DCTP-A and DCTP-I: Collection Tree Protocols for Dual Radio Platforms

Gabriel S. Luz
Universidade Federal de Minas Gerais
Belo Horizonte, Minas Gerais, Brazil
gabrielluz@dcc.ufmg.br

Luiz F. M. Vieira
Universidade Federal de Minas Gerais
Belo Horizonte, Minas Gerais, Brazil
lfvieira@dcc.ufmg.br

Marcos A. M. Vieira
Universidade Federal de Minas Gerais
Belo Horizonte, Minas Gerais, Brazil
mmvieira@dcc.ufmg.br

Omprakash Gnawali
University of Houston
Houston, Texas, USA
gnawali@cs.uh.edu

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, DCTP-A and DCTP-I. DCTP-A builds the collection tree alternating the radio band each node while DCTP-I builds two independent collection trees. The protocols were implemented in TinyOS and evaluated experimentally in a testbed in the physical world using the 900MHz and 2.4GHz radio bands, compared to the state of the art (CTP and CTP-Multi) and to each other, considering the metrics delivery rate, latency, throughput in a saturated network 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. DCTP-A achieved almost 100% of delivery rate, while DCTP-I achieved up to 90% delivery rate with less number of beacons messages.

CCS CONCEPTS

• **Computer systems organization → Sensor networks**; • **Networks → Network protocol design**; **Network layer protocols**.

KEYWORDS

wireless sensors networks, routing, dual radio, ctp, throughput

ACM Reference Format:
Gabriel S. Luz, Luiz F. M. Vieira, Marcos A. M. Vieira, and Omprakash Gnawali. 2019. DCTP-A and DCTP-I: Collection Tree Protocols for Dual Radio Platforms. In *22nd Int'l ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '19), November 25–29, 2019, Miami Beach, FL, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3345768.3355912

MSWiM '19, November 25–29, 2019, Miami Beach, FL, USA

ACM ISBN 978-1-4503-6904-6/19/11…$15.00
https://doi.org/10.1145/3345768.3355912

1 INTRODUCTION

A wireless sensors network (WSN) is a network composed of distributed sensor nodes. Each sensor node is equipped with a variety of application specific sensors, a microprocessor and a radio transceiver, allowing the direct communication between the network elements. Each node can act as a collector and forwarder of data [15]. With these characteristics, WSNs are easy to deploy, have great capacity for distributed sensing and are widely used in a variety of applications.

A large part of the WSNs applications involve sending data from the other nodes to specific nodes that are in charge of centralizing the collected information, this important process is called data collection. Several data collection protocols have been proposed in the literature, among them the Collection Tree Protocol (CTP) [3] gained recognition for its efficiency and reliability. CTP is briefly explained in the Related Work Section.

Recently, aiming to increase throughput and maintain a low power consumption per transmitted byte, WSN platforms were equipped with two radios transceivers in each sensor node. The cost of adding a second radio to a sensor node is small and can greatly improve network performance and power consumption. As an example we can cite the Opal platform [5], whose sensor node, shown in Figure 1, has a SAM3U Cortex-M3 MCU Atmel processor and two radios, a AT86FR212 that operates in the 900 MHz band (which has 10 channels) and a AT86RF231 that operates in the 2.4 GHz (which has 16 channels). As each radio operates in different radio bands, it is possible to prevent interference between the radios.

The use of multiple radios allows simultaneous transmissions between the nodes, which increases network throughput, stability, delivery rate and decreases power consumption per transmitted byte. As an illustration, the Opal mote radios consume 0.669 and 0.659 pj/bit/m^2, while the TelosB CC2420 consumes 11.89 pj/bit/m^2 [6]. The gains in energy consumption in the Opal platform have been shown with depth in [5]. In addition, Yin et al. [16] demonstrated that the 900 MHz ISM band has better connectivity than the 2.4 GHz ISM band. Consequently, using two radios also has benefits to link connectivity and quality. It is important to highlight that using multiple radios is different from using multiple channels. The latter approach does not enable a mote to transmit and receive simultaneously as in the first one. Furthermore, it is possible to use multiple channels when using multiple radios.

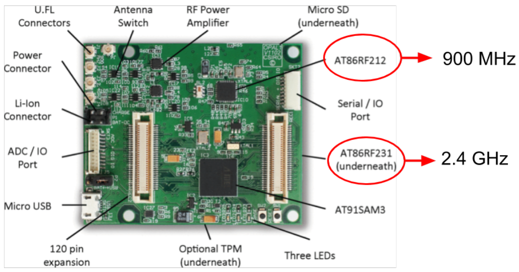

Figure 1: The Opal mote and its components. Source: [5]

The main contributions of this paper can be summarized as follows. We present DCTP-A and DCTP-I, two novel collection tree protocols for dual-radio networks. DCTP-A builds the collection trees alternating the radio bands while DCTP-I builds two independent collection trees. The protocols have low memory footprint and were implemented in the *TinyOS* platform. We compared the protocols with each other and with the state-of-art protocols, the original CTP and Multi-CTP. We present real world results, evaluating the protocols in a testbed composed of 100 Opal motes. We show that both protocols present gains in the delivery rate, throughput and latency.

The two new protocols aim to explore the effects of dual radio in data collection performance. CTP-Multi is a modification of CTP to use two radio bands, proposed in [6]. DCTP-A and DCTP-I are different from CTP-Multi in the way the radios are used, the new ones use the modification to improve throughput, reliability, latency and performance in general, while the other only uses it to find a better link for a single hop transmission. The novel protocols advantages were tested and evaluated in a real world testbed.

This paper is organized as follows. The next section 2 presents the related work and 3 describes the problem tackled by the protocols. The following two sections 4 and 5 present the new protocols, Dual Radio Collection Tree Protocol with Independent Trees (DCTP-I) and Dual Radio Collection Tree Protocol with Radio Alternation (DCTP-A). Section 6 discusses the experiments and results obtained. Finally, Section 7 concludes the paper.

2 RELATED WORK

2.1 Collection Tree Protocol (CTP)

The CTP was presented in the paper [3], along with two principles for routing protocols: *Datapath Validation* and *Adaptative Beaconing*. The protocol was implemented for the *TinyOS* platform and its description can be found in the document TEP 123 in the *TinyOS* documentation.

CTP is a data collection protocol based on tree routing. Each tree is rooted in a collection node, which is in charge of receiving and storing data collected by the nodes of its tree. A sensor node belongs to only one tree. A collection node announces itself as a root node, informing CTP, which takes on the role of creating a tree for the new root. A node does not know to which tree it belongs, it only knows which node is its father in the tree. A node sends all the collected data and forwards the packets it receives to its father. To build the routes (trees), the CTP estimates link quality using the Expected Transmission Count (ETX) metric. To each direct transmission between two nodes it is associated with an ETX value, representing the quality of this link (the higher is the ETX, the worst is the link). To the nodes an ETX value is also associated, it is the sum of the ETX values of all the links that compose the path from this node to the root, as a result a root's ETX is zero. Figure 2 shows the tree generated by CTP for a 7 nodes network, the first node is the root.

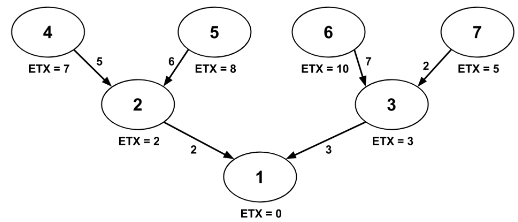

Figure 2: Example of a tree built by the CTP, with the nodes and links ETX values.

The operation of CTP can be divided into three main components: Link estimator, Routing and Forwarding.

The role of the Link Estimator is to estimate the transmission ETX value of a link between two neighbor nodes. To calculate the ETX, the estimator uses data packets sent by the forwarding component and the beacons sent by the routing. The beacons are sent as broadcast periodically according to a timer that increases exponentially to a max value and can be reduced to a minimum value if certain events occur. The longer the period of the timer, the fewer beacons are sent, and when it is set to the minimum value the send rate of beacons is the highest so that the network can adapt to changes in the topology. A change in a node or link is announced by setting some bits in the beacon header. This mechanism is called *Adaptative Beaconing* and was inspired by the *Trickle* [7] algorithm. A beacon contains the identifier and the ETX of the node that sent it, when a node receives a beacon it updates its routing table with the new ETX informed. The Link Estimator monitors whether the data packets reach or not the next hop, and use the aggregated information each time five data packets are sent to calculate the link ETX. The beacons are also used to estimate link quality, using a sequence number that is incremented at each transmission, the missing numbers indicate the number of lost beacons. Therefore, node u can estimate route quality to the root passing through node v, adding the ETX value of v, which was sent by a beacon from v to u, with the ETX of the link $u \rightarrow v$ that was obtained by monitoring data packets and beacons.

The Routing is responsible for choosing the next hop for a data packet transmission and for controlling the sending of beacons. The next hop, which is the father of the node in the tree, is the node that if chosen will provide the route with the smallest ETX to the root. The sending of beacons is done in the way explained in the Link Estimator.

The Forwarding controls the sending and forwarding of data packets. This component consults the Routing to get the next hop address, chooses when to send a packet, passes information to the Link Estimator and implements the principle of *Datapath Validation*.

A problem in WSN routing is the presence of loops in a route, which will confine the packets sent through this route preventing them from reaching the root. To deal with this problem, the packets contain the ETX value of the last node that transmitted it, as the ETX must decrease in direction to the root, then if a node receives a data packet carrying an ETX value lower than its value, then it detects a loop and sets the beacon timer to the minimum to correct this inconsistency. As a result, it is possible to validate the route using data packets. Another problem is the duplication of packets, which occurs when a node re-transmit a packet that had successfully arrived on the next hop, generating two equal packets in the network. This problem might grow exponentially and must be treated, CTP does it using a message cache in each node. If the message received is already in the node's cache then it discards it and does not forward it.

2.2 Protocols Similar to CTP

Many modifications to CTP have been proposed in the literature. The XCTP [12] provides in addition to the communication from nodes to root, the communication from root to nodes, extending the features of CTP. Another examples are *Matrix* [9] and *Mobile Matrix* [11] which uses data collection tree routing protocol, as CTP, as a base for a routing protocol with the hierarchical allocation of IPv6 addresses for mobile networks. Funneling Wider Bandwidth (FWB) [13] uses wider bandwidth channels to improve data collection and reduce the overall number of time slots. FlushMF [14] uses multiple frequencies as transport protocol. Other protocols use multiple channels to create better routes and reduce interference, an example is the protocol *Multi-channel CTP* [10].

However there is very little in the literature about using multiple paths to improve data collection. One example of work in this area is *MMCR* [1] that uses multiple radio interfaces and multiple channels.The main protocol found in the literature that adapts the CTP to use multiple radio interfaces and multiple channels is the CTP-Multi [6], which is a dual radio protocol that modifies the Link Estimator to choose the best radio band for a link and modifies the Routing to send beacons by multiple radios. It was implemented using a separate table for each radio band in the Link Estimator, obtaining different estimations for the link quality using different radio bands. The Routing only uses one routing table filled with the ETX values of the neighbor nodes obtained from the beacons received. At the moment of updating the route the neighbor with the best link and ETX and the respective radio are chosen for the next transmissions.

In this paper we explored two approaches that allow improving network performance not by choosing the best radio for a link as done by CTP-Multi, but by building two independent trees or alternating the radio band used at each transmission.

3 PROBLEM DEFINITION

The main goals that CTP [3] aims to achieve are:

- Reliability : A data collection protocol must be able to obtain a delivery rate higher than 90%.
- Robustness: A protocol must be able to operate without settings or configurations in a variety of topology, conditions and environments.
- Efficiency: The number of transmissions and states required to transmit a packet must be minimized..
- Hardware Independence: It should not assume specific hardware or radio chip characteristics.

Other factors are also important in data collection, specially latency and throughput. Low latency is very important for real time monitoring and applications that require a fast response to certain events. In [8] the researchers developed solutions for radio selection and data partitioning in one hop transmissions in multiple radio platforms, the aim was to fulfill real-time data transfer constraints and maximize energy efficiency. In the paper they used the example of medical applications that require real-time constraints to justify the importance of real-time protocols. Many others applications also require fast and efficient data transfer and to achieve this, data collection must have low latency and high throughput.

The problem this work aims to solve is creating new protocols using two radio bands that improve CTP in regard to reliability, robustness and efficiency, and also improve the end to end throughput and delay.

4 DUAL RADIO COLLECTION TREE PROTOCOL WITH INDEPENDENT TREES (DCTP-I)

DCTP-I expands the CTP concept from one to two trees rooted on the same root node. In this protocol each node belongs to two independent trees, the first is only used by radio 1 and the second is used only by radio 2. To create the trees, the Routing and Link Estimator were duplicated.

When a packet is created it has a bit that indicates which radio must be used to transmit it. A packet with the radio bit set to one can only be transmitted by radio 1, as a result this packet will remain in the same tree from the moment it was created until it reaches the root. The trees are designated to each packet in an alternate way, the first packet is assigned to the tree 1, the second to tree 2, the third to tree 1 , the fourth to tree 2 and so on. Consequently each tree receives half of the packets.

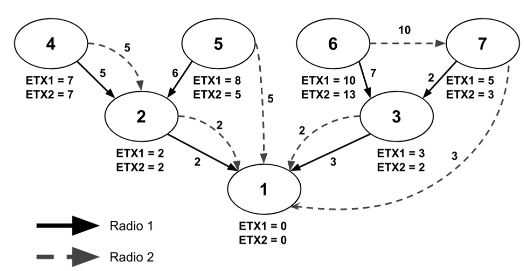

Figure 3: Example of the two trees formed by DCTP-I, with the ETX values of the nodes and links for each radio band.

As the Routing and Link Estimator components were duplicated, the trees are created independent of each other, but with the same root node. The two routing tables and the sending of beacons are independent for each radio. As a result, each node belongs to two trees and has two fathers, which can be different or not. As the radios operate at different bandwidth, their reach is also different.

Figure 3 shows the two trees created by DCTP-I for the same network of Figure 2. It shows that the nodes share the same root and each node has two fathers and two ETX values. The Forwarding sends packets according to the radio bit. Before sending, it verifies the bit radio and consults the respective Routing component to know which father to send the packet to.

5 DUAL RADIO COLLECTION TREE PROTOCOL WITH RADIO ALTERNATION (DCTP-A)

DCTP-A uses a different approach from DCTP-I. Instead of always using both radio bands to send and receive packets, this protocol makes a node always receives by one radio and sends by another. Figure 4 shows the tree formed by DCTP-A in the same network used in the last figures. Alternating radios follow the principle used by the routing protocols [2] and [4], which improved the throughput by enabling that a node sends and receives packets simultaneously.

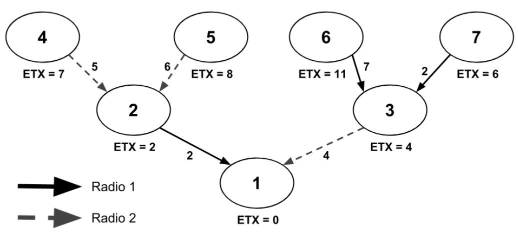

Figure 4: Example of the tree formed by DCTP-A, with the ETX values of nodes and links.

To implement DCTP-A the main components of CTP were modified. It uses two Link Estimators, one for each radio, which inform the link ETX values to the Routing component. The Routing stores a variable that indicates which radio band should be used to send packets and creates a routing table for each radio. Routing table X contains the ETX values of the neighbors to which radio X is used for the transmission. The Forwarding is similar to the CTP one, but it connects to both Link Estimators and before sending a packet it consults the Routing to know which radio to use. Therefore, the tree is formed so that a node only receives packets by one radio and only sends data packets and beacons by the other. However, it is possible that the tree is not perfect and a node may receive and send using the same radio band. A node always receives beacons by both radios.

To understand how the radios are alternated it is necessary to understand how the Routing behaves when a beacon is received and how the route is altered in each node. When a beacon is received by radio 2 containing the ETX value of a neighbor, we know that this ETX value is associated to radio 2, since a node only sends beacons and data packets with the same radio. To alternate the radio bands used, this ETX value and the neighbor identifier are placed into the radio 1 routing table. When a node updates its route (father node and ETX) it scans the two routing tables and picks the best neighbor from each one, to the table 1 neighbor's ETX we add the ETX value of the link using radio 1, to the table 2 neighbor's ETX we add the ETX value of the link using radio 2. We choose the best neighbor from them, if the best neighbor ETX is better than the current ETX value subtracted from a certain threshold value, which indicates how much better the new ETX must be, it becomes the new father and the ETX value is updated. If the radio used to reach the new father is different from the last radio, then the radio used to transmit is updated and the beacon timer period is set to the minimum value, increasing the beacon sending rate to warn the neighbors about the change. Altering the radio band results in the node receiving and sending by the same band, creating a bottleneck in the path, in order to fix this all the nodes in the sub-tree of that node must change the radios they use to transmit, making this an expensive operation with a higher amount of beacons. Algorithm 1 shows the procedure that updates the father node and the ETX of a node. When a node needs to update its route, it executes this algorithm using its own routing tables.

Algorithm 1 Update Route

▷ Procedure used by a node u to update its father node and its ETX value
Scans routing table 1 and picks the neighbor $v1$ with the smallest ETX
SmallestETX1 = $v1$ ETX + link ETX $u \rightarrow v1$
Scans routing table 2 and picks the neighbor $v2$ with the smallest ETX
SmallestETX2 = $v2$ ETX + link ETX $u \rightarrow v2$
if (*SmallestETX1* $\geq$ *SmallestETX2*) **then**
    SmallestETX = SmallestETX2
else
    SmallestETX = SmallestETX1
if (*SmallestETX* < (*CurrentETX* − *Father changing threshold*)) **then**{
    Father = Best Neighbor
    CurrentETX = SmallestETX
    Timer = Minimum Value} ▷ Increases the beacon sending rate

6 EXPERIMENTS AND RESULTS

6.1 General Experiments

In order to test and compare the protocols CTP, CTP-Multi, DCTP-I and DCTP-A, two types of experiments were made. Longer ones with a duration of 45 minutes with a moderate packet generation rate and shorter ones with a duration of 5 minutes in which the packet generation rate was altered from experiment to experiment. Experiments measured throughput, latency, number of data packets sent in the network, number of beacons sent and delivery rate.

The experiments were made in the Twonet *testbed*, which is placed in a building of the University of Houston. It contains 100 Opal motes and is subjected to interference, specially with *Wi-Fi*. The 5 minutes interval was chosen because it is the shortest duration possible in the *testbed*, it allows the formation of the trees and the sending of a considerable amount of data packets. The 45 minutes interval was chosen to compare the protocols with a moderate packet generation rate and for a longer duration, in which

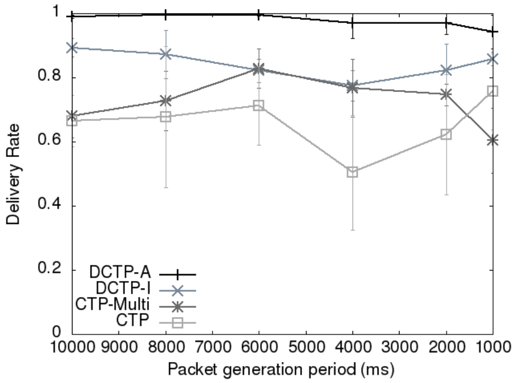

Figure 5: Delivery rate for packet generation period of 10,000 to 1,000 ms.

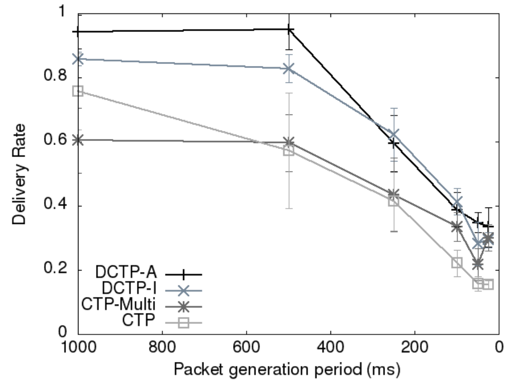

Figure 6: Delivery rate for packet generation period of 1,000 to 25 ms.

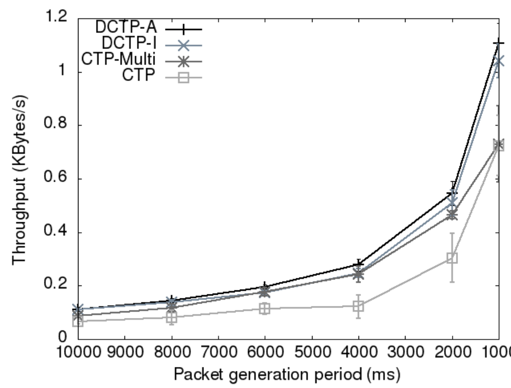

Figure 7: Throughput for packet generation period of 10,000 to 1,000 ms.

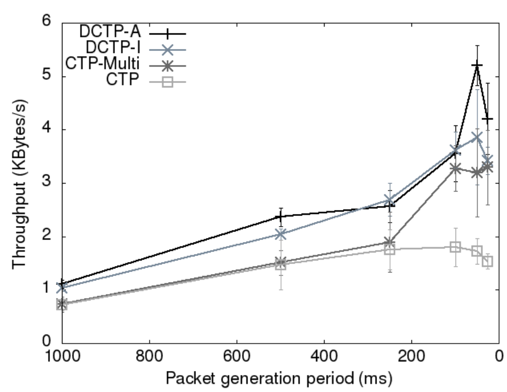

Figure 8: Throughput for packet generation period of 1,000 to 25 ms.

the network might suffer more changes than a shorter duration and be able to repair from them.

The 5 minutes experiments generate data packets at a constant rate defined by the packet generation period. This period varied from 10,000 milliseconds to 25 milliseconds, the shorter the period the higher the generation rate of packets. All of the 100 nodes of the *testbed* were used, but only half of them was in charge of generating data packets.

The short duration experiments were repeated 6 times and a confidence interval of 95%. The charts that show these results were divided into two, the first one shows from 10,000 milliseconds to 1,000 milliseconds and the second one shows from 1,000 milliseconds to 25 milliseconds, which is a more extreme situation for the protocols and with shorter intervals between the measurements. The protocols source code and the programs used to make experiments are available in a public repository in *GitHub*[1]. The delivery rate was calculated dividing the number of packets that arrived at the root by the number of data packets generated in the other nodes. When a node generates a packet it tries to send it, if it fails it tries again. Consequently the number of generated packets is not always the same. Duplicated packets are a problem, to minimize it every node was equipped with a packet cache.

All the protocols support multiple roots, which means that they can have more than one collection node. Using only one root makes the data collection harder, makes the trees deeper and, as a result, allows us to analyze the protocols in a harsher environment, in which the differences between them become more exposed than in a multiple roots environment. The use of more than one root is very advantageous, specially if the collection nodes are chosen in a way that almost the entire network is reached by a few multi-hop transmissions.

Figures 5 and 6 show the delivery rate of each protocol, which represents the reliability of each one. We can notice that DCTP-A is superior, achieving a delivery rate of almost a 100% until the 5,000 period and maintaining more than 90% until the 500 period (two packets generated per second in each node). It is followed by DCTP-I, which maintained rates between 80% and 90% until the 500 period. The protocols CTP and CTP-Multi started with a lower delivery rate, but CTP-Multi performed better. This can be explained by the fact that CTP-Multi has more possibilities to pick from and to avoid interference. After the 500 period, the delivery

[1]https://github.com/gabrielsluz/CTP-ALL

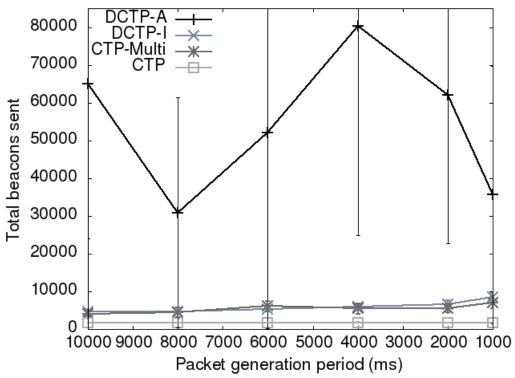

Figure 9: Total beacon sent for packet generation period of 10,000 to 1,000 ms.

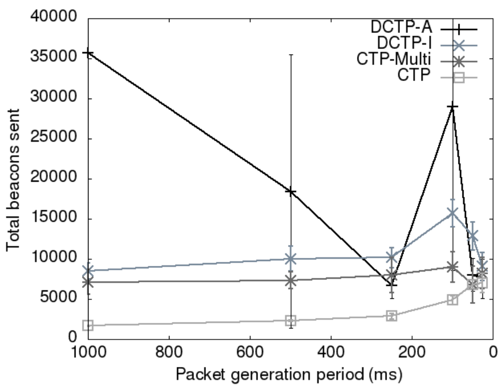

Figure 10: Total beacon sent for packet generation period of 1,000 to 25 ms.

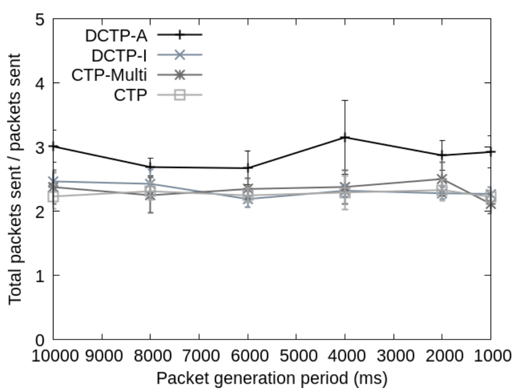

Figure 11: Total packets sent per packet generated for packet generation period of 10,000 to 1,000 ms.

rate in all protocols fall sharply as the network saturates, but the ones that were better remained superior.

Figures 7 and 8 show the throughput obtained by each protocol. The throughput corresponds to the number of data packets that

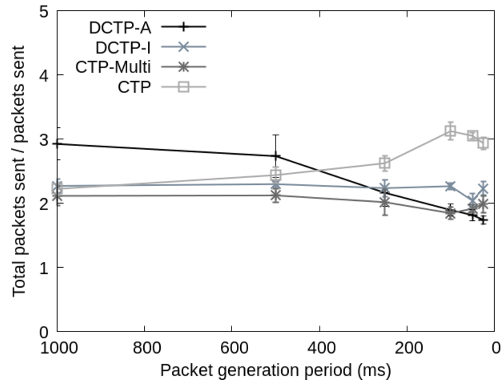

Figure 12: Total packets sent per packet generated for packet generation period of 1,000 to 25 ms.

reached the root divided by the time interval between the first and the last packet to arrive. It is important to point out that the graphics are on different scales, the throughput in the Figure 8 is much higher than the other. At the beginning (10,000) the throughput depends more on the packet generation rate than in the protocols, but shortly after the differences start to appear. The difference between DCTP-A and DCTP-I in relation to delivery rate is larger than in relation to throughput. The reason is that DCTP-A uses two radio bands to transmit, whereas the other only uses one, allowing DCTP-A to initiate a transmission through a radio while the other radio is still transmitting. CTP-Multi achieved a higher throughput than CTP, as expected due to the higher reliability of the dual radio protocol. The throughput of all protocols is improved til the mark of 50 milliseconds is reached, when it drops in all protocols.

The number of beacons sent is shown in Figures 9 and 10. This metric indicates the cost of maintaining the trees. The graphics reveal that DCTP-A sends, on average, much more beacons than the others. The higher cost of that protocol can be explained by the necessity of updating the entire sub-tree of a node when it changes its radio in order to avoid bottlenecks. A change in the radio used by a node closer to the root will cause that node to send beacons at a much higher rate and in order to propagate the change through the sub-tree all the nodes from it will change their radios and will send beacons at a higher rate. Consequently a change in the network caused by interference or by other reasons can cause a momentary but drastic increase in the beacon send rate, this explains the large variance of the results obtained. In some experiments the protocol operated on a low cost but in others the cost was much higher. DCTP-A minimum beacon send period was lower than the one used in the other protocols. The lowest cost protocol is CTP, followed by CTP-Multi. As DCTP-I uses two Routing components, it was expected to have a higher cost, but the difference between DCTP-I and CTP-Multi was small. For packet generation periods smaller than 1,000 milliseconds, DCTP-A number of beacons send lowered, while the other protocols kept on a similar rate to the other graphic.

Figures 11 and 12 are related to the protocols' efficiency, because it shows the number of data packets sent in the entire network divided by the number of data packets generated, which means that re-transmissions, packets forwarded and duplicated packets were counted, obtaining how many packets, in average, were necessary to send one generated packet to the root. For a period of more than 1,000 milliseconds, DCTP-A is the least efficient, while the others are very similar. However, due to the previous results, we know that DCTP-A performs better in relation to throughput and delivery rate. When the packet generation rate is increased, DCTP-A becomes more efficient.

The 45 minutes experiments were performed in the same way as the shorter ones, but with a packet generation rate fixed of 1,500 milliseconds and with a longer duration. The experiments were repeated 28 times for each protocol and used a 95% confidence interval. Table 1 shows the results for each protocol.

In relation to delivery rate, the results were similar to the 5 minutes experiments, DCTP-A maintained its superiority and, on average, CTP-Multi had a worse delivery rate than CTP, even though it was able to achieve a better throughput than the single radio protocol. We expected that CTP-Multi would obtain better reliability than CTP, but as it sent a larger volume of packets, obtaining a better throughput, it compensated the gains of radio diversity lowering its delivery rate. In relation to throughput, alternating radios and using two radios to send and receive achieved the best results.

The number of beacons also confirmed the results of the shorter experiments and shows the large variance of DCTP-A. The metric cost of each packet indicates the average number of packets necessary to send one generated data packet to the root, it counts packets forwarded, re-transmissions, duplicates and beacons, and is related to the protocols cost. We can notice that DCTP-A is the most costly protocol, due to the larger number of beacons and to the result obtained from the Figure 11. The cost of CTP-Multi is close to the cost of CTP, because the number of re-transmissions in CTP is bigger than the dual radio one.

Metric \ Protocol	DCTP-A	DCTP-I	CTP-MULTI	CTP
Delivery rate (%)	93.27 ± 3.57	82.89 ± 4.16	74.25 ± 4.32	74.53 ± 2.92
Throughput(KBytes/s)	0.673 ± 0.027	0.648 ± 0.037	0.565 ± 0.035	0.440 ± 0.020
Number of Beacons	63,051 ± 48,044	51,045 ± 5,153	36,293 ± 4,126	3,082 ± 241
Cost of each packet	4.286 ± 0.954	3.386 ± 0.155	2.844 ± 0.177	2.753 ± 0.157

Table 1: Results of the 45 minutes experiments. The confidence interval is of 95%

6.2 Latency Results

In a single experiment of 45 minutes the arrival times of the first 2,000 packets to reach the root were measured. To show these results we constructed a cumulative distribution function (CDF) for each protocol, exposed in Figure 13. The more vertical is the function curve, the more packets were received in less time, which means that the protocol with the most vertical function curve is the protocol with the lowest latency. Consequently, DCTP-A and DCTP-I achieved a very similar latency and were better than the others. CTP-Multi achieved a lower latency than CTP.

6.3 Robustness to Failure

We tested the robustness to failure of each protocol using 10 minutes experiments in which 9 nodes, of the 100 nodes, were turned

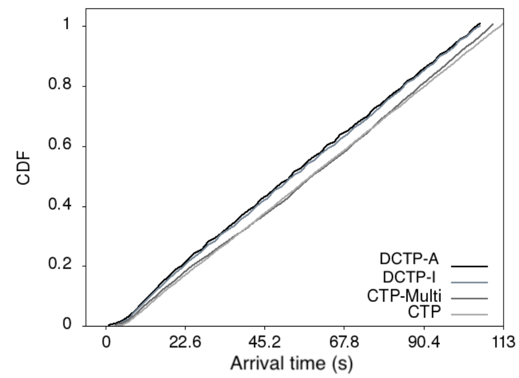

Figure 13: Cumulative distribution functions of the arrival times of the first 2,000 packets.

off at the fifth minute. We used a packet generation period of 1.000 ms and only half of the nodes were in charge of generating packets. The experiments were repeated 12 times for each protocol. At each minute all the packets generated in the network and all the packets received at the root were counted, we obtained the delivery rate of that minute in the experiment by dividing the number of packets generated by the number of packets received at the root. We calculated the average delivery of each minute and used a confidence interval of 95%, the results are exposed in figure 14.

The protocols DCTP-A and DCTP-I obtained better delivery rates than the other ones and, together with CTP, were not affected by the removal of the 9 nodes at minute 5. The only protocol that suffered heavily from the failure was CTP-Multi.

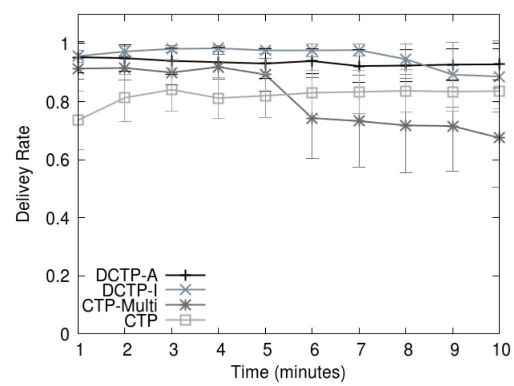

Figure 14: Delivery rate measured at each minute. Nodes fail at 5 minutes.

6.4 DCTP-I with Load Balancing

In an attempt to improve DCTP-I, we modified the Forwarding component of the protocol to use the best suited available radio to send a packet. In the original version DCTP-I a packet generated is sent by the same radio until it reaches the root, but in the modified version DCTP-I LB (Load balancing) a node can choose to send via another radio if the preferred one is busy. We compared DCTP-I

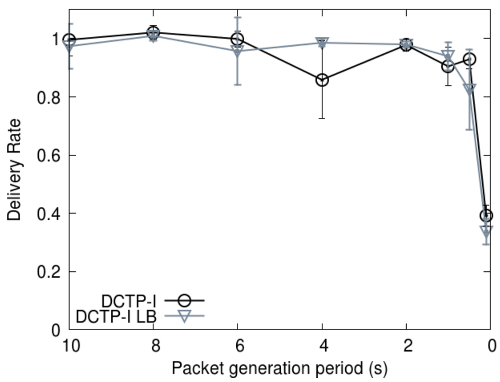

Figure 15: Delivery rate of DCTP-I with load balancing and DCTP-I for packet generation periods of 10 to 0.1 seconds.

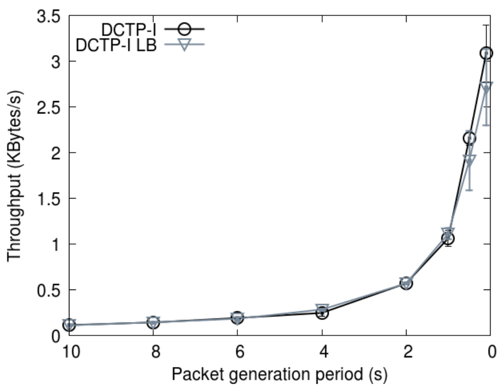

Figure 16: Throughput of DCTP-I with load balancing and DCTP-I for packet generation periods of 10 to 0.1 seconds.

LB with the original version using 5 minutes experiments, varying the packet generation period from 10 seconds to 0.1 second. We executed each protocol 10 times for each generation period and used a confidence interval of 95%. We evaluated the delivery rate (Figure 15), throughput (Figure 16) and number of beacons sent, in which the protocols had the same cost, and concluded that the modification did not improve the original version.

7 CONCLUSION AND FUTURE WORK

Based on the results from the experiments we can conclude that the new data collection protocols, DCTP-A and DCTP-I, were able to outperform, in delivery rate, throughput and latency, CTP and CTP-Multi, which are the current state-of-the-art in data collection. DCTP-A had the best performance, obtaining the best delivery rate and obtaining better but close results do DCTP-I in throughput and latency.

DCTP-I revealed to be a good alternative to DCTP-A, since it achieved good performance and does not have a cost too elevated. CTP-Multi obtained better routes than CTP, but its performance was exceeded by the new protocols, and the lower cost in relation to DCTP-I might not overshadow the benefits of the novel protocol. As a result, the new protocols were able to better explore the benefits of using two radio bands in data collection.

As future work we can investigate ways to maintain the performance and reduce the number of beacons. Another possible future work is to explore the use of multiple channels, which in addition to the use of two radio bands can have a great impact in reducing interference and, consequently, improving data collection.

ACKNOWLEDGEMENTS

The authors would like to thank the research agencies CAPES, CNPq and FAPEMIG for their financial support.

REFERENCES

[1] R. Anguswamy, M. Zawodniok, and S. Jagannathan. 2009. A Multi-Interface Multi-Channel Routing (MMCR) Protocol for Wireless Ad Hoc Networks. In *2009 IEEE Wireless Communications and Networking Conference*. 1–6. https://doi.org/10.1109/WCNC.2009.4917512

[2] G. Ekbatanifard, P. Sommer, B. Kusy, V. Iyer, and K. Langendoen. 2013. Fast-Forward: High-Throughput Dual-Radio Streaming. In *2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems*. 209–213. https://doi.org/10.1109/MASS.2013.41

[3] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. 2009. Collection Tree Protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*.

[4] Nildo Santos Ribeiro Júnior, Luiz Filipe Menezes Vieira, and Marcos Augusto Menezes Vieira. 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.

[5] Raja Jurdak, Kevin Klues, Brano Kusy, Christian Richter, Koen Langendoen, and Michael Brünig. 2011. Opal: A multiradio platform for high throughput wireless sensor networks. *Embedded Systems Letters, IEEE* 3, 4 (2011), 121–124.

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

[7] P Levis, N Patel, David Culler, and S Shenker. 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)* (01 2004), 15–28.

[8] Di Mu, Mo Sha, Kyoung-Don Kang, and Hyungdae Yi. 2019. Energy-Efficient Radio Selection and Data Partitioning for Real-Time Data Transfer.

[9] Bruna Peres, Bruno P. Santos, Otavio A. de O. Souza, Olga Goussevskaia, Marcos A. M. Vieira, Luiz F. M. Vieira, and Antonio A. F. Loureiro. 2018. Matrix: Multihop Address allocation and dynamic any-To-any Routing for 6LoWPAN. *Computer Networks* 140 (2018), 28 – 40. https://doi.org/10.1016/j.comnet.2018.04.017

[10] A. Phokaew, C. Tanwongvarl, and S. Chantaraskul. 2014. Adaptive multi-channel CTP for Wireless Sensor Networks. In *2014 International Electrical Engineering Congress (iEECON)*. 1–4. https://doi.org/10.1109/iEECON.2014.6925917

[11] Bruno P. Santos, Olga Goussevskaia, Luiz F.M. Vieira, Marcos A.M. Vieira, and Antonio A.F. Loureiro. 2018. Mobile Matrix: Routing under mobility in IoT, IoMT, and Social IoT. *Ad Hoc Networks* 78 (2018), 84 – 98. https://doi.org/10.1016/j.adhoc.2018.05.012

[12] B. P. Santos, M. A. M. Vieira, and L. F. M. Vieira. 2015. eXtend collection tree protocol. In *2015 IEEE Wireless Communications and Networking Conference (WCNC)*. 1512–1517. https://doi.org/10.1109/WCNC.2015.7127692

[13] Rodrigo C. Tavares, Marcos Carvalho, Marcos A. M. Vieira, Luiz F. M. Vieira, and Bhaskar Krishnamachari. 2018. FWB: Funneling Wider Bandwidth Algorithm for High Performance Data Collection in Wireless Sensor Networks. In *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM '18)*. ACM, New York, NY, USA, 9–16. https://doi.org/10.1145/3242102.3242112

[14] R. C. Tavares, M. A. M. Vieira, and L. F. M. Vieira. 2016. FlushMF: A Transport Protocol Using Multiple Frequencies for Wireless Sensor Network. In *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 192–200. https://doi.org/10.1109/MASS.2016.033

[15] Marcos Augusto M. Vieira, Claudionor N Coelho Jr., Diógenes Cecilio da Silva Jr., and José M da Mata. 2003. Survey on wireless sensor network devices. In *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA'03. IEEE Conference*, Vol. 1. 537–544.

[16] S. Yin, O. Gnawali, P. Sommer, and B. Kusy. 2014. Multi channel performance of dual band low power wireless network. In *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*. 345–353. https://doi.org/10.1109/MASS.2014.120