Wireless Networks (2024) 30:1591–1600
https://doi.org/10.1007/s11276-023-03594-0

ORIGINAL PAPER

Latency minimizing in two paths dual radio networks

Gabriel Santos Luz[1] · **Nildo dos Santos Ribeiro Junior**[1] · **Luiz F. M. Vieira**[1] · **Marcos A. M. Vieira**[1] · **Omprakash Gnawali**[2]

Accepted: 14 November 2023

Abstract

Aiming to increase throughput in Wireless Networks, such as in Wireless Sensor Network and the Internet of Things, platforms emerged in which devices have two radios, and also data transfer protocols that prioritize maximum throughput and energy efficiency, using two different paths simultaneously. The usage of dual radios allowed simultaneous transmissions between wireless devices, which, besides increasing network throughput, can also improve network stability, delivery rate, transmission cost, and energy consumption per transmitted byte. However, one path may be much longer than the other, causing high latency. First, in this work, we present the problem formulation to find two disjoint paths with the same parity size for platforms with two heterogeneous radios to reach the network maximum flow, while also minimizing the longest path, which reduces latency. Second, we show that the problem is NP-Complete. Next, we present a solution based on integer linear programming. Moreover, we tested the solution on almost 5,700 instances obtained from an actual testbed and the results show a reduction in latency while maintaining the high throughput.

Keywords Latency · Dual-path routing · Wireless sensor networks · Dual-radio

1 Introduction

Wireless Networks, such as Wireless Sensor Networks (WSNs) [1] and the Internet of Things (IoT) [2], are composed of embedded devices with network capabilities [3]. Wireless Sensor Networks are usually composed of a large number of distributed sensor nodes equipped with a variety of sensors. Besides having one or more sensors, each node has a microprocessor and one or more radio transceivers, which allow elements of a WSN to exchange data directly with each other, acting as data collectors and data transmitters in the network. Therefore, WSNs can be easily implemented and possess great potential in distributed sensing.

WSNs can be applied in different areas [4], ranging from the control of an industry assembly line to medical and biological applications in the human body. This type of network is also often applied in a natural environment (e.g. sensors for monitoring fauna and flora, atmospheric conditions, temperature, atmospheric pressure, etc.), security (e.g. monitoring commercial centers), traffic, and many other applications in industrial sectors. Therefore, we can conclude that WSNs are an important technology.

Even though this type of network has many advantages, it also has some limitations that must be considered. An important characteristic of WSNs is the great influence of the environment and of the main goal, which determines the restrictions imposed on a WSN [5]. However, the biggest limitations of this type of network are cost and energy consumption. As it is usually composed of a large number of sensor nodes, which can reach up to tens of

✉ Luiz F. M. Vieira
lfvieira@dcc.ufmg.br

Gabriel Santos Luz
gabrielluz@dcc.ufmg.br

Nildo dos Santos Ribeiro Junior
nildo@dcc.ufmg.br

Marcos A. M. Vieira
mmvieira@dcc.ufmg.br

Omprakash Gnawali
gnawali@cs.uh.edu

[1] Computer Science Department, Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais 31270-90, Brazil

[2] Department of Computer Science, University of Houston, Houston, TX 77204-3010, USA

thousands of sensor nodes, it is essential to minimize the cost of each element in the network. As a result, WSNs usually have a reduced processing and memory capacity. Energy consumption must be minimized, as well, since sensor nodes are usually supplied by batteries, which is why lower power consumption is important to extend a sensor node's service life. This aspect is even more relevant in applications whose sensor nodes are not easily accessed, such as forest applications [4]. At the beginning of the development of WSNs, these two factors guided applications of this technology, in which only small amounts of data were collected. However, nowadays, some applications that use images, videos, and larger amounts of data, are being developed and another factor must be taken into consideration: network throughput.

In the recent past, different wireless networks, such as the Internet of Things (IoT) and the Internet of Mobile Things (IoMT) have shown a demand for more throughput [2]. Underwater optical-acoustic sensor networks could also benefit from dual radio networks [6]. Aiming to increase throughput in the network, while still maintaining power efficiency, there were developed embedded devices with more than one radio, also known as multi-radio platforms for WSN [7–10], and for the Internet of Things (IoT), including: the IoT DevKit-LoRaWAN [11], Multi-Transceiver consisting of LoRa and ESP8266-Wifi Communication Module [12], Wi-Fi and LoRa radios [13]; Pycom's FiPy [14], which is a device with multiples radios for LoRa, Sigfox, WiFi, and Bluetooth; Dual-radio motes, such as Waspmote [15], OpenMote B [16], and Firefly [17], which has 2.4 GHz short-range and 920 MHz long-range radios; Narrowband Internet of Things (NB-IoT) (an emerging cellular technology) and 802.15.4 for sink nodes [18].

For this work, we adopted the Opal mote [19], shown in Fig. 1. It possesses a processor SAM3U Cortex-M3 MCU of Atmel and two radios, an AT86FR212 that operates in the 900MHz band and has 10 channels, and an AT86RF231 radio that operates in the 2.4GHz and has 16 channels. As each radio operates in a different band, it is possible to impede interference between the two radios. The usage of multiple radios allowed simultaneous transmissions between sensor nodes, which, besides increasing network throughput, can also improve network stability, delivery rate, transmission cost, and energy consumption per transmitted byte. To illustrate, the radios of Opal mote consume 0.669 and 0.659 pJ/bit/m^2 whereas the CC2420 of TelosB consumes 11.89 pJ/bit/m^2 [20]. The gains in energy consumption of the Opal platform have been demonstrated with depth in [21]. Besides that, Yin et al. [22] showed that the ISM band of 900MHz presents better connectivity than the ISM band of 2.4GHz. Therefore, using two radios also brings benefits in link quality and connectivity.

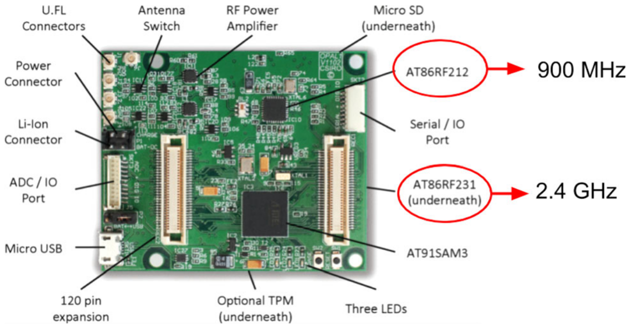

Fig. 1 Opal Mote operates with two heterogeneous radios in different bands. Source: [21]

Recently, techniques and routing protocols to support higher throughput were developed for dual radio WSN platforms, among them FastFoward [23] and Two Paths [24]. FastForward uses both radios and one path. TwoPaths improves FastForward throughput by using two disjoint paths with the same parity size. However, one path may be too long, which will cause high latency. The work developed in this paper follows the principle of Two Paths, which finds two disjoint paths with the same parity, but we also minimize latency by reducing the longest path.

The main contributions of this paper are: (i) we formalize the problem of finding two disjoint paths with the same parity size and minimizing the longest path and we prove that the problem is NP-Complete; (ii) we provide an integer linear programming model that solves the problem; (iii) we experimented with almost 5,700 instances from a real-world testbed; (iv) our results demonstrate that the latency is reduced while maintaining a high throughput.

This paper is organized as follows. Section 2 presents the Related Work. Section 3 formalizes the problem definition and its complexity. Section 4 details the Integer Linear Programming Model. In Sect. 5, we explain the model. Section 6 brings the experiments and results. Finally, Sect. 7 concludes the paper.

2 Related work

Multiple path systems have been studied in *Mesh* networks [25–27], but they assume homogeneous radios, and not heterogeneous as in this work, as consequence, they do not need to treat the parity restriction. They also do not use two different channel ranges.

In the context of platforms with more than one radio, the first massive data transfer protocol was FastFoward [23]. Figure 2 illustrates how FastFoward works. This protocol utilizes only one path to transmit data packets from the source node to the destination node, in a way that relay

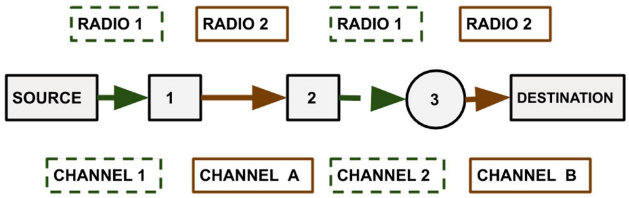

Fig. 2 An example of the radio and channel allocation in dual-radio networks with only one path protocol, such as FastFoward

nodes receive packets through one radio and, simultaneously, transmit through the other radio. To reduce the effect of interference between transmissions, FastFoward not only uses two radios in different bands but also alternates radio channels of the same band. As a consequence, relay nodes use all of the radio resources, but the source and destination nodes use only half of the available resources. To use all the available radio resources Ribeiro et al. created the TwoPaths [24] protocol.

In the TwoPaths protocol, as in other traditional WSNs protocols, every time the network needs to carry out a massive data transfer, the protocol is called, then the paths are computed and only the nodes contained in these paths will come out of the battery saving mode. Figure 3 illustrates how TwoPaths works. The source node only transmits data packets, the destination node only receives packets, and relay nodes receive and transmit data packets simultaneously through different radios. Consequently, both radios are used by all nodes. However, this is only possible when the two chosen paths are disjoint and have the same parity, otherwise, bottlenecks would appear in the network, which would compromise the protocol performance. Figure 4 illustrates a situation in which the chosen paths have the same parity and the destination node receives through only one radio, compromising the protocol's efficiency.

Recently, a centralized solution using an integer linear programming model was developed [28]. The objective of the model is to find two disjoint paths with the same parity, minimizing the sum of the costs of the paths. However, there can exist a long path that leads to increased latency. Therefore, we worked on maintaining the two disjoint paths with the same parity size routing while minimizing latency.

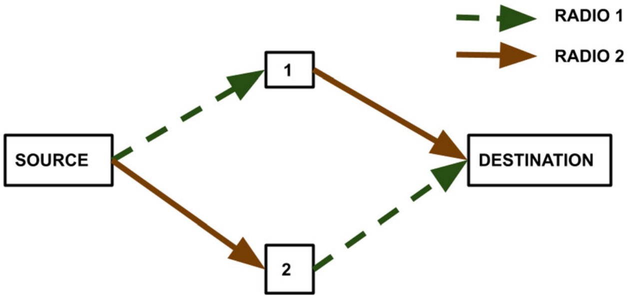

Fig. 3 An example of radio allocation in two disjoint paths with the same parity by the TwoPaths protocol

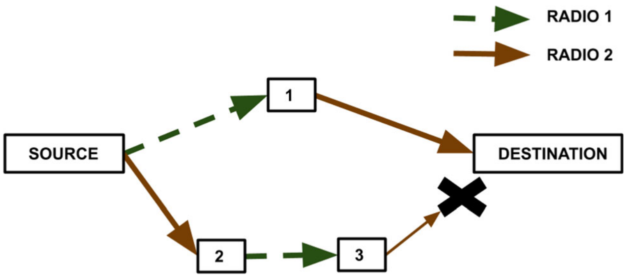

Fig. 4 An example that shows two disjoint paths, that do not have the same parity, as a consequence the destination node only receives through one radio

The TwoPaths protocol was implemented using TinyOS 2.1.2 for the Opal platform [21] and tested in a real-world testbed, in which the protocol was able to reach 96% of the theoretical limit for the network throughput, doubling the throughput in comparison to FastFoward. In this paper, we present a solution that maintains the throughput gain, doubled in comparison to the FastFoward protocol, and, at the same time, minimizes the latency by minimizing the longest path.

Multi-Radio and Multi-Channel Assignment Algorithms have been explored in the past, for instance, in Maritime Wireless Mesh Networks [29] or joint optimization of scheduling and power control in Wireless Networks [30]. Previous work on multi-channel and multi-radio does not consider dual radio platforms with heterogeneous radios (e.g. one in each band), as it happens with the Opal mote platform. This makes the problem different, requiring two paths with the same parity in size to obtain a better throughput.

The use of two radios in different bandwidths can improve throughput. More recently, link aggregation and variable bandwidth have emerged as solutions to also improve throughput. Junior et al. [31] propose intra and inter-flow aggregation in SDN networks. By aggregating flows, one can achieve higher throughput similar to using the idea of two radios in parallel. Milanez et al. [32] use variable bandwidth to improve the network throughout by allowing more bandwidth to the network bottleneck, providing the same benefit as the two radio platforms do. However, they do not develop a latency-minimizing algorithm for two paths in dual-radio platforms as shown here.

3 Problem definition

Given a graph $G = (V, E)$, in which V is a set of vertices and E is a set of edges, the problem of finding two disjoint paths with the same parity and with the lowest latency is to find two paths P_1 and P_2, beginning in the same source vertex *s* and ending in the same destination vertex *t*, with all relay nodes disjoint, both of the paths with the same parity in the number of hops, i.e., $|P_1| \bmod 2 = |P_2| \bmod 2$, and minimizing the longest path. We call this problem 2Min-Max-CD-Parity. We clarify that $\bmod q$ is the modulo operator, which returns the remainder of the division by *q*, in this problem, $\bmod 2$ returns if a number is even or odd.

The problem here called 2Min-Max-CD, of finding two disjoint paths to minimize the longest path in graphs with disjoint vertices, was proven to be NP-Complete [33]. Following, we prove that the problem 2Min-Max-CD-Parity is NP-Complete reducing from 2Min-Max-CD.

Theorem 1 *Given a graph* $G = (V, E)$ *and* $s, t \in V$, *it is NP-Complete to find two simple paths from s to t, with all relay vertices disjoint and with the same parity size, while minimizing the longest one.*

Proof First, we prove that 2Min-Max-CD-Parity is in NP, since, in polynomial time, we can verify the solution by comparing the parity of the number of hops in both paths, checking if the relay vertices are disjoint and appear no more than once.

Suppose that we have an algorithm that solves the 2Min-Max-CD-Parity problem. We can, through a polynomial reduction, use it to solve 2Min-Max-CD as seen next. We create a new graph G' from the original graph adding 2 vertices v_1, v_2, and edges from all the vertices that connect to *t* to v_1, v_2. An edge from *t* to v_1, v_2 is also added. The vertices v_1, v_2 and the edges connected to them were created so that a solution in G' does not need to have the same parity in *G*. The reduction is illustrated in Fig. 5. If the algorithm solves 2Min-Max-CD-Parity in G', then it finds a solution for 2Min-Max-CD in *G*. The same way, if it finds a solution for 2Min-Max-CD in *G*, this solution solves 2Min-Max-CD-Parity in G' using v_1 or v_2 to obtain the same parity if necessary, proving that the problem with parity is also NP-Complete. ☐

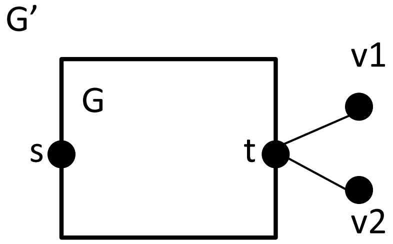

Fig. 5 Reduction between the problem of finding two disjoint paths minimizing the longest path and the problem of finding two disjoint paths with the same parity size and minimizing the longest path

4 The integer linear programming model

Here we present the integer linear programming model, whose objective is to find two disjoint paths with same parity size while minimizing the longest path. As a consequence, it reduces the time between the transmission and the reception of messages sent from the source to the destination node. Therefore, the solution can obtain a smaller delay while it maintains throughput.

To create the model we used the following representation: a directed weighted graph $G = (N, A)$, in which N is the set of all nodes in the network and A is the set of edges that represents connections between two nodes. For each edge (*i*, *j*), from *i* to *j*, a weight $c^1_{i,j}$ is associated and represents the cost of transmitting a data packet from the node *i* to the node *j* through radio 1, for the cost of transmitting through radio 2 there is the weight $c^2_{i,j}$. To each edge, it is also associated, binary variables represented as follows: $x^{\mathrm{Radio,Path}}_{i,j}$. They are: $x^{1,1}_{i,j}$, $x^{2,1}_{i,j}$, $x^{1,2}_{i,j}$, $x^{2,2}_{i,j}$, given that $x^{1,1}_{i,j}$ equals one *if and only if the edge (i,j) belongs to path 1 and radio 1 is being used to communicate from i to j*, $x^{1,2}_{i,j}$ equals 1 *if and only if the edge (i,j) belongs to path 2 and radio 1 will be used.*

$S(i)$ represents the set of nodes that can be directly accessed (can be accessed through one edge from i) from the i node, in other words, if $(i,j) \in A$ then $j \in S(i)$. $E(i)$ represents the set of nodes from which i can be directly accessed, in other words, if $(j, i) \in A$ then $j \in E(i)$.

The source node is called *s* and the destination node is called *d*. We define *I* as the set of all intermediate nodes (nodes that are neither the source nor the destination). The variables *r* and *p*, respectively, indicate the radio and the path associated with a certain edge, as a consequence, *r, p* ∈ {1,2}.

The integer linear programming model was implemented using the GMPL language (GNU Mathematical Programming Language) and solved using the GLPK (GNU Linear Programming Kit), an open-source tool to solve linear programming problems. The source code of the solution is available in a public repository on GitHub.[1] Any ILP solver can be used to solve the model.

The Model

minimize

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

subjected to

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

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

---

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

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

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

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

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

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

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

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

5 Model explanation

The objective function minimizes the longest path, which is the same as minimizing the sum of the costs of all the edges that are part of the path. The edges that constitute Path 1 are indicated by the binary variables $x_{i,j}^{1,1}$ and $x_{i,j}^{2,1}$, which are equal to 1 if used in the path. The same happens to the second path indicated by the binary variables $x_{i,j}^{1,2}$ and $x_{i,j}^{2,2}$.

Restrictions (I), (II), and (III) define the beginning of the paths, ensuring that the source node does not receive from any node and that the paths start with different radios.

Restriction (IV) guarantees the destination node does not transmit to any other node. While restriction (VIII) assures that the paths have the same parity, as a consequence the paths end with different radios avoiding a bottleneck at the destination node.

Restrictions (V) and (VI) coordinate continuity in the paths, in a way that the paths start from the source node and can only end at the destination node. These conditions also guarantee that the radios are alternated on each hop in a path.

Restriction (VII) ensures that all nodes but the destination node receive from at most one radio.

Restriction (IX) restates the binary character of the variables associated with the edges.

6 Experiments and results

The experiments were done using instances created from a real-world *testbed* called *Twonet* [34] which contains 100 sensor nodes of platform Opal, shown in Fig. 1. The testbed is composed of 100 Opal Motes, The sensor node is a platform that has an Atmel's SAM3U Cortex-M3 MCU processor and two radios: an AT86FR212 that operates in the 900MHz band (which has 10 channels), and an AT86RF231 radio that operates in the 2.4 GHz (which has 16 channels) for IEEE 802.15.4.

Figure 6 shows an illustration of the *testbed*, each dot in the image represents a sensor node, and a dot was enlarged to show a node with two antennas. The *testbed* occupies four floors of a building at the University of Houston.

The IEEE802.15.4 standard only defines and implements PHY and MAC layers. For testing, we used TCP at the transport layer. They operate independently on their respective layers. TCP, at the transport layer, will be responsible for end-to-end reordering. IEEE802.15.4 radios just transmit and receive packets. We created one TCP flow that was splitted into two sub-flows at the MAC layer.

The cost of each link and the network topology was collected from the *testbed* in the same way as in [28]. From the collected topology, instances were obtained by varying the source and destination nodes, in a way that the topology is always the same. The metric used for the links cost is the ratio of the number of packets sent to the number of packets received, which is the inverse of the packet delivery rate. The larger the number of packets received, the smaller the cost, given that the number of packets sent stays the same. We used a blacklist to remove unstable links that have PRR below 80% as indicated by [35].

The cost of a path is defined as the sum of the costs of each link used in the path. Latency is defined as the cost of the most costly path between two nodes and is the main metric used in this paper. We also used the total cost of the solution, which is the sum of the cost of both paths, minimizing this metric does not necessarily minimize the latency of a solution.

The experiments were done in 5,700 instances. We used the integer linear programming model described in the last section of this paper. We compared our model with the integer linear programming model from Two Paths, as it also builds two disjoint paths with the same parity but minimizes the total cost.

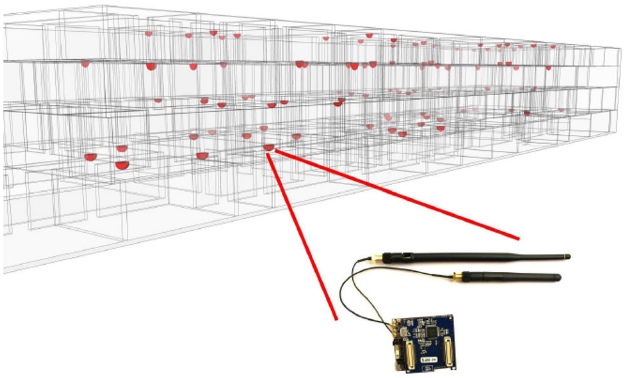

Fig. 6 Testbed TwoNet com 100 Opal motes

The models were implemented using the language GMPL (GNU Mathematical Programming Language) and solved using GLPK (GNU Linear Programming Kit), an open-source tool to solve linear programming problems. The experiments were done in a machine with the following configuration: Ubuntu operating system, x86 64 architecture, Intel(R) Core(TM) i7-4790 CPU with a clock of 3.60 GHz, 8GB RAM. The model proposed in this paper will be called "MinLatency" to facilitate the results' display.

The collected data were: total cost, costs of paths 1 and 2, the difference between the costs of the paths, and the average time taken to compute the solution for an instance of the model. To see the difference in the number of hops, in one experiment the models were modified to minimize the number of hops instead of the cost. To display the results we used cumulative distribution functions, which allow us to analyze large amounts of results. It is important to note that we refer to the path with the higher cost as the longest path. The longest path is not used to refer to the one with more hops, but the one with the highest cost.

Figure 7 shows the frequency of the cost of each path for the solution that minimizes latency. Figure 8 shows the same for the TwoPaths solution. In both plots, path number 2 is the path with a higher cost. These plots allow us to compare the two models concerning the cost of each path separately. The lines of the MinLatency plot are closer to each other than the ones of the TwoPaths figure, which shows that the model reduces the delay between the paths. The figure also shows that the maximum cost found by MinLatency, which was 105, is significantly smaller than the one found by TwoPaths, which was 135.

Figure 9 enables a better analysis of the difference in the costs of the paths in each instance. The plot shows that MinLatency obtains a much better balance in path cost and the worst case is also much lower than TwoPaths. This is important because big differences between the costs of the paths imply big differences in latency time for each path. The figure shows that the line of MinLatency begins higher and stays higher than the TwoPaths line, showing that its values are generally lower than the ones on the other line. The solution that minimizes latency has a larger concentration of values between 0 and 20 than the other model. In addition, TwoPaths has many results with a difference higher than 90, achieving more than 120, which means that it has extreme cases much worse than the ones from MinLatency.

Figure 10 allows us to observe the behavior of both models concerning the total cost (sum of the cost of both paths). Both models achieve very similar results, as lines overlap in a major part of the plot. We can verify that TwoPaths obtains better results, as was expected. However, MinLatency can achieve results close to the model that

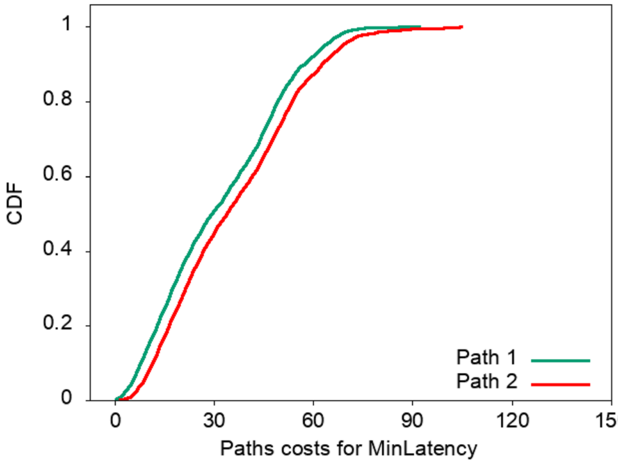

Fig. 7 Cost of each path found by MinLatency (proposed)

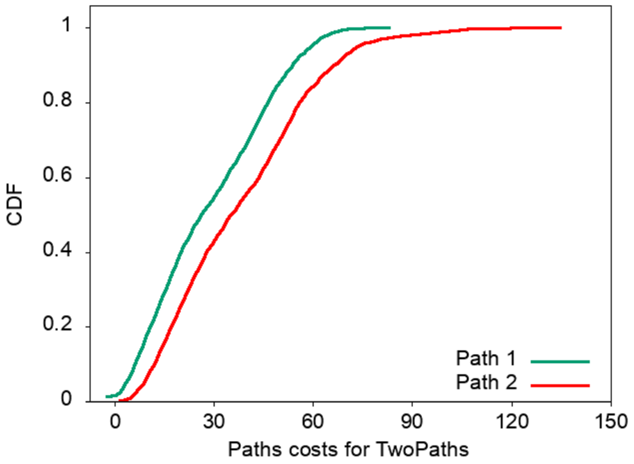

Fig. 8 Cost of each path found by TwoPaths (benchmark)

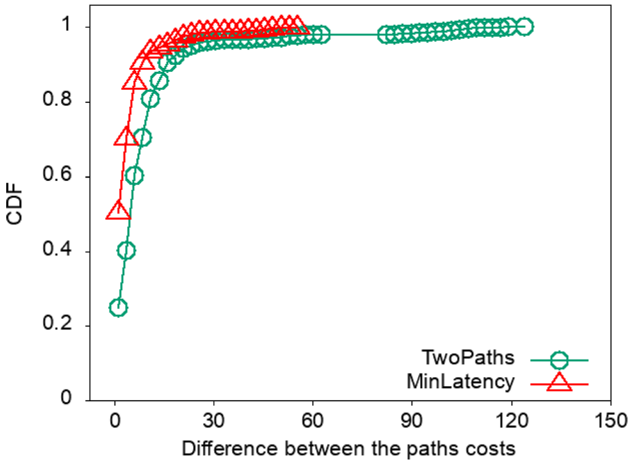

Fig. 9 Difference between the costs of the paths for each solution, where MinLatency is our proposed protocol and TwoPaths is the benchmark

minimizes the total cost, in other words, it is close to the optimal solution.

Figure 11 allows us to directly compare the model's performance concerning latency. We can verify that the results are similar but MinLatency can achieve lower latency than TwoPaths.

Figure 12 displays the results in regards to the cost of the longest path when all the hops have the same cost, equal to 1. This scenario is the same as when the models only consider the number of hops instead of the cost. Around 85% of the results are the same for both models, but MinLatency is better as it achieves a better distribution of results and a lower worst case, equal to 4, in comparison to TwoPaths, whose worst case is equal to 5 hops.

Figure 13 displays a general view of the solutions considering the averages of the metrics of throughput and latency. For some applications, it is better to have higher throughput and, also, to have a better latency, represented as the cost of the longest path. FastForward achieves the lowest latency, as it only uses the shortest path, but it also has the lowest throughput. The throughput is doubled by using two paths with the same parity (from 30 kBps to 60 kBps), as happens in the TwoPaths algorithm. The solution proposed in this paper is able to maintain the throughput gain achieved by using two paths with the same parity and to achieve a better latency than TwoPaths.

Table 1 shows the average time taken by GLPK to solve an instance of each model. We can see that MinLatency takes significantly more time to compute. This difference can be explained by the fact that the MinLatency model has more restrictions than the one that minimizes the total cost, resulting in a larger input to an integer linear programming problem and consequently to a much higher computational cost. Another factor is the high standard deviation in both

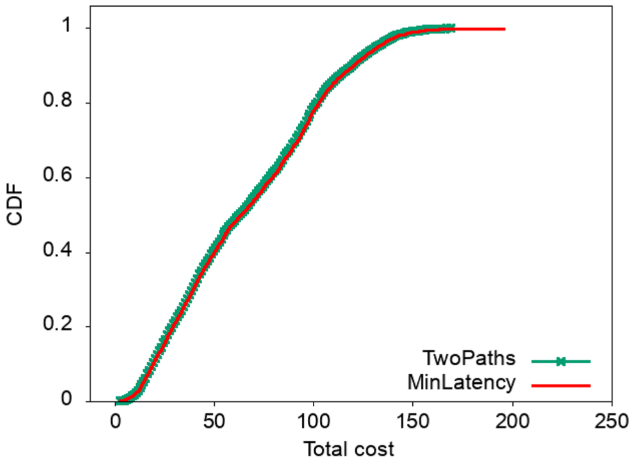

Fig. 10 Total cost, where MinLatency is our proposed protocol and TwoPaths is the benchmark

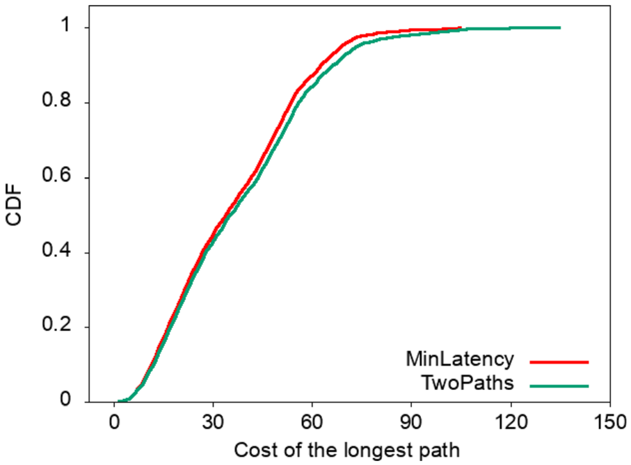

Fig. 11 Cost of the longest path, where MinLatency is our proposed protocol and TwoPaths is the benchmark

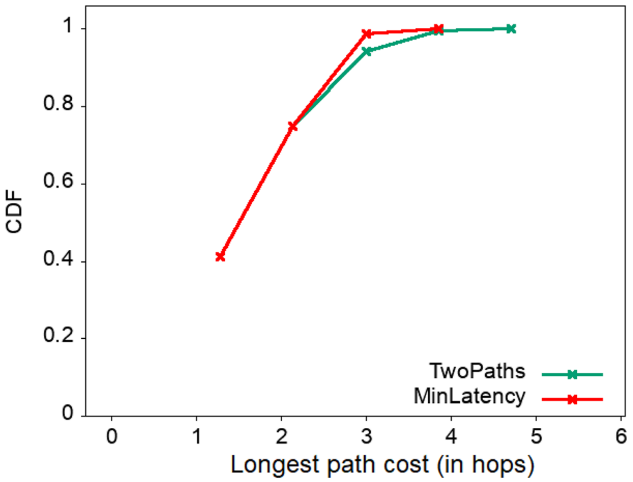

Fig. 12 Cost of the longest path using the model that only considers the number of hops, where MinLatency is our proposed protocol and TwoPaths is the benchmark

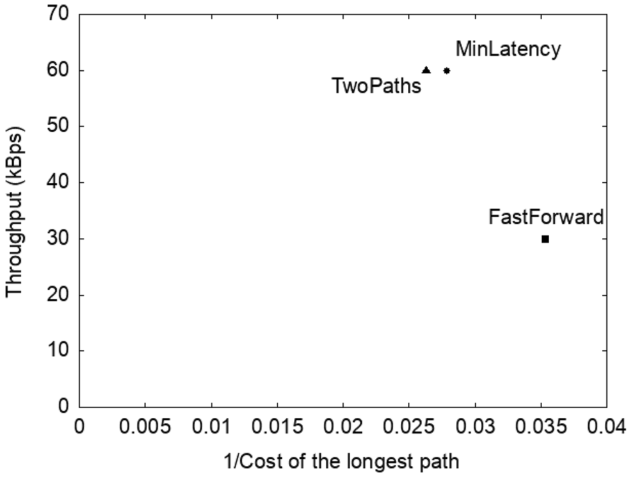

Fig. 13 General view of the solutions

Table 1 Average time to solve each instance of the models

Model	TwoPaths	MinLatency
Average	0,1 s	6,7 s
Standard deviation	0,3	11,6

models, showing that the time to solve the models is very sensitive to different inputs.

Different latencies cause packets to be ordered. This is done at the transport layer, e.g. using TCP.

As it is an exact solution to an NP-Complete, in a real sensor nodes network with a large number of nodes approximate solutions may be better suited, as the cost may become too high. In tests with randomly generated graphs, the model was tested in topologies of 10 to 1500 nodes, reaching 75 thousand edges.

It is important to note that the execution time of 6,7 s passed in a computer that computes the routes outside of the sensor network, so it does not have any carry-over to the power consumption of the network nodes. As it was mentioned in the Introduction section, the gains concerning the power consumption of the Opal platform were analyzed in-depth in [21].

Another factor is that, even though we proved that the problem is NP-Complete, the model was capable of solving instances of a real testbed with 100 nodes.

A heuristic algorithm could solve the problem in less time and replace the generic ILP model solver. However, we still want to know the optimal solution. The ILP model can give the optimal solution, while the heuristic does not guarantee that.

7 Conclusion

Using two disjoint paths with the same parity size in dual heterogeneous radio platforms was recently shown to double the network throughput, close to achieving the maximum theoretical network throughput limit. However, one path could be longer than the other, causing high latency.

In this paper, we defined the problem of finding two disjoint paths with the same parity size while minimizing the latency, which we proved to be NP-Complete. We presented a solution based on integer linear programming that allows us to keep the gains of using two radios, doubling throughput, and reducing latency.

The experiments used data from a real testbed in 5,700 instances. The results allowed us to conclude that the model advances the state-of-the-art in the reduction of latency, doing so without sacrificing total cost.

For future work, we could develop a routing protocol for two paths with the same parity minimizing latency that can be computed by the network sensors themselves. Also to use network coding to improve network throughput, as in CodeDrip [36].

Acknowledgements The authors would like to thank the research funding agencies CNPq processes 309496/2019-5 and 305979/2022-1, FAPESP 2020/05183-0, and FAPEMIG 02400-18 and APQ-00823-18 for their financial support.

Funding Fundaço de Amparo à Pesquisa do Estado de São Paulo, Fundação de Amparo à Pesquisa do Estado de Minas Gerais, Conselho Nacional de Desenvolvimento Científico e Tecnológico.

Declarations

Conflict of interest This work was supported by CNPq, FAPEMIG and FAPESP research agencies. The authors have no relevant financial or non-financial interests to disclose. All authors contributed to this paper. On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Akyildiz, I. F., & Vuran, M. C. (2010). *Wireless sensor networks* (Vol. 4). Wiley.
2. Santos, B. P., Goussevskaia, O., Vieira, L. F., Vieira, M. A., & Loureiro, A. A. (2018). Mobile matrix: Routing under mobility in IoT, IoMT, and social IoT. *Ad Hoc Networks, 78*, 84–98.
3. Vieira, M. A. M., Coelho Jr. C. N., da Silva Jr. D. C., & da Mata, J. M. (2003). Survey on wireless sensor network devices. In *Emerging technologies and factory automation, 2003. Proceedings. ETFA'03. IEEE conference* (vol. 1, pp. 537–544).
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., & Fernandes, A. O. (2004). Arquiteturas para redes de sensores sem fio. In *Tutorial no simpósio brasileiro de redes de computadores (SBRC)* (pp. 167–218).
5. Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks, 52*(12), 2292–2330.
6. Câmara Júnior, E. P., Vieira, L. F., & Vieira, M. A. (2020). Captain: A data collection algorithm for underwater optical-acoustic sensor networks. *Computer Networks, 171*, 107145.
7. Beutel, J. (2006). Fast-prototyping using the btnode platform. In *Design, automation and test in Europe, 2006, DATE'06. Proceedings* (vol. 1, pp. 1–6). IEEE.
8. Kohvakka, M., Arpinen, T., Hännikäinen, M., & Hämäläinen, T. D. (2006). High-performance multi-radio wsn platform. In *Proceedings of the 2nd international workshop on multi-hop ad hoc networks: from theory to reality* (pp. 95–97). ACM.
9. Burns, A., Greene, B. R., McGrath, M. J., O'Shea, T. J., Kuris, B., Ayer, S. M., Stroiescu, F., & Cionca, V. (2010). Shimmer^{TM}-a wireless sensor platform for noninvasive biomedical research. *Sensors Journal IEEE, 10*(9), 1527–1534.
10. Popovici, E., Boyle, D., O'Connell, S., Faul, S., Angove, P., Buckley, J., O'Flynn, B., Barton, J., & O'Mathúna, C. (2011). The s-mote: A versatile heterogeneous multi-radio platform for wireless sensor networks applications. In *Circuit theory and design (ECCTD), 2011 20th European conference on* (pp. 421–424). IEEE.

11. Alliance, L. (2021). IoT DevKit–LoRaWAN. https://lora-alliance.org/
12. Azmi, N., Sudin, S., Munirah Kamarudin, L., Zakaria, A., Visvanathan, R., Chew Cheik, G., Mamduh Syed Zakaria, S. M., Abdullah Alfarhan, K., & Badlishah Ahmad, R. (2018). Design and development of multi-transceiver Lorafi board consisting LoRa and ESP8266-Wifi communication module. In: *Materials science and engineering conference series* (vol. 318, pp. 012051).
13. Kim, D. H., Lim, J. Y., & Kim, J. D. (2016). Low-power, long-range, high-data transmission using wi-fi and lora. In *2016 6th international conference on IT convergence and security (ICITCS)* (pp. 1–3).
14. Ferreira, P., Miranda, R. N., Cruz, P. M., & Mendonça, H. S. (2019). Multi-protocol lorawan/wi-fi sensor node performance assessment for industry 4.0 energy monitoring. In *2019 IEEE-APS topical conference on antennas and propagation in wireless communications (APWC)* (pp. 403–407).
15. libelium. (2021). Waspmote. http://www.libelium.com/products/waspmote
16. Tuset, P., Vilajosana, X., & Watteyne, T. (2021). OpenMote B. https://openmote.com/product/openmote-b/
17. Zolertia. (2021). Firefly mote. http://zolertia.io/product/hardware/firefly
18. Li, H., Dong, W., Wang, Y., Gao, Y., & Chen, C. (2020). Enhancing the performance of 802.15.4-based wireless sensor networks with NB-IoT. *IEEE Internet of Things Journal, 7*(4), 3523–3534.
19. Jurdak, R., Klues, K., Kusy, B., Richter, C., Langendoen, K., & Brünig, M. (2011). Opal: A multiradio platform for high throughput wireless sensor networks. *Embedded Systems Letters IEEE, 3*(4), 121–124.
20. Kusy, B., Richter, C., Hu, W., Afanasyev, M., Jurdak, R., Brünig, M., Abbott, D., Huynh, C., & Ostry, D. (2011). Radio diversity for reliable communication in wsns. In *Information processing in sensor networks (IPSN), 2011 10th international conference* (pp. 270–281). IEEE.
21. Jurdak, R., Klues, K., Kusy, B., Richter, C., Langendoen, K., & Brünig, M. (2011). Opal: A multiradio platform for high throughput wireless sensor networks. *Embedded Systems Letters IEEE, 3*(4), 121–124.
22. Yin, S., Gnawali, O., Sommer, P., & 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* (pp. 345–353).
23. Ekbatanifard, G., Sommer, P., Kusy, B., Iyer, V., & Langendoen, K. (2013). Fastforward: High-throughput dual-radio streaming. In *IEEE 10th international conference on mobile Ad-Hoc and sensor systems* (pp. 209–213). IEEE.
24. Ribeiro Junior, N. D. S., Vieira, M. A. M., Vieira, L. F. M., & Gnawali, O. (2021). Dual radio networks: Are two disjoint paths enough? *IEEE Internet of Things Magazine, 4*(1), 67–71.
25. Draves, R., Padhye, J., & Zill, B. (2004). Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th annual international conference on mobile computing and networking* (pp. 114–128). ACM.
26. Marina, M. K., Das, S. R., & Subramanian, A. P. (2010). A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. *Computer Networks, 54*(2), 241–256.
27. Raniwala, A., & Cker Chiueh, T. (2005). 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, pp. 2223–2234).
28. dos Santos Ribeiro, N., Vieira, M. A., Vieira, L. F., & Gnawali, O. (2022). Splitpath: High throughput using multipath routing in dual-radio wireless sensor networks. *Computer Networks, 207*, 108832.
29. Xing, J., Li, Q., Sun, H., Qi, J., Zhu, Z., & Wang, J. (2019). Multi-radio multi-channel assignment algorithm in maritime wireless mesh networks. *IEEE Access, 7*, 85 903-85 912.
30. Liu, L., Cheng, Y., Cao, X., Zhou, S., Niu, Z., & Wang, P. (2019). Joint optimization of scheduling and power control in wireless networks: Multi-dimensional modeling and decomposition. *IEEE Transactions on Mobile Computing, 18*(7), 1585–1600.
31. Junior, R. R., Vieira, M. A., Vieira, L. F., & Loureiro, A. A. (2022). Intra and inter-flow link aggregation in SDN. *Telecommunication Systems, 79*(1), 95–107.
32. Milanez, G. A. S., Vieira, M. A., Vieira, L. F., & Nacif, J. A. (2023). Tsch-varba-dynamic channel allocation with variable bandwidth in Tsch networks. *Computer Networks, 2023*, 4171549.
33. Li, C.-L., McCormick, S., & Simchi-Levi, D. (1990). The complexity of finding two disjoint paths with min-max objective function. *Discrete Applied Mathematics, 26*(1), 105–115.
34. Li, Q., Han, D., Gnawali, O., Sommer, P., & 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* (pp. 89). ACM.
35. Son, D., Krishnamachari, B., Heidemann, J. (2004). Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks. In *2004 1st annual IEEE communications society conference on sensor and Ad Hoc communications and networks. IEEE SECON 2004* (pp. 289–298). IEEE.
36. dos Santos-Ribeiro Júnior, N., Tavares, R. C., Vieira, M. A., Vieira, L. F., & Gnawali, O. (2017). Codedrip: Improving data dissemination for wireless sensor networks with network coding. *Ad Hoc Networks, 54*, 42–52.

Gabriel Santos Luz is a master student at UFMG. He received his undergraduate at the Universidade Federal de Minas Gerais in Belo Horizonte. His research interests are in Computer Networking and Robotics.

Nildo dos Santos Ribeiro Junior received his M.Sc. in Computer Science at the Universidade Federal de Minas Gerais (UFMG). His research interests are in Wireless Sensor Networks and Computer Networking.

Luiz F. M. Vieira is an Associate Professor of the Computer Science Department at the Universidade Federal de Minas Gerais (UFMG). He received his undergraduate and M.S. at the Universidade Federal de Minas Gerais in Belo Horizonte, and the Ph.D. degree in Computer Science from the University of California Los Angeles (UCLA). His research interests are in Computer Networking and Wireless Networks.

Marcos A. M. Vieira is an Associate Professor of the Computer Science Department at the Universidade Federal de Minas Gerais (UFMG). He received his undergraduate and M.S. at the Universidade Federal de Minas Gerais in Belo Horizonte, and M.S. and Ph.D. degrees in Computer Science from the University of Southern California (USC). His research interests are in Computer Networking and Wireless Networks.

Omprakash Gnawali is an Assistant Professor at the University of Houston, USA. He was a Postdoctoral Scholar at Stanford University, got his Ph.D. from the University of Southern California, and received his Masters and Bachelors degrees from the Massachusetts Institute of Technology. His research lies at the intersection of low power wireless networks and embedded sensing systems.