

# PAC-A e PAC-I: Protocolos de Árvore de Coleta para Plataformas com Dois Rádios

Gabriel S. Luz<sup>1</sup>, Luiz F. M. Vieira<sup>1</sup>, Marcos A. M. Vieira<sup>1</sup>, Omprakash Gnawali<sup>2</sup>

<sup>1</sup>Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais – Belo Horizonte, MG – Brasil

<sup>2</sup>Computer Science Department  
University of Houston, Houston, Texas, USA

**Abstract.** *The use of two radios per node increases the energy efficiency of wireless sensor networks. Given that data collection is one of the most important functions in wireless sensor networks, this paper presents and compares two new data collection protocols for wireless sensor networks with two radios. The protocols were implemented in TinyOS and evaluated experimentally in a testbed in the physical world, compared to the state of the art (CTP and CTP-Multi) and to each other, considering the metrics delivery rate, latency, flow in a network saturation scenario, total number of messages and the cost of maintaining routes. The results show the gain of the protocols for wireless sensor networks with two radios.*

**Resumo.** *O uso de dois rádios por nó aumenta a eficiência energética das redes de sensores sem fio. Dado que a coleta de dados é uma das funções mais importantes em redes de sensores sem fio, este artigo apresenta e compara dois novos protocolos de coleta de dados para redes de sensores sem fios com dois rádios. Os protocolos foram implementados no TinyOS e avaliados experimentalmente em um testbed no mundo físico, comparados com o estado da arte (CTP e CTP-Multi) e entre si, em relação à taxa de entrega, latência, vazão em cenário de saturação da rede, número total de mensagens e ao custo de manter as rotas. Os resultados mostram o ganho dos protocolos para redes de sensores sem fio com dois rádios.*

## 1. Introdução

Uma rede de sensores sem fios (RSSF) é composta por um conjunto de nós sensores distribuídos. Cada nó sensor possui uma variedade de sensores específicos para sua aplicação, um microprocessador e um transceptor de rádio, permitindo que os elementos da rede se comuniquem diretamente entre si, com cada nó podendo agir como coletor e retransmissor de dados [Vieira et al. 2003]. Com essas características, as RSSFs são redes de fácil implementação e de grande capacidade para sensoriamento distribuído, sendo amplamente usadas em diversas aplicações.

Em [Ruiz et al. 2004] foi mostrado que aplicações de RSSFs podem ser importantes no monitoramento de variáveis críticas em linhas de montagem em grandes indústrias, sensoriamento de grandes áreas ambientais (e.g. fauna, flora, temperatura, umidade, incêndios, etc.), monitoramento do trânsito em grandes centros urbanos e até em

aplicações médicas e biológicas no corpo humano. Com diversas aplicações no mundo real, RSSFs são muito relevantes no cenário atual.

Grande parte das aplicações de redes de sensores sem fios envolvem enviar dados dos demais nós da rede para nós encarregados de centralizar as informações coletadas, esse processo configura a coleta de dados. Por sua importância em RSSFs, diversos protocolos para coleta de dados foram propostos na literatura, dentre os quais o *Collection Tree Protocol* (CTP) [Gnawali et al. 2009] se destacou por sua eficiência e confiabilidade. O funcionamento do CTP está exposto na seção de trabalhos relacionados.

Recentemente, com o objetivo de melhorar a vazão e manter um baixo consumo de energia por byte transmitido, foram criadas plataformas para RSSFs com dois rádios por nó sensor. O custo de se adicionar um segundo rádio ao nó sensor é pequeno e gera benefícios no desempenho da rede e também melhora o consumo energético. Como exemplo podemos citar a plataforma Opal [Jurdak et al. 2011], cujo nó sensor possui um processador SAM3U Cortex-M3 MCU da Atmel e dois rádios, um AT86FR212 que opera na banda de 900MHz (que possui 10 canais), e um rádio AT86RF231 que opera na banda de 2.4 GHz (que possui 16 canais). Como cada rádio opera em uma banda diferente, pode-se impedir a interferência de um rádio com o outro. O uso de múltiplos rádios permite transmissões simultâneas entre os nós sensores, que além de aumentar a vazão na rede, também pode melhorar a estabilidade da rede, a taxa de entrega, reduzir o custo de transmissão e melhorar o consumo de energia por byte transmitido. Só para ilustrar, os rádios do Opal mote consomem 0.669 e 0.659 pJ/bit/m<sup>2</sup> enquanto o CC2420 do TelosB consome 11.89 pJ/bit/m<sup>2</sup> [Kusy et al. 2011]. Os ganhos no consumo de energia da plataforma Opal já foram demonstrados com profundidade em [Jurdak et al. 2011]. Além disso, Yin et al. [Yin et al. 2014] mostraram que a banda ISM de 900MHz apresenta melhor conectividade que a banda ISM de 2.4GHz. Portanto, o uso de dois rádios também traz benefícios na qualidade e conectividade dos enlaces.

O trabalho descrito neste artigo busca explorar os benefícios de plataformas de dois rádios na coleta de dados. Para isso, foram implementados dois novos protocolos, PAC-A e PAC-I, inspirados no CTP, usando os dois rádios de diferentes maneiras. Os protocolos foram implementados no *TinyOS* e testados no *testbed Twonet* [Li et al. 2013], que é composto por 100 *Opal Motes*. Os protocolos foram comparados entre si, ao CTP original e à modificação para dois rádios CTP-Multi detalhada em [Kusy et al. 2011].

As principais contribuições do artigo são: (i) os algoritmos PAC-A e PAC-I para coleta de dados em redes de sensores sem fio com dois rádios, (ii) a avaliação experimental em ambiente real dos algoritmos propostos, (iii) o avanço no estado da arte mostrado pelos resultados, em termos de taxa de entrega, vazão e número de *beacons*.

O artigo está organizado da seguinte forma. Na próxima seção são apresentados os trabalhos relacionados. Na seção 3 é descrito a definição do problema. A seção 4 traz o protocolo de Árvore de Coleta com Duas Árvores Independentes (PAC-I). A seção 5 detalha o Protocolo de Árvore de Coleta com Alternância de Rádios (PAC-A). A seção 6 discute os experimentos e resultados. Finalmente, a seção 7 conclui o artigo.

## 2. Trabalhos Relacionados

### 2.1. Collection Tree Protocol (CTP)

O CTP foi apresentado no artigo [Gnawali et al. 2009], juntamente com dois princípios para protocolos de roteamento : *datapath validation* e *adaptative beaconing*. O protocolo foi implementado para o *TinyOS* e sua descrição pode ser encontrada no documento TEP 123, pertence à documentação do *TinyOS*. Nos próximos parágrafos é dada uma visão geral do funcionamento do protocolo.

O CTP é um protocolo de coleta de dados baseado em roteamento por árvores. Cada árvore está enraizada em um nó de coleta, que é encarregado de receber e guardar os dados coletados pelos nós de sua árvore. Um nó sensor pertence à uma única árvore. Um nó de coleta se anuncia como raiz, informando ao CTP, que, se encarrega de criar uma árvore para essa raiz. Um nó não sabe a qual árvore pertence, sabe apenas quem é seu pai, que é para quem ele envia os pacotes recebidos de outros nós e os dados coletados. Para criar as rotas (árvores) o CTP estima a qualidade dos enlaces usando a métrica *Expected Transmission Count* (ETX), sendo que à cada transmissão direta entre dois nós vizinhos é associado um valor de ETX, representando a qualidade dessa transmissão (quanto maior o ETX, pior o enlace). Cada nó também possui um ETX associado, que é a soma dos ETX dos saltos que compõem o caminho desse nó até a raiz, sendo que o ETX de um nó raiz é igual a zero. A figura 1 mostra a árvore gerada pelo CTP para uma rede com sete nós, sendo o nó 1 a raiz.

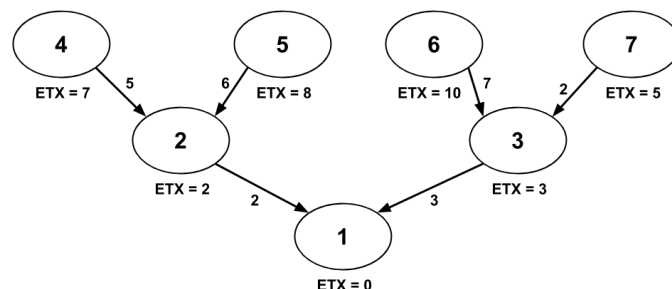


Figura 1. Exemplo de uma árvore formada pelo CTP, com os valores de ETX dos nós e dos enlaces.

O funcionamento do CTP pode ser dividido em três componentes principais: Estimador de enlaces, Roteamento e Encaminhamento de pacotes de dados. Na figura 2 pode-se ter uma visão simplificada de como esses componentes estão associados.

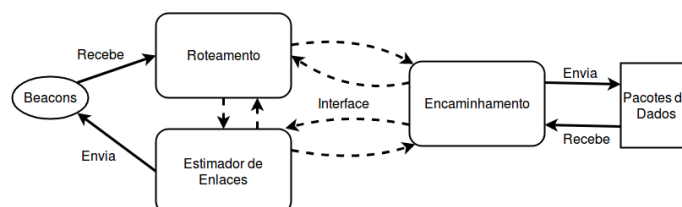


Figura 2. Esquema que ilustra as associações entre os três principais componentes do CTP.

A função do Estimador de enlaces é estimar o ETX de transmissões de um salto entre nós vizinhos. Para calcular o ETX, o estimador de enlaces usa os pacotes de

dados enviados pelo componente de encaminhamento e os *beacons* enviados pelo roteamento. Os *beacons* são enviados por *broadcast* periodicamente de acordo com um cronômetro que aumenta de forma exponencial até um valor máximo e pode ser reduzido a um valor pequeno caso alguns eventos específicos ocorram. Quanto maior o período do cronômetro, menos *beacons* são enviados, e quando o cronômetro é reduzido a um valor pequeno, significa que a taxa de envios aumenta para que topologia se ajuste à alguma mudança na rede. Uma mudança em um nó ou enlace é anunciada quando alguns bits são modificados no cabeçalho de um *beacon*. A esse mecanismo foi dado o nome de *adaptive beaconing* e foi inspirado no algoritmo *Trickle* [Levis et al. 2004]. Um *beacon* contém o identificador e o ETX do nó que o enviou, assim, ao receber um *beacon*, um nó atualiza a sua tabela de roteamento com o novo ETX do vizinho. O Estimador de Enlaces monitora se os pacotes de dados chegam ou não ao próximo salto e, baseando-se nessas informações, calcula o ETX de uma transmissão a cada cinco pacotes enviados. Os *beacons* também são usados para estimar a qualidade de um enlace, usando um número de sequência que é incrementado a cada envio, os números faltantes indicam os *beacons* perdidos. Dessa maneira, o nó  $u$  pode estimar a qualidade da rota até a raiz passando pelo nó  $v$ , somando o ETX de  $v$  até a raiz (ETX de  $v$ ) enviado pelos *beacons* e o ETX do enlace  $u \rightarrow v$  obtido pelos pacotes de dados e *beacons*.

O Roteamento é responsável por escolher o próximo salto para a transmissão de um pacote de dados e por controlar o envio de *beacons*. O próximo salto, que é o pai do nó na árvore, é aquele que se escolhido providenciará a rota com o menor ETX até a raiz. O envio de *beacons* é feito da maneira explicada no Estimador de Enlaces.

O Encaminhamento controla o envio de pacotes de dados criados no próprio nó ou recebidos de outros nós. Esse componente consulta o Roteamento para saber o próximo salto, escolhe quando enviar um pacote, passa informações para o Estimador de Enlaces e implementa o princípio do *datapath validation*. Um problema no roteamento em RSSFs é a presença de laços nas rotas fazendo com que um pacote nunca chegue a seu destino final. Para lidar com isso, os pacotes de dados contêm o ETX do nó que o retransmitiu pela última vez, como o ETX sempre deve diminuir em direção à raiz, então se um nó recebe um pacote com ETX menor do que o dele há uma inconsistência na rota e *beacons* devem ser transmitidos para reparar o laço. Dessa maneira, é possível validar o caminho de dados usando os pacotes de dados. Outro problema é a duplicação de pacotes que ocorre quando um nó retransmite um pacote que havia chegado com sucesso em seu destino, gerando dois pacotes iguais na rede. Esse problema pode crescer exponencialmente e deve ser tratado, no CTP isso é feito usando uma cache de mensagens em cada nó. Caso a mensagem recebida por um nó já esteja em sua cache, ele a descarta e não a passa adiante.

## 2.2. Protocolos Similares ao CTP

Já foram propostas diversas modificações para o CTP. O XCTP [Santos et al. 2015] provê além da comunicação dos nós para a raiz, também da raiz para os nós, estendendo as funcionalidades do CTP. Outro exemplo é o *Matrix* [Peres et al. 2018] que utiliza um protocolo de coleta de dados baseado em árvores, como o CTP, como base para um protocolo de roteamento com alocação hierárquica de endereços IPv6 para redes móveis. Também foram criados protocolos que exploram o uso de múltiplos canais criando rotas melhores e reduzindo os efeitos da interferência, como o protocolo *Multi-channel CTP* [Phokaew et al. 2014].

Entretanto há muito pouco na literatura sobre o uso de múltiplos rádios para melhorar a coleta de dados. Exemplos de trabalhos nessa área são: o *MMCR* [Anguswamy et al. 2009] que utiliza múltiplas interfaces de rádio e múltiplos canais, e o [Jin et al. 2018] que propôs um algoritmo de escalonamento de *convergecast* com múltiplas interfaces de rádio. O principal protocolo encontrado na literatura que adapta o CTP para usar múltiplos rádios foi o CTP-Multi [Kusy et al. 2011], que modifica o Estimador de Enlaces para escolher a melhor banda de rádio para um enlace e modifica o Roteamento para enviar *beacons* por múltiplos rádios. Ele foi implementado usando uma tabela separada para cada banda de rádio no Estimador de Enlaces, obtendo uma estimativa para a qualidade do enlace entre o nó e seus vizinhos usando bandas diferentes, enquanto, no Roteamento usa-se apenas uma tabela de roteamento preenchida a partir dos valores de ETX dos vizinhos até a raiz recebidos por meio dos *beacons*. No momento de atualizar as rotas escolhe-se o melhor vizinho e o melhor rádio para transmitir um pacote de dados. Neste artigo exploramos duas abordagens que permitem melhorar o desempenho da rede não só escolhendo o melhor rádio como faz o CTP-Multi, mas construindo duas árvores independentes ou usando os rádios de forma alternada na coleta de dados.

### 3. Definição do Problema

Os principais objetivos que o CTP [Gnawali et al. 2009] busca alcançar são:

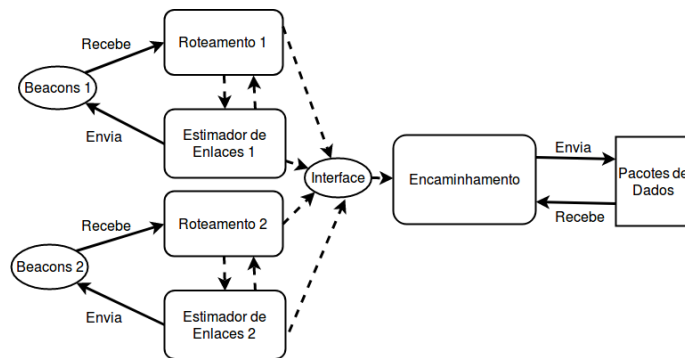
- **Confiabilidade** : Um protocolo de coleta de dados deve ser capaz de obter uma taxa de entrega maior que 90%.
- **Robustez** : Um protocolo deve conseguir operar sem ajustes e configurações em diversos tipos de topologias, condições de rede e ambientes.
- **Eficiência** : Deve-se minimizar o número de transmissões e de estados para transmitir um pacote pela rede.
- **Independência de Hardware**: Não deve assumir alguma característica específica de algum chip de rádio.

Outros fatores também são importantes na coleta de dados, especialmente a latência e a vazão. Quando se trata de monitoramento em tempo real e outras aplicações que precisam de uma resposta rápida para certos eventos, uma baixa latência é importante e, por isso alcançá-la em protocolos de coleta de dados já foi objeto de estudo [Adelani et al. 2018, Xu et al. 2011]. A vazão também é uma importante métrica para protocolos de coleta de dados, já que é necessário suportar um alto volume de dados na rede, algo que pode melhorar a confiabilidade e a eficiência do protocolo.

O problema que este trabalho busca resolver é o de criar novos protocolos que melhorem o CTP em relação à confiabilidade, robustez e eficiência na coleta de dados, e que também melhorem a vazão e a latência, aproveitando os dois rádios disponíveis.

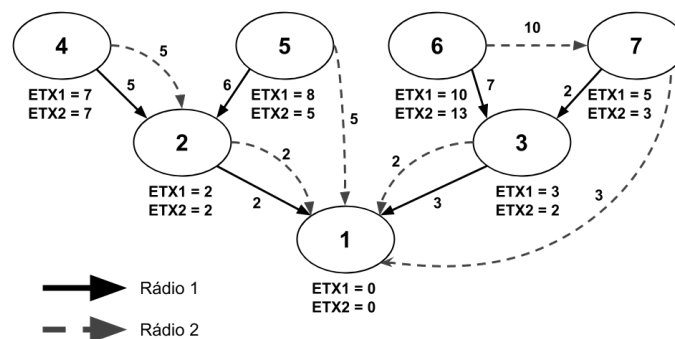
### 4. Protocolo de Árvore de Coleta com Duas Árvores Independentes (PAC-I)

O PAC-I expande o conceito do CTP de uma para duas árvores com um mesmo nó raiz. Nesse protocolo cada nó pertence a duas árvores independentes, sendo que a primeira é usada apenas pelo rádio 1 e a segunda é usada apenas pelo rádio 2. Para criar as duas árvores o Roteamento e o Estimador de Enlaces foram duplicados da maneira representada na figura 3.



**Figura 3. Esquema que ilustra as associações entre os principais componentes do PAC-I.**

Quando um pacote é criado nesse protocolo, ele guarda um bit que indica qual rádio deve ser usado para transmiti-lo. Um pacote com o bit de rádio igual a 1 só poderá ser transmitido pelo rádio 1 e, por isso, esse pacote permanecerá na árvore 1 do momento em que foi criado até chegar no nó raiz. As árvores são designadas aos pacotes de forma alternada, ou seja, o primeiro pacote é da árvore 1, o segundo é da árvore 2, o terceiro da árvore 1, o quarto é da árvore 2 e assim sucessivamente. Dessa maneira, cada árvore recebe metade dos pacotes.



**Figura 4. Exemplo das duas árvores formadas pelo PAC-I, com os valores de ETX dos nós e dos enlaces para cada rádio.**

Como o Roteamento e o Estimador de enlaces foram duplicados, as árvores são criadas de forma independente uma da outra, porém com o mesmo nó raiz. Assim a construção das duas tabelas de roteamento e o envio de *beacons* são independentes para cada rádio. Dessa maneira, um nó pertence a duas árvores, possuindo dois pais, que podem ser distintos ou não. Como os rádios operam em larguras de bandas diferentes, o alcance dos rádios não é igual e, portanto, podem ter pais diferentes. A figura 4 mostra as duas árvores criadas pelo PAC-I para a mesma rede da figura 1. Nela pode-se perceber que árvores compartilham a mesma raiz e cada nó possui dois pais e dois valores de ETX. O Encaminhamento envia pacotes de acordo com o bit de rádio. Antes de enviar, verifica o bit de rádio e consulta o respectivo componente do Roteamento para saber para qual pai enviar esse pacote.

## 5. Protocolo de Árvore de Coleta com Alternância de Rádios (PAC-A)

O PAC-A utiliza uma abordagem diferente do PAC-I. Ao invés de sempre usar os dois rádios para enviar e receber pacotes, esse protocolo faz com que um nó receba por um rádio e envie por outro. A figura 5 mostra como fica a árvore encontrada pelo PAC-A na mesma rede das figuras anteriores. Alternar os rádios segue o princípio usado pelos protocolos para encontrar caminhos em RSSFs com dois rádios *FastForward* [Ekbatanifard et al. 2013] e *Dois Caminhos* [Júnior et al. 2017], que conseguiram melhorar a vazão fazendo com que um nó envie e receba pacotes ao mesmo tempo usando os dois rádios simultaneamente.

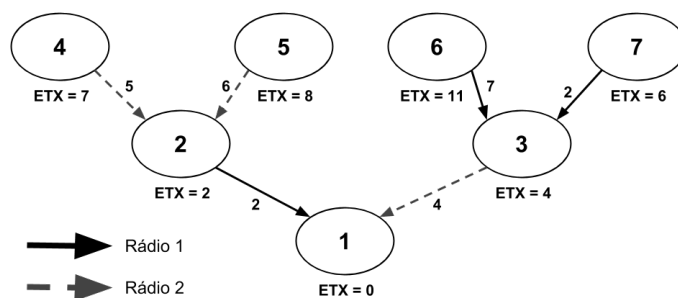
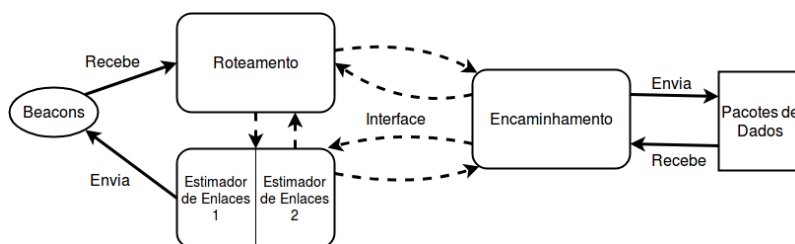


Figura 5. Exemplo da árvore formada pelo PAC-A, com os valores de ETX dos nós e dos enlaces para seus respectivos rádios.

Para implementar esse protocolo foram feitas alterações nos principais componentes do CTP, como ilustrado na figura 6. Usa-se dois estimadores de enlaces, um para cada rádio, que fornecem os valores de ETX de enlaces de um salto para o Roteamento. O Roteamento guarda uma variável que indica qual rádio é usado para enviar pacotes e cria uma tabela de roteamento para cada rádio, sendo que a tabela  $X$  contém os valores de ETX dos vizinhos para os quais se usa o rádio  $X$  para enviar a eles. O Encaminhamento é muito semelhante ao usado pelo CTP, porém ele se conecta aos dois Estimadores de Enlace e antes de enviar qualquer mensagem consulta o Roteamento para saber qual rádio usar. Dessa maneira, a árvore é construída para que um nó só receba pacotes de dados por um rádio e só envie pacotes de dados e *beacons* pelo outro, porém existe a chance de que ele receba pacotes de dados pelo mesmo rádio que envie caso a árvore não esteja perfeita. Um nó sempre recebe *beacons* pelos dois rádios.

Para entender como a alternância dos rádios é feita é preciso entender como o Roteamento se comporta ao receber um *beacon* e como a rota (árvore) é alterada em cada nó. Ao receber um *beacon* pelo rádio 2 contendo o ETX de um vizinho, sabe-se que esse ETX está associado ao rádio 2 do vizinho, já que um nó só envia *beacons* e pacotes de dados pelo mesmo rádio. Assim, para fazer a alternância dos rádios, esse valor de ETX e o identificador do vizinho são colocados na tabela de roteamento do rádio 1. Quando um nó atualiza a sua rota (pai e ETX) ele percorre as duas tabelas de roteamento e pega o melhor vizinho de cada uma, ao ETX do vizinho da tabela 1 soma-se o ETX do enlace entre o nó e o vizinho pelo rádio 1 e faz-se o mesmo para o outro vizinho obtido da tabela 2 porém usando o ETX pelo rádio 2. Em seguida escolhe-se o melhor dos vizinhos entre os dois, caso o vizinho vencedor seja melhor que o ETX atual subtraído de um certo valor, que indica o quão melhor o novo vizinho deve ser do pai atual, ele se torna o novo pai e o valor de ETX é atualizado. Caso o rádio usado para chegar ao novo pai

seja diferente do usado para chegar no antigo, então o rádio usado para envios é alterado e o cronômetro que controla o envio de *beacons* é reiniciado para seu valor mínimo, aumentando significativamente a taxa de envio de *beacons* para alertar os vizinhos sobre a mudança. Com a mudança do rádio é importante que toda a subárvore enraizada nesse nó também atualize seus enlaces, porém nem sempre é possível avisar os filhos de um nó que o rádio foi alterado fazendo com que esse nó receba e envie pelo mesmo rádio.



**Figura 6. Esquema que ilustra as associações entre os principais componentes do PAC-A.**

---

### Algoritmo 1 Atualiza Rota

---

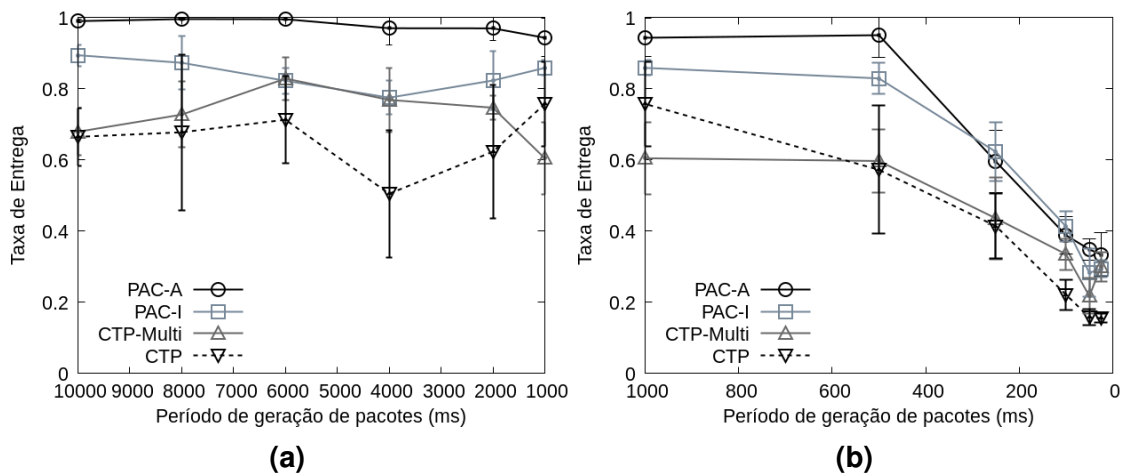
- 1: ▷ Procedimento que atualiza o pai e o ETX do nó  $u$
  - 2: Percorre a tabela de roteamento 1 e escolhe o vizinho  $v1$  com o menor ETX
  - 3:  $MenorETX1 = ETX \text{ de } v1 + ETX \text{ do enlace } u \rightarrow v1$
  - 4: Percorre a tabela de roteamento 2 e escolhe o vizinho  $v2$  com o menor ETX
  - 5:  $MenorETX2 = ETX \text{ de } v2 + ETX \text{ do enlace } u \rightarrow v2$
  - 6: **if** ( $MenorETX1 \geq MenorETX2$ ) **then**
  - 7:      $MenorETX = MenorETX2$
  - 8: **else**
  - 9:      $MenorETX = MenorETX1$
  - 10: **if** ( $MenorETX \text{ é } < (ETX_{Atual} - \text{Limite para mudar de Pai})$ ) **then**{
  - 11:      $Pai = \text{Melhor Vizinho}$
  - 12:      $ETX_{Atual} = MenorETX$
  - 13:      $Cronômetro = \text{Valor Mínimo}$  } ▷ Aumenta a frequência de envio de *beacons*
- 

## 6. Experimentos e Resultados

Para testar e comparar os protocolos CTP, CTP-Multi, PAC-I e PAC-A, foram feitos dois tipos de experimentos. Um mais longo com 45 minutos de duração com uma taxa de criação de pacotes média moderada e outros menores com 5 minutos de duração nos quais a taxa de criação de pacotes foi alterada de experimento em experimento. As métricas usadas foram: vazão, atraso na entrega dos pacotes, número de pacotes de dados enviados em toda a rede, número de *beacons* enviados e taxa de entrega.

Os experimentos foram feitos no *testbed Twonet* [Li et al. 2013] localizado em um prédio da *University of Houston*. Ele contém 100 nós sensores da plataforma *Opal* e está sujeito à interferência, especialmente com o *Wi-Fi*. O intervalo de 5 minutos foi escolhido por ser a menor duração possível no *testbed* e por permitir a formação das árvores de coletas e o envio de um número considerável de pacotes. Já a duração de 45 minutos foi escolhida para comparar os protocolos com uma taxa de geração de pacotes





**Figura 7. Gráficos que mostram a variação da taxa de entrega com a mudança no período de geração de pacotes.**

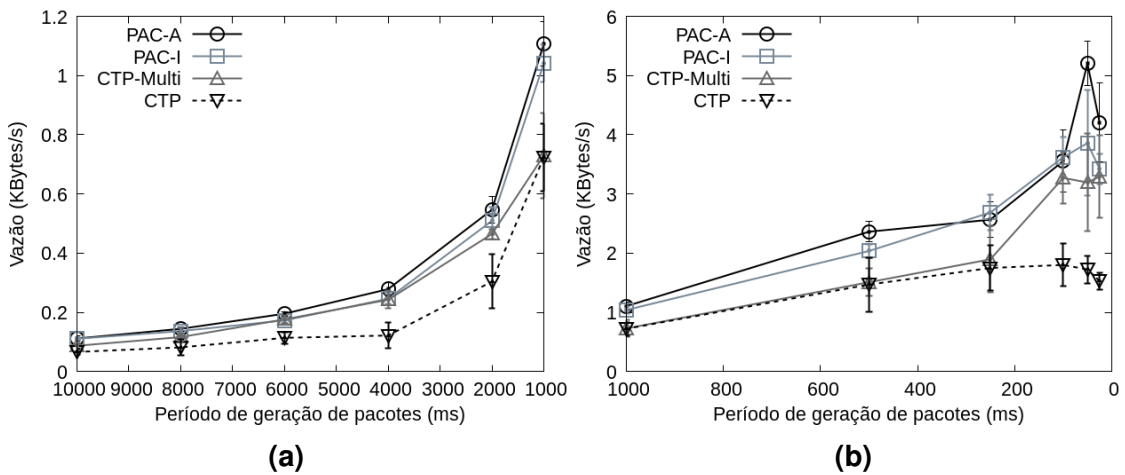
moderada e por uma duração maior, na qual a rede pode sofrer mudanças que não teriam ocorrido em um intervalo de tempo menor.

Os experimentos de 5 minutos geram pacotes de dados à uma taxa constante que é definida pelo período de geração de pacotes. Esse período foi variado de 10.000 milissegundos até 25 milissegundos, sendo que quanto menor o período maior a taxa de geração de pacotes. Todos os 100 nós do *testbed* foram utilizados, porém apenas metade foi encarregada de enviar pacotes. Os experimentos de curta duração foram repetidos 6 vezes e foi usado um intervalo de confiança de 95%. Os gráficos que expõem esses resultados foram divididos em dois, o primeiro mostra de 10.000 até 1.000 milissegundos e o segundo mostra de 1000 até 25 milissegundos, que é uma situação mais extrema e com intervalos menores entre cada medida, por isso foi necessário dividir os gráficos. O código fonte dos protocolos e dos programas usados para fazer os experimentos estão em um repositório público no *GitHub*<sup>1</sup>.

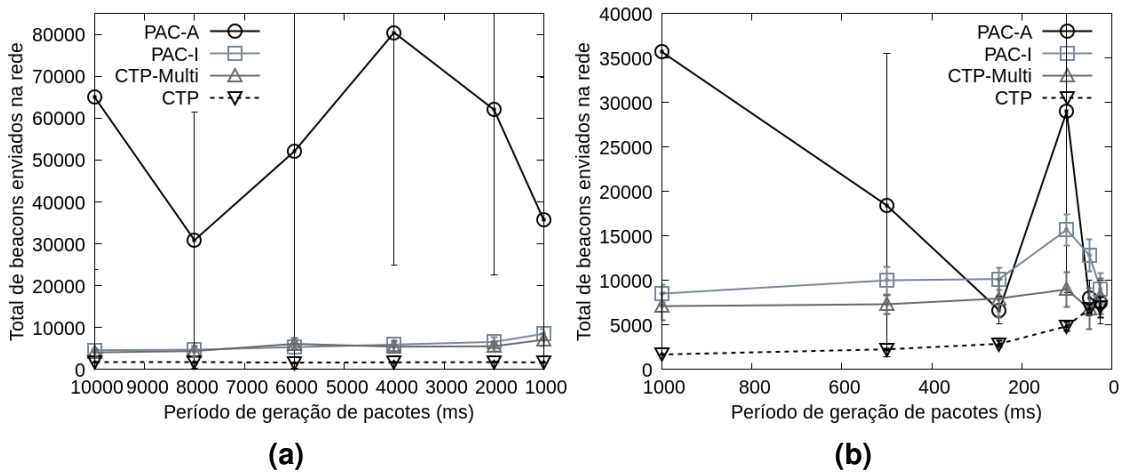
Todos os protocolos suportam múltiplas raízes, ou seja, podem ter mais de um nó de coleta. Usar apenas uma raiz dificulta a coleta de dados, faz com que as árvores geradas tenham mais níveis e, por isso, permite avaliar os protocolos em um cenário mais difícil em que as diferenças entre os protocolos serão mais visíveis. A taxa de entrega foi calculada dividindo o número de pacotes que chegaram na raiz pelo número de pacotes que foram gerados e enviados nos demais nós. Quando um nó gera um pacote ele tenta enviá-lo, caso não seja possível ele tenta novamente depois. Dessa maneira, o número total de mensagens geradas e enviadas pelos nós não é sempre o mesmo. Um problema encontrado foi a chegada de pacotes duplicados, para diminuir esse problema cada nó possui uma cache de pacotes recebidos.

Na figura 7 estão os dois gráficos que mostram a taxa de entrega de cada protocolo, a qual representa a confiabilidade de cada um deles. Percebe-se que o PAC-A é superior aos demais, conseguindo uma taxa de entrega de quase 100% até o período de 5.000, e mantendo mais de 90% até o período de 500 (dois pacotes gerados por segundo em cada nó). Ele é seguido pelo PAC-I, que manteve taxas entre 80% e 90% até o período

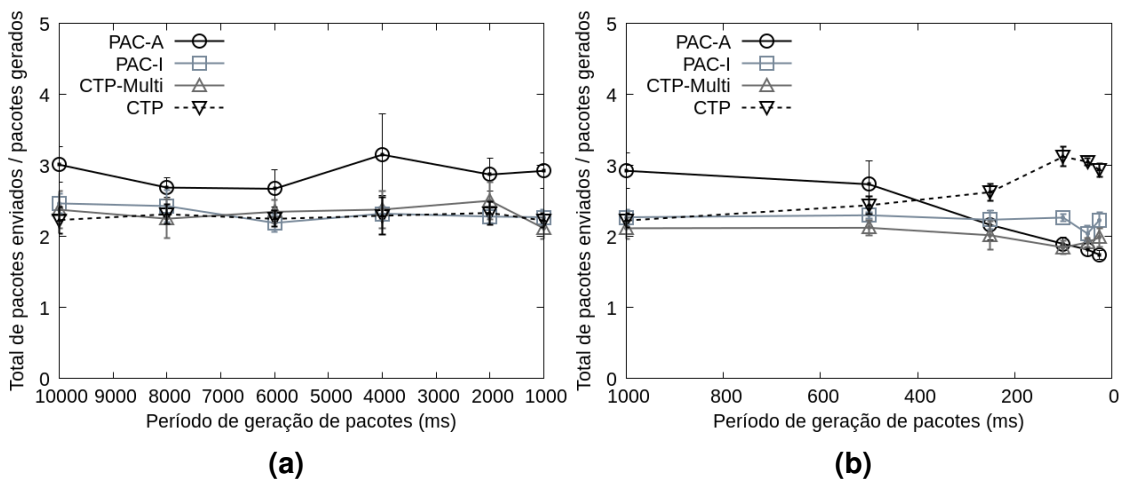
<sup>1</sup><https://github.com/gabrielsluz/CTP-ALL>



**Figura 8.** Gráficos que mostram a variação da vazão com a mudança no período de geração de pacotes.



**Figura 9.** Gráficos que mostram a variação no número total de *beacons* com a mudança no período de geração de pacotes.



**Figura 10.** Gráficos que mostram a variação na razão do número total de pacotes de dados enviados na rede sobre o número de pacotes recebidos na raiz com a mudança no período de geração de pacotes.

de 500. Os protocolos CTP e CTP-Multi já começaram com uma baixa taxa de entrega, porém percebe-se que o CTP-Multi é melhor. Isso pode ser explicado, pois a modificação que usa duas bandas de rádio tem mais possibilidades de caminhos para escolher e pode evitar uma banda com alta interferência em alguma parte do *testbed*. Após o período de 500, a taxa de entrega de todos os protocolos cai significativamente, porém os que eram melhores no início continuam superiores.

A figura 8 mostra a vazão obtida por cada protocolo. A vazão corresponde ao número de pacotes de dados que chegaram no nó raiz dividido pelo intervalo de tempo entre o primeiro pacote e o último a chegar. É importante ressaltar que os gráficos estão em uma escala diferente, a vazão obtida no segundo (período de 1.000 até 25) é muito maior do que a obtida no primeiro. No começo a vazão depende mais da taxa de geração de pacotes do que do protocolo, porém logo pode-se ver as diferenças entre cada um. A diferença entre o PAC-A e o PAC-I em relação à taxa de entrega é maior do que a diferença entre a vazão dos dois. Isso pode ser explicado pelo fato de que o PAC-I usa dois rádios para enviar pacotes, enquanto o outro usa apenas um, permitindo que se inicie uma transmissão por um rádio mesmo que o outro rádio ainda não tenha acabado de transmitir. O CTP-Multi teve melhor vazão do que o CTP. A vazão cresce até o período de 50 milissegundos, quando começa a cair em todos os protocolos.

O número total de *beacons* enviados é mostrado na figura 9. Essa métrica indica o custo para se manter as árvores. Nela percebe-se que o PAC-A é muito mais custoso do que os outros, isso se deve ao fato de que o período mínimo de envio *beacons* foi reduzido e que sempre que um nó troca de banda de rádio toda a sua sub-árvore deve ser atualizada para evitar gargalos em que um nó recebe e envia pelo mesmo rádio. Assim uma pequena mudança na árvore pode causar um efeito cascata que alterará grande parte das rotas, algo que é custoso em relação ao número de *beacons* necessário. Outro fator é o amplo intervalo de confiança que indica que o custo de controle do PAC-A varia muito, provavelmente devido à interferência no *testbed* que pode fazer com que uma banda de rádio seja afetada no meio do experimento. O protocolo com menor custo de controle é o CTP, seguido pelo CTP-Multi. Por usar dois componentes de Roteamento era esperado que o PAC-I tivesse um custo maior, porém a diferença com o CTP-Multi não foi muito grande. Para os períodos de geração de pacotes menores que 1.000 percebe-se que o PAC-A é o que teve uma maior queda no número de *beacons*, enquanto os outros continuaram com um número próximo ao do gráfico anterior.

A figura 10 está relacionada à eficiência dos protocolos, pois mostra o número de pacotes de dados enviados por todos os nós dividido pelo número de pacotes gerados, ou seja, retransmissões, encaminhamentos e mensagens duplicadas foram contabilizados obtendo quantas mensagens, em média, são necessárias para enviar um pacote de dados. Para um período de geração de pacotes maior (entre 10.000 e 1.000 ms), percebe-se que o PAC-A é menos eficiente em relação ao número de mensagens e que os outros são praticamente iguais. Entretanto, devido os resultados anteriores, sabe-se que o desempenho do PAC-A é melhor para vazão e taxa de entrega. Quando se aumenta a frequência de geração de pacotes, a eficiência do PAC-A melhora, inclusive em relação aos demais protocolos.

Os experimentos de 45 minutos foram feitos da mesma forma que os de 5 minutos, porém com período de geração de pacotes de 1.500 milissegundos e uma duração maior.

Os experimentos foram repetidos 28 vezes para cada protocolo e foi usado um intervalo de confiança de 95%. Na tabela 1 estão expostos os resultados para cada um dos protocolos. Em relação a taxa de entrega, os resultados foram semelhantes aos obtidos nos testes de 5 minutos para 1.000 ms, o PAC-A manteve sua superioridade e, no caso médio, o CTP-Multi foi pior do que o CTP, apesar de que, em relação à vazão, percebe-se que a modificação que pode usar dois rádios enviou mais pacotes do que o protocolo original. Era de se esperar que o CTP-Multi obtivesse uma melhor taxa de entrega, mas o maior volume de pacotes enviados compensou o ganho obtido com a diversidade de rádios. Em relação à vazão, a alternância de rádios e o uso dos dois rádios para enviar e receber, se mostraram superiores em relação à vazão. O número de *beacons* também corrobora com os resultados vistos nos experimentos de 5 minutos, sendo que o PAC-A apresenta grande variação. A métrica custo por envio indica o número total de pacotes necessários para enviar um pacote até a raiz, contabilizando encaminhamentos, retransmissões, duplicatas e *beacons*, ela está diretamente relacionada ao custo energético de cada protocolo. Percebe-se que o PAC-A possui o maior custo, devido ao maior número de *beacons* e ao resultado obtido na figura 10. O custo para o CTP-Multi é muito próximo ao do CTP, pois o número de retransmissões no protocolo original é maior do que no modificado.

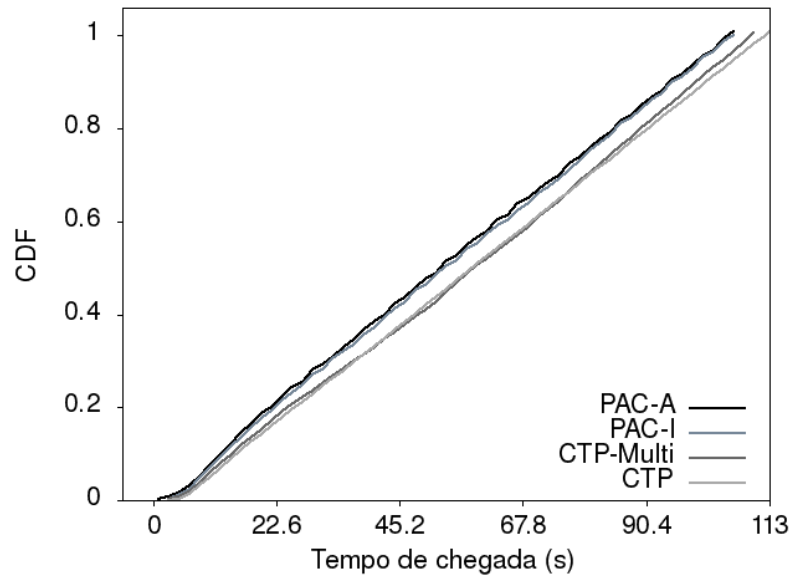
Protocolo	PAC-A	PAC-I	CTP-MULTI	CTP
<b>Métrica</b>				
Taxa de Entrega (%)	93,27 ± 3,57	82,89 ± 4,16	74,25 ± 4,32	74,53 ± 2,92
Vazão (KBytes/s)	0,673 ± 0,027	0,648 ± 0,037	0,565 ± 0,035	0,440 ± 0,020
Número de <i>Beacons</i>	63.051 ± 48.044	51.045 ± 5.153	36.293 ± 4.126	3.082 ± 241
Custo por envio	4,286 ± 0,954	3,386 ± 0,155	2,844 ± 0,177	2,753 ± 0,157

**Tabela 1. Resultados dos experimentos de 45 minutos. Intervalo de confiança de 95%**

Em um experimento de 45 minutos os tempos de chegada das 2000 primeiras mensagens na raiz foram medidos. Para expor esses resultados foi construída uma função de distribuição acumulada para cada protocolo, expostas na figura 11. Quanto mais vertical a linha é, mais pacotes foram recebidos em menos tempo, ou seja, o protocolo com a linha mais vertical corresponde ao com menor atraso. Assim verifica-se que o PAC-A e o PAC-I alcançaram quase a mesma latência e foram melhores que os outros dois. Já o CTP-Multi teve menor latência do que o CTP.

## 7. Conclusão e Trabalhos Futuros

Baseando-se nos resultados dos experimentos é possível concluir que os novos protocolos de coleta de dados, PAC-A e PAC-I, foram capazes de superar o CTP e o CTP-Multi, que são o atual estado da arte na coleta de dados, em relação à taxa de entrega, vazão e latência. O PAC-A obteve o melhor desempenho, sendo o protocolo com a melhor taxa de entrega entre os quatro e conseguindo, próximo ao PAC-I, a melhor vazão e latência. O PAC-I mostrou ser uma boa alternativa ao PAC-A, já que alcançou bom desempenho e não possui um custo muito elevado. O CTP-Multi consegue obter melhores rotas do que o CTP, porém seu desempenho foi superado pelos novos protocolos e, como possui um custo muito próximo ao do PAC-I, não apresentou vantagens em relação ao novo protocolo de árvores independentes nos experimentos. Portanto, os protocolos propostos



**Figura 11. Gráfico da função de distribuição acumulada dos tempos de chegada das primeiras 2.000 mensagens.**

neste trabalho conseguiram aproveitar melhor os benefícios de se usar dois rádios para a coleta de dados.

Como trabalho futuro pode-se investigar, tentar manter o desempenho e reduzir o número de *beacons*. Outra possibilidade pode ser a exploração do uso de múltiplos canais, algo que junto com o uso de duas bandas de rádios, pode ter grande impacto na redução dos efeitos da interferência aumentando a confiabilidade da coleta de dados.

### Agradecimentos

Os autores gostariam de agradecer as agências de fomento à pesquisa, CAPES, CNPq e FAPEMIG.

### Referências

- Adelani, Z., Mirjalily, G., and HajMirzaei, M. (2018). Balanced multi-channel data collection in wireless sensor networks. *IJCNIS*, 10.
- Anguswamy, R., Zawodniok, M., and Jagannathan, S. (2009). A multi-interface multi-channel routing (mmcr) protocol for wireless ad hoc networks. In *2009 IEEE Wireless Communications and Networking Conference*, pages 1–6.
- Ekbatanifard, G., Sommer, P., Kusy, B., Iyer, V., and Langendoen, K. (2013). Fast-forward: High-throughput dual-radio streaming. In *2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems*, pages 209–213.
- Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., and Levis, P. (2009). Collection Tree Protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*.
- Jin, X., Xu, H., Xia, C., Wang, J., and Zeng, P. (2018). Convergecast scheduling and cost optimization for industrial wireless sensor networks with multiple radio interfaces. *Wirel. Netw.*, 24(8):3205–3219.

- Jurdak, R., Klues, K., Kusy, B., Richter, C., Langendoen, K., and Brünig, M. (2011). Opal: A multiradio platform for high throughput wireless sensor networks. *Embedded Systems Letters, IEEE*, 3(4):121–124.
- Júnior, N. S. R., Vieira, L. F. M., and Vieira, M. A. M. (2017). Maximizando a vazão através de múltiplos caminhos em plataformas de dois rádios. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. SBRC.
- Kusy, B., Richter, C., Hu, W., Afanasyev, M., Jurdak, R., Brünig, M., Abbott, D., Huynh, C., and Ostry, D. (2011). Radio diversity for reliable communication in wsns. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference*, pages 270–281. IEEE.
- Levis, P., Patel, N., Culler, D., and Shenker, S. (2004). Trickle: A selfregulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proceedings of the first USENIX/ACM symposium on networked systems design and implementation (NSDI)*, pages 15–28.
- Li, Q., Han, D., Gnawali, O., Sommer, P., and Kusy, B. (2013). Twonet: Large-scale wireless sensor network testbed with dual-radio nodes. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 89. ACM.
- Peres, B., Santos, B. P., de O. Souza, O. A., Goussevskaja, O., Vieira, M. A. M., Vieira, L. F. M., and Loureiro, A. A. F. (2018). Matrix: Multihop address allocation and dynamic any-to-any routing for 6lowpan. *Computer Networks*, 140:28 – 40.
- Phokaew, A., Tanwongvarl, C., and Chantaraskul, S. (2014). Adaptive multi-channel ctp for wireless sensor networks. In *2014 International Electrical Engineering Congress (iEECON)*, pages 1–4.
- Ruiz, L. B., Correia, L. H. A., Vieira, L. F. M., Macedo, D. F., Nakamura, E. F., Figueiredo, C. M. S., Camara, D., Maia, E. H. B., Vieira, M. A. M., Nogueira, J. M. S., Loureiro, A. A. F., da Silva, D. C., and Fernandes, A. O. (2004). Arquiteturas para redes de sensores sem fio. In *Tutorial no Simpósio Brasileiro de Redes de Computadores (SBRC)*, pages 167–218.
- Santos, B. P., Vieira, M. A. M., and Vieira, L. F. M. (2015). extend collection tree protocol. In *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1512–1517.
- Vieira, M. A. M., Coelho Jr., C. N., da Silva Jr., D. C., and da Mata, J. M. (2003). Survey on wireless sensor network devices. In *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA'03. IEEE Conference*, volume 1, pages 537–544.
- Xu, X., Li, X. Y., Mao, X., Tang, S., and Wang, S. (2011). A delay-efficient algorithm for data aggregation in multihop wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(1):163–175.
- Yin, S., Gnawali, O., Sommer, P., and Kusy, B. (2014). Multi channel performance of dual band low power wireless network. In *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, pages 345–353.