Minimizando a Latência entre Dois Caminhos em Plataformas com Dois Rádios

Gabriel Santos Luz*, Nildo dos Santos Ribeiro Júnior*, Luiz F. M. Vieira*, Marcos A. M. Vieira*, and Omprakash Gnawali†,
*Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte MG, Brazil
†Computer Science Department, University of Houston, Houston, Texas, USA.

***Resumo*—Visando aumentar a vazão em redes de sensores surgiram plataformas em que os dispositivos possuem dois rádios e, também, protocolos de transferência de dados que priorizam a vazão e a eficiência energética, utilizando dois caminhos distintos simultaneamente. Neste trabalho apresentamos a formulação do problema para encontrar dois caminhos para plataformas com dois rádios e alcançar a vazão máxima da rede, enquanto também é minimizado o maior caminho, reduzindo também a latência. Mostramos que o problema é NP-Completo e uma solução baseada no modelo de programação linear inteira. Testamos a solução em mais de 5000 instâncias obtidas de um testbed real e os resultados mostram uma redução na latência sem comprometer o custo total.**

I. Introdução

REDES de Sensores sem Fio (RSSFs) são normalmente constituídas por um grande número de nós sensores distribuídos equipados com uma variedade de sensores. Além de possuir um ou mais sensores, cada nó possui um microprocessador e um ou mais transceptores de rádio, que permitem que os elementos de uma RSSF troquem dados diretamente entre si, agindo como coletores de dados e retransmissores de dados na rede [1]. Por isso, as RSSFs podem ser de fácil implementação e possuem grande capacidade de sensoreamento distribuído. Como consequência de tais atributos, as RSSFs podem ser uma poderosa ferramenta em diversas áreas de aplicação [2].

Como mostrado em [2], as RSSFs podem ser aplicadas em diversas áreas, que vão desde o controle de uma linha de montagem em uma indústria até aplicações médicas e biológicas no corpo humano. Esse tipo de rede também é comumente aplicado no meio ambiente (e.g. sensores para monitoramento da fauna e flora, condições atmosféricas, temperatura, pressão, etc.), segurança(e.g. monitoramento em centros comercias), tráfego e outras diversas aplicações em setores industriais. Portanto, pode se perceber que as RSSFs são uma importante tecnologia.

Apesar de possuir grandes vantagens esse tipo de rede possui algumas limitações que precisam ser levadas em consideração em sua implementação. Uma característica importante das RSSFs é a grande influência do cenário em sua implementação, sendo que o objetivo, assim como o ambiente em que será implementado, determinam as restrições impostas para uma RSSF [3]. Porém as maiores limitações das RSSFs são o custo e o consumo de energia. Como normalmente é utilizado um número muito grande de nós sensores, que pode chegar a dezenas de milhares de nós sensores, é essencial minimizar o custo de cada elemento da rede, como resultado as RSSFs geralmente possuem uma capacidade reduzida de processamento e de memória. O consumo de energia também deve ser minimizado, já que os nós sensores são normalmente alimentados por pilhas ou baterias e por isso, um consumo de energia mais baixo é importante para prolongar sua vida útil. Esse aspecto é ainda mais relevante em aplicações cujos nós sensores estão em lugares de difícil acesso, como em aplicações florestais [2]. No início do desenvolvimento das RSSFs, esses dois fatores guiaram as aplicações dessa tecnologia, nas quais apenas dados em pequena quantidade eram coletados. Entretanto, hoje em dia, algumas aplicações que usam imagens, vídeos e dados em maiores quantidades, estão sendo desenvolvidas e por isso um outro fator também deve ser levado em consideração: a vazão na rede.

Com o intuito de aumentar a vazão na rede e ainda manter a eficiência energética foram desenvolvidas RSSFs com mais de um rádio em cada nó sensor. Um exemplo é no sensor Opal Mote [4], mostrado na figura 1. Ele possui um processador SAM3U Cortex-M3 MCU da Atmel e dois rádios, um AT86FR212 que opera na banda de 900MHz e que possui 10 canais, e um rádio AT86RF231 que opera na banda de 2.4 GHz e que possui 16 canais. Como cada rádio operando em uma banda diferente, pode-se impedir a interferência de um rádio com o outro. O uso de múltiplos rádios permitiu 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^2 enquanto o CC2420 do TelosB consome 11.89 pJ/bit/m^2 [5]. Os ganhos no consumo de energia da plataforma Opal já foram demonstrados com profundidade em [4]]. Além disso, Yin et al. [6] 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.

Para essas plataformas RSSFs de dois rádios foram desenvolvidos técnicas e protocolos de roteamento para suportar a alta vazão, entre eles o FastFoward [7] e o "Dois Caminhos" [8]. O trabalho desenvolvido neste artigo segue o princípio do protocolo "Dois Caminhos" de achar dois caminhos disjuntos com mesma paridade, porém também foca em minimizar o atraso entre os dois caminhos minimizando o

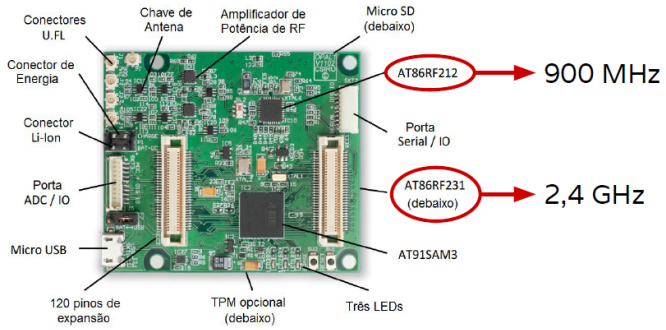

Figura 1. Opal Mote que opera com dois rádios heterogêneos em bandas diferentes.

maior caminho.

As principais contribuições desse artigo são: (i) a definição do problema de achar dois caminhos disjuntos de mesma paridade reduzindo a latência; (ii) a prova que o problema é NP-Completo; (iii) uma solução ótima baseada em programação linear inteira; (iv) resultados com dados de testbed reais; (v) comparação com o protocolo "Dois Caminhos" [8] e análise experimental para mais de 5000 instâncias com 100 nós.

O artigo está organizado da seguinte forma: na seção II apresentamos os trabalhos relacionados. Na seção III definimos o problema e provamos que ele é NP-Completo. A seção IV detalha a solução via o algoritmo de programação linear inteira. Na seção VI, os resultados dos testes em diferentes instâncias são apresentados. Finalmente, na seção VII temos a conclusão e os trabalhos futuros.

II. Trabalhos Relacionados

Sistemas de múltiplos caminhos foram estudados em redes *Mesh* [9], [10], [11], porém eles assumem que os rádios são homogêneos, e não heterogêneos como neste trabalho, então não precisam tratar da restrição de paridade. Também não utilizam duas faixas de canais diferentes.

No cenário de plataformas com mais de um rádio, o primeiro protocolo de transferência massiva de dados foi o FastFoward [7]. A figura 2 ilustra o funcionamento do FastFoward. Esse protocolo utiliza-se apenas um caminho para transmitir os pacotes de dados entre um nó fonte e um nó destino, sendo que os nós intermediários recebem os pacotes por um rádio e, simultaneamente, transmitem por outro rádio. Para diminuir o efeito da interferência entre as transmissões, o FastFoward além de utilizar dois rádios em bandas diferentes, também alterna canais em rádios de uma mesma banda. Os nós intermediários utilizam todo o recurso de rádio, porém os nós fonte e destino utilizam apenas metade do recurso disponível. Para utilizar todo o recurso de rádio disponível foi criado o protocolo "Dois Caminhos" [8].

O protocolo "Dois Caminhos" , assim como outros protocolos tradicionais de RSSFs, sempre que a rede precise realizar uma transferência massiva de dados o protocolo é chamado e assim os caminhos entre os nós são calculados e apenas os nós contidos nesses caminhos sairão do modo de economia de energia. A figura 3 ilustra o funcionamento do "Dois Caminhos" . O nó fonte apenas transmite pacotes de dados, o

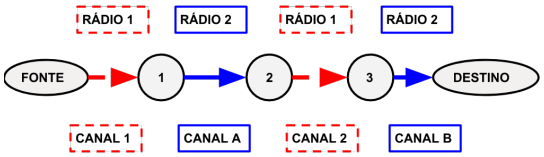

Figura 2. Esquema que exemplifica a alocação de rádios e canais em protocolos de dois rádios e só um caminho, como o FastFoward.

nó de destino apenas recebe pacotes e os nós intermediários recebem e transmitem pacotes de dados simultaneamente por rádios diferentes. Dessa maneira, ambos os rádios são usados em todos os nós. Entretanto, para que isso seja possível os caminhos escolhidos devem ser disjuntos e possuir a mesma paridade. Os caminhos precisam ser disjuntos pois são usados apenas dois rádios e por isso os caminhos não podem compartilhar nós, caso contrário os nós compartilhados precisariam receber e transmitir mais de um pacote de dados por vez, o que não é possível com dois rádios. Então criaria-se um gargalo que prejudicaria em muito o desempenho do protocolo. Os caminhos precisam ser da mesma paridade pois o nó fonte deve transmitir usando ambos os rádios e o nó destino deve receber usando ambos os rádios, algo que só é possível se os caminhos tiverem a mesma paridade. A figura 4 exemplifica o motivo dos caminhos precisarem da mesma paridade em seu comprimento.

Para criar um solução centralizada para o problema dos caminhos foi desenvolvido um modelo de programação linear inteira apresentado no artigo [8]. Esse modelo tem como objetivo achar dois caminhos disjuntos de mesma paridade, minimizando o custo total (soma dos custos dos dois caminhos).

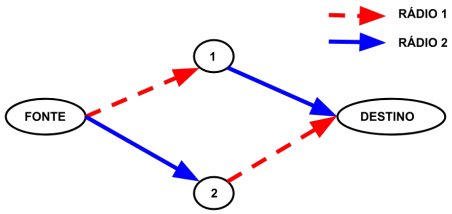

Figura 3. Esquema que exemplifica o alocamento de rádios em dois caminhos disjuntos de mesma paridade pelo "Dois Caminhos" .

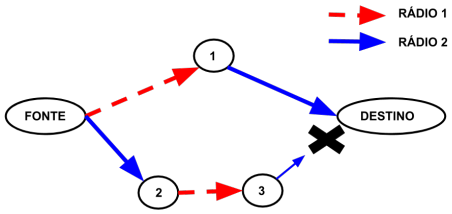

Figura 4. Esquema que mostra a escolha de dois caminhos disjuntos porém de paridades diferentes, com isso o nó destino recebe por apenas um rádio.

O protocolo "Dois Caminhos" foi implementado usando o TinyOS 2.1.2 para a plataforma Opal [4] e testado em um tesbed no mundo físico, no qual conseguiu atingir 96% do limite teórico para a vazão na rede, dobrando a vazão em

relação ao protocolo FastFoward. Neste artigo, é apresentado uma solução que mantem o ganho de vazão, dobrando em relação ao FastForward, e ainda minimiza a latência ao reduzir o maior caminho.

III. Definição do Problema

Dado um grafo $G = (V, E)$, onde V é um conjunto de vértices e E é um conjunto de arestas, o problema de encontrar dois caminhos disjuntos com a mesma paridade e menor latência é encontrar dois caminhos P_1 e P_2, iniciando no mesmo vértice fonte s e terminando no mesmo vértice destino t, com todos os nós intermediários disjuntos, com ambos os caminhos com a mesma paridade no número de saltos, i.e., $|P_1| \mod 2 = |P_2| \mod 2$, e minimizando o maior caminho. Chamamos esse problema de 2Min-Max-CD-Paridade. Esclarecemos que $\mod q$ é o operador módulo, que retorna o resto da divisão por q, ou seja, $\mod 2$ retorna se é par ou ímpar.

O problema, aqui chamado de 2Min-Max-CD, de encontrar dois caminhos disjuntos com o objetivo de minimizar o maior caminho em grafos com vértices disjuntos, foi provado ser NP-Completo [12]. A seguir provamos que o problema 2Min-Max-CD-Paridade é NP-Completo fazendo uma redução de 2Min-Max-CD.

Teorema 1. *Dado um grafo $G = (V, E)$ e s, $t \in V$, é NP-completo encontrar dois caminhos simples de s para t, com todos os vértices intermediários disjuntos, e tamanhos de mesma paridade, e o maior deles sendo minimizado.*

Demonstração. Primeiro provamos que 2Min-Max-CD-Paridade está em NP, visto que, em tempo polinomial, pode ser verificada a solução comparando a paridade do tamanho dos caminhos, e checando se os vértices intermediários são disjuntos e aparecem no máximo uma vez.

Suponha que temos um algoritmo que soluciona o problema 2Min-Max-CD-Paridade. Podemos, através de uma redução polinomial, usá-lo para resolver 2Min-Max-CD como a seguir. Criamos um novo grafo G' a partir do grafo original G adicionando 2 vértices v_1, v_2, e arestas de todos os vértices que se ligam em t a v_1 e v_2. Também é adicionado uma aresta de t a v_1 e a v_2. Os vértices v_1 e v_2 e as arestas ligadas a eles foram criados para que uma solução com paridade em G' não precise ter paridade em G. A redução é ilustrada na figura 5. Se o algoritmo solucionar 2Min-Max-CD-Paridade em G', então ele acha uma solução para 2Min-Max-CD em G. Da mesma forma, se for encontrado uma solução para 2Min-Max-CD em G, essa soluciona 2Min-Max-CD-Paridade em G' utilizando v_1 ou v_2 para obter a mesma paridade caso necessário, provando assim que o problema com paridade também é NP-Completo. □

IV. O Modelo de Programação Linear Inteira

Aqui apresentamos o modelo de programação linear inteira cujo objetivo é minimizar o maior caminho. Dessa maneira, limita-se o atraso envolvendo os dois caminhos disjuntos, ou seja, limita-se o tempo entre a transmissão e a recepção das

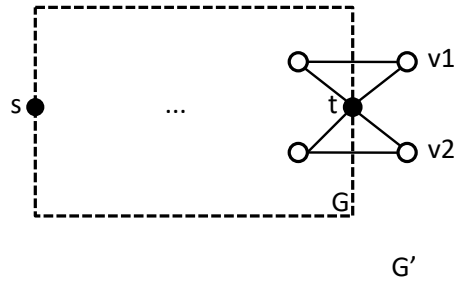

Figura 5. Redução entre o problema de achar dois caminhos disjuntos minimizando o maior caminho e o problema de achar dois caminhos disjuntos com mesma paridade minimizando o maior.

mensagens. Além disso, essa solução mantem a vazão, pode-se obter um menor atraso sem sacrificar a vazão total.

Para criar o modelo utilizamos a seguinte representação: um grafo direcionado G = (N,A), sendo N o conjunto de todos os nós e A o conjunto de todas as arestas. Para cada aresta (i,j) , de i para j, é associado um peso $c^1_{i,j}$ que representa o custo de transmitir um pacote de dados do nó i para o nó j através do rádio 1, e para o custo de transmitir pelo rádio 2 existe o peso $c^2_{i,j}$. Também é associado a cada aresta as variáveis binárias que seguem a seguinte representação : $x^{\text{Rádio,Caminho}}_{i,j}$. São elas: $x^{1,1}_{i,j}$, $x^{2,1}_{i,j}$, $x^{1,2}_{i,j}$, $x^{2,2}_{i,j}$, sendo que $x^{1,1}_{i,j}$ é igual a 1 se e somente se a aresta (i,j) faz parte do caminho 1 e será usado o rádio 1, já $x^{1,2}_{i,j}$ é igual a 1 se e somente se a aresta (i,j) faz parte do caminho 2 e o rádio 1 será usado. $S(i)$ representa o conjunto de nós que podem ser acessados diretamente (por uma única aresta) a partir do nó i, ou seja, se $(i, j) \in A$ então $j \in S(i)$. Já $E(i)$ representa o conjunto de nós a partir dos quais pode-se acessar diretamente o nó i, ou seja, $(j, i) \in A$ então $j \in E(i)$. Denota-se s como o nó fonte e d como o nó destino. Define-se I como o conjunto dos nós intermediários. Além disso, também são usadas as variáveis r e p, que indicam, respectivamente, o rádio e o caminho, portanto r e p só podem assumir os valores 1 e 2. A partir dessa notação pode-se entender o modelo que será apresentado a seguir.

O modelo de programação linear inteira foi implementado utilizando a linguagem GMPL (GNU Mathematical Programming Language) e resolvido utilizando o GLPK (GNU Linear Programming Kit), ferramenta de código aberto para resolver problemas de programação linear. O código fonte da solução está disponível em repositório público no GitHub[1].

O Modelo

minimizar

$$max(\sum_{(i,j)\in A} c^1_{i,j}x^{1,1}_{i,j} + c^2_{i,j}x^{2,1}_{i,j}, \sum_{(i,j)\in A} c^1_{i,j}x^{1,2}_{i,j} + c^2_{i,j}x^{2,2}_{i,j})$$

sujeito à

$$\text{(I)} \sum_{j\in E(i)} (x^{1,1}_{j,i} + x^{2,1}_{j,i} + x^{1,2}_{j,i} + x^{2,2}_{j,i}) = 0, \quad \text{se } i = s$$

$$\text{(II)} \sum_{j\in S(i)} x^{1,p}_{i,j} + x^{2,p}_{i,j} = 1, \quad \text{se } i = s, p \in \{1,2\}$$

[1]https://github.com/gabrielsluz/SolucaoMinimizaLatencia

(III) $\sum_{j \in S(i)} x_{i,j}^{r,1} + x_{i,j}^{r,2} = 1, \quad \text{se } i = s, r \in \{1,2\}$

(IV) $\sum_{j \in S(i)} x_{i,j}^{1,1} + x_{i,j}^{2,1} + x_{i,j}^{1,2} + x_{i,j}^{2,2} = 0, \quad \text{se } i = d$

(V) $\sum_{j \in S(i)} x_{i,j}^{1,p} - \sum_{j \in E(i)} x_{j,i}^{2,p} = 0, \quad \text{se } i \in I, p \in \{1,2\}$

(VI) $\sum_{j \in S(i)} x_{i,j}^{2,p} - \sum_{j \in E(i)} x_{j,i}^{1,p} = 0, \quad \text{se } i =\in I, p \in \{1,2\}$

(VII) $\sum_{j \in E(i)} x_{j,i}^{1,1} + \sum_{j \in E(i)} x_{j,i}^{1,2} + \sum_{j \in E(i)} x_{j,i}^{2,1} + \sum_{j \in E(i)} x_{j,i}^{2,2} \leq 1, \quad \text{se i } \neq d$

(VIII) $\sum_{i,j \in A} x_{j,i}^{1,1} + \sum_{i,j \in A} x_{j,i}^{1,2} - \sum_{i,j \in A} x_{j,i}^{2,1} - \sum_{i,j \in A} x_{j,i}^{2,2} = 0$

(IX) $x_{i,j}^{1,2}, x_{i,j}^{2,2}, x_{i,j}^{1,1}, x_{i,j}^{2,1} \in \{0,1\}$

V. Explicação do Modelo

A função objetivo minimiza o maior caminho. O caminho 1 é utilizado se as variáveis binárias $x_{i,j}^{1,1}$ e $x_{i,j}^{2,1}$ são 1 e o caminho 2 é utilizado se as variáveis binárias $x_{i,j}^{1,2}$ e $x_{i,j}^{2,2}$ são 1.

As restrições (I), (II) e (III) definem o começo dos caminhos, sendo que o nó fonte não pode receber de nenhum outro e os caminhos começam com rádios diferentes.

A restrição (IV) garante que o nó destino não transmite para nenhum outro nó. Enquanto a restrição (VIII) garante que os caminhos terão a mesma paridade, assim os caminhos terminam com rádios diferentes, evitando um gargalo no nó destino.

As restrições (V) e (VI) coordenam o fluxo dos caminhos, ou seja, os caminhos são iniciados no nó fonte e só podem terminar no nó destino. Além disso, essas restrições garantem que os rádios são alternados a cada salto em um caminho.

A restrição (VII) faz com que cada nó diferente do nó destino receba dados por no máximo 1 rádio.

Já a restrição (IX) reafirma o caráter binário das variáveis associadas às arestas.

VI. Experimentos e Resultados

Os experimentos foram feitos utilizando instâncias criadas a partir de um testbed de redes de sensores sem fio chamado Twonet [13] que contém 100 nós sensores com dois rádios da plataforma Opal. A figura 6 mostra um diagrama do testbed, cada ponto na figura representa um nó sensor, um ponto foi ampliado na figura para mostrar um nó com suas duas antenas. Os nós ficam presos no teto e o testbed ocupa quatro andares de uma universidade.

O custo de cada enlace e a topologia da rede foram coletados no testbed da mesma forma que em [8], as instâncias foram obtidas a partir topologia coletada do testbed Twonet na qual variou-se os nós fonte e os nós destino. Os nós sensores enviaram pacotes periodicamente para seus vizinhos. Os nós montam uma tabela com seus vizinhos, e esta é usada para montar uma tabela que representa a topologia da rede. A métrica para o custo do enlace é a razão entre o número de pacotes enviados pelo número de pacotes recebidos naquele enlace, ou seja, o inverso da taxa de entrega de pacotes. Quanto

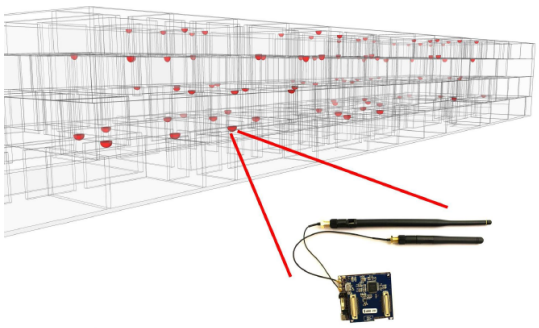

Figura 6. Testbed TwoNet com 100 Opal motes.

maior a taxa de pacotes recebidos, menor o custo, e quanto menor a taxa de pacotes recebidos, maior o custo.

O custo do caminho é definido como a soma do custo dos enlaces utilizados pelo caminho. O custo da latência da solução é definido como o maior custo entre os dois caminhos da solução e é a métrica utilizada na solução desse artigo. Para efeito de comparação, também usamos o custo total da solução que é definido como a soma do custo dos dois caminhos da solução, porém note que esse custo total não necessariamente minimiza a latência.

As instâncias foram geradas alterando os nós fonte e destino, de modo que a topologia do testbed é sempre mantida e o que muda para cada instância são os nós fonte e destino, únicos para cada uma. Dessa maneira, pode-se obter um grande número de diferentes instâncias.

Os experimentos foram feitos em 5.700 instâncias. Foi usado o modelo de programação linear inteira descrito na seção anterior desse artigo. Também foi usado o modelo do "Dois Caminhos" como comparação para o novo modelo, visto que ele também encontra dois caminhos disjuntos e com mesma paridade. Os modelos foram implementados utilizando a linguagem GMPL (GNU Mathematical Programming Language) e resolvidos utilizando o GLPK (GNU Linear Programming Kit), ferramenta de código aberto para resolver problemas de programação linear. Os experimentos foram realizados em uma máquina com a seguinte configuração: sistema operacional Ubuntu, arquitetura x86 64, processador Intel(R) Core(TM) i7-4790 CPU com clock de 3.60GHz, memória de 8GB. O modelo proposto neste artigo será denominado "Minimiza Latência" para facilitar a exposição dos resultados.

Os dados coletados foram: custo total (soma dos custos dos dois caminhos), custos dos caminhos 1 e 2, diferença entre o custo dos caminhos, e o tempo médio de execução do modelo. Além disso, para um experimento, os modelos foram modificados para minimizar o número de saltos nos caminhos. Para expor esses dados foram usadas funções de distribuição acumulada, que possibilitam a análise de grandes quantidades de resultados.

Para todas as instâncias, a figura 7 mostra a frequência do custo de cada caminho para a solução que minimiza a latência. Já a figura 8 mostra a frequência do custo de cada caminho para a solução Dois Caminhos. Sendo que para ambas o caminho 2 é o maior caminho. Elas permitem uma comparação entre os dois modelos em relação ao custo de cada caminho

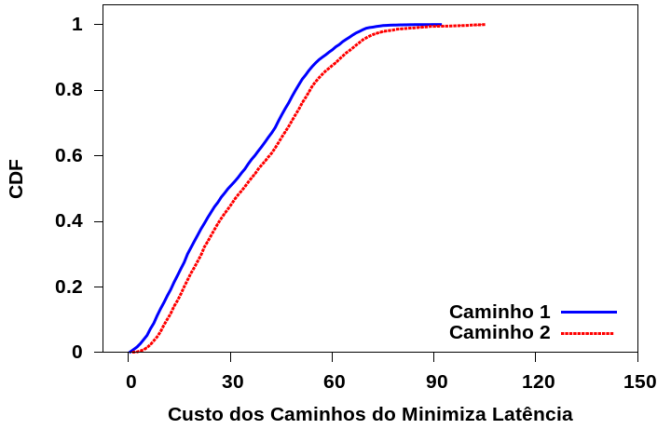

Figura 7. Custo de cada caminho para o Minimiza Latência.

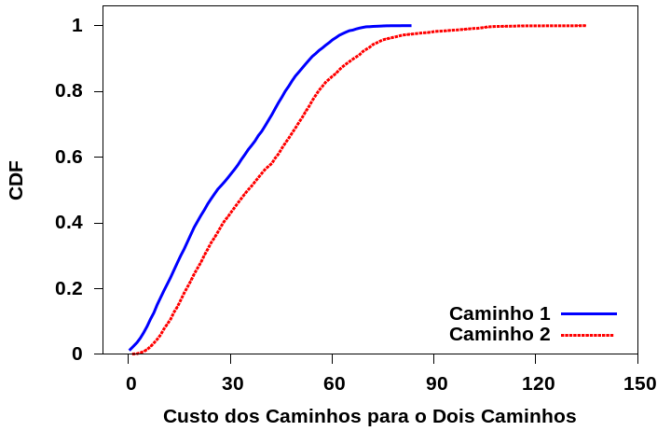

Figura 8. Custo de cada caminho para o Dois Caminhos.

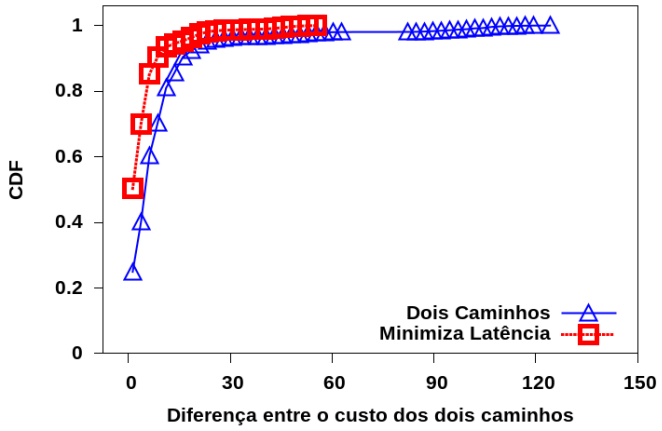

Figura 9. Diferença entre os custos dos caminhos.

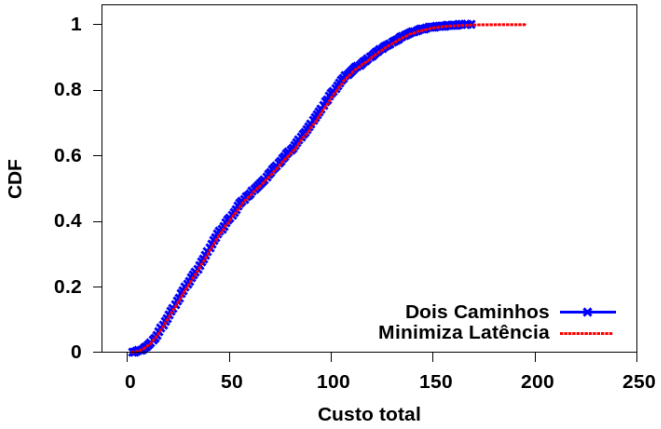

Figura 10. Custo Total.

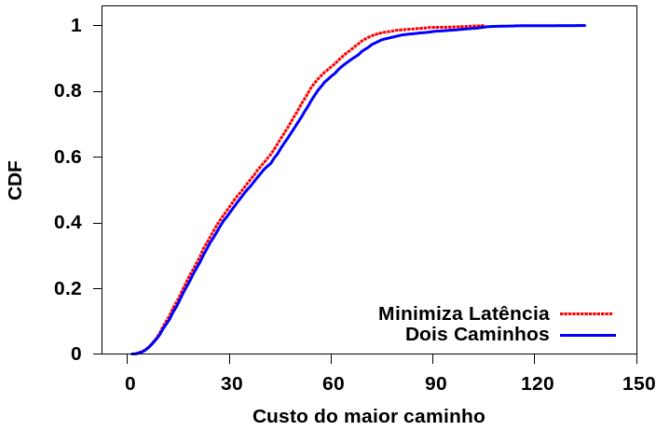

Figura 11. Custo do maior caminho.

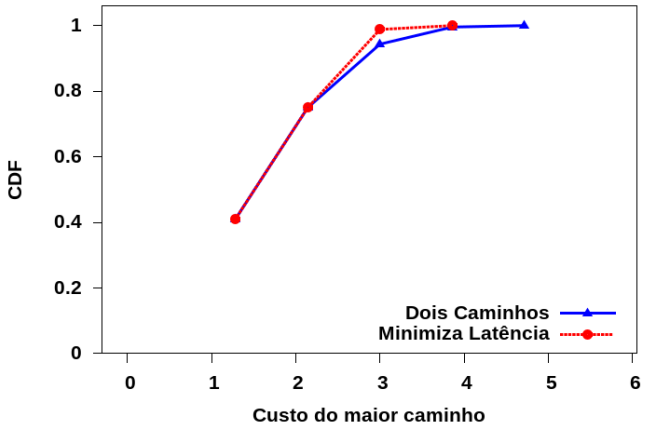

Figura 12. Custo do maior caminho considerando apenas o número de saltos.

separadamente. Percebe-se que as curvas do modelo deste artigo estão mais próximas do que as curvas do outro, isso demonstra que esse modelo consegue reduzir o atraso entre os caminhos. Além disso, no Dois Caminhos o custo máximo do maior caminho foi igual a 135, enquanto no Minimiza Latência foi igual a 105, demonstrando que o modelo deste artigo consegui minimizar o maior caminho e chegar a resultados melhores neste quesito.

A figura 9 apresenta a diferença entre os custos dos dois caminhos em cada instância. Com isso, pode-se verificar de uma forma direta a diferença entre a latência no uso dos dois modelos. Pela figura, conclui-se que o modelo deste artigo é melhor para minimizar a latência, já que a curva de sua função está muito acima da outra curva na parte inicial (diferença de custos de 0 até 20), indicando que o modelo que minimiza a latência possui uma maior concentração de resultados entre 0 e 20, do que o modelo do Dois Caminhos, sendo que grande parte está no zero, ou seja, a menor diferença possível. Além disso, o Dois Caminhos apresenta muitos resultados com diferença maior que 90, chagando até mais que 120, ou seja, possui casos extremos em que a diferença entre os custos dos caminhos é muito grande. Portanto, o modelo deste artigo, possui melhor desempenho em minimizar a latência do que o "Dois Caminhos" .

A figura 10 permite observar o comportamento de ambos os modelos em relação ao custo total (soma do comprimento dos dois caminhos). Nela percebe-se que ambos os modelos alcançam resultados muito próximos, já que as curvas quase se sobrepõem. Verifica-se que o " Dois Caminhos " consegue resultados um pouco melhores do que o " Minimiza Latência". Portanto, o modelo deste artigo foi capaz de se aproximar dos resultados do modelo "Dois Caminhos" que minimiza o custo total e acha a solução ótima, ou seja, o "Minimiza Latência" se aproxima muito da solução ótima para o custo total dos caminhos.

A figura 11 possibilita observar o comportamento de ambos os modelos em relação à latência. Nela percebe-se que o modelo aqui desenvolvido consegue melhores resultados na minimização da latência em relação ao "Dois Caminhos", apesar das curvas estarem próximas a diferença ainda é visível.

A figura 12 demonstra o custo do maior caminho quando todos os saltos possuem custo igual a 1. Esse cenário é o mesmo de quando os modelos não levam em consideração o custo dos saltos, apenas o número de saltos. Nela percebe-se que cerca de 85% dos resultados encontrados são iguais, porém o modelo que minimiza a latência é melhor, já que possui uma melhor distribuição dos resultados e possui o maior caminho no máximo com custo 4, enquanto o "Dois Caminhos" chega a custo 5.

A figura 13 ilustra uma parte da topologia da rede e a solução para os dois caminhos encontrada pelo modelo nessa topologia. O nó 1 é o nó fonte e o nó 9 é o nó destino. Também são mostrados os custos das arestas que compõem os caminhos da solução ao lado da aresta a que se referem. O custo dos caminhos da solução foi 24 e 21.

A figura 14 mostra uma visão geral das soluções considerando as médias nas métricas de vazão e latência. Para as aplicações, é melhor ter maior vazão e, ao mesmo tempo, é

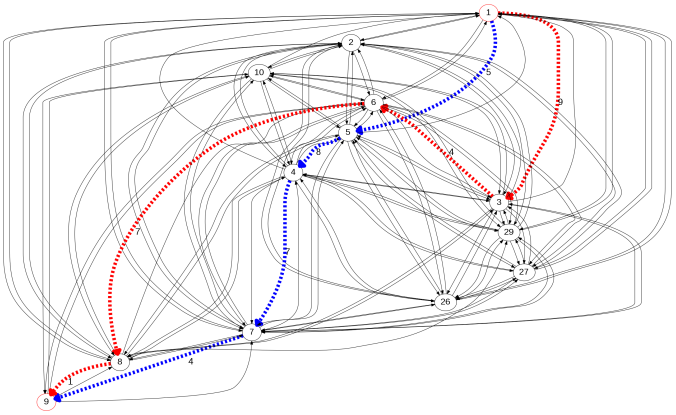

Figura 13. Ilustração da solução encontrada que minimiza a latência. Nós pelo caminho vermelho: 1,3,6,8,9. Nós pelo caminho azul: 1,5,4,7,9.

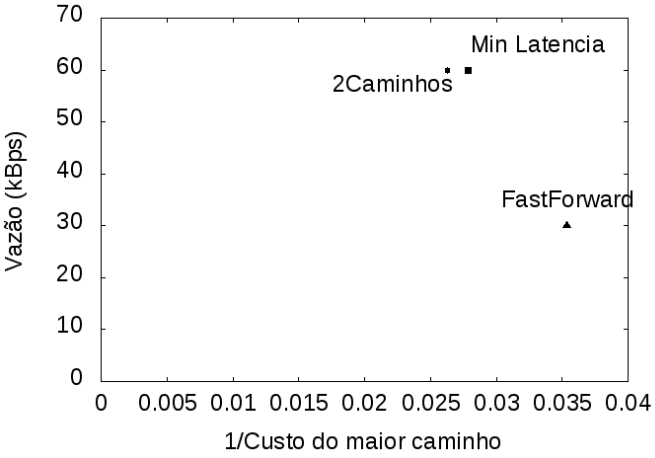

Figura 14. Visão geral das soluções

melhor ter menor latência, representado como o custo do maior caminho. O FastForward apresenta o menor maior caminho porque utiliza apenas um caminho, o menor caminho, porém tem a menor vazão. A vazão é dobrada com o uso de dois caminhos (de 30kBps para 60kBps), como ocorre no algoritmo DoisCaminhos. Conforme mostra a figura 14, a solução aqui proposta (Min Latência) atinge seu objetivo, mantendo a vazão dobrada e reduzindo o custo do maior caminho.

Modelo	Dois Caminhos	Minimiza Latência
Média	0,1s	6,7s
Desvio Padrão	0,3	11,6

Tabela I
TEMPO MÉDIO DE EXECUÇÃO DE CADA MODELO PARA CADA INSTÂNCIA.

A tabela I traz o tempo médio de execução e o desvio padrão de cada modelo para cada instância. Nela percebe-se a diferença entre o tempo de execução dos modelos. Com um tempo médio de 6,7 s o modelo que minimiza a latência demora, em média, mais do que o tempo do modelo do "Dois Caminhos", que é de 0,1 s. Outro fator é o desvio padrão no tempo de execução para ambos os modelos, isso mostra que para determinadas instâncias o tempo de execução dos modelos é muito maior do que o tempo médio, ou seja, a

instância influencia diretamente no tempo de execução dos modelos. O modelo que minimiza o maior caminho possui mais restrições do que o modelo que minimiza a soma dos caminhos, resultando em uma entrada maior para o problema de programação linear inteira e consequentemente um aumento no volume de processamento.

Por ser uma solução exata de um problema NP-Completo, em uma rede real de sensores sem fio com muitos nós pode ser mais indicado usar soluções aproximadas, já que o tempo de execução que depende do número de vértices pode se tornar extremamente alto. Em testes com grafos gerados aleatoriamente, o modelo foi testado em topologias de 10 até 1500 vértices, chegando a 75 mil arestas. Nas instâncias de 1000 a 1500 vértices o tempo de execução pode variar muito dependendo dos nós fonte e destino, sendo que nos piores casos testados o algoritmo executou por muitas horas. Dessa maneira, a solução possui limitações em relação ao custo de execução, sendo que para casos com mais de 1000 nós o tempo de execução pode ser inviável.

Nota-se que o tempo de execução de 6,7s se passou no sorvedouro que calcula as rotas, o que não acarreta custo energético à rede de sensores. Além disso, como já mencionado na Introdução, os ganhos em relação ao consumo energético da plataforma Opal foram discorridos com profundidade em [4]. Outro fator quanto ao tempo de execução é que, apesar de provarmos que o problema é NP-Completo, o modelo foi capaz de resolver instâncias de um testbed real com 100 nós.

VII. Conclusão

Nesse artigo foi introduzida a definição do problema de achar dois caminhos disjuntos de mesma paridade reduzindo a latência. Foi provado que esse problema é NP-Completo. Foi apresentado uma solução ótima baseada em programação linear inteira que permite manter os ganhos com uso de dois rádios, dobrando a vazão, e, ao mesmo tempo, reduzindo a latência.

Os experimentos utilizaram dados de testbed reais em 5.700 instâncias. Os resultados permitiram concluir que o modelo deste artigo avança o estado da arte na redução da latência, sendo que o faz sem sacrificar totalmente o custo total.

Para trabalhos futuros, pode-se desenvolver um protocolo de roteamento para dois caminhos de mesma paridade minimizando latência que possa ser calculado de dentro da própria rede. Também pretende-se testar esse protocolo na plataforma Opal no testbed Twonet.

Agradecimentos

Os autores gostariam de agradecer às agências de fomento à pesquisa CNPq, CAPES e FAPEMIG pelo apoio financeiro.

Referências

[1] M. A. M. Vieira, C. N. Coelho Jr., D. C. da Silva Jr., and J. M. da Mata, "Survey on wireless sensor network devices," in *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA'03. IEEE Conference*, vol. 1, 2003, pp. 537–544.

[2] L. B. Ruiz, L. H. A. Correia, L. F. M. Vieira, D. F. Macedo, E. F. Nakamura, C. M. S. Figueiredo, D. Camara, E. H. B. Maia, M. A. M. Vieira, J. M. S. Nogueira, A. A. F. Loureiro, D. C. da Silva, and A. O. Fernandes, "Arquiteturas para redes de sensores sem fio," in *Tutorial no Simpósio Brasileiro de Redes de Computadores (SBRC)*, 2004, pp. 167–218.

[3] J. Yick, B. Mukherjee, and D. Ghosal, *Wireless sensor network survey.*, 52nd ed. Computer networks, 2008.

[4] R. Jurdak, K. Klues, B. Kusy, C. Richter, K. Langendoen, and M. Brünig, "Opal: A multiradio platform for high throughput wireless sensor networks," *Embedded Systems Letters, IEEE*, vol. 3, no. 4, pp. 121–124, 2011.

[5] B. Kusy, C. Richter, W. Hu, M. Afanasyev, R. Jurdak, M. Brünig, D. Abbott, C. Huynh, and D. Ostry, "Radio diversity for reliable communication in wsns." in *Information Processing in Sensor Networks (IPSN),2011 10th International Conference*. IEEE, 2011, pp. 270–281.

[6] S. Yin, O. Gnawali, P. Sommer, and B. Kusy, "Multi channel performance of dual band low power wireless network," in *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, Oct 2014, pp. 345–353.

[7] G. Ekbatanifard, P. Sommer, B. Kusy, V. Iyer, and K. Langendoen, "Fastforward: High-throughput dual-radio streaming," in *IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems*. IEEE, 2013, pp. 209–213.

[8] N. S. R. Júnior, L. F. M. Vieira, and M. A. M. Vieira, "Maximizando a vazão através de múltiplos caminhos em plataformas de dois rádios," in *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. SBRC, 2017.

[9] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2004, pp. 114–128.

[10] M. K. Marina, S. R. Das, and A. P. Subramanian, "A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks," *Computer Networks*, vol. 54, no. 2, pp. 241 – 256, 2010.

[11] A. Raniwala and T. cker Chiueh, "Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3, March 2005, pp. 2223–2234 vol. 3.

[12] C.-L. Li, S. McCormick, and D. Simchi-Levi, "The complexity of finding two disjoint paths with min-max objective function," *Discrete Applied Mathematics*, vol. 26, no. 1, pp. 105 – 115, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0166218X90900247

[13] Q. Li, D. Han, O. Gnawali, P. Sommer, and B. Kusy, "Twonet: Large-scale wireless sensor network testbed with dual-radio nodes," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2013, p. 89.