Directional Antennas for Convergecast in Wireless Sensor Networks: Are They a Good Idea?

Giovanni Tarter
University of Trento, Italy
Email: giovanni.tarter@gmail.com

Luca Mottola
Politecnico di Milano, Italy and SICS Swedish ICT
Email: luca.mottola@polimi.it

Gian Pietro Picco
University of Trento, Italy
Email: gianpietro.picco@unitn.it

***Abstract*—Directional antennas improve network performance by increasing the communication range and alleviating contention as proven, e.g., in cellular and ad-hoc networks. In principle, one may reap similar benefits in wireless sensor networks (WSNs), where energy concerns and reliability requirements make this antenna technology even more desirable. However, it is unclear how the shortcomings of directional antennas, e.g., increased likelihood of hidden terminals, affect WSNs. We *quantitatively* study these aspects for *convergecast*, a staple network functionality popular in WSN applications, e.g., for data collection. The integration of directional communication in convergecast protocols is non-trivial: probing wireless links between neighboring nodes is no longer feasible with single broadcast transmissions, as the antenna configuration depends on the target neighbor. This bears a great impact on the efficiency in building and maintaining the routing topology. We perform our study in simulation, based on an empirical model of an existing antenna prototype. This allows us to explore the parameter space efficiently yet realistically; a goal otherwise impossible without several antenna prototypes that, unlike WSN motes, are not readily available. Our results point to a negative answer; directional antennas, when used for WSN convergecast, provide limited benefits, appreciable only when certain specific conditions are met.**

I. Introduction

> *Research papers usually unfold by first eliciting a problem, then describing its solution, and finally presenting evidence that the solution improves over existing ones. This paper is different. We* quantitatively *demonstrate that directional antennas do* not *play well with the dominating design of convergecast. Our aim is to inspire work in directions* other *than those we unsuccessfully attempted.*

Directional antennas can concentrate the radiated energy only in given directions, typically using a software control to determine the direction of maximum gain on a per-packet basis. Because of the broadcast nature of wireless communications, this ability may provide key benefits; for example, by alleviating channel contention in directions other than the intended receiver, and by increasing the communication range.

Directional antennas are commonplace in cellular and ad-hoc networks. The benefits they offer apply in principle also to wireless sensor networks (WSNs); however, their application to WSNs is comparatively limited, as we illustrate in Section II. Directional antenna prototypes apt to WSNs do exist; however, protocols taking advantage of directional transmissions are often based on abstract antenna models, which hardly match the behavior of real-world prototypes.

To bridge this gap, this paper reports on our attempts at employing directional antennas in a tree-based convergecast WSN protocol. We focus on the staple MultihopLQI [26], described in Section III. Convergecast is arguably the most common traffic pattern in WSNs, as it underpins many applications including data collection [12]. MultihopLQI is one of the most stable convergecast protocols; it is both widely employed in real deployments and amenable to be modified with reasonable effort to use directional transmissions. The latter aspect holds particularly w.r.t. protocols (e.g., CTP [14]) using variations of ETX or other metrics whose convergence time would challenge the use of directional transmissions; some of our results confirm this argument quantitatively.

We employ an electronically switchable directional (ESD) antenna [15] called SPIDA [21], also described in Section III. ESD antennas are a good match to WSNs, as they are cheap to manufacture and provide many of the benefits of directional transmissions essentially at no additional energy cost. Our study is performed in simulation and compares a custom version of MultihopLQI leveraging directional transmissions against the original design employing only omni-directional ones. As shown in Section IV, we consider key WSN performance metrics and employ an existing link-layer empirical model of the SPIDA antenna [20]. The latter allows us to strike a balance between the accuracy of the results—still realistic as the model faithfully adheres to real-world dynamics—and the practical need to sweep the parameter space, including system scale. The latter is particularly difficult to achieve given that ESD antennas prototypes are not commercially produced at scale.

Directional antennas, including ESD ones, can shape their lobe both when transmitting and when receiving. In the latter, the antenna increases the gain to handle transmissions from a given direction, simultaneously shielding the transceiver from transmissions from different directions. In a tree-based routing topology, it is much simpler to leverage directional transmissions than receptions. The latter would require parents in the tree to know the time every child is going to start transmitting, to properly configure the antenna beforehand. In the unsynchronized, CSMA setting employed by most convergecast protocols, this is extremely difficult to achieve. Therefore, we only consider directional transmissions and omni-directional receptions, as in existing works [10], [11].

We describe our findings in Section V. Papers usually focus on a solution and rarely linger into the often tortuous path of failures that led to it. This paper takes the opposite approach and unfolds as a tell-tale of several episodes. Each episode begins with a "solution" we thought would unleash the potential of directional antennas for WSN convergecast, and ends with the "problem" that instead surfaced, leading to a new episode. Throughout 10,000 simulated hours of experiments, the outcome is that we were unable to find a *practical* protocol configuration that would reap the benefits of directional antennas.

As discussed in Section VI, the evidence we collect suggests that the *tree-shaped routing topology* used to implement WSN convergecast, in combination with *opportunistic packet schedules*, appear to place an inherent limit on the improvements attainable with directional transmissions. Based on our results, a design exploiting directional antennas demands to overhaul these techniques, despite them being a cornerstone in the state of the art for WSN convergecast [5], [14], [19], [30].

We end by discussing potential threats to the validity of our results in Section VII, and with concluding remarks in Section VIII.

II. Related Work

Several antenna technology enabling directional transmissions exist, e.g., adaptive beamforming antennas. Unlike ESD antennas, these are widely employed in cellular and ad-hoc networks [8]. Recent examples are the work by Arslan et al. [2], who design an efficient Wi-Max beamformer antenna together with a real prototype, and works applying adaptive beamforming to indoor wireless LANs [6]. Liu et al. [17] design WiFi access points equipped with phased array directional antennas to achieve high throughput in dense networks, and further develop a protocol that leads to network capacity improvement [18].

These solutions, however, are ill-suited to WSNs [3], as the key performance requirements are sharply different. Cellular networks and wireless LANs seek to achieve high throughput and low latency, whereas in WSNs reliability and energy consumption are paramount. Traffic patterns and network topologies also differ: cellular networks and wireless LANs are mainly characterized by one-to-one or one-to-many traffic atop star-shaped topologies. In WSNs, traffic typically flows in a *many-to-one* fashion across an unstructured *multi-hop* topology.

Compared to other kinds of wireless networks, the WSN literature about directional antennas is rather limited. The existing works mainly belong to two categories: individual antenna prototypes or clean-slate designs of network protocols employing directional transmissions.

Works in the first category demonstrate the viability of ESD antennas for WSNs, yet rarely explore their integration in a concrete network stack. For example, Giorgetti et al. [13] combine four patch antennas to achieve directional transmissions and assess the improvements in single-link performance. Viani et al. [29] employ parasitic elements and analyze the antenna's ability to reduce internal interference. Parasitic elements are often instrumental to reduce cost and size [6]. Nilsson [21] employs parasitic elements in the SPIDA design as well, whose link-layer performance is assessed in ad-hoc experiments [22].

As for network protocols, they are often designed based on idealized antenna models defined purely by geometrical properties. For example, Felemban et al. [10] present a clean-slate protocol stack that solely employs directional transmissions [11]. Although this can in principle be realized atop ESD antennas, their simulation results would hardly translate directly to a real antenna. Other works focus on specific network services in isolation, e.g., neighbor discovery [28] and MAC [10], seldom including real-world validations. A partial exception is the work by Mottola et al. [20] that, based on an empirical link-layer model, evaluates the impact of directionally forwarding packets atop an omni-directional convergecast tree. In this configuration, the routing links are constrained by the omni-directional range, in spite of directional transmissions reaching farther. Our work is more general, as we evaluate the impact of *both* forwarding applications packets *and* the significantly more complex task of building the routing tree.

III. Antenna and Protocol Stack

We describe next the specific antenna and protocol stack we choose for our study.

A. Directional Antenna and Model

SPIDA is an *electronically switched parasitic element* antenna [25] designed by Nilsson [21]. It consists of a central active element surrounded by parasitic elements. The former is a quarter-wavelength whip antenna, i.e., a traditional omni-directional antenna. The parasitic elements can be switched between ground and isolation: when grounded (isolated) they work as reflectors (directors) of radiated power.

The SPIDA antenna has six individually controllable parasitic elements, yielding six possible "switches" to control the shape and direction of the main lobe; when all isolated, SPIDA behaves as an omni-directional antenna. The cost, size, and radiation characteristics of SPIDA are comparable with the state of the art in directional antennas for WSNs [6], [13], [29], rendering our findings of general applicability.

An empirical link-layer model for SPIDA exists [20], enabling one to synthetically reproduce in simulation the antenna characteristics without excessive processing overhead. The model is successfully validated along a number of dimensions, including packet reliability and link fluctuations, and is based on real-world RSSI packet traces from long-term experiments where the antenna is configured to achieve directional transmission and omni-directional reception, as in our setting.

B. Protocol Stack

We focus on the protocol stack differences between the omni-directional baseline and its directional variants.

Routing. Our implementation of MultihopLQI reproduces the operation of the original protocol, e.g., it employs the same techniques for loop avoidance and for detection of link failures. Nevertheless, the empirical model in [20] only provides $RSSI$ traces, not LQI; therefore, we resort to a different metric, based on a simple model of free space propagation [23]. We define the probability of successful reception $p_i(RSSI)$ of a packet of m bytes sent on a link i as:

$$p_i(RSSI) = (1 - BER(RSSI))^{8m}$$

where the bit error rate (BER) is computed as

$$BER(RSSI) = \frac{1}{2} erfc\left(\sqrt{\frac{10^{\frac{RSSI-RSSI_{min}}{10}} \times B_n}{D}}\right)$$

where $RSSI_{min} = -98.7$ dBm is the radio sensitivity threshold, $B_n = 194$ MHz is the noise bandwidth, and $D = 250$ kbps is the data rate of the CC2420 radio chip used to derive the empirical radio model [20]. To account for link dynamics, $p_i(RSSI)$ is computed as the per-sender and per-sector average of the $RSSI$ values of received packets, over a window of 30 samples. We verified these parameters strike a balance between stability and convergence speed, also with variable channel conditions.

The end-to-end probability M_L of delivering a packet to the sink over a multi-hop path of L links is

$$M_L = \prod_{i \in L} p_i(RSSI)$$

Since the metric is a probability and not a score, it is multiplied (rather than summed) along the path from the node to the sink. As the length of the path is not implicitly encoded as part of the metric, this may yield unnecessarily long routes in the presence of several neighbors with good links. In these cases, we choose the route with the smaller hop count among routes with the same metric. Although sub-optimal, such an approach is often employed in practice [3]. The other alternative would be to derive a higher-level metric that combines end-to-end probability and hop count, which would present the problem of properly tuning the weight associated to either factor [3].

Directional transmissions and MAC. MultihopLQI is designed to work atop a contention-based MAC protocol supporting link-layer acknowledgments and automatic re-transmissions. We provide these functionality in the stack we use in our study. Directional transmissions, however, complicate support for the latter functionality. Indeed, a packet reception using an omni-directional configuration does not provide information on what is the best direction to use for communicating back to the transmitter. On the other hand, the SPIDA antenna does not provide angle-of-arrival information [21].

We address this problem by embedding the identifier of the sector used for the transmission within every beacon used for link probing. Based on the received beacons, every node maintains a mapping $\langle id, sector \rangle$ that describes what $sector$ the node with identifier id employed for transmitting a received beacon. In case a node receives beacons from the same node over multiple sectors, we only keep the sector with the highest $RSSI$. We then piggyback these tuples on every outgoing beacon; this way, all reachable nodes eventually learn what sector yields a successful transmission to a nearby device. The tuple size is 2 B. If the network is very dense, the packet size may increase drastically. We choose to include only the 10 tuples with the highest $RSSI$, which represent the best-quality links most likely used for routing and impose a modest 20 B per-packet overhead.

IV. EXPERIMENTAL SETTINGS

We use Castalia [4], a WSN simulator built upon the Omnet++ platform. Compared to other WSN simulators [9], Castalia features much faster running times at the cost of not modeling the hardware layers. This allows us to sweep the parameter space in reasonable time. We discuss in Section VII the potential impact of this choice on our conclusions.

As the baseline, we consider a version of MultihopLQI that exclusively operates with the SPIDA antenna constantly set in the omni-directional mode. This version, hereafter called OMNI, models the protocols' behavior in the original form, i.e., without directional transmissions.

Metrics. We consider staple WSN performance metrics [1]. We measure the packet *delivery at the sink*, defined as the fraction of application packets successfully received at the sink over those sent by all sources. This figure determines the level of service provided by the WSN, as it is directly

Dimension	Values	Unit metric
Number of nodes	**300**	nodes
Network density	4..**12**..20 (3.0)..(8.6)..(11.8)	nodes / 10.000 m^2 (neighbors in omni)
Packet generation rate	2..**4**..8	pkt/min

Table I: System dimensions; default values are in bold.

proportional to the amount of sensed data that reach the user. The delivery's counterpart is the *routing efficiency*, indicating the cost the system incurs for delivering a packet to the sink across multiple hops. Typically, this is measured in terms of a node's energy consumption, which represents the most precious resource in WSNs. This figure, however, is mainly determined by the MAC layer used. Designing an efficient low-power MAC layer for directional antennas is, however, a challenge per se [11], [24] and any choice would, in a way or the other, significantly bias the results. We thus resort to measuring the cost of successful packet delivery in terms of the network-layer packets it generates, which still provides an indication of the system effort. The routing efficiency is therefore the ratio between the application packets delivered to the sink and the overall packets generated within the network to this end, including retransmissions and link probing beacons.

Settings. We explore different system dimensions, as illustrated in Table I. The number of nodes purposely exceeds the size of real installations, yielding challenging conditions. As for network density, Table I reports the number of nodes over the 100×100 m^2 area we use. The actual network connectivity, however, is a function of the antenna. To relate our values to existing literature, Table I also reports the average number of neighbors when using SPIDA in omni-directional mode. This figure, along with the packet generation rates we use, match existing deployments [12].

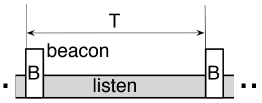
Throughout the experiments, application packets are 80 B long. Beacon packets used for link probing, instead, are of 30 B for the omni-directional case and 51 B for the directional one, as the latter carry additional control information. The node placement and physical orientation of the SPIDA antenna are random. For each setting, the metrics above are computed on at least 48 statistically-independent experiments. We run simulations until the variation of metrics around their average value is within a 5% bound. Unless specified otherwise, the following charts report network-wide averages, along with their 95% confidence interval.

V. FINDINGS

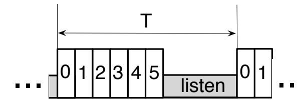
We retrace the sequence of solutions and problems we investigated, towards the goal of determining to what extent directional antennas provide benefits for WSN convergecast.

A. Directional Link Probing: Convergence Time or Overhead?

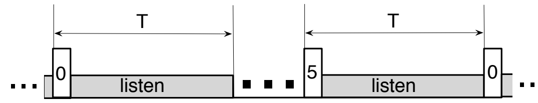
Mainstream WSN convergecast protocols, including MultihopLQI, build a tree topology rooted at the sink. This topology is built by performing *link probing* via beacons, which provides the information to build the multi-hop routes to the sink. In omni-directional protocols, link probing bears great impact on the performance. In particular, the frequency of beacon exchanges determines the tradeoff between quickly detecting changes in link quality and



(a) Omni-directional (OMNI).



(b) Directional (DIR-SWEEP).



(c) Directional (DIR-SPREAD).

Figure 1: Link probing: node schedule.

communication overhead. Directional communication further complicates the problem.

Link probing schemes based on omni-directional antennas rely on the fact that a *single* broadcast beacon is sufficient to *simultaneously* probe the sender's neighborhood. The simplest schemes, commonly used in WSN convergecast, are periodic; as shown in Figure 1a, in each period T, a node first transmits a beacon then listens for beacons from neighbors. This allows a node to gather neighbor identifiers and link quality information. To avoid systematic packet collisions, in our OMNI scheme each node periodically nudges the start of its interval by a random value in $(0, T]$, as done in many existing solutions.

Simultaneously probing a node's neighborhood with a single packet is not an option with directional antennas. As shown in Figure 2, if the latter is dynamically configured in omni-directional mode, the neighborhood reached by the beacon is a subset of the one available when the antenna is configured to radiate energy only on a given sector. On the other hand, when the beacon is sent on a given sector, the rest of the neighborhood (i.e., the other sectors) is not reached. Therefore, if the antenna support N sectors, discovering the neighborhood requires N beacons, increasing both contention and overhead.

However, protocol designers can choose to trade the fast *convergence time* given by the simultaneous link probing offered by the broadcast scheme vs. the extra *overhead* imposed by the directional scheme, by properly playing with the *schedule* of the N beacon transmissions. We focus on two opposite alternatives, shown in Figure 1b and 1c. The first one privileges convergence time by replacing the omni-directional beacon with N directional beacons sent back-to-back. This

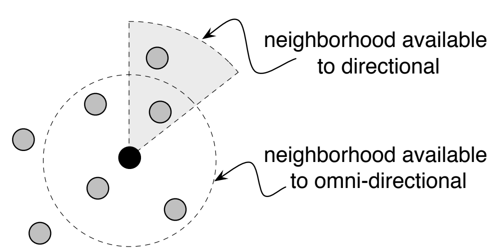


Figure 2: Omni-directional vs. directional link probing.

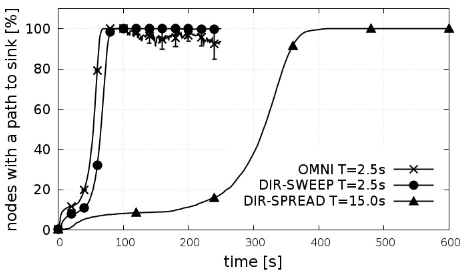


Figure 3: Link probing: convergence time vs. overhead.

approximates the simultaneity of omni-directional schemes by "sweeping" the neighborhood as quickly as possible. This strategy, hereafter called DIR-SWEEP, constructs the topology almost as fast as the omni-directional case, at the cost of an N-fold increase in the beaconing overhead.

However, is this overhead really required? What if we trade-off the spatial resolution provided by directional antennas for the simultaneity provided by omni-directional ones, and "spread" the link probing over time? The second alternative, hereafter called DIR-SPREAD, investigates this aspect by sending the N directional beacons over a longer $N \times T$ interval. Over time, this incurs the *same* overhead as OMNI but increases convergence time, therefore delaying topology construction.

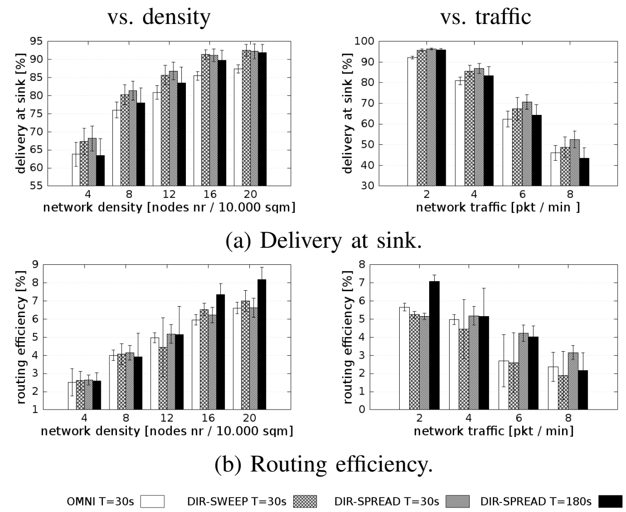
> **Tradeoffs of directional link probing:** *Broadcast-based link probing can be replaced at the cost of an N-fold increase in either* i) *beaconing overhead, to retain the same convergence time, or* ii) *convergence time, to retain the same beaconing overhead.*

The pros and cons of these alternatives are illustrated in Figure 3, showing results from a set of simulations where the tree topology is constructed without application traffic. The simulations use a baseline OMNI beaconing period T=2.5 s; the other parameters are the default ones in Table I. Both directional alternatives construct a tree reliably: all nodes are eventually connected to the sink. However, while the 6-fold beaconing overhead allows DIR-SWEEP to achieve this goal almost as fast as OMNI, DIR-SPREAD reaches 100% connectivity with almost a 6-fold delay, but with the same network overhead as OMNI.

A 6-fold delay in *building* the tree is not a problem, as this happens only at startup. However, can the less aggressive directional link probing of DIR-SPREAD match (or improve) the performance of the omni-directional one in *maintaining* the tree? The question cannot be answered in isolation, as the impact of link probing depends on the use that higher-level protocols make of the information made available by it.

B. Link Probing Convergence Time: Does it Matter?

Figure 4 analyzes the impact of the strategies for directional link probing on the performance of convergecast, expressed by the metrics we defined in Section IV, on the



(a) Delivery at sink.

(b) Routing efficiency.

Figure 4: Impact of link probing.

modified version of MultihopLQI also described there. The charts are derived by considering *only* the traffic after the tree topology is built. Beacons are nonetheless used to reconfigure the tree against changes in link quality. Here we use a baseline beaconing period T=30 s for OMNI that, unlike the one in Figure 3, is representative of the values found in real deployments [30]. We compare against DIR-SWEEP with T=30 s, and DIR-SPREAD with T=N×30=180 s. Moreover, we also compare against DIR-SPREAD with T=30 s (i.e., one beacon every 5 s) as this provides us with the complementary view where we have the same overhead as DIR-SWEEP but spread over T.

The left-hand side of Figure 4a shows the delivery rate at sink as a function of density. The performance of DIR-* alternatives increases with density, as expected due to their ability to reduce contention; the improvement, however, is slightly above 10% at best. The lowest performance among DIR-* is the one of DIR-SPREAD at T=180 s, due to its sparse beaconing being too slow to react to changes in the link quality. This is evident at the lowest density, where the performance of DIR-SPREAD is equivalent to OMNI. On the other hand, the difference among DIR-* is minimal at highest density.

Similar remarks hold for delivery at sink vs. traffic, shown in the right-hand side of Figure 4a. At low traffic, replacing the OMNI beacon with a single DIR one (DIR-SPREAD at T=180 s) is beneficial, and achieves the same benefits as the other DIR-* alternatives. As the traffic increases, however, the likelihood of this beacon to be lost in collisions is higher; this slows down adaptation to changes in connectivity and undermines the advantages due to the reduction of contention, explaining the performance of DIR-SPREAD at T=180 s. In any case, the improvement of DIR-* vs. OMNI w.r.t. traffic is even smaller than the one w.r.t. density. Figure 4b, however, tells a different story by looking at routing efficiency. For the alternatives with high beaconing, routing efficiency remains at or below the one of OMNI, while for DIR-SPREAD at T=180 s is always at or above OMNI—e.g., 27% better at highest density.

Summing up, DIR-SPREAD with a setting T=180 s offers the best tradeoff between delivery at sink and routing efficiency, *always* providing a performance equivalent or superior to OMNI.

> **Higher probing convergence time is acceptable:** *The benefits of directional transmissions can be reaped without increasing the overhead w.r.t.* OMNI.

The other directional alternatives perform better in some cases, but worse in others, especially w.r.t. routing efficiency, which has a direct impact on lifetime. For this reason, hereafter we consider only DIR-SPREAD at T=180 s.

The results we obtained use directional transmissions for both the link probing necessary to build and maintain the convergecast tree and the forwarding of application messages along it. What if directional transmissions were used only for the latter, based on a convergecast tree built using only omni-directional communication? In principle, this should abate the tree construction overhead of the high beaconing alternatives, and could possibly improve over the high convergence time DIR-SPREAD scheme we identified as the best tradeoff thus far. We verify this hypothesis next.

C. What About Directional Forwarding Only?

Exploiting directional transmissions for message forwarding still requires the sender to determine the right sector to use. Several strategies are possible, and some practical ones are explored in recent work [20].

To determine the maximum improvement achievable, we place ourselves in an ideal condition where the sector for transmitting to a given parent is the optimal one, computed offline with global knowledge about node placement. In this scheme, called hereafter DIR-FWD, directional forwarding takes place based on the same convergecast tree generated by the OMNI baseline, where forwarding is omni-directional. As a term of comparison, we also report the results for DIR-SPREAD at T=180 s from the previous section.

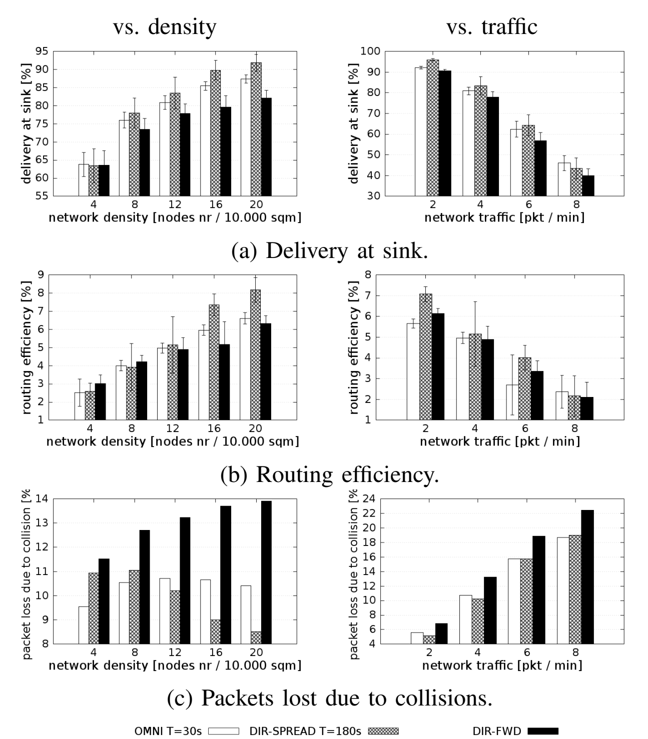
Figure 5a shows the delivery at sink as a function of density and traffic. In both cases, DIR-FWD is at best equivalent—and often worse—than OMNI; the latter is always equal or worse than DIR-SPREAD, as discussed. Therefore, DIR-FWD does not bring *any* advantages w.r.t. delivery. The situation is slightly different for routing efficiency, shown in Figure 5b. At low to medium densities, DIR-FWD improves over both competitors, up to 20% in the best case. Instead, at the two highest densities DIR-FWD is significantly worse than the others. For what concerns traffic, instead, DIR-SPREAD appears to be the best option.

> **Directional forwarding alone is not an option:** *To reap the benefits of directional transmissions, these must be employed also during topology construction and maintenance.*

What is the reason? First, our RSSI-based metric may yield rather long paths. The simulation logs show that the route stretch of DIR-* is only 1–2 hops shorter than OMNI, which significantly reduces the potential advantages. This is expected, as using RSSI as a routing metric suffers from similar problems [3]. On the other hand, link estimators like ETX [7] are not an option, as they need tens of rounds of link probing to converge [3]; these would greatly amplify the N-fold increase in either convergence time or overhead, ultimately yielding unacceptable performance.

Second, directional antennas exacerbate hidden terminals. Figure 5c shows the fraction of packets lost due to collisions. An increase in traffic increases collisions, as shown in the right-hand side, but leaves unaltered the relative performance of OMNI and DIR-SPREAD. Indeed, for a given density, both protocols build and maintain the best tree possible; the bigger incidence of hidden terminals in DIR-SPREAD is compensated by longer links, which decrease the number of hops and thus increase both path reliability and reduce contention.

The hybrid solution DIR-FWD, instead, also operates with the longer range of directional forwarding, but over the shorter links generated by OMNI. This results in higher chances of collisions, amplified as traffic increases. The latter situation is more evident in the left-hand side of Figure 5c, where an increase in density significantly degrades the performance of DIR-FWD w.r.t. the alternatives. At low density, few forwarding options are available, and DIR-SPREAD pays a higher toll to hidden terminals. At the



(a) Delivery at sink.

(b) Routing efficiency.

(c) Packets lost due to collisions.

Figure 5: Directional forwarding.

other extreme, when density is high, the broadcast-based OMNI protocol suffers from collisions, while the latter are reduced in DIR-SPREAD by the combination of directional communication and spatial diversity.

The two problems are related: short links amplify the impact of the hidden terminal problem. If we were able to construct a tree whose links are "as long as possible", this could potentially unlock the full potential of directional antennas. We investigate next to what extent this holds.

D. Are Long Links the Key?

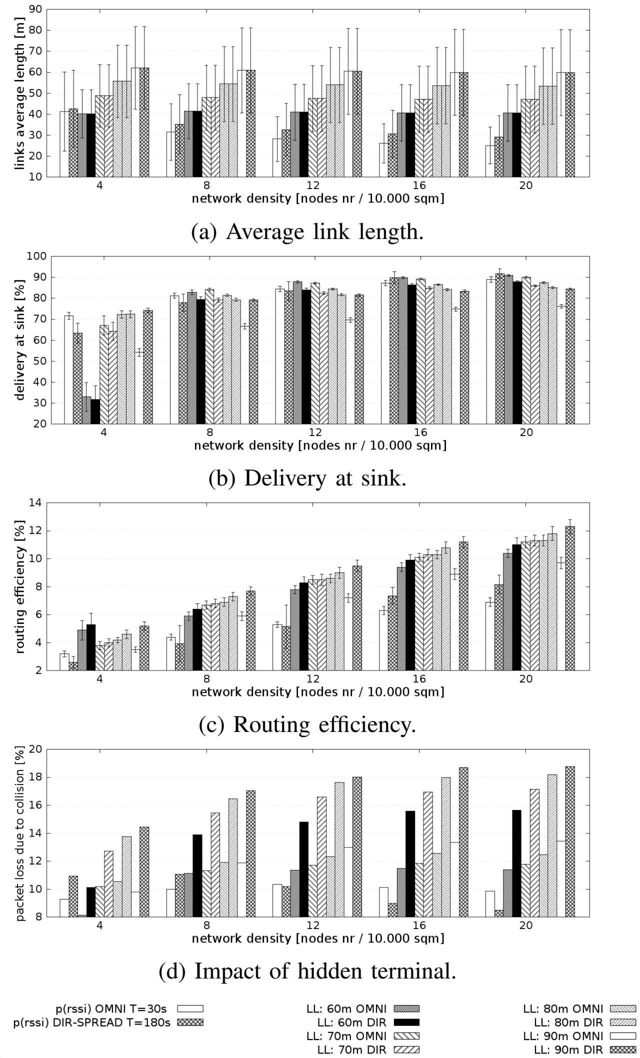
To investigate this aspect, we exploit the asset of simulation. Instead of letting MultihopLQI build the routing topology, we use global knowledge about the physical placement of nodes to artificially generate a topology whose links have a desired average distance. A graph is constructed whose vertexes are the input network nodes, edges are all the possible links among nodes, and weights are assigned to edges corresponding to the geographical distance between nodes. The Dijkstra algorithm is then applied to this graph, yielding the paths from the root to all other nodes that minimize the sum of link lengths.

We generate different tree topologies by controlling their *average link length*, that is, by filtering out links exceeding a *maximum*[1] threshold $LL = d$. This procedure yields a topology that has links as long as possible, but always shorter than LL. Figure 6a shows the actual average link length (with min-max bars) for each value of LL. This and the following charts report also the link length when using the $p_i(RSSI)$ versions of OMNI and DIR-SPREAD, identified in Section V-B and V-C as providing the best performance for each antenna configuration. The LL versions of OMNI and DIR correspond instead to the cases where the tree parent is determined by the offline algorithm, rather than the $p_i(RSSI)$ metric. As expected, the average link length of $p_i(RSSI)$ protocols is much shorter than the one we artificially generated.

These topologies are computed offline, their knowledge hard-wired in the nodes, and used throughout a simulation run. Nevertheless, the nodes still send beacons in the same way as the OMNI (and DIR-SPREAD at T=180 s) case. These beacons are ignored; their only purpose is to make the simulations more realistic by taking into account that, in practice, the topology used has a maintenance overhead.

Figure 6b and 6c show the delivery at sink and routing efficiency vs. density. The dual charts vs. traffic are omitted because the latter bears a more limited impact on performance, as evidenced thus far. These charts show that, in comparison with the $p_i(RSSI)$ versions of the protocols,

[1]It should be noted that the dual constraint of limiting the *minimum* distance is in general unfeasible. Based on the input placement of nodes, it is often the case that short links are key to maintain the network connected. The opposite case, where *long* links are key to avoid partitioning, occurs rarely in our setting; nevertheless, it is the cause for the drop in delivery at density=4 in Figure 6b.



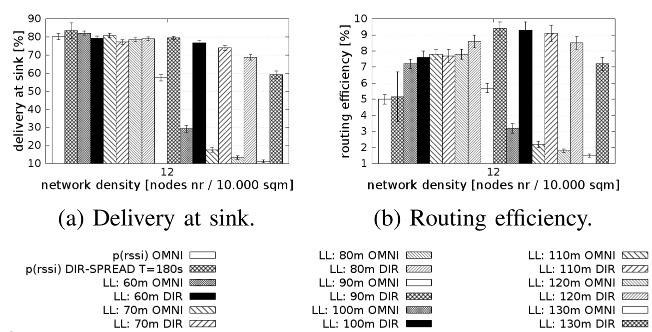
(a) Average link length.

(b) Delivery at sink.

(c) Routing efficiency.

(d) Impact of hidden terminal.

Figure 6: Impact of long links.



(a) Delivery at sink.

(b) Routing efficiency.

Figure 7: Performance with links up to 130 m, density=12.

constraining the topology to longer links yields a slightly higher delivery and a significantly higher routing efficiency, confirming our hypothesis. Moreover, beyond LL=90 m, the delivery of OMNI degrades dramatically. This can be appreciated better in Figure 7, where we show results with LL values higher than those in Figure 6, omitted there to reduce cluttering. The reason lies in the longer range (∼50% more in our case) of directional antennas. Otherwise, for LL <90 m, the difference between OMNI and DIR is less prominent; OMNI yields marginally better (<6.5%) delivery at all densities, while DIR provides better (<10%) routing efficiency at all densities. Therefore, the key finding is the fact that *unless* the routing protocol is able to make directional antennas operate in conditions where the extra range is exploited, OMNI is still a viable solution.

> **Directional links should be long enough:** *For directional transmissions to provide benefits, and in absence of further coordination among nodes, the links used for routing should stretch beyond the omni-directional range.*

At first sight, one may think that this finding is actually obvious, given that a reason to use directional antennas is to achieve a longer range with the same energy. However, their directionality also reduces contention; in principle, this should enable improvements also within the omni-directional range, although the finding above says otherwise. Further, why the difference between OMNI and DIR is not higher, even for LL <90 m? Figure 6d shows that the culprit is the hidden terminal problem that, as well-known in the literature, is exacerbated by directional antennas. For instance, at the reference density of 12 and LL=80 m, the incidence of packet loss due to collisions is 34% higher in DIR.

E. What is the Impact of Hidden Terminals?

As the links employed for directional routing become longer, packets are increasingly lost due to collisions. We conjectured that in these cases, scenarios akin to Figure 8 manifest more prominently. At the sender, the CCA check before transmission occurs with the antenna in *omni-directional* mode; performing the same CCA once per sector would delay every transmission excessively. The sender is then unable to detect ongoing *directional* transmissions and, if the receiver is the same, a packet collision likely occurs. Hidden terminal issues are, in fact, the motivation underlying the design of many MAC protocols for directional antennas [11], [24].

Provided our conjecture holds, correctly handling scenarios akin to Figure 8 should provide improvements. We re-run

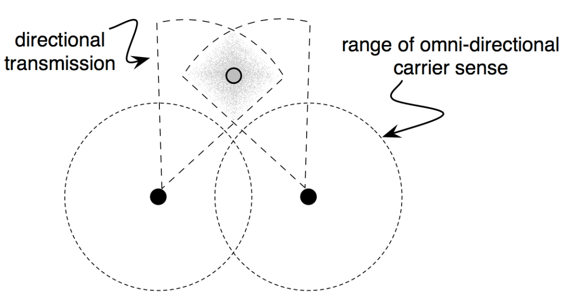
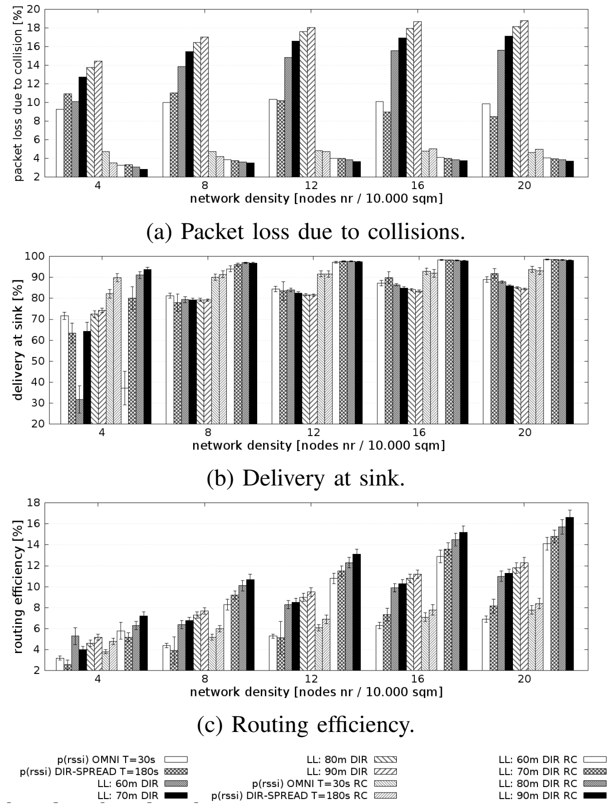


Figure 8: Two nearby transmitters address the same receiver in a directional manner; due to omni-directional CSMA, carrier sense fails and packets collide at the receiver.



(a) Packet loss due to collisions.

(b) Delivery at sink.

(c) Routing efficiency.

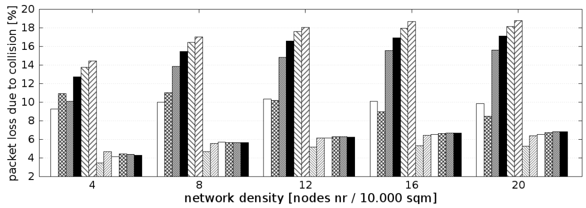
Figure 9: Artificially coordinating directional senders.

the experiments in Section V-D by performing a CCA check *also on the receiver* and *artificially coordinating* all senders to prevent collisions. This essentially entails that, if an ongoing transmission exists towards a receiver, the simulator temporarily suspends all other transmissions towards it.

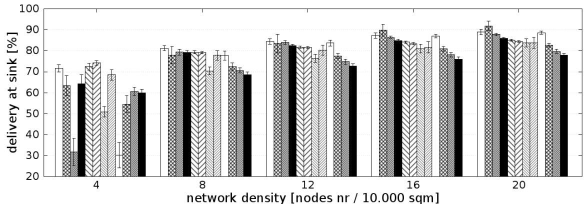
Figure 9a confirms our conjecture. With the artificial coordination we introduce via receiver checks (RC), the number of packets lost due to collisions drastically lowers. This observation applies across all settings we explore. The remaining losses are due to collisions among the (uncoordinated) link probing beacons. In essence, Figure 9 shows that coordinating packet transmissions in DIR provides better performance than the uncoordinated DIR or the OMNI configuration. This applies also when the maximum link length is the same for omni- and directional configurations. In particular, Figure 9c shows that the advantages in routing efficiency are more marked with higher network density, where channel contention is more likely.

These results show that the limiting factor for the performance of DIR within the omni-directional communication range are, in fact, hidden-terminal problems between children and parents in the routing tree.

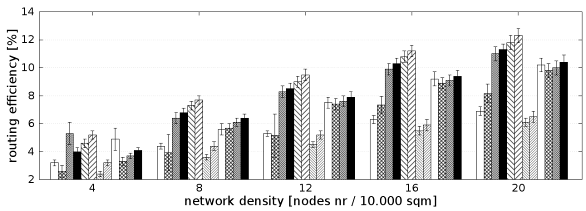
> **Impact of hidden terminals:** *In principle, solving hidden terminal problems improves performance with directional transmissions also within the omni-directional communication range.*



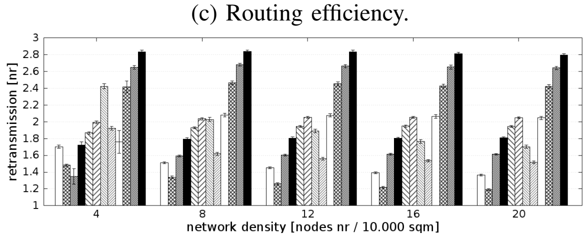
(a) Packet losses because of collisions.



(b) Delivery at sink.



(c) Routing efficiency.



(d) Packet retransmissions.

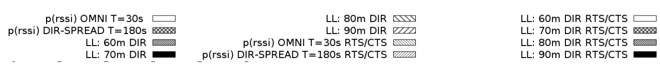


Figure 10: Using RTS/CTS to cope with hidden terminals.

These improvements, however, are only theoretical: we avoid collisions by manipulating the way the simulator schedules transmissions, which is not feasible in reality.

F. Can We Address Hidden Terminals Practically?

Motivated by these considerations we seek a *practical distributed* schema to avoid hidden terminal problems akin to Figure 8. Two options are available. One one hand, we could design a TDMA-like schema that, by design, avoids overlapping transmissions. However, similar solutions hardly operate efficiently in WSNs: the required time synchronization is energy-hungry, and packet schedules must be continuously re-computed as the underlying topology changes.

The other option is to retain opportunistic channel access. In this case, we exploit a custom RTS/CTS schema to coordinate packet transmissions. Before the actual transmission, the sender generates a Ready-To-Send packet explicitly addressed to the receiver, which replies with a Clear-To-Send packet if not already busy transmitting[2]. Data transmission and acknowledgement happen following the RTS/CTS exchange. Thus, *four* packet transmissions are necessary for every data packet. For this reason, this scheme is rarely employed in WSNs, due to energy consumption. However, it is worth investigating, in our specific setting enabled by directional transmissions, if the extra overhead due to coordination can be compensated by the potential gains it unlocks by preventing hidden terminals.

We re-run the experiments again, obtaining the results in Figure 10. The number of packets lost because of collisions in Figure 10a is comparable to the one obtained with the artificial schema in Figure 9a. This demonstrates that the custom RTS/CTS schema we employ does solve most of the hidden terminal problems involving directional transmissions.

The toll to pay, however, is a hefty one: the performance of DIR becomes again comparable to OMNI, and the benefits of the former vanish. The delivery with the RTS/CTS schema (Figure 10b) is equal or worse than the case where we accept packets collisions when multiple senders cannot hear each other (Figure 9b). Similar considerations hold by comparing the routing efficiency in Figure 10c against Figure 9c.

Figure 10d shows the likely culprit: the number of packet retransmissions increases dramatically with the RTS/CTS schema. Besides packet failures that would normally occur on the channel, transmitting 4 packets per application packet increases contention and therefore the chance that any of them fails—and each transmission failure causes the entire RTS/CTS procedure to re-start, further aggravating channel contention. This observation is valid throughout all settings, leading to the following final insight that adds to the list of reasons for *not* employing RTS/CTS in low-power multi-hop wireless.

> **Avoiding hidden terminals:** *When used for coordinating transmissions with directional transmissions, RTS/CTS causes excessive overhead, yielding worse performance than simply accepting packet collisions.*

VI. Final Considerations and Outlook

The quantitative evidence we gathered suggests that directional transmissions are at odds with two fundamental cornerstones of WSN convergecast protocols.

1) **Tree-shaped routing topologies.** They are often used, as they naturally support the many-to-one traffic pattern [5], [14], [19], [30]. However, the link quality metrics used to build and maintain the trees do not consider the characteristics of directional antennas. Moreover, directional transmissions exacerbate the hidden terminal

[2]The transmission of both RTS and CTS is handled through the (simulated) network stack (i.e., as if it were implemented in software) as Castalia is unable to emulate the radio hardware directly. A hardware implementation would probably perform better in terms of latency, yet it would likely not impact the conclusions we draw later in this section.

problems already present with omni-directional transmissions.

2) **Opportunistic channel access.** Most WSN convergecast protocols rely on opportunistic channel access, as it deals more easily than TDMA with time-varying topologies and can be easily extended with radio duty-cycling. On the other hand, opportunistic channel access makes coordinating concurrent transmitters, and therefore avoiding hidden terminals, very challenging.

Therefore, we argue that to reap the most benefits from the use of directional transmissions in WSN convergecast, protocol designers need to abandon either or both these cornerstones.

Novel ways to build and maintain routes are a primary need. One possibility is to devise routing metrics limiting the children for every parent. Indeed, the ideal situation is one where paths proceed in "parallel" up to the sink; the antenna's ability to concentrate the radiated energy only towards the receiver would increase the chance of the capture effect [16], implicitly resolving potential collisions on unintended receivers.

However, achieving this configuration is non-trivial. Let apart fully centralized approaches shown to scale reasonably only when a single source-sink path is to be configured [27], identifying multiple parallel paths requires the coordination across a minimum of two hops to avoid inter- and intra-path interference. Further, the routing configuration must keep up with the underlying channel dynamics, which is generally challenging in a low-power setting [3] and even more so when operating across multiple hops.

To deal with hidden terminals, a possibility is to flip the usual convergecast operation where the sources implicitly schedule packet transmissions along the entire path. One could charge the sink to coordinate the paths leading to it, granting permission to funnel data only to a subset of nodes at a time, carefully chosen to minimize collisions. However, achieving this behavior in a distributed manner and without excessively sacrificing bandwidth is also going to be challenging.

VII. Threats to Validity

We discuss next how the scope and methodology we adopt may alter our conclusions.

Antenna and link-layer model. The SPIDA antenna is one of the *very few prototypes existing* and, to the best of our knowledge, the only one with an empirical link-layer model that enables *realistic large-scale* simulations. We argue that SPIDA represents a meaningful design point. Antennas such as the one by Giorgetti et al. [13] feature less directional propagation, thus limiting the potential gains due to directional transmissions. On the other hand, further increasing the directionality of the SPIDA would entail a complete redesign including more sophisticated control circuitry or the use of dedicated signal processors, which would greatly affect energy consumption and/or size [21].

The link-layer model does have a few limitations, mainly that it is obtained in an outdoor environment free from external interference [20]. While our insights directly apply to many outdoor WSN scenarios [12], the characteristics of indoor radio propagation may actually reinforce some of our conclusions. Section V-B describes one key finding as the fact that maintaining a routing tree using directional transmissions is challenging. As churn increases due to unpredictable wireless dynamics, the routing tree must be reconfigured more frequently and more rapidly. This increases the burden on link probing that, when using directional transmissions, leads to slower convergence or increased overhead, as discussed in Section V-A.

Simulator and MAC. Castalia is an event-driven *network* simulator that, similar to other network simulators like ns-2, does *not* account for the local processing on the individual nodes. In Castalia this happens instantly, which renders simulations less time-accurate but speeds up run times by orders of magnitude. It is general wisdom that these simulators tends to be optimistic on many accounts, e.g., they hide race conditions. The results supporting many of our findings are probably only going to become worse if the local processing times are accounted for.

Finally, the MAC protocol we use is admittedly not as sophisticated as state-of-the-art ones. Besides not performing radio duty-cycling, as discussed in Section IV, it does not model hardware-level operations, e.g., to handle acknowledgments via the radio chip. This is due to a limitation of Castalia, which does not model hardware layers. This could affect only our findings in Section V-F, whose results are however so macroscopic that we believe their essence would not be altered otherwise.

VIII. Conclusions

The characteristics of directional antennas suggest that they can significantly improve the performance of WSNs protocols in reliability and energy preservation. In this paper we *quantitatively* investigated this hypothesis, hitherto largely unverified in the literature, by focusing on the convergecast functionality. Unfortunately, the outcome is a negative one. Directional transmissions, once applied to a mainstream convergecast protocol, bring marginal improvements and only in specific conditions. We showed that the reason is that they are fundamentally at odds with two cornerstones of mainstream convergecast protocols: tree-shaped routing topologies and opportunistic channel access. By sharing these findings, albeit negative, we aim to inspire researchers to critically revisit common practices and explore radically different alternatives, able to harvest the potential of directional antennas in WSNs.

REFERENCES

[1] J. Al-Karaki and A. E. Kamal. Routing techiniques in wireless sensor networks: A survey. *IEEE Wireless Comm.*, 11(6), 2004.

[2] M. Arslan et al. Design and implementation of an integrated beamformer and uplink scheduler for OFDMA femtocells. In *Proc. of MobiHOC*, 2012.

[3] N. Baccour et al. Radio link quality estimation in wireless sensor networks: A survey. *ACM Trans. on Sensor Networks*, 8(4), 2012.

[4] A. Boulis. *Castalia user's Manual*. NICTA, March 2011. Version 3.2.

[5] N. Burri, P. von Rickenbach, and R. Wattenhofer. Dozer: Ultra-low power data gathering in sensor networks. In *Proc. of IPSN*, 2007.

[6] R. Choudhury et al. Beamnet: An ad hoc network testbed using beamforming antennas. In *Proc. of VTC*, 2005.

[7] D. D. Couto et al. A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4), 2005.

[8] H. Dai et al. An overview of using directional antennas in wireless networks. *Communication Systems*, Nov. 2011.

[9] J. Eriksson et al. COOJA/MSPSim: Interoperability testing for wireless sensor networks. In *SIMUTools*, 2009.

[10] E. Felemban et al. Samac: A cross-layer communication protocol for sensor networks with sectored antennas. *IEEE Trans. on Mobile Computing*, 9(8), 2010.

[11] E. Felemban et al. Sand: Sectored-antenna neighbor discovery protocol for wireless networks. In *Proc. of SECON*, 2010.

[12] E. Gaura et al. *Wireless Sensor Networks: Deployments and Design Frameworks*. Springer, 2010.

[13] G. Giorgetti et al. Exploiting low-cost directional antennas in 2.4 GHz IEEE 802.15.4 wireless sensor networks. In *European Conf. on Wireless Technologies*, 2007.

[14] O. Gnawali et al. Collection tree protocol. In *Proc. of SenSys*, 2009.

[15] R. C. Hansen. *Electrically small, superdirective, and superconducting antennas*. John Wiley & Sons, 2006.

[16] K. Leentvaar and J. Flint. The Capture Effect in FM Receivers. *IEEE Trans. on Communications*, 24(5), 1976.

[17] X. Liu et al. DIRC: Increasing indoor wireless capacity using directional antennas. *ACM SIGCOMM Computer Communication Review*, 39(4), 2009.

[18] X. Liu et al. Pushing the envelope of indoor wireless spatial reuse using directional access points and clients. In *Proc. of MobiCom*, 2010.

[19] S. Moeller et al. Routing without routes: The backpressure collection protocol. In *Proc. of IPSN*, 2010.

[20] L. Mottola, T. Voigt, and G. Picco. Electronically-switched directional antennas for wireless sensor networks: A full-stack evaluation. In *Proc. of SECON*, 2013.

[21] M. Nilsson. Directional antennas for wireless sensor networks. In *Scandinavian Workshop on Wireless Adhoc Networks*, 2009.

[22] E. Öström, L. Mottola, and T. Voigt. Evaluation of an electronically switched directional antenna for real-world low-power wireless networks. In *Proc. of REALWSN*, 2010.

[23] T. S. Rappaport et al. *Wireless communications: principles and practice*, volume 2. Prentice Hall, 1996.

[24] S. Roy et al. A Network-aware MAC and Routing Protocol for Effective Load Balancing in Ad Hoc Wireless Networks with Directional Antenna. In *Proc. of MobiHoc*, 2003.

[25] D. Thiel and S. Smith. *Switched parasitic antennas for cellular communications*. Artec House, London, 2002.

[26] TinyOS Community Forum. TinyOS TEP 119 - Collection. www.tinyos.net/tinyos-2.x/doc/html/tep11.html.

[27] A. Varshney et al. Directional transmissions and receptions for high-throughput bulk forwarding in wireless sensor networks. In *Proc. of SenSys*, 2015.

[28] S. Vasudevan et al. On neighbor discovery in wireless networks with directional antennas. In *Proc. of INFOCOM*, 2005.

[29] F. Viani et al. Exploitation of parasitic smart antennas in wireless sensor networks. *Electromagnetic Waves and Applications*, 24(7), 2010.

[30] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proc. of SenSys*, 2003.