

Directional Transmissions and Receptions for High-throughput Bulk Forwarding in Wireless Sensor Networks

Ambuj Varshney
Uppsala University, Sweden
ambuj.varshney@it.uu.se

Luca Mottola
Politecnico di Milano, Italy and
SICS Swedish ICT, Sweden
luca@sics.se

Mats Carlsson
SICS Swedish ICT, Sweden
matsc@sics.se

Thiemo Voigt
Uppsala University, Sweden
SICS Swedish ICT, Sweden
thiemo@sics.se

ABSTRACT

We present DPT: a wireless sensor network protocol for bulk traffic that uniquely leverages electronically switchable directional (ESD) antennas. Bulk traffic is found in several scenarios and supporting protocols based on standard antenna technology abound. ESD antennas may improve performance in these scenarios; for example, by reducing channel contention as the antenna can steer the radiated energy only towards the intended receivers, and by extending the communication range at no additional energy cost. The corresponding protocol support, however, is largely missing. DPT addresses precisely this issue. First, while the network is quiescent, we collect link metrics across all possible antenna configurations. We use this information to formulate a constraint satisfaction problem (CSP) that allows us to find *two multi-hop disjoint paths* connecting source and sink, along with the corresponding antenna configurations. Domain-specific heuristics we conceive ameliorate the processing demands in solving the CSP, improving scalability. Second, the routing configuration we obtain is injected back into the network. During the actual bulk transfer, the source funnels data through the two paths by quickly alternating between them. Packet forwarding occurs deterministically at every hop. This allows the source to implicitly “clock” the entire pipeline, sparing the need of proactively synchronizing the transmissions across the two paths. Our results, obtained in a real testbed using 802.15.4-compliant radios and custom ESD antennas we built, indicate that DPT approaches the maximum throughput supported by the link layer, peaking at 214 kbit/s in the settings we test.

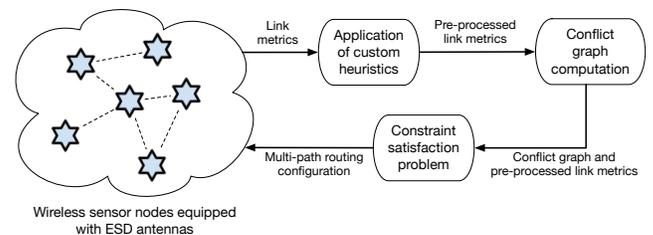


Figure 1: DPT operation.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Wireless Communication

Keywords

Directional antennas, Bulk data transmissions, Wireless sensor networks, Electronically controlled antennas

1. INTRODUCTION

Bulk traffic is arguably a frequent pattern in wireless sensor networks (WSNs), often found in scenarios such as structural health monitoring [4, 27, 36], wildlife monitoring [13] and volcano monitoring [47]. In these scenarios, tens of resource-constrained sensor nodes quietly monitor some event of interest. Upon recognizing such an event, they gather large amounts of data; for example, high-frequency acceleration samples, which are then sent over multiple hops to a powerful sink node for further processing.

Challenges. The literature includes several WSN protocols to support the bulk traffic pattern [9, 39]. Most of them are, however, based on omni-directional antennas, that is, transmissions are thought to spread the radiated energy equally in every direction. Differently, electronically switchable directional (ESD) antennas can concentrate the radiated energy only in given directions [21], using a software control to determine the direction of maximum antenna gain on a per-packet basis. This allows one to reduce channel contention by directing the antenna only towards the intended receivers and to increase the communication range with no impact on energy consumption. This antenna technology

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author(s). Copyright is held by the owner/author(s).

SenSys '15, November 1–4, 2015, Seoul, South Korea..

ACM 978-1-4503-3631-4/15/11.

DOI: <http://dx.doi.org/10.1145/2809695.2809720>.

recently proved to be viable also for WSN devices [34]: antennas like the one we use in this work can indeed meet the relevant energy and form factor constraints.

In principle, the features of ESD antennas would make them ideal to support bulk traffic. The reality, however, is that network designers are unable to reap the benefits of ESD antennas for bulk traffic because of the lack of protocol support. Existing solutions [9, 39] are indeed unsuited to operate with ESD antennas, as they simply do not consider antenna directionality. As a result, they cannot work outright without undergoing significant changes in their design.

Conceiving multi-hop wireless protocols using ESD antennas is challenging. For example, these antennas introduce additional complexity in topology construction and maintenance because of the large number of antenna configurations possible between nodes. Existing work using directional antennas, on the other hand, typically target traffic patterns other than those germane to WSNs, and are often limited to one-hop configurations [30, 31, 33]. The few attempts in WSNs are, however, limited to using directional antennas only when transmitting [32], hence missing out on part of the benefits brought by this antenna technology.

Moreover, when forwarding data over multiple hops, intra-path and inter-path interference significantly affects the attainable throughput [26]. Existing approaches to avoid intra-path interference use wireless channel diversity [35, 39] or separate interfering flows in time [26]. The former approaches are limited by the availability of interference-free channels, which are in fact difficult to identify in the increasingly crowded radio environments where WSNs are deployed [49]. Time-multiplexing the flows, on the other hand, is typically detrimental to the overall throughput.

Contribution. To tackle these challenges, we design DPT (Directional Pipelined Transmission): a multi-hop wireless protocol we expressly conceive to support bulk traffic using ESD antennas. At the core of DPT is the use of multiple disjoint paths for multi-hop routing and a custom approach to synchronize packet transmissions all the way up to the data sink. The protocol operation unfolds in several distinct steps, shown in Figure 1:

1. While the system is quiescent; for example, whenever nodes monitor their sensors awaiting an event [47], we gather link metrics for all possible antenna configurations. As soon as we probe every antenna configuration for a node, thus having complete information for a given device, these metrics are communicated to the sink. This process repeats periodically.
2. Upon a request of bulk transfer from any of the nodes, the sink first applies domain-specific heuristics to filter out link metrics likely immaterial to determining the routing configuration; next, it uses the processed link metrics to derive a model of the network as a function of antenna configurations and to determine the *conflict graph*. The latter identifies the packet transmissions that would interfere with each other in the absence of capture effect [29].
3. The link metrics and conflict graph are input to a constraint satisfaction problem (CSP) to identify a routing configuration with two node- and link-disjoint paths. Such configuration minimizes the overall cost of the routes according to the chosen link metric. The

domain-specific heuristics we apply at the previous step ease the processing demands in solving the specific problem instance at the sink, improving scalability.

4. We inject the solution returned by the CSP solver into the network, including the node-by-node paths and the corresponding antenna configurations. The source funnels data through the two paths by alternating between them, as shown in Figure 2. Along each path, we make packet forwarding occur within deterministic latencies [35]. As a result, each path is implicitly “clocked” by the source when injecting the packet. The sink operates opposite to the source, and alternates the receiving antenna configuration between the two downstream nodes. This works only if the two paths differ by an even number of hops, which we give as an additional constraint to the CSP solver.

The advantages of our solution are several: *i*) unlike the state of the art, it makes full use of directional transmission on *both* the transmitter and receiver end, harvesting the most benefits from ESD antennas; *ii*) it does not require multiple wireless channels to operate, which are increasingly hard to identify in many settings; and *iii*) it spares the need for a separate time-synchronization protocol to schedule transmissions, reducing the overhead.

These advantages materialize in a performance near the maximum throughput possible over the link-layer technology we use. Our results indicate that DPT can send bulk data at a maximum throughput of 214 Kbit/s and at a mean throughput of 203 Kbit/s, with a minimum packet reception rate of 90%. This performance holds as long as the state of the network at the time of collecting the link metrics remains comparable to that at the time of performing the bulk transfer, which is expected to be the case in most of the target application scenarios [4, 13, 27, 36, 47]. In such scenarios, indeed, links are stable over periods of hours [46]. Nonetheless, we also investigate how DPT’s performance changes whenever these conditions do not hold.

The paper proceeds as follows. We discuss related work in Section 2. Section 3 presents DPT’s design, whereas implementation details are briefly discussed in Section 4. Section 5 presents our experimental evaluation. Before concluding, we place DPT’s performance in the perspective of existing literature on bulk transfer protocols in Section 6,

2. BACKGROUND

We present background material to provide a foundation for the remainder of the paper.

2.1 Applications and Protocols

Applications characterized by bulk traffic include structural health monitoring [4, 27, 36], wildlife monitoring [13], and volcano monitoring [47]. In these scenarios, the need for high-throughput bulk transfers extends beyond the mere necessity of delivering data, as the protocol efficiency may also affect the data quality. For instance, Werner et al. [47] report that communicating to the sink a single volcanic event, which alone generates around 50 Kbytes of data per sensor node, may take so long to prevent the detection of many back-to-back seismic events.

When deploying these applications, the system is typically partitioned into a dense sensing network and a sparse backbone of forwarding nodes [2]. The latter are equipped

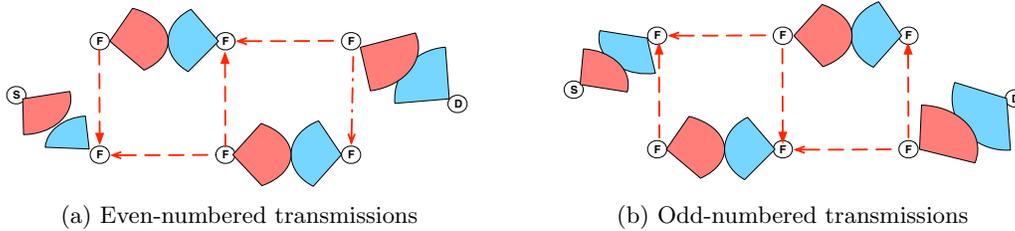


Figure 2: DPT: transmissions occur in parallel over the two paths at non-interfering links. *The nodes change antenna orientation based on which of the two paths is active. S indicates the source node, D indicates the sink, and F nodes are forwarding. Dashed arrows indicate possible interference at the receiving end. Darker (red) arcs indicate directional transmissions, lighter arcs (blue) are directional receptions.*

with external omni-directional antennas and power amplifiers, which allow them to cover hundred(s) of meters in a single hop, thus reducing the overall number of hops within the forwarding backbone [38]. Precisely because of this, the use of ESD antennas is appealing in these scenarios, in that they can increase communication range at the same energy cost, sparing the need for energy-consuming power amplifiers. In addition, the ability to reduce contention on the wireless channel might further reduce energy consumption by increasing the reliability of the individual transmissions.

Among existing bulk traffic protocols, many employ various forms of packet scheduling often combined with conditional immediate transmission (CIT) [35]: a forwarding mechanism that achieves high end-to-end throughput. For example, Flush [26] achieves a throughput of 4.8 Kbit/s by controlling the rate of transmissions to ensure packets on the same path do not interfere. PIP [39] uses CIT and multiple channels to reach a throughput of roughly 58.4 Kbit/s. Burst forwarding [12] shows lower throughput than PIP, but uses radio duty cycling and hence gains in energy efficiency. Zhang et al. use CIT and optimal scheduling to reach a throughput of up to 202.4 Kbit/s [48]. Chen et al. develop a collection protocol that uses multiple channels and multiple paths [5]. P3 [9] uses CIT, multiple channels, and synchronous transmissions [17] to forward packets using a subset of the nodes identified from global topology information.

In contrast to DPT, these approaches are all based on omni-directional antennas and may require multiple channels to operate. The latter is also a requirement for multi-radio platforms [14]. Given the crowded radio spectrum [49], this may become an issue: it will be increasingly difficult to identify multiple interference-free channels, and yet if only one of them is interfered, the whole protocol operation may suffer. For example, Doddavenkatappa et al. state “(if) one of the channels used in a pipeline is bad, the pipeline stalls and throughput degrades significantly” [9].

2.2 Directional Antennas

Directional antennas that can dynamically direct transmissions such as switched and adaptive beamforming are widely employed in cellular and ad-hoc networks. A significant body of work exists in this field [8]. Recent examples include the work by Arslan et al. who design an efficient Wi-Max beamformer antenna and implement real prototypes [1] as well as works on adaptive beamforming for indoor wireless LANs [7]. Liu et al. design a wireless network whose access points (AP) use phased array directional antennas to achieve high throughput in dense, indoor environments [30]. They further develop a distributed protocol called SPEED

and demonstrate in a testbed with directional antennas that it can lead to capacity improvement [31]. Along the same lines, Ramanathan et al. implement and evaluate a complete system solution for ad-hoc networks [40]. These kinds of antennas, however, are not suitable for sensor network applications; for example, because of physical size constraints.

Existing literature investigating the use of ESD antennas in sensor networks can be roughly divided into two categories, focusing on the design of either antenna prototypes or network protocols. The former works demonstrate the viability of ESD antennas, showing that current technology can meet the requirements of form factor and energy consumption. However, these works rarely assess the impact of the prototypes on the network performance. Specifically, Giorgetti et al. [19] assess the improvements in link performance with a prototype designed by combining four patch antennas. Viani et al. [44] present a design based on parasitic elements, and analyze its ability to reduce interference and to support localization. Parasitic elements are a common choice to reduce cost and size [7] and are also used by Nilsson [34] in the design of SPIDA, the specific ESD antenna we employ, further described in Section 4.

In the second category, protocols are often designed based on idealized antenna models that are rarely evaluated with real hardware [20, 41]. An exception is Felemban et al. [15], who also present a testbed evaluation of their neighbor discovery protocol. We use a modified version of this protocol to collect link metrics when the network is quiescent. Other works focus on specific network services considered in isolation, such as neighbor discovery [43] and MAC [16]. Mottola et al. demonstrate how slight modifications to an existing collection protocol designed for omni-directional antennas can lead to significant performance improvements using ESD antennas [32]. They use simulations based on an empirical link-layer model of the SPIDA antenna.

Overall, our work complements the existing literature by tackling the problem of employing ESD antennas for bulk traffic, an area where dedicated solutions are missing.

3. DPT: DIRECTIONAL PIPELINED TRANSMISSION

First, we provide an overview of the protocol operation. Next, we describe how we gather the necessary link metrics and how we pre-process them by applying custom heuristics to aid scalability. Then, we illustrate the CSP formulation [18] we use to identify the packet routes. We conclude by describing how the identified routing configurations reach the WSN nodes, and then used for bulk transmissions.

3.1 Overview

We adopt a centralized approach. Target applications usually feature a powerful sink to process the sensed data [38], which we leverage also to identify the routing configuration.

Many wireless sensor network applications requiring support for bulk transfers also feature large periods of inactivity between sensing cycles; for example, as in volcano monitoring [47]. As we describe in Section 3.2, we take advantage of such inactive periods to probe the wireless links across all possible antenna configurations. This step outputs a link metric depending on the given antenna configurations. For example, if an antenna features k possible directions of maximum gain, every link is in principle characterized by k^2 metrics, corresponding to every transmitter-receiver antenna configuration. However, due to directional transmissions and receptions, many of these combinations are unfeasible, since the nodes can only communicate when the sender and the receiver antenna lobes sufficiently overlap. We then funnel the link metrics to the sink. We use subsequent rapid floods with the antennas in omni-directional mode to this end, as in existing work [9].

Section 3.3 describes how the sink pre-processes the link metrics. First, the sink applies dedicated heuristics to simplify this information before using them to establish the routing paths. Indeed, we experimentally verify that out of the feasible transmitter-receiver antenna configurations, only a small portion is typically part of an efficient routing configuration. The remaining link metrics can be disregarded without significantly affecting the overall efficiency, yet greatly easing the processing demands at the sink, thus improving scalability. Next, the sink determines possibly interfering links, that is, pairs of links and corresponding antenna configurations that—if used simultaneously—would cause the transmissions to collide without the capture effect to possibly redress the situation [29]. We call this information *conflict graph*.

As described in Section 3.4, the link metrics and conflict graph are then fed as input to a CSP solver, which determines the two non-interfering paths based on a CSP formulation of the routing problem. This includes, among the stated constraints, the conditions that allow the two paths to be both link- and node-disjoint, as well as the restrictions on the difference in the length of the two paths. In principle, our approach would also work for more than two disjoint paths, but that would not provide further improvements, because: *i*) two paths are already sufficient to approach the maximum throughput supported by the link-layer, *ii*) at the source, packets cannot be generated faster than this. In a sense, with DPT it is the source’s ability to quickly inject packets in the pipelines that becomes the limitation.

The solution the CSP solver returns is then injected back to the network, where every node configures its directional antenna accordingly. We use subsequent network floods [17] with antennas set in omni-directional configuration to communicate these data to the nodes. As we illustrate in Section 3.5, at this point the source can start funneling data through the two paths, alternating the upstream node with the corresponding antenna configuration. Transmissions along the two paths will then occur in parallel until they join at the sink, which switches the receiving antenna configuration by alternating between the two downstream nodes.

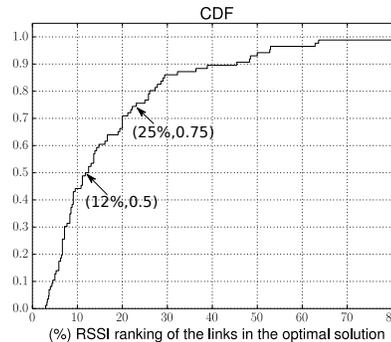


Figure 3: Ranking of antenna configurations based on RSSI values against whether the antenna configuration becomes part of an optimal solution. *Notably, 75% of the antenna configurations in an optimal solution lie within the top 25% antenna configurations, according to RSSI values, for the same link.*

3.2 Collecting Link Metrics

To provide the input to determine the two node- and link-disjoint paths, we need to gather the link metrics. Compared to networks with omni-directional antennas, the procedure to do so changes in the presence of ESD antennas. In principle, we need to probe every links for every possible antenna configuration.

To this end, we use a neighbor discovery protocol that employs a token-based approach, similar to SAND [15]. The token is released by the sink. A node holding the token performs the neighbor discovery by sending HELLO messages along every possible antenna configuration. The neighbors send replies in a time-slotted manner. Once a node finishes neighbor discovery, the collected link metrics are sent to the sink. We repeat the procedure across the entire network until a node requests a bulk transfer.

For every transmitter-receiver antenna configuration, as link metrics we estimate the packet delivery rate (PDR), as well as the mean signal strength (RSSI) of the received packets, as returned by the radio chip. To determine the PDR, we send a fixed number of REPLY messages along each antenna configuration in response to a single HELLO message. The PDR is used as a link metric to find disjoint paths with highest reliability. Differently, the RSSI values are used to identify transmissions that would collide if scheduled simultaneously, which is eventually represented as the conflict graph.

For every node, the sink attempts a round of neighbor discovery for a maximum number of times, by regenerating the token upon expiration of a timer. Beyond the maximum number of retries, the sink considers a node as failed and proceeds with the next one.

3.3 Pre-processing Link Metrics

In response to a request from a source to perform a bulk transfer, the sink pre-processes the collected link metrics before triggering the CSP solver. In doing so, the sink considers the last hour of link metric data, based on evidence found in the literature on the link dynamics in the scenarios we target [46]. This pre-processing includes two steps.

Applying heuristics. Based on experimental evidence, we recognize that not all the feasible transmitter-receiver antenna configurations are to be considered by the CSP solver.

We rather observe that only a small fraction of such antenna configurations tend to become part of the optimal solution, and that this fraction is identifiable beforehand.

Figure 3 verifies this observation, based on a dataset we collect from real-world experiments, further illustrated in Section 5. We obtain the chart by considering any two nodes connected by a link in the optimal solution, and by ordering the antenna configurations between these nodes according to the RSSI value. The X axis therefore indicates the percentage of antenna configurations with highest RSSI value out of all antenna configurations for a link. The Y axis plots the cumulative distribution function (CDF) of such antenna configurations in the optimal solutions, that is, what fraction of optimal solutions includes the corresponding percentage of antenna configurations with highest RSSI value.

The key insight from Figure 3 is that 75% of the antenna configurations for a link that is part of an optimal solution lie among the 25% antenna configurations with highest RSSI values for the same link. Further, we observe that almost half of the antenna configurations for a link that is part of an optimal solution lie within the 12% antenna configurations with highest RSSI values for the same link.

In essence, the plot tells us that the antenna configurations ultimately part of an optimal solution are likely those with highest RSSI, whereas antenna configurations corresponding to medium or low RSSI are rarely included. Based on this observation, we develop two heuristics to filter out likely immaterial antenna configurations. This reduces the size of the problem’s input, speeding up the execution of the CSP solver and hence improving scalability.

One possibility is to only keep the M antenna configurations with highest RSSI. We call this heuristic *top M configurations*. For example, if between two given nodes we initially find 20 feasible antenna configurations and $M = 5$, we only keep the 5 highest RSSI antenna configurations and discard all others. The value of M is experimentally derived from the same dataset used to create Figure 3. This approach should be especially effective for many outdoor scenarios where nodes are kept in line of sight, as in many of the applications we target [13, 47]. In these scenarios, the antenna orientation directed straight towards the receiver typically results in the highest signal strength [37].

Another possibility is to filter out antenna configurations only between the node pairs with a large number of feasible antenna configurations, so not to remove antenna configurations between nodes that enjoy fewer choices. To this end, we first compute the average number of feasible antenna configurations between any pair of nodes. Then, across the nodes with more feasible configurations than the average, we keep only the $N\%$ best configurations, based on RSSI values. We call this heuristic *weighted top $N\%$* .

For both heuristics we quantitatively measure, in Section 5, both the speed-up in processing time to identify a solution to the routing problem and their impact on a solution’s optimality. Indeed, both heuristics do not guarantee to retain an antenna configuration that would be included in the optimal solution. Running the risk of removing a useful antenna configuration is, however, the cost to pay for increased scalability in solving the CSP.

Computing the conflict graph. Notions akin to our conflict graph are widely used in wireless networks to identify interfering links [25, 30, 31]. In this work, we compute a form of conflict graph from the link metrics output by either

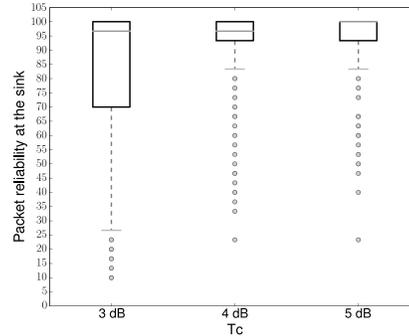


Figure 4: Packet reliability at the sink for different values of T_c . In the box plot, the top gray horizontal bar represents the median and the lower gray horizontal bar the minimum (excluding outliers) while the circles represent the outliers. The upper hinge of the box represents the 75 percentile, and the lower hinge represents the 25 percentile. A setting close to the co-channel interference tolerance level leads to high variance and low efficiency.

heuristic. We check, for every link in the network and associated antenna configurations, the RSSI value corresponding to every potentially concurrent transmission towards a common receiver r . If two links and associated antenna configurations are found to yield RSSI values whose difference is below a threshold T_c , the links and their antenna configurations are added to the conflict graph. We use T_c to determine whether capture effect [29] takes place and thereby either packet would be correctly received nonetheless.

The key question is how to set T_c . Using CC2420 radios, T_c is found to be around 3 dB for single transmissions, which equals the co-channel interference tolerance level of the radio chip [6]. However, RSSI values may vary over time also with directional antennas [45]. Setting T_c exactly to 3 dB when computing the conflict graph—which must remain valid for the duration of the bulk transfer—might lead to packet corruption, as the RSSI values of concurrent transmissions may temporarily fall below the radio’s co-channel interference tolerance level.

Figure 4 provides evidence of this phenomenon, based on an excerpt of the real-world experiments we discuss in Section 5. The chart shows that, as T_c increases, the system performs increasingly better in terms of packet reliability at the sink. Moreover, the variance between different experiments also decreases as T_c increases, as higher values of T_c offer added resilience to RSSI fluctuations. Care must be taken, however, in increasing the value of T_c . An excessively high value may needlessly reduce the solution space, excluding the use of links that would rather correctly operate simultaneously because of capture effect.

To complicate matters, we also observe that some of the links can be interfered by more than one concurrent transmission. In these situations, the combined signal strength of the interfering transmissions also increases, possibly requiring a further increase in T_c . To address this issue, we may adjust T_c as a function of how many links are possibly interfering, or modify the formulation of the CSP to prevent the concurrent use of links possibly interfered by more than one transmission. Both approaches, however, would increase the complexity of the processing at the sink and be detrimental to overall scalability.

Because of the considerations above, we set $T_c = 6$ dB and use this setting for the evaluation in Section 5. The evidence we collect in our experimental setting demonstrates that this configuration strikes a good balance between resilience to RSSI fluctuations, also in the presence of multiple interfering links, and the reduction of the solution space.

3.4 Optimization problem

Given the pre-processed link metrics and the conflict graph, we formulate the routing problem as a Constraint Satisfaction Problem (CSP) and attempt to optimally solve it.

We model the network as a digraph. The node set \mathcal{N} is composed of the sensor nodes with designated source node $o \in \mathcal{N}$ and sink node $d \in \mathcal{N}$. Let \mathcal{C} denote the configuration set of the transmitter antenna of every node (except the sink) as well as of the receiver antenna of every node (except the source). In our experiments, $|\mathcal{C}| = 6$, that is, for any given link, there are 36 combined configurations.

We set $w_{i,j,t,r}$ to be inversely proportional to the packet delivery rate on link (i, j) with antenna configuration (t, r) , and use an infinite cost for antenna configurations with insufficient connectivity. We model the conflict graph as a set \mathcal{F} of tuples $\langle i, j, t, r, i', j', t', r' \rangle$ where $i, j, i', j' \in \mathcal{N}$ and $t, r, t', r' \in \mathcal{C}$, denoting that link (i, j) with combined configuration (t, r) is in conflict with link (i', j') with combined configuration (t', r') if used at the same time.

Our goal is to find a disjoint path route from the source to the sink at minimal cost subject to specific constraints. Our CSP model uses the following decision variables, where $i \in \mathcal{N}$ and $p \in [1, 2]$ distinguishes the path:

$S_{i,p} \in \mathcal{N}$: successor of node i on path p , if i is on path p , otherwise i .

$T_{i,p} \in \mathcal{C}$: transmitter configuration of node i on path p , if i is on path p , otherwise 1.

$R_{i,p} \in \mathcal{C}$: receiver configuration of node i on path p , if i is on path p , otherwise 1.

$D_{i,p} \in \mathbb{N}$: distance in hops, at most $|\mathcal{N}| - 1$, from the source to node i on path p , if i is on path p , otherwise 0.

We are to find an assignment to these variables that minimizes the overall cost of the routing configuration, which is a function of the S, T, R variables:

$$\text{Minimize} \quad \sum_{i \in \mathcal{N} \setminus \{d\}, p \in [1, 2]} w_{i, S_{i,p}, T_{i,p}, R_{S_{i,p}, p}} \quad (1)$$

Whether the solution is admissible depends on the following constraints:

C1: There should be two link- and node-disjoint paths from the source to the sink.

C2: We must *not simultaneously* use pairs of links in the conflict graph.

C3: The length of the paths must differ by an even number.

Constraint **C1** guarantees that each node and each link is only part of one path while constraint **C2** ensures that concurrent transmissions do not collide destructively. Note that, however, constraint **C2** does not prevent the solution from using links in the conflict graph at different times. This

is intentional, in that it fosters higher parallelism throughout the two source-sink paths. On the other hand, without constraint **C3**, packets from the two disjoint paths would arrive simultaneously at the sink and collide. We now explain how the above constraints are modeled in our CSP.

Constraint **C1** is modeled by (2-7) in the following. In doing so, we use the successor representation of paths, as defined for $S_{i,p}$ variables. By using the convention that *i*) the path must originate from the source (2); *ii*) the source is the successor of the sink (3); and *iii*) any node that does not take part in a given path is its own successor, we have the necessary condition that both sets of successors are permutations of \mathcal{N} (4). Such permutation conditions are propagated very efficiently by CSP solvers. Link- and node-disjointness are straightforward to express (5). Equation (6) and (7) prevent cycles, by ensuring that all $D_{i,p}$ values are valid.

$$S_{o,1} \neq o \neq S_{o,2} \quad (2)$$

$$S_{d,1} = S_{d,2} = o \quad (3)$$

$$\{S_{i,1} \mid i \in \mathcal{N}\} = \{S_{i,2} \mid i \in \mathcal{N}\} = \mathcal{N} \quad (4)$$

$$\forall i \in \mathcal{N} \setminus \{o, d\} : S_{i,1} = i \vee S_{i,2} = i \quad (5)$$

$$D_{o,1} = D_{o,2} = 0 \quad (6)$$

$$\forall i \in \mathcal{N} \setminus \{d\}, \forall p \in [1, 2] : \quad (7)$$

$$D_{i,p} = \begin{cases} 0, & \text{if } S_{i,p} = i \\ D_{S_{i,p}, p} - 1, & \text{otherwise} \end{cases}$$

Constraint **C2** is modeled by (8), which needs to be transformed into a form suitable for the CSP solver, because the solver has no efficient way of expressing that a tuple is a member of the set. Thus, in a pre-solving phase, we transform (8) into an and-or-tree with equalities $x = v$ at the nodes, where x is a decision variable and v is an integer:

$$\forall i, j \in \mathcal{N} \setminus \{d\}, p, q \in [1, 2] : \quad (8)$$

$$(D_{i,p} + D_{j,q} + p + q) \bmod 2 = 0 \implies \langle i, S_{i,p}, T_{i,p}, R_{S_{i,p}, p}, j, S_{j,q}, T_{j,q}, R_{S_{j,q}, q} \rangle \notin \mathcal{F}$$

Constraint **C3** is straightforward to express:

$$(D_{d,1} + D_{d,2}) \bmod 2 = 0 \quad (9)$$

Finally, as is often the case in combinatorial optimization, the problem contains an obvious symmetry: any solution can be transformed to an equivalent solution by swapping the two paths. To avoid this useless degree of freedom, we break the symmetry with (10):

$$S_{s,1} < S_{s,2} \quad (10)$$

By using only a subset of the constraints, it is straightforward to get the optimal solution for a single path with the corresponding antenna configurations.

3.5 Bulk Transfer

Once the CSP solver identifies a solution to the routing problem, we encode the individual node and antenna configurations in dedicated control packets, which we inject into the network using rapid network floods [17] with the antennas in omni-directional mode. By doing so, every node

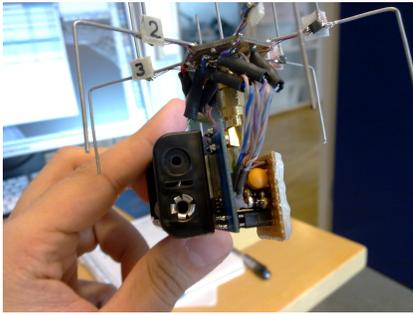


Figure 5: The ESD antenna we use in our experiments. The antenna is integrated with a standard TMote Sky node. Contiki drivers allow to control the direction of maximum gain in software.

on either path knows what is the receiving/transmitting antenna configuration as well as the identity of the upstream node, and prepares for the bulk transfer. We use a similar packet to trigger the actual bulk transfer at the source.

As hinted before, the source alternates between the two upstream nodes, funnelling data along either path. By doing so, the source implicitly “clocks” the entire pipeline: from now on, the forwarding transmissions at the intermediate nodes proceed with deterministic latency, using techniques similar to CIT [35]. At the sink, we apply the opposite mechanism, and alternate the antenna configurations between the two downstream nodes to receive data from either path¹.

As the bulk transfer proceeds, DPT continues to operate efficiently as long as the network state reflects the link metrics used to determine the two disjoint paths. Although this is expected to be the case in most of the applications we target [4, 13, 27, 36, 47]—where link qualities are observed to be stable over periods of hours [46]—we design a simple mechanism to find out whenever such conditions cease to hold and a routing reconfiguration is necessary.

Specifically, we allocate one byte per hop in the packet header to store RSSI information. At the intermediate forwarding nodes, we append the RSSI of the received packet before forwarding the packet further. On receiving the packet at the sink, we inspect the RSSI information collected at all hops along the forwarding path. The sink can then compare this information with that used earlier to determine the routing paths. In case of a significant mismatch, the sink temporarily stops the bulk transfer, re-initiates the process of collecting link metrics, computes a new optimal solution to the same CSP given the new input data, and resumes the transfer.

4. PROTOTYPE

We create a hardware/software prototype of DPT including custom-built ESD antennas. We use the prototype to perform a real-world evaluation, reported in Section 5.

Antenna. We use an ESD antenna similar to the one of Nilsson [34, 50], connected to a standard TMote Sky node that features a CC2420 radio [6]. The antenna has six par-

¹In the case of paths of unequal lengths, packets on the shorter path start to arrive earlier than on the longer path, as the pipeline in the latter needs more time to fill up. In this case, we only start to alternate antenna configuration at the sink once we have received a number of packets from the shorter path equal to the difference in the number of hops.

asitic elements surrounding a quarter wavelength monopole antenna. As a result of this design, the antenna gain varies as an offset circle from -4 to +7 dB in the horizontal plane. The six parasitic elements can be individually grounded or isolated using a software control. When all the parasitic elements are isolated, the antenna operates in omni-directional mode. If all the parasitic elements except one are grounded, the direction of the maximum antenna gain is towards the isolated element. This results in six directional configurations of the antenna, plus the omni-directional mode. The change of the direction of maximum gain happens within a few microseconds.

Contiki drivers allow a programmer to control the grounding or isolation of individual parasitic elements via software. The resulting antenna prototype is comparable to state-of-the-art dynamically steerable antennas available for WSNs [7, 19, 44]. We construct a number of these antennas for our experiments.

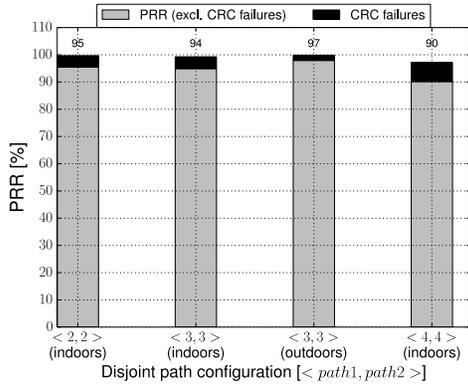
DPT implementation. We implement DPT on top of Contiki [11], along with a technique similar to CIT [35]. To further improve the throughput, we pipeline packet transmissions on the nodes. When we start to receive a packet and the start of frame delimiter (SFD) pin goes high, we send an already buffered packet from the microcontroller’s memory to the radio transmit buffer. As we finish receiving the packet and the SFD pin goes down, we immediately strobe the radio to transmit the packet we transferred to the radio transmit buffer in the meantime. As the radio prepares the packet for transmission, we change the antenna direction towards the upstream node.

We use fixed-size packets for pipelined transmissions, which enables fast and deterministic-time forwarding of the packet as well as ensures that we can meet the timing requirements for the capture effect. This requires that concurrent packet transmissions start within 160us of each other [28]. While the packet is being transmitted over the air, we copy the previously received packet from the radio’s receive buffer to the microcontroller’s memory. These techniques are also used by other high-throughput bulk transfer protocols [9, 39].

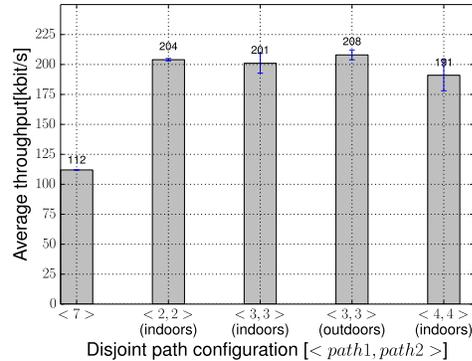
At the source node, when forwarding packets over the disjoint paths, we ensure that the packets are sent back-to-back with minimal delay. This is required to improve the throughput since the delay that occurs when loading packets into the radio transmit buffer at the source is lost once and for all, as it cannot be gained back. For this, we employ a technique similar to Glossy [17] to quickly load a packet at the source: as soon as we finish transmitting a packet, we immediately strobe the radio for the next transmission over the disjoint path. While the radio calibrates and begins the transmission of the preamble, we load the packet to be transmitted. As transmitting is slower than packet loading over the SPI bus that connects the radio to the microcontroller on the TMote Sky, we do not cause buffer under-runs.

The target applications can tolerate some packet loss [13, 47]. Thus, in our current implementation, we do not use link-layer acknowledgments, which improves the end-to-end throughput. If needed, one can send negative acknowledgments after the bulk transfer finished to retrieve lost packets from the source, as done in Flush [26].

CSP solver. We use MiniZinc 2.0 to solve the CSP. The process includes two phases: the flattening [18] of the MiniZinc model to a FlatZinc program, and the solving of the FlatZinc program. Flattening expands all quantified expres-



(a) PRR



(b) Throughput

Figure 6: PRR and throughput for paths of equal length. *DPT achieves a PRR above 90% and a maximum throughput of 214 Kbit/s, close to the theoretical maximum. The number of hops between source and sink have little impact on the performance.*

sions, nested logical expressions, and macros into a sequence of variable declarations and constraints. In Section 5, we report the processing times for the two phases separately. The flattening phase still presents room for improvements: increasing its efficiency is largely a matter of effective engineering. In contrast, the solving phase is difficult to further optimize, as the problem is inherently NP-hard.

5. EVALUATION

This section presents the experimental results we collect by running DPT in real-world testbeds we build, considered the absence of public testbed infrastructures featuring ESD antennas. Overall, our results indicate that:

- DPT approaches the maximum throughput allowed by the underlying link layer in most of the settings we test, reaching a maximum performance of 214 kbit/s;
- the routing configuration output by the solver remains usable, that is, the conflict graph still applies, much longer than the duration of a typical bulk transfer;
- our custom heuristics more than halve the solving times at the sink, with negligible impact on the optimality of the obtained solution;
- in most scenarios, particularly with a larger number of nodes, disjoint paths can be determined with high probability, confirming the feasibility of our approach.

The remainder of this section illustrates the details leading to the aforementioned results. Section 5.1 reports on the performance of DPT in throughput and packet reception rate (PRR). We study the effect on the link dynamics on our approach in Section 5.2, whereas Section 5.3 analyzes the efficiency of the solving process at the sink. Finally, Section 5.4 offers some considerations on how often disjoint paths do exist in a general network. Complementing this discussion, Section 6 places DPT’s performance in the larger perspective of existing literature on bulk transfer protocols.

Testbeds. We run experiments in a 32 sqm office environment as well as in an 80 sqm outdoor area in front of our institution’s main building. The office environment features a complex radio setting, with metallic objects in the corners and in the ceiling. Other interfering wireless networks, such as WiFi, Bluetooth, and appliances operating in the ISM 2.4 GHz band, are also abundantly present. This environment is harsh compared to that of our target applications, whose

Table 1: The number of different routing configurations we observe in our experiments. *We use $\langle a, b \rangle$ to denote path 1 of length a and path 2 of length b .*

$\langle 2,2 \rangle$	$\langle 3,3 \rangle$	$\langle 4,4 \rangle$	$\langle 4,2 \rangle$	$\langle 5,3 \rangle$
3	9	5	3	3

characteristics are better approximated by the cleaner radio environment we find outdoors. The results we report next for the office environment thus represent a worst case.

In both testbeds, we deploy up to twelve WSN nodes equipped with our ESD antennas. To create a multi-hop settings, we set the radio transmit power to -25 dBm. We use 802.15.4 channel 15, as it is the least interfered channel in the office environment. In the indoor experiments, we place the nodes randomly and differently for each experiment run, but always approximately one meter above the ground. In the case of outdoor experiments, the nodes are again randomly placed in a different way for every experiment run, approximately half a meter above the ground and in line of sight of each other.

5.1 Bulk Transfer Performance

We first analyze whether we achieve our primary goal, that is, achieving high-throughput bulk transfer.

Settings and metrics. Given a random placement of the nodes, we collect link metrics for about an hour and then trigger a bulk transfer of 350 Kbytes, using 123 bytes packets. Out of the 123 bytes, 1 byte is used to store the length of the packet, 2 bytes for the 802.15.4 frame check sequence field, and 5 bytes for control information. The remaining 115 bytes constitute the application payload. To calculate the throughput we consider all the application data transmitted within the duration of a bulk transmission, as done in related work [9, 39].

In our setup, and even more in real applications, it is difficult to control or to predict the shape of the routing solution output by the CSP solver. The experiments we run yield, as a result, different routing configurations with different frequencies. Table 1 summarizes how many occurrences of which routing configuration we observe in our experiments. The results include configurations where the disjoint paths have equal or different lengths. We analyze these separately, to understand the influence of the routing configuration on

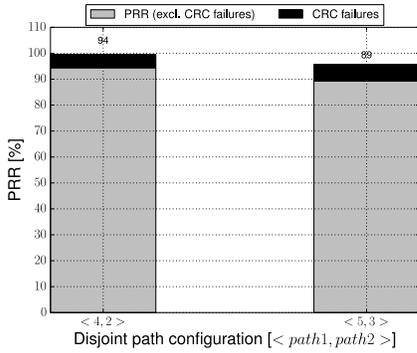


Figure 7: PRR for paths of unequal length. Paths of different lengths do not cause synchronization problems, as the performance remains similar to the case of equal length paths, shown in Figure 6(a). Similar considerations apply to throughput as well.

the synchronization along the parallel paths. We also investigate the impact of lossy links.

We take throughput and PRR as performance metrics, similar to existing literature on bulk transfer protocols [9, 39]. At the sink and at all intermediate nodes, we measure these quantities by keeping track of sequence numbers embedded within the packets, whether the received packet is corrupt, and the time when we start to receive the first and the last packet in each experiment. When computing both metrics, we exclude corrupted packets from the statistics, which then cover only useful application data even though approaches for recovering packets with bit-errors exist [22].

Equal length paths. Figure 6 shows DPT’s performance in throughput and PRR for different disjoint path configurations of equal length. For both metrics, an increasing number of hops has little impact on the performance. This provides evidence of the scalability of our approach against the depth of the network.

Figure 6(a) shows that in most settings, DPT receives packets with a PRR above 90%. We also observe that a small percentage of packets have CRC failures. This is likely due to external interference from other wireless devices in the office environment, as the fraction of packets with CRC failures is comparatively higher than in the outdoor setting. Considered the intense levels of interference in the office environment, nonetheless, we can maintain that their impact on DPT’s performance is arguably limited.

Figure 6(b) shows the mean and maximum throughput we measure in our experiments. Our results demonstrate that DPT achieves a maximum throughput of 214 Kbit/s, approximating the maximum link-layer throughput [35]. As we mentioned, the source cannot generate packets faster than said maximum. Because of this, DPT’s performance, as well as that of any other bulk transfer protocol based on the same radio technology, is upper-bound by the source’s ability to generate packets.

As a reference, Figure 6(b) also includes the throughput we achieve by operating DPT over a single path. As one would indeed expect in case the synchronization across the two paths would be accurate, the throughput when using two paths is almost twice that of the single path.

Unequal length paths. Figure 7 shows DPT’s performance in throughput and PRR in case the routing configuration includes paths of different length. The plot demon-

strates that DPT achieves a PRR above 90% in this case as well. Worth noticing is also that the PRR of the configuration with six nodes ($\langle 4, 2 \rangle$) is similar to the PRR of the configuration with the same number of nodes but equal length paths ($\langle 3, 3 \rangle$ in Figure 6(a)). The same holds for the configuration with eight nodes in Figure 7 ($\langle 5, 3 \rangle$) compared to the configuration with the same number of nodes and equal length paths ($\langle 4, 4 \rangle$ in Figure 6(a)). These observations provide evidence that even with disjoint paths of unequal length, the fast deterministic packet forwarding keeps the jitter at the forwarding nodes to the minimum and avoids synchronization problems that would lead to colliding transmissions. The same considerations apply to the throughput performance, which we do not show for brevity.

Lossy links. While the CSP solver finds the optimal disjoint paths, it is still possible that one of them includes a lossy link. The reason for this could simply be that there is indeed no better path available. To understand how this occurrence may affect the performance, we intentionally make one of the links drop packets randomly. Specifically, we consider two given disjoint paths: $source \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow sink$ and $source \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow sink$, and make node 5 drop 25% of the packets—chosen randomly—when it forwards packets to node 6.

Figure 8 shows the results of this experiment. Figure 8(a) shows the PRR on a per-node basis. As expected, the PRR at node 6 is lower than at other intermediate nodes, because of the losses on the link connecting node 5 to node 6. In turn, this affects the PRR at the sink. Similar considerations apply to the throughput figures, as Figure 8(b) demonstrates. These results show that the presence of a lossy link affects the performance in an appreciable manner.

We acknowledge this limitation of DPT, essentially caused by the absence of redundancy in the routing configuration. Nevertheless, there are ways to redress the situation. One possibility, for example, is to implement a local repair mechanism that, upon recognizing a lossy link, attempts to locally change the routing configuration based on inputs provided by the CSP solver, such as alternative upstream nodes that can be output together with the optimal solution.

5.2 Link Dynamics

In this experiment we investigate the effect of link dynamics. To this end, we collect the link metrics over a period of more than 24 hours in our indoor office environment on a working day using 10 ESD antennas. We process the link metrics over the past hour and compute the solution to the multi-path routing problem.

The results indicate that the optimal solution changes quite rapidly after it has been obtained, in particular during daytime when people move and more interference is present. This is expected in an indoor environment. Nevertheless, more important for DPT is the timescale at which the solution obtained remains *usable*, since this determines if we can successfully complete a bulk transfer. A solution is usable as long as the transmissions which happen simultaneously can leverage the capture effect on a given receiver, and thus do not collide. This holds true if the difference in the RSSI measures of all pairs of involved links, with given antenna configurations, are still above the threshold T_c .

To investigate this aspect, we choose an upper bound for the times it takes to solve the CSP, inject the solution into the network, and trigger the source to start the bulk trans-

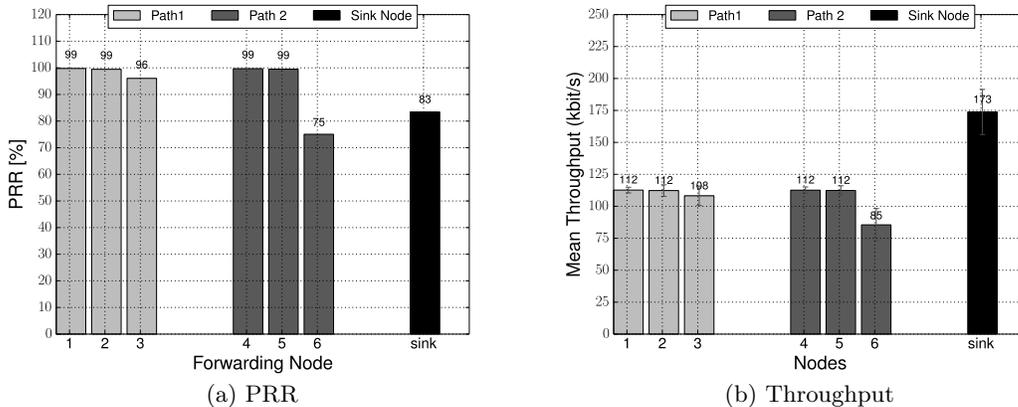


Figure 8: Throughput and PRR in the presence of a lossy link, in a configuration with eight nodes. To emulate a bad link, 25% of the packets are dropped randomly as they are forwarded from node 5 to node 6. The lossy link significantly affects the PRR and throughput.

Heuristic	Running time improvement (%)	No solution/Sub-optimal	β (avg)
Top M=5	57.09	2/2	57
Top M=10	34.37	0/0	81
Weighted top N=25%	44.33	0/0	76

Table 2: The performance of different heuristics in running times and impact on a solution’s optimality.

We indicate under “No solution/Sub-optimal” the number of times when no solution or a sub-optimal solution is found. A sub-optimal solution has a higher cost than the optimal solution according to Equation (1). Both heuristics provide drastic improvements in running time, whereas only using top M with $M = 5$ we observe an effect on the optimality of the routing configuration.

fer as described in Section 3.5. We select a conservative upper bounds of 5 minutes. Thus, we shift the collected link-metrics by an interval of 5 minutes and compute the solution at every such interval over the entire period of 24 hours. For each such solution, we investigate at which time scale the solution is still usable.

Results. Figure 10 depicts the result of the experiment. The figure indicates that the routing solutions found are usable during the duration of a typical bulk transfer. Further, it shows that for almost 50% of the solutions the disjoint path remains usable for a time period larger than 50 minutes, which suggests that in many cases the collection of links metrics can proceed leisurely.

5.3 CSP Solving

We study the efficiency of the solving process at the sink. First, we measure the running times without applying the heuristics of Section 3.3. We consider these figures as a baseline. Next, we measure the improvements brought by the heuristics against the impact in terms of optimality of the obtained routing configurations.

All measurements are taken using MiniZinc 2.0 on an Intel Core i7 at 3.5 GHz with 16 Gbytes of RAM running OS X 10.10.2. As input, we use the link metrics collected in thirteen real-world experiments among those used to understand the bulk transfer performance, reported in Section 5.1. This data set, albeit not vast, represents very diverse scenarios

in terms of shape and characteristics of the network topology. We thus maintain that the results we describe next are already conducive to valuable observations.

Results. Figure 9 depicts the results we collect. Based on our data set, the running times without applying any heuristics, are not prohibitive, yet still significant. Moreover, as discussed in Section 4, the process consists of two distinct phases, namely the flattening of the CSP formulation, and the solving itself. The flattening step greatly dominates the total running time—by orders of magnitude—despite the solving step being NP-hard, whereas the flattening step is not. This crucially indicates that the largest room for improvement is in the flattening step.

Confronted with these results, we devise the heuristics illustrated in Section 3.3 to reduce the size of the CSP input to the flattening step, aiming at lowering the overall running times. Figure 9 confirms that this strategy is indeed effective. The running times are often more than halved compared to the baseline. The gains also appear more marked at larger scales. In particular, for 12 nodes, top M with $M = 5$, that is, by keeping only the five strongest antenna configurations according to RSSI values between a pair of nodes, we can reduce the solving time to 58 seconds: a 66% improvement over the baseline.

When applying the heuristics, however, we may mistakenly filter out links that would be included in the optimal solution if they were available to the CSP solver. In turn, this would inevitably result in a sub-optimal output by the CSP solver. To understand this trade-off, we consider the difference in the value of the objective function in our CSP formulation, specified in Equation (1).

Moreover, even if two solutions have the same overall cost, they may still differ in the links or antenna configurations employed. To this end, we defined a metric β as the percentage of links and corresponding antenna configurations found in both solutions. In a sense, β indicates the extent of topological overlapping between two solutions; thus, $\beta = 100\%$ if two routing configurations are identical, whereas $\beta = 0\%$ if two solutions are different in all links or antenna configurations.

Table 2 presents the average results across all network topologies we test. The numbers indicate that our heuristics

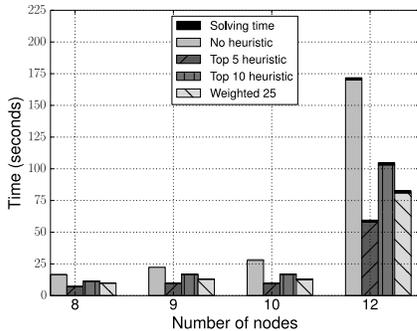


Figure 9: Running times of the solving process at the sink. The flattening step dominates the solving times. The heuristics reduce the running times by more than half, with greater improvements with more nodes.

do reduce the running times at the sink, but sometimes at the cost of sub-optimal routing configurations. For example, using *top M* with $M = 5$, we reduce the running times by 57%, on average. However, the optimality of the solution is partly affected: in two instances out of thirteen, the solution is sub-optimal, that is, Equation (1) evaluates to a higher value. Moreover, in other two instances, MiniZinc cannot find a solution, likely because the heuristic removes a link that is fundamental to identify the disjoint paths.

Increasing M to 10 reduces the improvement in running times to around 34%, but does *not* affect the optimality of the solution. Indeed, the cost returned by MiniZinc by solving Equation (1) is exactly the same with and without using *top M*. Interestingly, the routing configuration is not identical. As β equals 81, we conclude that some links or antenna configurations differ. Applying *weighted top N%* with $N = 25\%$ leads to similar conclusions, with better improvements in running time compared to *top M* with $M = 10$.

In conclusion, we argue that both heuristics are useful to the practical applicability of DPT. Crucially, both offer tunable parameters that designers may use as a knob to explore the trade-off between running times at the sink and optimality of the routing configuration.

5.4 Feasibility

DPT operates by forwarding packets over two node- and link-disjoint paths. A natural question is then to test how likely it is to find two such paths between a source and a sink in a general network. To find an answer, we use a synthetic network model generator we implement. This allows us to quickly test a sizable number of network topologies.

Our tool generates the network model by randomly distributing a configurable number of nodes over a limited area as follows. We place the initial node completely random. After placing a node, we place the next one at a distance from the previous node that follows a Gaussian distribution with a mean roughly three quarters of the communication range when our ESD antenna is set to omni-directional mode. Following the idea of the Poisson Disc distribution, we also enforce a minimum distance between the new node and all previously placed nodes.

When all nodes are distributed in the area, we compute the physical distance between each pair of nodes and assign it an omni-directional RSSI value based on the log-distance path loss model, using a path loss exponent of two, typical

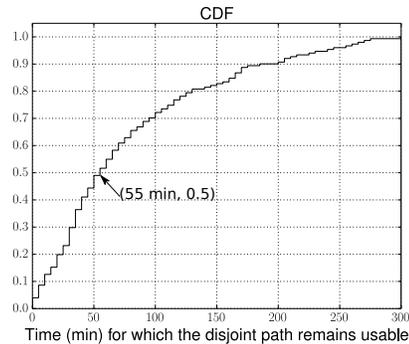


Figure 10: The timescale at which the disjoint path remains usable. For over 50% of the solutions the disjoint path is usable for over 55 minutes.

of free space [23, 42]. Given a directional antenna configuration for the two nodes, we manipulate the omni-directional RSSI based on empirical values that we have gathered during our experiments. For the antenna configurations in the direction of maximum gain, we increase the directional RSSI compared to the omni-directional value, and vice versa in the opposite antenna direction. Based on this network model, we construct the conflict graph and input it to the solver, as if the link metrics were collected from a real network. The solver then searches for the optimal solution to CSP formulation in Section 3.4.

Results. We check networks of 12 and 18 nodes. For these networks, disjoint paths form in more than 60% and 80% of the 50 simulations, respectively. The evidence shows that the 1-hop neighbors of the sink and of the source are crucial. If these do not offer sufficient options to form the two node- and link-disjoint paths, finding a solution to the CSP becomes difficult. On the other hand, if the network is sufficiently dense, there are typically more than enough options in the more central parts of the network.

While these results are based on random placement, real deployments rarely consist of randomly-scattered nodes. Nodes are rather placed carefully to cover the events of interest and/or to obtain a well-connected network [38]. In this regard, we believe that existing tools for pre-deployment, such as Scale [3] and Trident [24], can be fruitfully employed to ensure that disjoint paths can be formed in most cases; for example, by also checking how the network topology changes by switching to a different radio channel. The fallback solution, always available with DPT, is to use just one path.

6. DISCUSSION

An experimental comparison of DPT against existing protocols for bulk transmissions, such as P3 [9] or PIP [39], is hardly feasible. These protocols are indeed developed and evaluated on testbeds with omni-directional antennas. They would then need significant redesigns to work with ESD antennas. Moreover, even if such re-designs were possible, the protocol logic would fundamentally mismatch the antenna operation. In simpler terms, it would be a case of “fitting a square peg into a round hole”.

What we can do, however, is to place DPT’s performance into the larger perspective of the state of the art, as reported in the relevant literature. We discuss this on two accounts: throughput and energy consumption.

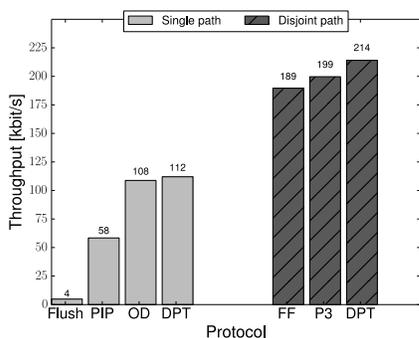


Figure 11: Maximum throughput of DPT and existing bulk transfer protocols. The performance figures for the latter are taken from literature. The absolute numbers suggest that DPT outperforms existing protocols in terms of maximum performance, while using only one channel.

Throughput. Figure 11 reports the maximum throughput figures for DPT and existing bulk transfer protocols, as found in the literature.

The left part of the chart shows the throughput of DPT and of other bulk transfer protocols when operating on a single path. The highest throughput found in the literature using this configuration is that of Österlind and Dunkels [35], indicated as OD, who achieve 108.8 kbit/s. This is 97% of the upper bound with 802.15.4 radios over single paths and multiple hops [35]². Flush [26] is considerably slower with a throughput of only 4.8 kbit/s. By using a custom TDMA mechanism, multiple channels, and CIT, PIP [39] achieves a throughput of 58.4 kbit/s, which corresponds to 52% of the upper bound. DPT, on the other hand, achieves a throughput slightly higher than that of Österlind and Dunkels. The reason for this is that switching radio channel, as they do, takes slightly more time than reconfiguring our ESD antenna, hence the higher throughput.

Using multiple paths, as in the protocols on the right part of Figure 11, the source can inject packets into the network as quickly as possible. Hence, the maximum throughput can approach the theoretical radio bandwidth, barring any processing delay between consecutive packets. With this configuration, the highest throughput reported in the literature is that of P3 [9], which attains 199.68 kbit/s. Ekbatani-fard et al.’s FastForward [14] leverages a dual-radio platform, which in some sense also uses two paths. With this technique, they achieve a throughput of 189.6 kbit/s. Our approach achieves a maximum throughput of 214 kbit/s, hence markedly higher than existing protocols.

Energy consumption. Table 6 reports the energy consumed by DPT, P3, and PIP in a bulk transfer of 100 Kbytes across 4 hops, as found in the literature for the latter two.

In these settings, these protocols differ in throughput and number of nodes required for the transfer. For example, DPT’s two disjoint paths require together eight nodes, while PIP uses five nodes on a single path. P3 makes use of a modified form of Glossy floods [17], which results in employing way more nodes than both PIP and DPT. In a 4 hops setting, P3 uses approximately 26 nodes [9].

²The upper bound of 112.5 kbit/s is simply half of the theoretical radio bandwidth, due to the half-duplex operation.

Metric	DPT	P3	PIP
throughput (kbit/s)	203	199	58
nodes employed	8	26	5
per-node energy (mJ)	236.5	241.2	810
network-wide energy (mJ)	1891.6	6271.1	4050
energy ratio to DPT	1	3.32	2.14

Table 3: Energy consumption for transferring 100 Kbytes of data over 4 hops using DPT and existing bulk transfer protocols. The performance figures for the latter are taken from literature. DPT outperforms both P3 and PIP in network-wide energy consumption.

All three protocols, nonetheless, keep the radio always on during a bulk transfer. By making the established assumption of a 60 mW power consumption when the radio is turned on [10], DPT and P3 would enjoy a lower per-node energy consumption compared to PIP. This is essentially because, due to higher throughput, the bulk transfer takes less time in DPT and P3, roughly 4 seconds for 100 Kbytes of data.

On a network-wide scale, however, P3’s use of Glossy floods is detrimental to the overall energy figures. By employing more nodes than DPT for the same bulk transfer, a system running P3 would consume more than three times the energy. This is essentially the price to pay for the added resilience provided by Glossy floods. As for the network-wide energy consumption of PIP, this is still more than twice that of DPT. This time, the result is due to the larger latency to complete the transfer, which forces nodes to keep their radios on much longer.

Note that, as in P3 and PIP, we do not consider the energy cost to acquire the topology information necessary to setup the bulk transfer, as in P3, or to establish the multi-hop path from source to sink in a distributed fashion, as in PIP. Essentially, the problem of topology discovery is orthogonal to that of achieving energy-efficient high throughput, and dedicated solutions are needed.

7. CONCLUSION

Directional antennas reduce contention while extending the communication range. In this paper, we use these features to devise DPT, the first high-throughput bulk transfer protocol that leverages ESD antennas. We transmit data over two multi-hop disjoint paths that we find by solving a CSP formulation of the routing problem at hand. Our results, obtained in a testbed using 802.15.4 radios and ESD antennas, indicate that DPT achieves a maximum throughput of 214 kbit/s, close to the maximum supported by the link layer, in the settings we test.

Acknowledgements

This work was partly supported by the Swedish Research Council, VINNOVA, SSF, and the Technological Cluster Project “Zero-energy Buildings in Smart Urban Districts” (EEB) of the Italian Ministry for University and Research.

8. REFERENCES

- [1] M. Y. Arslan, K. Sundaresan, S. V. Krishnamurthy, and S. Rangarajan. Design and implementation of an integrated beamformer and uplink scheduler for ofdma femtocells. In *13th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MOBIHOC '12, pages 215–224, 2012.
- [2] M. Buevich, D. Schnitzer, T. Escalada, A. Jacquiau-Chamski, and A. Rowe. Fine-grained remote monitoring, control and pre-paid electrical service in rural microgrids. In *13th International Symposium on Information Processing in Sensor Networks*, IPSN '14, pages 1–12, 2014.
- [3] A. Cerpa, N. Busek, and D. Estrin. Scale: A tool for simple connectivity assessment in lossy environments. Technical report, Center for Embedded Network Sensing, 2003.
- [4] K. Chebrolu, B. Raman, N. Mishra, P. K. Valiveti, and R. Kumar. Brimon: A sensor network system for railway bridge monitoring. In *6th International Conference on Mobile Systems, Applications, and Services*, MOBISYS '08, pages 2–14, 2008.
- [5] Y. Chen, P. H. Gomes, and B. Krishnamachari. Multi-channel data collection for throughput maximization in wireless sensor networks. In *11th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, MASS '14, pages 443–451, 2014.
- [6] Chipcon AS. *CC2420 datasheet - 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver (Rev. B)*.
- [7] R. Choudhury, T. Ueda, J. Bordim, and N. Vaidya. Beamnet: An ad hoc network testbed using beamforming antennas. In *Vehicular Technology Conference*, 2005.
- [8] H. Dai, K.-W. Ng, M. Li, and M.-Y. Wu. An overview of using directional antennas in wireless networks. *International Journal of Communication Systems*, 26(4):413–448, 2013.
- [9] M. Doddavenkatappa and M. C. Chan. P3: A practical packet pipeline using synchronous transmissions for wireless sensor networks. In *13th International Symposium on Information Processing in Sensor Networks*, IPSN '14, pages 203–214, 2014.
- [10] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes. Powertrace: Network-level power profiling for low-power wireless networks. Technical Report T2011:05, SICS Swedish ICT, Mar. 2011.
- [11] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, LCN '04, pages 455–462, 2004.
- [12] S. Duquennoy, F. Österlind, and A. Dunkels. Lossy links, low power, high throughput. In *9th ACM Conference on Embedded Networked Sensor Systems*, SENSYS '11, pages 12–25, 2011.
- [13] V. Dyo, S. A. Ellwood, D. W. Macdonald, A. Markham, C. Mascolo, B. Pásztor, S. Scellato, N. Trigoni, R. Wohlers, and K. Yousef. Evolution and sustainability of a wildlife monitoring sensor network. In *8th ACM Conference on Embedded Networked Sensor Systems*, SENSYS '10, pages 127–140, 2010.
- [14] G. Ekbatanifard, P. Sommer, B. Kusy, V. Iyer, and K. Langendoen. Fastforward: High-throughput dual-radio streaming. In *Proceedings of the 10th International Conference on Mobile Ad-Hoc and Sensor Systems*, MASS '13, pages 209–213, 2013.
- [15] E. Felemban, R. Murawski, E. Ekici, S. Park, K. Lee, J. Park, and Z. Hameed. Sand: Sectorized-antenna neighbor discovery protocol for wireless networks. In *7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks*, SECON '10, pages 1–9, 2010.
- [16] E. Felemban, S. Vural, R. Murawski, E. Ekici, K. Lee, Y. Moon, and S. Park. Samac: A cross-layer communication protocol for sensor networks with sectorized antennas. *IEEE Transactions on Mobile Computing*, 9(8):1072–1088, Aug. 2010.
- [17] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with glossy. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks*, IPSN '11, pages 73–84, 2011.
- [18] T. Feydy, Z. Somogyi, and P. Stuckey. Half reification and flattening. In *Principles and Practice of Constraint Programming CP 2011*, volume 6876 of *Lecture Notes in Computer Science*, pages 286–301. Springer Berlin Heidelberg, 2011.
- [19] G. Giorgetti, A. Cidonali, S. Gupta, and G. Manes. Exploiting low-cost directional antennas in 2.4 ghz ieee 802.15.4 wireless sensor networks. In *European Microwave Week*, 2007.
- [20] H. Gossain, T. Joshi, C. D. M. Cordeiro, and D. P. Agrawal. Drp: An efficient directional routing protocol for mobile ad hoc networks. *IEEE Trans. on Parallel & Distributed Systems*, 17(12):1438–1541, 2006.
- [21] R. C. Hansen. *Electrically small, superdirective, and superconducting antennas*. John Wiley & Sons, 2006.
- [22] F. Hermans, H. Wennerström, L. McNamara, C. Rohner, and P. Gunningberg. All is not lost: Understanding and exploiting packet corruption in outdoor sensor networks. In *11th European Conference on Wireless Sensor Networks - Volume 8354*, EWSN 2014, pages 116–132, 2014.
- [23] S. L. Howard, C. Schlegel, and K. Iniewski. Error control coding in low-power wireless sensor networks: When is ecc energy-efficient? *EURASIP Journal on Wireless Communications and Networking*, 2006(2):29–29, 2006.
- [24] T. Istomin, R. Marfievici, A. L. Murphy, and G. P. Picco. Trident: In-field connectivity assessment for wireless sensor networks. In *ExtremeCom*, 2014.
- [25] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *9th Annual International Conference on Mobile Computing and Networking*, MOBICOM '03, pages 66–80, 2003.
- [26] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica. Flush: A reliable bulk transport protocol for multihop wireless networks. In *5th International Conference on Embedded Networked Sensor Systems*, SENSYS '07, pages 351–365, 2007.

- [27] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *6th International Conference on Information Processing in Sensor Networks*, IPSN '07, pages 254–263, 2007.
- [28] O. Landsiedel, F. Ferrari, and M. Zimmerling. Chaos: Versatile and efficient all-to-all data sharing and in-network processing at scale. In *11th ACM Conference on Embedded Networked Sensor Systems*, SENSYS '13, pages 1:1–1:14, 2013.
- [29] K. Leentvaar and J. Flint. The capture effect in fm receivers. *IEEE Transactions on Communications*, 24(5):531–539, 1976.
- [30] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste. Dirc: Increasing indoor wireless capacity using directional antennas. *SIGCOMM Comput. Commun. Rev.*, 39(4):171–182, Aug. 2009.
- [31] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste. Pushing the envelope of indoor wireless spatial reuse using directional access points and clients. In *16th International Conference on Mobile Computing and Networking*, MOBICOM '10, pages 209–220, 2010.
- [32] L. Mottola, T. Voigt, and G. Picco. Electronically-switched directional antennas for wireless sensor networks: A full-stack evaluation. In *10th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, SECON '13, pages 176–184, 2013.
- [33] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-Giel, and S. Das. Mobisteer: Using steerable beam directional antenna for vehicular network access. In *5th International Conference on Mobile Systems, Applications and Services*, MOBISYS '07, pages 192–205, 2007.
- [34] M. Nilsson. Directional antennas for wireless sensor networks. In *Proc. 9th Scandinavian Workshop on Wireless Adhoc Networks (Adhoc'09)*, 2009.
- [35] Österlind F. and Dunkels A. Approaching the maximum 802.15.4 multi-hop throughput. In *5th ACM Workshop on Embedded Networked Sensors*, EMNETS '08, 2008.
- [36] J. Paek, K. Chintalapudi, R. Govindan, J. Caffrey, and S. Masri. A wireless sensor network for structural health monitoring: Performance and experience. In *2nd IEEE Workshop on Embedded Networked Sensors*, EMNETS '05, pages 1–9, 2005.
- [37] B. Raman and K. Chebrolu. Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks. In *11th Annual International Conference on Mobile Computing and Networking*, MOBICOM '05, pages 156–169, 2005.
- [38] B. Raman and K. Chebrolu. Sensor networks: A critique of "sensor networks" from a systems perspective. *SIGCOMM Comput. Commun. Rev.*, 38(3):75–78, July 2008.
- [39] B. Raman, K. Chebrolu, S. Bijwe, and V. Gabale. Pip: A connection-oriented, multi-hop, multi-channel tdma-based mac for high throughput bulk transfer. In *8th ACM Conference on Embedded Networked Sensor Systems*, SENSYS '10, pages 15–28, 2010.
- [40] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit. Ad hoc networking with directional antennas: a complete system solution. *IEEE Journal on Selected Areas in Communications*, 23(3):496–506, 2005.
- [41] S. Roy, D. Saha, S. Bandyopadhyay, T. Ueda, and S. Tanaka. A network-aware mac and routing protocol for effective load balancing in ad hoc wireless networks with directional antenna. In *4th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MOBIHOC '03, pages 88–97, 2003.
- [42] N. Scholastica. Pathloss prediction model of a wireless sensor network in an indoor environment. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 3(9), 2014.
- [43] S. Vasudevan, J. Kurose, and D. Towsley. On neighbor discovery in wireless networks with directional antennas. In *IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, INFOCOM '05, pages 2502–2512 vol. 4, 2005.
- [44] F. Viani, L. Lizzi, M. Donelli, D. Pregolato, G. Oliveri, and A. Massa. Exploitation of Parasitic Smart Antennas in Wireless Sensor Networks. *Journal of Electromagnetic Waves and Applications*, 24(7):993–1003, 2010.
- [45] T. Voigt, L. Mottola, and K. Hewage. Understanding link dynamics in wireless sensor networks with dynamically steerable directional antennas. In *10th European Conference on Wireless Sensor Networks*, EWSN'13, pages 115–130, 2013.
- [46] H. Wennerstrom, F. Hermans, O. Rensfelt, C. Rohner, and L.-A. Norden. A long-term study of correlations between meteorological conditions and 802.15.4 link performance. In *10th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, SECON '13, pages 221–229, 2013.
- [47] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. In *7th Symposium on Operating Systems Design and Implementation*, OSDI '06, pages 381–396, 2006.
- [48] H. Zhang, F. Osterlind, P. Soldati, T. Voigt, and M. Johansson. Rapid convergecast on commodity hardware: Performance limits and optimal policies. In *7th IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks*, SECON '10, pages 1–9, 2010.
- [49] G. Zhou, J. A. Stankovic, and S. H. Son. Crowded spectrum in wireless sensor networks. In *3rd IEEE Workshop on Embedded Networked Sensors*, EMNETS '06, 2006.
- [50] E. Öström, L. Mottola, and T. Voigt. Evaluation of an electronically switched directional antenna for real-world low-power wireless networks. In *Real-World Wireless Sensor Networks*, Lecture Notes in Computer Science, pages 113–125. Springer Berlin Heidelberg, 2010.