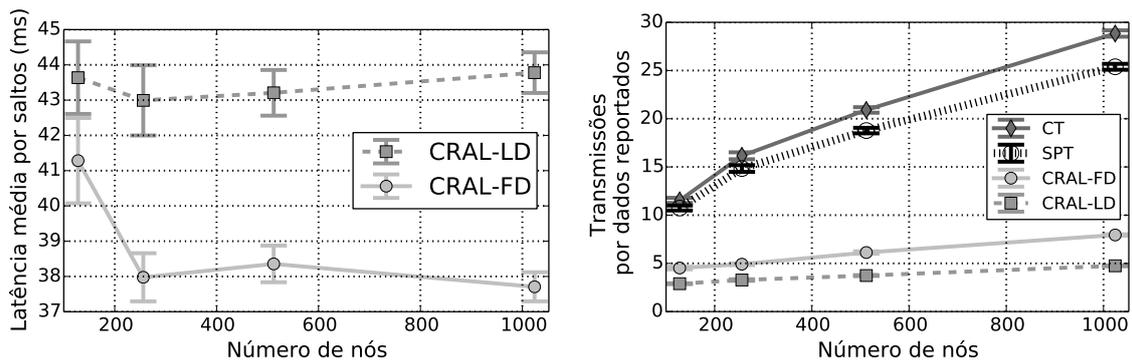
**Figura 8. Avaliação do consumo de energia e efeito *overhearing* dos algoritmos.**

Neste experimento é analisado o consumo de energia dos protocolos em termos de bateria e efeito *overhearing*. Na Figura 8(a) é apresentado o consumo em Joules dos protocolos, CRAL-LD e CRAL-FD se mostram mais econômicos do que SPT e CT. CRAL-FD supera o consumo de CRAL-LD, pois apresentar rotas com ETX médio ligeiramente superiores ao CRAL-LD (Figura 5). SPT e CT são similares para redes de porte pequeno e médio, porém CT apresenta maior consumo de energia em redes de grande porte, o que pode ser explicado pelo fato do uso do menor caminho em saltos até o sorvedouro. A Figura 8(b) exibe o *overhearing*<sup>3</sup> para os diferentes portes de rede. CRAL-LD é o algoritmo que apresenta menores valores, seguido por CRAL-FD e, por fim, CT. Estes resultados mostram o potencial do protocolo CRAL na economia de energia.

Para avaliar a eficiência (Seção 3) verificamos a latência<sup>4</sup> e o número de transmissões por dados gerados. A Figura 9(a) mostra a latência do CRAL-LD e CRAL-FD para diferentes portes da rede. CRAL-FD apresenta latência menor, pois escolhe rotas com maior vazão que CRAL-LD, visto que emprega o ETT como LQE. CT não é exibido, pois seus resultados apresentam valores, em média, 30% superiores aos aferidos por CRAL-LD ou CRAL-FD. A Figura 9(b) exibe o número de transmissões por dados

<sup>3</sup>*Overhearing*, neste trabalho, é definido como o número de mensagens escutadas pelo nó que não são destinadas ao mesmo acarretando em consumo de energia desnecessário.

<sup>4</sup>Neste trabalho, latência é o tempo médio por saltos das mensagens de dados.



(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 9. Avaliação eficiência dos algoritmos de roteamento em termos de latência por saltos e número de transmissões por dados reportados.**

gerados. CRAL-LD e CRAL-FD necessitam, em média, 8 transmissões por mensagem de dados, ao passo que SPT e CT superam 25 transmissões. Alto número de transmissões por dado implica em gasto de energia, aumento do *overhearing* e alta latência.

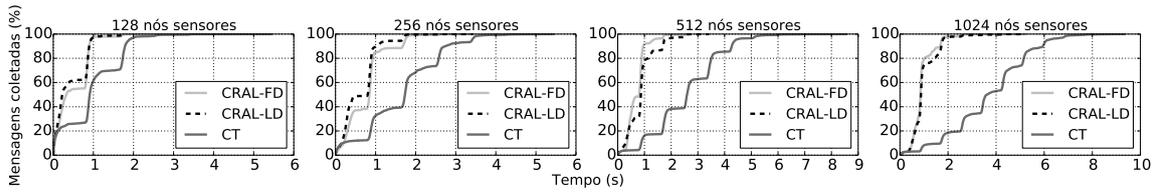
Para mostrar a flexibilidade do protocolo CRAL foram anexadas diferentes técnicas de fusão de dados (Seção 4.3). O desempenho dos algoritmos sobre o *freshness*<sup>5</sup> dos dados coletados é exibidos na Figura 10. A Figura 10(a) mostra a porcentagem de coletas e o tempo de entrega, usando *PerHopAd* e o sorvedouro no centro da rede (cenário que apresenta melhores resultados para todos protocolos). É fácil perceber o protocolo CRAL entrega rapidamente as mensagens. Em redes grandes, CRAL-LD e CRAL-FD entregam 90% ou mais das mensagens com apenas 2s, enquanto CT no máximo 20% com o mesmo tempo. Esta característica é similar em todos os cenários avaliados. Nas Figuras 10(b)(c) são apresentados os resultados do *freshness*<sup>6</sup> 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 no caso da taxa de entrega ser o fator mais relevante.

Em relação à 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*, a explicação disto é o efeito cascata da função *PerHopAd*, o qual favorece a fusão de dados ao passo que mantém baixo o tempo de entrega dos dados. A combinação do *PerHopAd* com CRAL torna a fusão de dados ágil em redes de todos os portes experimentadas. O emprego de técnicas de fusão de dados com o CT diminui consideravelmente a velocidade da entrega dos dados, sendo pouco ágil e escalável.

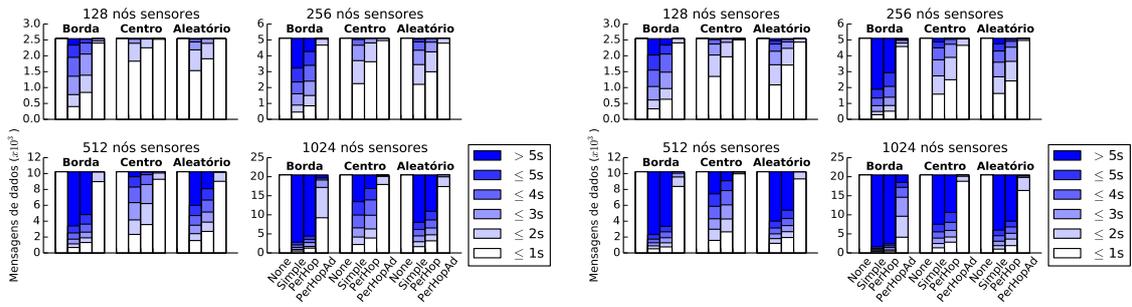
Um dos maiores benefícios das funções de fusão de dados é a redução da quantidade de transmissões na rede. Esta redução acarreta, por exemplo, em menor consumo de energia e atenuação do efeito *overhearing*. A Figura 11 mostra o número de transmissões algoritmos de roteamento em conjunto com as funções de fusão de dados. O compromisso entre transmissões e o *freshness* dos dados é analisado na Figura 11(a). CRAL-LD

<sup>5</sup>*Freshness* é o tempo que a mensagem foi gerada subtraído do tempo que foi coletada no sorvedouro.

<sup>6</sup>Neste experimento, o *freshness* foi discretizado como 1s, 2s, ..., assim para cada função as mensagens recebidas com até 1s, 2s, ... de atraso, o qual será chamado de idade da mensagem.



(a) Porcentagem de mensagens coletadas no tempo.



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

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

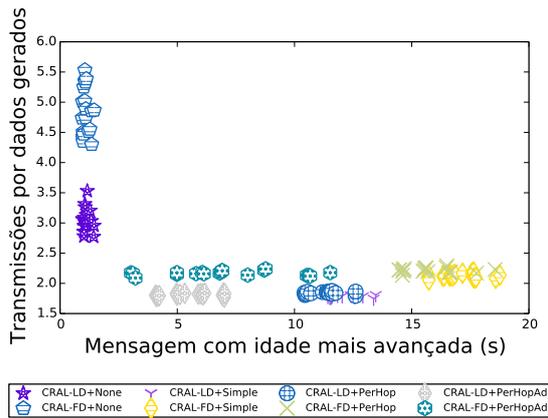
**Figura 10. Avaliação do comportamento dos algoritmos de roteamento sobre a idade dos dados coletados em face às funções de fusão de dados.**

e CRAL-FD sem técnicas de fusão apresentam baixo tempo de entrega dos dados, já quando se faz uso da fusão de dados tem-se maior tempo para entregar os dados porém com um número de transmissões maior. 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 ao do CRAL em todos os cenários avaliados. O número de transmissões por dados gerados para o da árvore CT é exibido na Figura 11(b). CT precisa de alto quantidade de transmissões para entregar dados, quando não são usadas funções de agregação. Já 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.

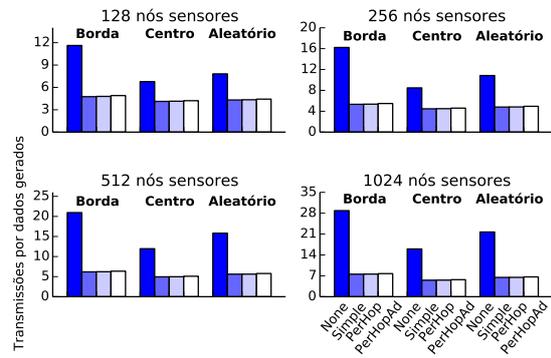
CRAL-LD e CRAL-FD se mostram mais econômicos do que CT em termos de transmissões por dados gerados. As Figuras 11(b)(c) mostram os resultados para respectivamente CRAL-LD e CRAL-FD, o número de transmissões dessas abordagens sem o uso de funções de fusão de dados é menor que CT com funções de fusão. Isto mostra que as abordagens são eficientes mesmo sem o uso de fusão de dados. Quando são empregadas as funções de fusão de dados, 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, apresentam substancial redução de transmissões e por consequência menor consumo de energia, tanto com e quanto sem o uso de fusão de dados.

## 6. Conclusão

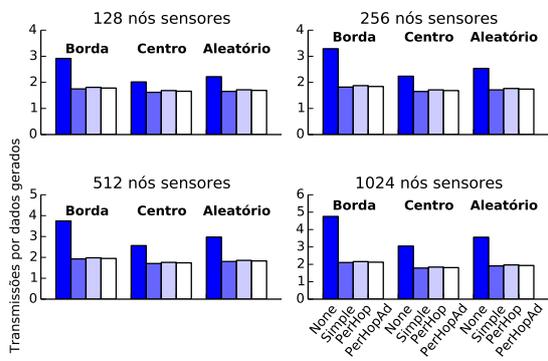
Neste artigo foi apresentado o protocolo CRAL que roteia mensagens baseado na importância topológica dos nós intermediários em RSSF. CRAL utiliza-se das métricas de



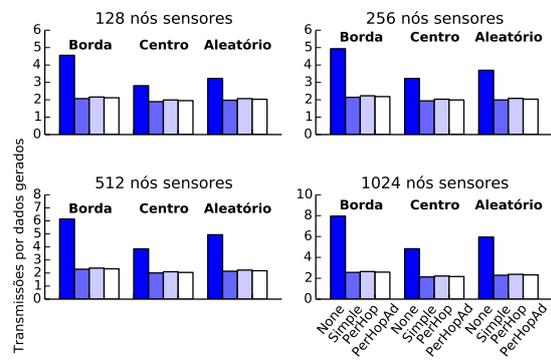
(a) Compromisso entre transmissões e *freshness*.



(b) Algoritmo CT.



(c) Algoritmo CRAL-LD.



(d) Algoritmo CRAL-FD.

**Figura 11. Avaliação do número de transmissões por dados gerados para cada algoritmo de roteamento em face às funções de fusão de dados.**

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 no trabalho. Os resultados mostram que CRAL apresenta alta taxa de entrega alcançando até a 99.6%, reduz o número de transmissão por dados em até 3 vezes. CRAL-LD e CRAL-FD operando em conjunto com técnicas de fusão de dados, em especial com a técnica PerHopAd, se mostram flexíveis e adaptativos, permanecendo eficientes, pois mais de 90% das mensagens levam até 2s para serem entregues, enquanto CT entrega 20% dos dados no mesmo tempo. Além disso, CRAL reduz drasticamente o número de transmissões por dados gerados e consumo de energia na presença ou não de técnicas de fusão de dados. Estes resultados colocam CRAL como uma alternativa para roteamento de baixo consumo de energia e para fusão de dados em RSSF.

Os itens discutidos na Seção 4.4 já encontram-se em desenvolvimento, como o balanceamento do consumo de energia. Além disso, desenvolvemos um protocolo que permite confirmação de dados [Santos et al. 2015], assim será possível solucionar o problema da perda de pacotes agregados.

## Referências

- Almiron, M. G., Ramos, H., OLIVEIRA, E., AO GM DE MENEZES, J., Guidoni, D., Stancioli, P., da Cunha, F., de Aquino, A., Mini, R., Frery, A., et al. (2010). Redes complexas na modelagem de redes de computadores. *Minicurso SBRC*.
- 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., Koubâa, 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.*
- Boukerche, A., Loureiro, A. A., Nakamura, E. F., Oliveira, H. A., Ramos, H. S., and Villas, L. A. (2014). Cloud-assisted computing for event-driven mobile services. *Mobile Networks and Applications*, 19(2):161–170.
- 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. (2008). 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*.
- IEEE 802.11b (1999). <http://standards.ieee.org/>.
- 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*.
- Orinoco wireless networks (2014). <http://www.orinocowireless.com/>.
- Ramos, H. S., Frery, A. C., Boukerche, A., Oliveira, E. M., and Loureiro, A. A. (2014). Topology-related metrics and applications for the design and operation of wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 10(3):53.
- Santos, B. P., Vieira, M. A. M., and Vieira, L. F. M. (2015). extend collection tree protocol. In *2015 WCNC: Mobile and Wireless Networks*, pages 1530–1535, New Orleans, USA.
- 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.