Low Power, Low Delay: Opportunistic Routing meets Duty Cycling

Olaf Landsiedel* [1], Euhanna Ghadimi[2], Simon Duquennoy[3], Mikael Johansson[2]
olafl@chalmers.se, euhanna@kth.se, simonduq@sics.se, mikaelj@kth.se

[1]Chalmers University of Technology, Sweden
[2]KTH Royal Institute of Technology, Sweden
[3]Swedish Institute of Computer Science (SICS), Sweden

ABSTRACT

Traditionally, routing in wireless sensor networks consists of two steps: First, the routing protocol selects a next hop, and, second, the MAC protocol waits for the intended destination to wake up and receive the data. This design makes it difficult to adapt to link dynamics and introduces delays while waiting for the next hop to wake up.

In this paper we introduce ORW, a practical opportunistic routing scheme for wireless sensor networks. In a duty-cycled setting, packets are addressed to sets of potential receivers and forwarded by the neighbor that wakes up first and successfully receives the packet. This reduces delay and energy consumption by utilizing all neighbors as potential forwarders. Furthermore, this increases resilience to wireless link dynamics by exploiting spatial diversity. Our results show that ORW reduces radio duty-cycles on average by 50% (up to 90% on individual nodes) and delays by 30% to 90% when compared to the state of the art.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication; C.2.2 [**Network Protocols**]: Routing Protocols

General Terms

Algorithms, Design, Experimentation, Measurement

Keywords

Wireless Sensor Network, Energy Efficiency, Opportunistic Routing, Duty Cycle

1. INTRODUCTION

In Wireless Sensor Networks (WSNs), forwarding of packets to their intended destination commonly resembles a two-step process: First, the routing protocol determines the next hop utilizing a routing metric and link estimations. Second, the MAC protocol waits for the intended destination to wake up and to successfully receive the packet.

In this paper, we depart from this unicast design paradigm. Instead, we transmit packets opportunistically in duty-cycled sensor networks: A packet is forwarded by the first awoken neighbor that successfully receives it and offers routing progress towards the destination (see Fig. 1). As a result, we significantly improve energy efficiency, reduce end-to-end delay, and increase resilience to wireless link dynamics when compared to traditional unicast routing in WSNs.

1.1 Significance and Distinction

Low-power links in WSNs are highly dynamic [27, 28]. Link estimation [13, 32] enables routing protocols in WSNs [14, 31] to limit forwarding to links of consistently high reliability, ensuring stable topologies. Our main departure from this work is that the opportunistic nature of our approach explicitly utilizes all neighbors, i.e, both stable and unstable links, for packet forwarding. As a result, we show significant improvements in terms of energy efficiency, delay, and resilience to link dynamics.

Originally, opportunistic routing [5, 7, 8, 19] was developed to improve throughput in multi-hop, mesh networks. It benefits from the fact that in wireless mesh-networks radios are always-on and hence can overhear messages at practically no additional cost. In contrast, sensor networks are commonly duty-cycled to ensure long node and network lifetime, limiting the use of overhearing for opportunistic routing. Moreover, WSN applications demand for high energy efficiency and low delays instead of high throughput. The main distinction of this work over existing work on opportunistic routing is that it adapts the concept of opportunistic routing to WSNs tailoring it to the specific demands of sensor networks and applications.

1.2 Contribution

This paper has four contributions: First, it presents Opportunistic Routing in Wireless sensor networks (ORW). Specifically, it adapts the concept of opportunistic routing to the particular requirements and challenges in WSNs, e.g., by focusing on energy as a key metric and incorporating duty-

*Work done while this author was at ACCESS Linnaeus Center, KTH Royal Institute of Technology, Sweden

IPSN'12, April 16–20, 2012, Beijing, China.
Copyright 2012 ACM 978-1-4503-1227-1/12/04 ...$10.00.

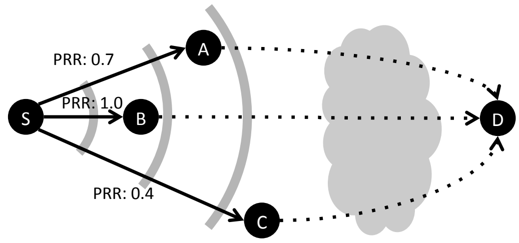


Figure 1: Opportunistic routing in ORW: The first awoken neighbor (A to C) that successfully receives a packet from S and provides routing progress, forwards it to the destination D. It utilizes all neighbors that provide routing progress independent of link quality.

cycled nodes. Second, it presents a novel anycast routing metric to build an anypath routing gradient and determine forwarder sets for opportunistic routing. Focusing on low-energy and low-delay routing, this metric estimates the delay in terms of wakeup periods required to deliver a packet to the sink. Third, we introduce a lightweight, coarse-grained link estimator that reduces probe traffic and state information. It reflects the reduced requirements of opportunistic routing in terms of timeliness and accuracy in link estimation. Fourth, it presents a practical realization of opportunistic routing and evaluates its benefits in both simulation and TinyOS-based testbed experiments. We show that ORW reduces radio duty cycles on average by 50% (up to 90% on individual nodes) and delays by 30 to 90% when compared to the state of the art. Additionally, we show an increased stability to link dynamics and churn.

The remainder of this paper is structured as follows: Section 2 provides the required background on opportunistic routing and introduces the basic concept of ORW. Next, we tailor opportunistic routing to the specific demands of WSNs and detail mechanisms for forwarder selection, our anycast routing metric, and link estimation (Section 3). We compare our design to the state of art in Section 4 and discuss related work in Section 5. Section 6 concludes.

2. ORW DESIGN OVERVIEW

In this section, we provide the required background on opportunistic routing in mesh networks and discuss why it cannot be directly utilized in wireless sensor networks. Next, we introduce the basic concept of our opportunistic routing scheme, show motivating examples, and outline its tailoring to the particular demands of wireless sensor networks.

2.1 Preliminaries

Opportunistic routing [5,7,8,19] improves network throughput in the context of multi-hop, mesh networks such as city-wide wireless networks. In contrast to traditional unicast routing, the underlying concept of opportunistic routing is to delay the forwarding decision until after the transmission. For example, in ExOR [5] each packet is addressed to set of potential forwarding nodes, prioritized by routing progress. Based on their priority, each node in the forwarder set is assigned a time slot for forwarding, which it only utilizes if it did not overhear the packet being forwarded in a previous time slot. Relying on such a consensus protocol or other approaches [7], opportunistic routing avoids duplicate forwarding.

Overall, opportunistic routing leverages spatial diversity to ensure high routing progress and to limit the impact of link dynamics. This leads to a significant throughput improvement when compared to traditional routing schemes [5, 7].

2.2 Opportunistic Routing in WSNs

Wireless sensor networks and their applications pose special requirements, such as low-power networking and resource constraints, that distinguish them from traditional multi-hop mesh networks. These limit the direct applicability of opportunistic routing in three key aspects: (1) reliability and energy efficiency vs. throughput, (2) duty cycling in sensor networks, and (3) complexity of unique forwarder selection.

Reliability, Energy Efficiency vs. Throughput: Opportunistic routing is designed to improve network throughput. However, WSN applications commonly demand reliable forwarding at high energy efficiency and not high throughput. In this paper, we show how opportunistic routing can be adapted to improve energy efficiency when compared to traditional WSN routing.

Duty Cycling in Sensor Networks: Sensor networks are commonly duty-cycled to ensure long node and network lifetime. Hence, nodes are in deep sleep states most of the time, with their radios turned off. Duty-cycling limits the number of nodes that concurrently overhear a packet (assuming no prior synchronization). As a result, it prevents the spatial reuse in the forwarding process, one of the key benefits of opportunistic routing. However, we show in this paper that opportunistic routing brings low latency to duty-cycled networks: Instead of waiting for a given forwarder to wake up, the anycast primitive allows a node to send to the first awoken parent.

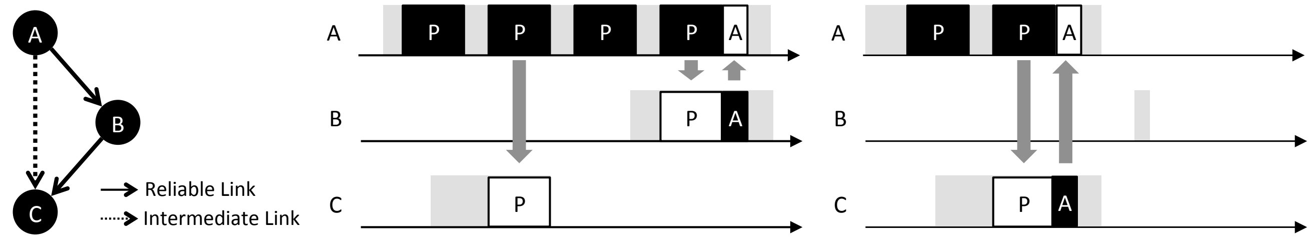
Complexity of Unique Forwarder Selection: Commonly, opportunistic routing relies on a consensus protocol to determine a unique forwarder among the receiving nodes. For example, each packet in ExOR contains a list of potential forwarders and their priorities. Due to the small packet size in sensor networks such forwarder lists are not feasible. Similarly, assigning time slots to each potential forwarder poses implementation challenges. We introduce a lightweight algorithm for unique forwarder selection tailored to the resource constraints in WSNs.

In this paper we argue that the concept of opportunistic routing, i.e., delaying the decision of selecting a forwarder until the packet has been received, is well suited for the large node densities and high link dynamics in WSNs. However, many aspects of its realization need to be revisited and adapted to the specific requirements of WSNs.

2.3 Basic Idea of ORW

ORW targets duty-cycled protocol stacks. For simplicity we here illustrate the basic concept of ORW utilizing an asynchronous low-power-listening MAC, such as in X-MAC [6][1]. In low-power-listening a sender transmits a stream of

[1] The concepts in ORW are generic and apply also to both synchronous (phase-locked) and receiver-initiated schemes. For example, in a synchronous MAC, ORW transmits when the first neighbor that provides routing progress is scheduled to wakeup up.



(a) Sample topology: Node A reaches C via B on reliable links or directly on an unreliable link.

(b) Traditional unicast routing in WSNs: Although C might overhear some transmission from A, packets are addressed to B to ensure stable routing.

(c) Opportunistic Routing in ORW: The first node that wakes up, receives a packet, and provides sufficient routing progress acknowledges and forwards it.

Figure 2: Basic idea of ORW: Utilizing the first woken neighbor as forwarder, ORW reduces energy consumption and delay. This exploiting of spatial and temporal link diversity also increase resilience to link dynamics.

packets until the intended receiver wakes up and acknowledges it (see Fig. 2b). To integrate opportunistic routing into duty cycled environments, we depart from this traditional unicast forwarding scheme in one key aspect: The first node that (a) wakes up, (b) receives the packet, and (c) provides routing progress, acknowledges and forwards the packet, see Fig. 2c. For example, in Figure 2a node A can reach node C either directly via an unreliable link or via B. Commonly, traditional routing ignores the unreliable link $A \to C$ and relies on $A \to B \to C$ for forwarding. ORW extends this, by also including $A \to C$ into the routing process: If $A \to C$ is temporary available and C wakes up before B, ORW will utilize it for forwarding. This reduces the energy consumption and delay (see Fig. 2c).

Our design enables an efficient adaptation of opportunistic routing to the specific demands of wireless sensor networks: (1) In contrast to opportunistic routing in mesh networks, forwarder selection in ORW focuses on energy efficiency and delay instead of network throughput: It minimizes the number of probes until a packet is received by a potential forwarder. (2) It integrates well into duty-cycled environments and ensures that many potential forwarders can overhear a packet in a single wakeup period. Thereby, ORW exploits spatial and temporal link-diversity to improve resilience to wireless link dynamics. (3) The fact that only a small number of nodes receive a probe at a specific point in time simplifies the design of a coordination scheme to select a single forwarder. This limits overhead of control traffic.

3. DESIGN

After discussing the basic concept of ORW, we present its core mechanisms: We introduce (1) EDC, an anycast metric to determine forwarders, (2) a new, coarse grained link estimator that reflects the requirements of opportunistic routing, and (3) unique forwarder selection tailored to wireless sensor networks. We highlight the differences to traditional routing in WSNs and discuss key challenges such as stability and avoiding routing loops, duplicates, and asymmetric links.

3.1 Anycast Routing Metric

In ORW, a packet is forwarded by the first awoken neighbor that provides routing progress. As a result, the routing topology towards a destination is not a tree anymore as in traditional unicast-based routing protocols. Instead, it assembles a Directed Acyclic Graph (DAG) with a single destination (Destination Oriented DAG, DODAG). In this DODAG, ORW allows each packet to traverse on a different route to the destination (anycast). Note that DODAGs are sometimes used instead of trees even in unicast-based routing protocols, such as RPL [31]. In this case, a single parent is selected before transmitting any packet.

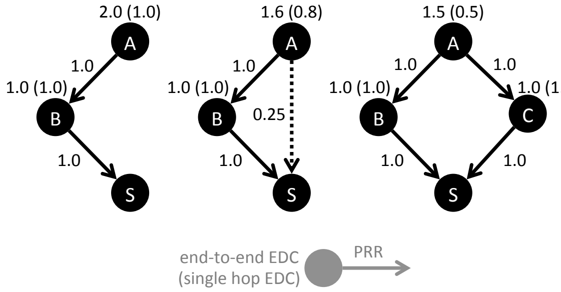
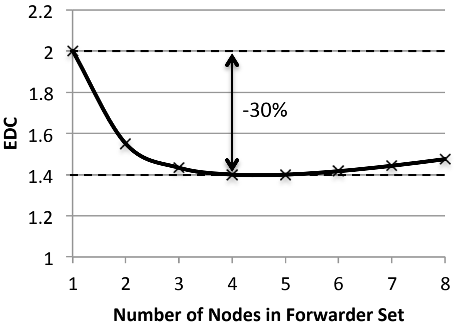


Figure 3: Example EDC for single and multi-path forwarding. EDC reduces delay and energy consumption by utilizing multiple, potentially unreliable links for forwarding. w is 0 in these examples. Please note that for a single path, EDC equals to ETX.

3.1.1 *Expected Duty Cycled Wakeups (EDC)*

ORW introduces EDC (Expected Duty Cycled wakeups) as routing metric. EDC is an adaptation of ETX [9] to energy-efficient, anycast routing in duty-cycled WSNs. EDC describes the expected duration, i.e., number of wakeups, until a packet has reached its intended destination, possibly across multiple hops. It is based on the following observation: Multiple routing choices decrease the waiting time until one of the potential forwarders wakes up and successfully receives the packet.

We define EDC as the sum of the expected time to reach a potential forwarder (also called single-hop EDC), the time to travel from the next hop to the final destination, plus a small constant accounting for the cost of forwarding. First, we focus on the single-hop EDC. To transmit a packet to the next hop over a reliable link, it takes on average one time unit, i.e., one average wakeup period, if there is one



Node	EDC	Use
1	1	✔
2	1.1	✔
3	1.2	✔
4	1.3	✔
5	1.4	✗
6	1.5	✗
7	1.6	✗
8	1.7	✗

Figure 4: Optimal forwarder set: the node has an EDC of 1.4 and includes 4 neighbors with an EDC strictly lower than 1.4 in its forwarder set. PRR is 1 and w is 0 in this example. When compared to single path routing, it reduces the number of expected wakeup periods to reach the sink by 30%.

neighbor[2]. It takes half a unit of time when two neighbors offer routing progress. EDC takes both the number of possible next hops and the quality of their links into account. We therefore define the single-hop EDC as the inverse of the sum of the link quality of the neighbors, as exemplified in Fig. 3, where:

- In the left case, A has a single neighbor with a perfect link, its single hop EDC is $1/1 = 1$;
- In the right case, A has two neighbors both having perfect links, its single hop EDC is $1/(1+1) = 0.5$;
- In the middle case, A has two neighbors with link qualities 1 and 0.25. Its single hop EDC is $1/(1+0.25) = 0.8$.

We define the EDC_i of node i for a given subset S_i of neighbors with link quality p_{ij} and EDC_j $(j \in S_i)$:

$$EDC_i(S_i) = \frac{1}{\sum_{j \in S_i} p_{ij}} + \frac{\sum_{j \in S_i} p_{ij} EDC_j}{\sum_{j \in S_i} p_{ij}} + w \qquad (1)$$

The first term is the aforementioned single hop EDC, it denotes how many units of time it requires on average to transmit a packet to one of the neighboring nodes in S_i. The second term describes the routing progress that the neighboring nodes in S_i offer, weighted by their link qualities p_{ij} (see Fig 3). The third term, w, adds a weight to reflect the cost of forwarding. Please note, for single path forwarding and a weight w of 0, EDC equals to ETX.

3.1.2 Forwarder Sets

We define the forwarder set F_i of a node i as the subset of its neighbor set that leads to the minimum EDC ($F_i \subseteq S_i$). Two key factors impact the forwarder set: (1) adding more neighboring nodes to the forwarder set reduces the time until one of the potential forwarders wakes up to receive. Hence, it decreases the single-hop EDC of the forwarding node and improves spatial diversity. However, (2) adding too many neighboring nodes to the forwarder set may decrease its average routing progress, as commonly not all neighbors provide good progress (see Eq. 1).

The forwarder set F_i is computed by adding nodes sorted by their EDC – starting with the lowest EDC – to the forwarder set and determining the set with the minimum EDC (see example in Fig. 4). F_i defines the EDC_i of a node i and all neighboring nodes j that provide routing progress, i.e., $EDC_j < EDC_i - w$, are utilized as potential forwarders. As a node only selects nodes that provide strictly more progress than itself, the resulting topology forms a loop free graph, i.e., is a DODAG. Due to space limitations, we leave a deeper discussion and analytical performance analysis of EDC as energy-efficient routing-metric as future work.

Overall, our design departs from traditional opportunistic routing such as ExOR or MORE [7] in the following key aspect: Commonly, opportunistic routing notes a prioritized forwarder set in the packet header. In contrast, in ORW all nodes providing routing progress potentially forward the packet. This leads to two important benefits: (1) instead of long address lists in the packet header denoting a forwarder set, which is not feasible in resource constrained WSNs, a single value that describes the EDC of the sender is sufficient. (2) It allows ORW to utilize spurious neighbors and neighbors it did not yet discover for forwarding.

3.1.3 Cost of Forwarding: w

The weight w is a constant value and describes the cost of forwarding a packet over one hop (see Eq. 1). Increasing w increases the routing progress that a forwarding neighbor j of a node i is required to provide to be included in the forwarder set F_i. Thus, increasing w leads to a smaller forwarder set. This is reflected in three effects: (1) It limits forwarding to nodes that provide high routing progress, leading to fewer hops until a packet reaches its destination. (2) However, reducing the size of the forwarder set increases delay and energy consumption for packet delivery. (3) A too low choice of w increases the risk of temporary routing loops, as packets are forwarded by nodes that provide even minimal routing progress (see Section 4.2). Overall, w allows ORW to balance delay and energy with routing progress and stability.

In our evaluation we determine a range of values for w that ensure stable routing (see Sec. 4.2). From this we choose a default configuration that provides both high performance and high stability. We show that this default is independent from individual deployments and holds across all our evaluation scenarios.

3.1.4 Loop Avoidance and Detection

Although the routing topology of ORW converges to a DODAG. i.e, is loop free, a slow spreading of updates, such as new EDC values of neighboring nodes, can lead to temporary loops. This is common in routing protocols and we address it with three standard techniques: (1) when a parent node is downgraded to a child node, a node observes this and forwards its packets without dropping duplicates. Additionally, these packets are delayed for a couple of milliseconds to allow the topology to stabilize. (2) A TTL field in each packet avoids infinite loops. (3) The forwarding cost w (see Sec. 3.1.3) ensures a threshold between forwarding nodes to avoid oscillating packets.

3.2 Link Estimation and Discovery

Anycast routing in ORW utilizes a pool of forwarders, where each packet potentially travels on a different route.

[2] For simplicity, we model both transmission success and the waking up of nodes to receive as Bernoulli processes.

Hence, when links to individual forwarders temporary fail or show reduced reliability, their impact on the overall quality of the forwarder set is limited. As a result, ORW does not require up-to-date estimates to each candidate forwarder, as traditional unicast routing.

Hence, we tailor link estimation and neighbor discovery in ORW to these specific demands. It mainly relies on overhearing: When a duty-cycled node in ORW wakes up to check for energy on the channel and subsequently receives a packet it (1) forwards it when providing routing progress, and (2) it updates its link quality estimate. For link estimation, a node maintains the link reception ratio from each neighbor. To this end, packets in ORW contain a header field that denotes the average rate at which a node is forwarding data. The link quality is obtained by dividing the rate of packets overheard from a neighbor by the forwarding rate of the same neighbor noted in the header field. As ORW operates with large forwarder sets and targets course grained link estimation, we argue that individual, asymmetric links have limited impact on the estimation and simplify link estimation by assuming $p_{ij} = p_{ji}$. Overall, this design departs from traditional link estimation used for unicast routing in WSNs in two key points: (1) stability, (2) limited use of probes.

Stability: While agility is a key design criteria for modern link estimators as they shall adapt quickly to changes in link quality, ORW may not even recognize when a link temporary fails, assuming it utilizes multiple links for forwarding. This is a design goal: as long as the aggregate of the neighbors performs stably, the dynamics of individual links will be masked. Aging slowly removes broken links from the forwarder set and neighbor table.

Limited Use of Probing: Traditional link estimation employs probing to determine the link qualities to neighboring nodes. In contrast, ORW commonly relies on overhearing during wakeups to update its neighbor table and link estimates. Probing is only utilized when not a single route is available. In our evaluation, we show that the routing topology in ORW converges quickly after boot-up and churn without the need for extensive probing. This reduces the overhead of control traffic in ORW.

To sum up, the bootstrap of a network using ORW is as follows: First, the nodes with no known parent probe their neighbors via a broadcast. The probing node i receives responses from a subset of its neighbors, depending on the link quality. For each received response from node j, node i adds the following entry to its neighbors table: (j, EDC_j, p_{ij}), where $p_{ij} = 1$. The link quality estimation, entirely based on overhearing, will refine the value of p_{ij} and may add new entries to the neighbor table, when overhearing a neighbor that didn't answer to the original probe. Node i won't use probing anymore, unless all its entries reach an estimated link quality of 0.

3.3 Unique Forwarder Selection

Once a packet has been received by one or more nodes in the forwarding set, the next step is to ensure that only a single one forwards it. In this section, we first show that in the majority of the cases a packet is only received by a single forwarder. Next, we introduce a lightweight coordination protocol to determine a unique forwarder in case the packet was received by multiple nodes.

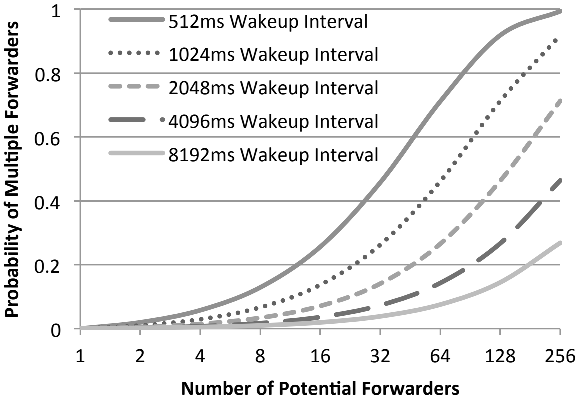


Figure 5: Probability that multiple forwarders receive the same packet for typical wakeup periods in ORW. For example, for 16 neighbors that provide routing progress, this probability is between 2% and 28% depending on the wakeup period.

3.3.1 Probability of Multiple Receivers

In ORW, a packet is forwarded by multiple nodes, if (1) multiple nodes are awake while the packet is transmitted and (2) more than one of these awake nodes successfully receives it and provides routing progress. This probability of multiple forwarders depends on two factors: the node density and the wakeup rate of each node. Both a high node density and a high wakeup rate increase the probability of a packet being received by multiple forwarders.

Figure 5 depicts the analytic probability of a packet being received by multiple forwarders for typical wakeup periods in ORW. For example, if 16 neighbors provide routing progress, the probability of multiple forwarders concurrently overhearing the same packet is between 2% and 28% for wakeup periods from 8192 to 512ms, respectively (assuming fully reliable links). The probability of multiple forwarders decreases with increasing wakeup intervals, a key benefit, as ORW targets low-power networking utilizing large wake-up intervals. Energy on the channel such as other data transmissions or noise, may extend the duration that a node is listening and hence may increase the risk of multiple receivers.

Overall, Figure 5 indicates that at our target wakeup rates a packet is received by only a small number of nodes, and commonly only by a single forwarder. This allows us to design a lightweight coordination protocol for ORW to determine a unique forwarder in case a packet was received by multiple nodes, which we introduce next.

3.3.2 Coordination Algorithm

The coordination protocol in ORW fulfills two tasks: (1) it determines the number of receivers of a packet, and (2) it ensures a unique forwarder in case of multiple receivers. It relies on three mechanisms:

Demand a Single Acknowledgment: Potentially, the sender receives multiple acknowledgments, one from each receiver[3]. In this case the sender retransmits the packet and both will (potentially) receive the packet again. Receiving

[3] The delay between a frame and its acknowledgement is bounded by the time for the receiver to take the forwarding decision, and an additional small random time we inject, to

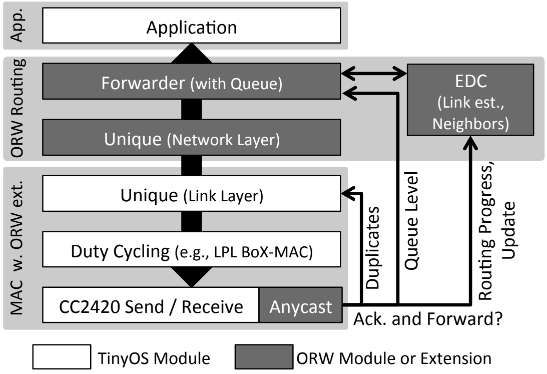


Figure 6: Cross-layer control flow in ORW: Before acknowledging and forwarding a packet, ORW checks whether (1) the node provides the requested routing progress, (2) has space in the queue, and (3) the packet is not a duplicate.

a link-layer duplicate, forwarders send a second acknowledgement only with 50% probability to reduce the number of duplicate acknowledgments. Furthermore, only the node that sent the final acknowledgment forwards the packet. The same mechanism is applied if acknowledgments collide, i.e., no acknowledgement is received (after a timeout) by the source.

Data Transmission Overhearing: When one nodes overhears another node forwarding the same packet while waiting for a clear channel, it cancels its own transmission.

Network-Layer Duplicate Detection: Network-layer duplicate detection serves as fall-back in case a packet slipped through the other mechanisms.

In corner case situations such as asymmetric or unstable links our practical design cannot guarantee a unique forwarder and packets may slip through. If they take different routes, this duplicate will only be detected at the sink. However, our evaluation shows that the lightweight mechanisms of ORW are sufficient to keep the duplicate rate at a level similar to traditional unicast routing such as CTP [14].

3.4 System Integration

ORW acts as replacement of the unicast forwarding logic of WSN routing protocols. As a case study we integrated ORW into CTP, the de-facto standard for collection protocol in TinyOS. In this section we discuss system integration, and the portability of our design.

ORW provides the same interfaces as CTP to the application, and uses its protocol headers, and in part its TinyOS modules such as the forwarder. Anycast routing in ORW relies on two headers: The EDC of a node and the required routing progress is stored as two 8-bit values in the 802.15.4 MAC header instead of the 16-bit destination address, which is not required for anycast routing. Thus, we allow EDC values from 0.0 to 25.5 at a granularity of 0.1. Overall, the integration into the 802.15.4 header allows a node to decide whether it provides the required routing progress after reading merely the header. Hence, it reduces energy consumption and ensures that 802.15.4 acknowledgments are triggered timely.

Additionally, we extend the CTP routing header with one field: we add a weighted average of the transmission rate to facilitate link estimation (see Sec. 3.2). ORW places a small interface between routing and MAC layer (see Fig. 6): To decide whether to accept, i.e., acknowledge and forward, a packet, it determines whether (1) the node provides routing progress, (2) has space in its queue, and (3) the packet is not a duplicate (see Sec. 3.3.1). Hence, although ORW places functionality on both the routing layer and the MAC layer, its design is not bound to a specific routing protocol, MAC layer, or duty cycling scheme. ORW including link estimation requires slightly less RAM and ROM than CTP and its link estimator 4BitLE [13] which is mainly due to our simplified link estimator.

4. EVALUATION

In this section we evaluate ORW: we show (1) a simulation-based evaluation focusing on the EDC metric and its differences to ETX, and (2) use two large testbeds for a detailed, experimental comparison of ORW and CTP. We focus on four key metrics: radio duty-cycle, end-to-end delay, reliability and transmission counts.

4.1 Simulation: Anycast EDC

In our simulation-based evaluation, we explore the potential of anycast routing with EDC in terms of delay, and hops when compared to unicast ETX. Also, we explore the impact of network density on the performance of EDC-based routing and we evaluate the influence of the transmission cost w on EDC (see Sec. 3.1.1).

4.1.1 Simulation Setup

In our simulations, we solely focus on EDC and ETX as routing metrics and not individual protocol implementations. Thus, we compare two idealized protocols in this section: anycast routing with EDC and unicast routing with ETX.

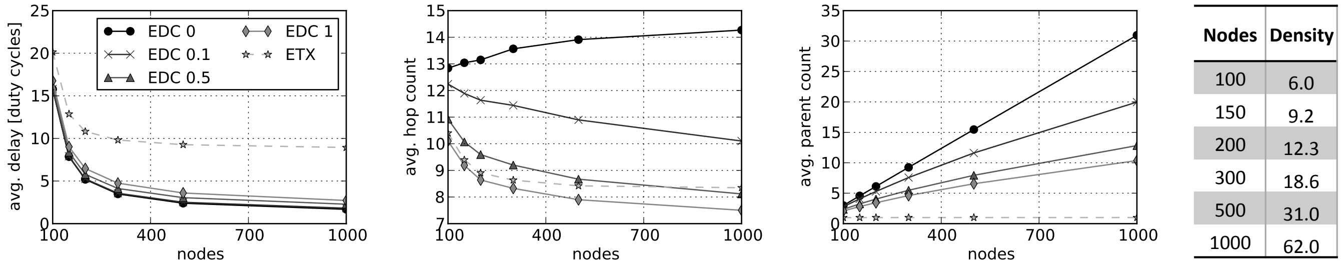
We use randomly generated topologies ranging from 100 to 1000 nodes placed in a fixed area and employ the Friis transmission model for radio propagation in a custom simulator; results are averaged over 100 random topologies per data point. For each topology we determine the neighbor sets of all nodes and link qualities, i.e., PRR, between them. On top of this, we deploy our idealized protocols. Hence, in this simulation-based evaluation we deliberately exclude protocol mechanisms outside of the routing metric itself such as link estimation or neighbor discovery. Additionally, this allows us to avoid protocol artifacts such the slow spreading of route updates or packet collisions.

Overall, our goal is to evaluate the underlying performance of our routing metric EDC independent of a specific protocol implementation, before we compare EDC-based anycast routing in ORW to CTP in our testbed based evaluation (see Sec. 4.2).

4.1.2 Anycast EDC and Unicast ETX

Figure 7 shows that anycast routing with EDC benefits from an increased network density much more than unicast-based ETX. Compared to ETX, EDC reduces the average delay for packet delivery by factors of 1.3 and 6 for network sizes of 100 and 1000, respectively (see Fig. 7a). We later show that this leads to energy savings on a similar scale (see

guarantee non-constant timing and make acknowledgement collision more unlikely.



Nodes	Density
100	6.0
150	9.2
200	12.3
300	18.6
500	31.0
1000	62.0

(a) **Delay:** EDC outperforms ETX by a factor of 1.3 to 6, depending on network size and choice of w. For w of 0, EDC shows the lowest delay.

(b) **Hops:** For w of 1, EDC outperforms ETX in hops. For other configurations it trades lower delay for higher hop counts.

(c) **Parents:** Increasing network size and density allows EDC to utilize more parents for forwarding. Increasing w reduces their number.

(d) **Density:** average node densities.

Figure 7: Simulation-based evaluation: The results show that anycast routing with EDC benefits from increased density much more than unicast routing with ETX. Nodes are placed in an area of fixed size, leading to increased network density when the number of nodes increases.

Sec. 4.2). Commonly, EDC leads to hop counts larger than ETX (see Fig. 7b). However, for w of 1 it outperforms ETX.

Figures 7c and 7d show that with $w = 0$, EDC utilizes about half of the neighboring nodes as parents. This decreases to 15% of the neighbors for $w = 1$. Overall, EDC with $w = 0$ utilizes all forwarders that provide even the smallest routing progress. Hence, while minimizing delay, this increases the hop count when network density increases and as a result the number of parents increases, too. However, this aggressive forwarding, also makes this configuration of EDC sensible to link dynamics and potentially leads to routing loops, as we show in Section 4.2. In the experimental evaluation we next show that a configuration of w slightly above 0, such as 0.1, avoids these problems while utilizing many parents and providing low delay and high energy-efficiency.

4.2 Testbed Based Evaluation of ORW

In this section we evaluate ORW on real-world testbeds and compare its performance to CTP, the de-facto standard collection protocol in TinyOS. Please, note that the concepts in ORW are generic and independent of the chosen duty cycling scheme: They apply to both asynchronous and synchronous (phase-locked) MAC schemes as well as receiver-initiated ones (see Section 2.3). However, to ensure a fair comparison with CTP, we use BoX-MAC in this evaluation. It is the default MAC in both TinyOS and CTP; and resembles a combination of X-MAC and B-MAC [25]. We also show results for CTP with A-MAC [12], a state-of-the art synchronous, receiver initiated MAC. We do not compare to SCP-MAC [33], another synchronous MAC. Although it promises high energy efficiency, it is not available for current TinyOS releases, and recent work [30] indicates that its energy efficiency in multi-hop collection trees is well below BoX-MAC.

4.2.1 *Testbeds and Metrics*

We base our evaluation on two testbeds: Indriya [10] and Twist [16], with 120 and 96 nodes, respectively. For each we use two levels of transmission power, resulting in four evaluation scenarios (see Table 1). We use the following setup for both ORW and CTP: Every node generates a packet randomly with an average interval of 4 minutes, and the network forwards it to the sink. Unless explicitly mentioned, we use wakeup interval of 2 seconds (with our settings, this leads to the optimal duty cycle in CTP, see Sec. 4.2.6). We use corner nodes 1 and 229 as sink nodes on Indriya and Twist, respectively.

Test- **bed**	**Size** nodes, m^3	**Sink** id	**Tx Power** dBm	⌀ hops
Indriya	120,	1	0	5.6
	50 x 25 x 20	1	-10	9.1
Twist	96,	229	0	3.4
	30 x 13 x 17	229	-25	7.1

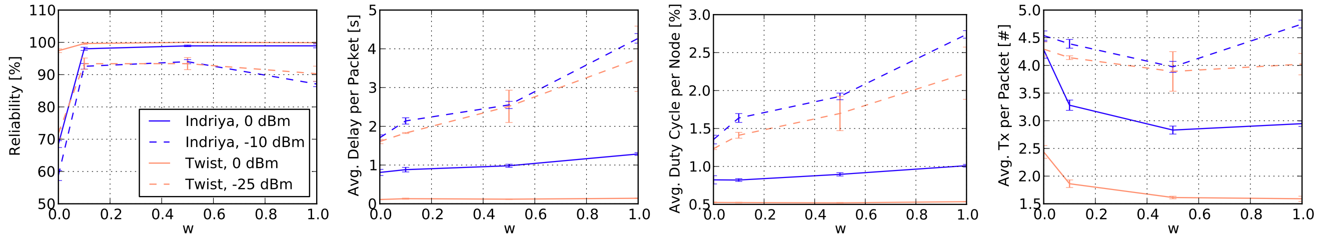
Table 1: We use four evaluation scenarios: two testbeds with two different power levels each. Each testbed contains about 100 nodes and the diameter ranges from 3 to 9 hops.

We evaluate energy consumption through the average duty cycle in the network, i.e., the portion of time spent with the radio chip turned on. The average duty cycle is a good proxy for energy consumption because (1) the radio chip consumes far more power than the other hardware components involved in our experiments and (2) low-power radio chips in sensor motes have a comparable power draw when transmitting or listening. This metric provides us with results that are independent from environmental conditions, hold across different hardware platforms, and are therefore reproducible. For a fair comparison we also skip the first two minutes when measuring duty cycles, as CTP shows a high duty cycle in this time due to its initial link probing.

For each data point, experiments are executed for a minimum of 30 minutes and are repeated three times; Experiments are executed at random times of the day, but back-to-back to ensure fairness. We display average results and error bars show standard deviations. Overall, the results shown are based on more than 300 individual experiments, each between 30 minutes and 2 hours.

4.2.2 *System Calibration: w*

We start the testbed-based evaluation by calibrating the transmission cost w. Figure 8 shows that in all four sce-



(a) **Reliability:** A w of 0 is prone to routing loops, causing packet loss.

(b) **Delay:** Increasing the transmission penalty w increases the delay.

(c) **Duty Cycle:** Increasing w increases the duty cycle.

(d) **Tx:** The transmission count decreases when increasing w.

Figure 8: **System Calibration**: a forwarding cost w of 0.1 is a good balance between energy efficiency, delay, and reliability.

narios, a low w leads to the best performance in terms of delay and duty cycle. A larger w reduces hop counts at the price of increased delay and duty cycle. However, reliability shows a sharp drop for w of 0. Our logs indicate that a w of 0 increases the risk of routing loops and duplicate packets, which increase packet drops and reduce reliability. Please note, that ORW limits us to evaluate w at a granularity of 0.1 (see Sec. 3.4).

Overall, all of our four evaluation scenarios show that a value of 0.1 for w provides a stable balance between reliability, delay, and duty cycle. As these scenarios cover a wide range of network densities and network diameters (see Table 1), we believe that a w of 0.1 is a good choice in general and use it as default in ORW.

4.2.3 *Per Node Comparison of ORW to CTP*

Next, we compare the performance of ORW and CTP in our four evaluation scenarios (see Table 1). Figure 9 and Table 2 show that ORW significantly improves duty cycles and delay. On average, ORW roughly doubles the energy efficiency, individual nodes show improvements up to 88%. The results show that ORW strongly benefits from network density: it shows the best results on the dense Twist deployment at a transmission power 0 dBm. Additionally, it improves delay by 30% to 90% depending on network density and achieves (re)transmission counts (unicast or anycast) that are similar, but slightly higher when compared to CTP.

In ORW, nodes in dense networks and the ones further away from the sink benefit the most from spatial diversity in anycast forwarding (see Fig. 10). We define spatial diversity as the number of different, i.e., unique, forwarders that are utilized per hop on the path from a node to the sink during the course of the experiment. As another benefit of anycast forwarding, ORW removes outliers both in terms of duty cycles and delay (see Fig. 9). This has two key advantages: (1) It reduces the time until the first node runs out of energy and hence, ensures that sensor coverage etc. can be maintained longer. (2) In terms of delay it allows us to switch to lower duty cycles in delay sensitive applications, what in turn reduces energy consumption even further.

4.2.4 *Impact of Churn on ORW and CTP*

We evaluate the impact of churn on both ORW and CTP. Each 15 minutes, we remove on average 10 nodes from the network. Throughout the course of two hours this reduces the number of nodes from 120 to about 30 on the Indriya testbed.

Figure 11 depicts the impact of churn on the key metrics of reliability, transmissions, duty cycles, and delay. Both protocols show spikes of reduced reliability under churn. For increased churn these spikes grow strongly for CTP. Hence, ORW maintains connectivity much longer in the resulting sparse network. Additionally, it shows the benefits of anycast routing in ORW over unicast routing in CTP: churn has only minimal impact on the duty cycle, delay, or transmission of ORW, while these show sharp peaks in CTP.

4.2.5 *Convergence of ORW*

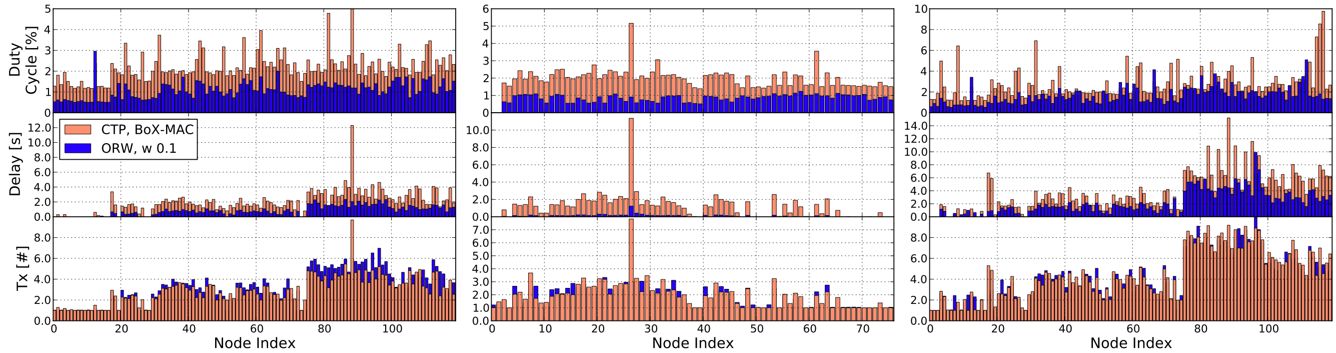
As ORW essentially operates without probing for link estimates (see Sec. 3.2), we discuss its convergence in this section. We track the evolution of the average EDC as well as number of neighbors and parents per node. Figure 12 shows that the routing metric EDC reaches an initial stable point within the first five minutes without the need for extensive beaconing or detailed link estimation. Over time it optimizes slightly.

Similarly, nodes in ORW continue to add new neighbors for routing. However, these lead to only minimal improvements, as we see only minimal changes to the duty cycle and delay over time (see Fig. 11). Overall, the results show that ORW stabilizes quickly without the need for expensive probing for link estimation and neighbor discovery. Additionally, Figure 11 shows that ORW maintains this stability even under churn.

4.2.6 *Choice of Wakeup Interval*

The previous experiments used a wakeup interval of 2 seconds, i.e., a node wakes up every two seconds to receive data from neighboring nodes. We used this interval to ensure a fair comparison, as it leads to the optimal duty cycle in CTP and its default BoX-MAC at an inter-packet interval of 4 min (see Fig. 13a).

In this section, we discuss the impact of the wakeup interval on duty cycle, delay, and reliability. Figures 13a and 13e show that ORW benefits much more than CTP from reduced wakeup intervals. The figures also depict the idle duty cycle, i.e., the energy that is consumed by just the wakeups of BoX-MAC without any data transfer. This baseline defines the lower bound for the duty cycle. For Indriya (see Fig. 13a) ORW stays closer to this line than CTP and in the dense Twist testbed (see Fig. 13e) ORW is marginally above



(a) Indriya, 0 dBm. (b) Twist, 0 dBm. (c) Indriya, -10 dBm.

Figure 9: **Per Node Comparison of ORW and CTP:** ORW improves duty cycles and delays while achieving slightly higher hop counts than CTP.

Testbed	Duty Cycle					Delay			Tx		
	Trace [%]		Improve [%]			Trace [s]		Impr. [%]	Trace [#]		Impr. [%]
	ORW	CTP	Avg.	Max.	Min.	ORW	CTP	Avg.	ORW	CTP	Avg.
Indriya, 0 dBm	1.1	2.2	50	79	-19	0.8	2.0	58	3.3	3.0	-11
Indriya, -10 dBm	1.6	2.8	41	88	-30	2.1	3.8	44	4.4	4.5	0
Twist, 0 dBm	0.8	2.0	57	82	0	0.1	1.2	91	1.8	2.0	10
Twist, -25 dBm	1.4	2.1	33	76	-39	1.8	2.4	29	4.1	3.8	-10

Table 2: **Testbed Experiments Summary:** ORW decreases average duty cycles up to 57% and delays up to 90%, while achieving similar, but slightly higher transmission count than CTP.

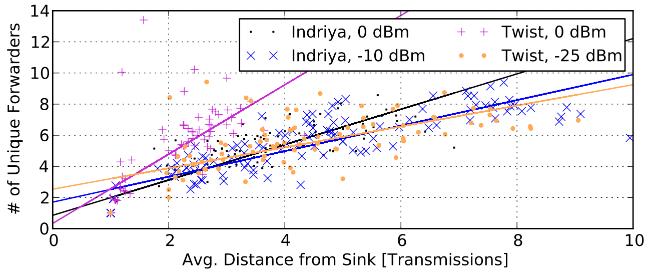


Figure 10: **Spatial Diversity (Number of Unique Forwarders):** nodes in dense networks and the ones further away from the sink exploit spatial diversity, i.e., using different forwarders, in anycast routing the most. We plot data points and their linear regression.

the baseline throughout all experiments. These results show that ORW efficiently exploits network density. Delay increases with increased wake-up intervals (see Fig. 13b and 13f). However, the increase for ORW is significantly lower than for CTP with BoX-MAC. Figures 13d and 13h show the resulting duty cycle for a given average delay. This underlines that ORW can operate at much lower duty cycles for a given average delay.

Both CTP and ORW show high reliability (see Fig. 13c and 13g). However, at high wakeup intervals reliability of CTP decreases due to queue overflows on individual nodes. In contrast, ORW avoids this by using multiple forwarders. The results for other inter-packet intervals, radio channels, and testbeds show similar performance gains of ORW over CTP. As reference, the figures also depict results for CTP on A-MAC: However, it does not reach the performance of BoX-MAC for multihop routing with CTP. While we cannot conclude on the exact reasons, our traces indicate that its probes and loss of synchronization at low wakeup-rates increase duty cycles and delays.

4.3 Discussion and Limitations

After evaluating our anycast routing scheme in simulation and comparing it to the state of the art CTP collection protocol in testbeds, we reflect on the results and discuss limitations in this section.

4.3.1 Discussion

Our results show that ORW improves duty cycles and delays significantly while achieving similar reliability and transmission counts when compared to the state of the art. Our results show an average decrease in duty cycle by about 50%, individual nodes improve up to 90%. Similarly, it decreases delay by a 30% to 90% depending on network density.

Anycast forwarding allows ORW to forward a packet faster than traditional unicast routing. Overall, such a design works best at high network densities, as this gives the most choices for forwarding. As a result, ORW shows the best results for dense topologies, i.e., in both testbeds at high transmission power. Similarly, its optimal duty cycle is at lower wakeup rates when compared to CTP. Hence, our results show that ORW can operate at much lower wakeup rates without major impact on reliability or delay (see Fig. 13). At lower densities, ORW still outperforms CTP, but its benefits decrease (see Table 2). Similarly, at high wakeup rates, the delay and energy advantages of ORW decrease.

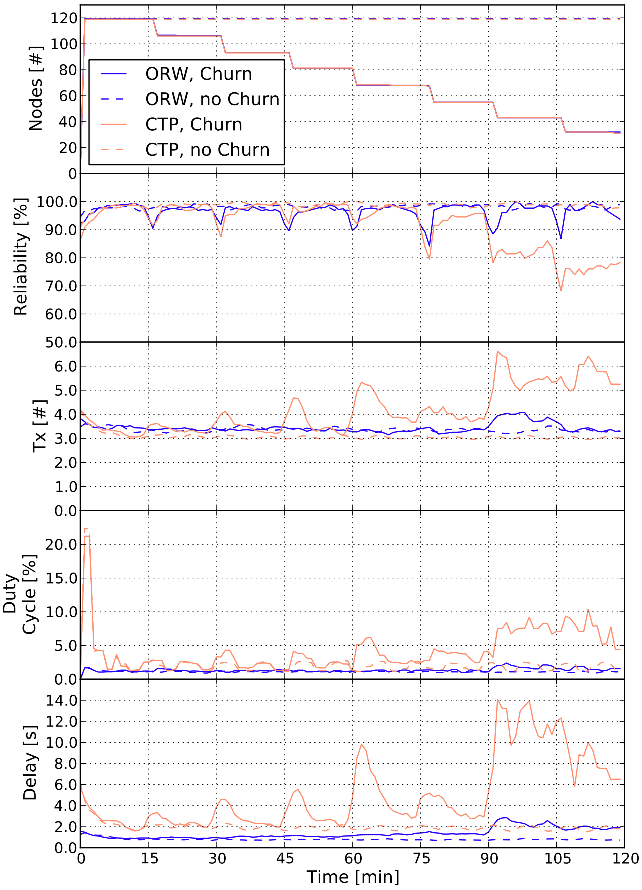


Figure 11: **Churn (Indriya, 0 dBm):** While both ORW and CTP achieve similar reliability under churn, CTP pays a higher price in terms of energy, delay, and transmissions.

Additionally, its opportunistic nature allows ORW to take the state of wireless link into account and delay the decision of selecting a forwarder until the packet has been received. As a result, it reflects temporal and spatial diversity of wireless links (see Fig. 10) and increases the resilience of routing to link dynamics and churn (see Fig. 11).

4.3.2 Limitations

ORW targets applications with lifetime demands in the order of month or years, which are typical deployment scenarios in WSNs. Commonly, such applications rely on duty-cycled low-power networking with wake-up rates in the order of seconds. Our evaluation shows that ORW achieves the strongest improvement at such low wake-up rates when compared to CTP (see Fig. 13). At higher wakeup rates the baseline cost of the MAC layer accounts for the majority of the cost, and both CTP and ORW show similar performance in terms of energy and delay (see idle line in Fig. 13a and 13e). Additionally, ORW focuses on collection applications with low data rates. Thus, we believe that its design is not well suited for high throughput settings such as bulk transfers.

While ORW is agnostic to the underlying MAC scheme, it shows the strongest improvements for asynchronous MAC layers. For synchronous MAC layers, i.e., with phase locking, we expect ORW to show similar improvements for delay but limited benefits in terms of energy. However, in dense deployments such as Twist (see Fig. 13e) the duty cycle in ORW closely approaches the idle base line of the MAC layer. This is also the cost of an ideal synchronous MAC, without considering its overhead in terms of time synchronization and guard times.

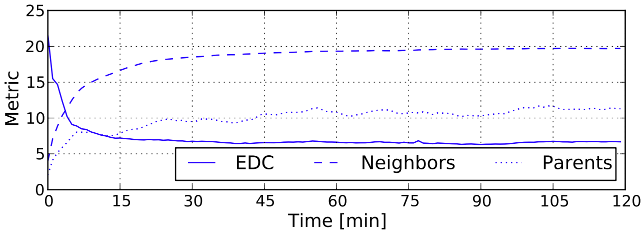


Figure 12: **Convergence of ORW (Indriya, 0 dBm):** Average per node values for EDC, neighbors in the routing table and parents selected in the forwarder set.

To ensure a fair comparison with CTP, our current implementation of ORW is tailored to collection applications with a single sink. Thus, we currently do not support mesh routing (or multiple sinks). However, the design of ORW is generic: When applications require it, this can be directly integrated by adding an extra header field that notes the intended destination next to the already existing requested routing progress (see Sec. 3.4).

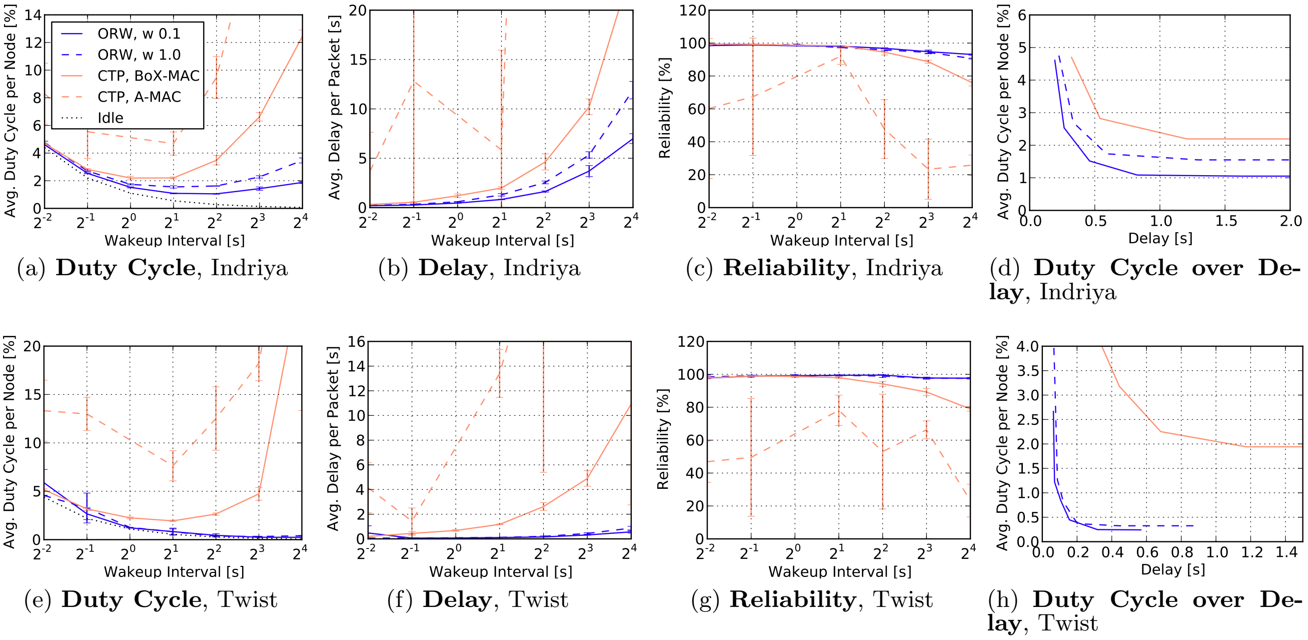
5. RELATED WORK

In this section we discuss related work on opportunistic and adaptive routing in WSNs. Opportunistic routing itself is discussed in the preliminaries in Sec. 2.1.

GeRaF [34] pioneered the concept of anycast routing in duty-cycled wireless sensor networks. It utilizes geographic routing to determine routing progress of its neighboring nodes and a busy tone protocol to ensure a unique forwarder. CMAC [20] combines the concepts of GeRaF and ExOR: It includes prioritized forwarders, slotted acknowledgments and overhearing of acknowledgments to determine a unique forwarder as in ExOR. Relying solely on geographic routing, both do not address the key challenges for opportunistic routing in duty-cycled WSNs such as anycast routing metrics and wireless link dynamics.

The application of opportunistic routing in WSNs also received great attention from a more theoretical perspective [2, 4, 11, 17, 18, 22, 24, 26, 29]. Similar to our work, these mainly focus on energy and delay instead of throughput as core metrics; and include duty-cycled nodes. Their models and simulation results show that opportunistic routing can improve energy efficiency and delay when compared to traditional unicast routing. While they omit the practical challenges that this paper addresses, their results strongly motivated our work. Other approaches to opportunistic forwarding [3, 21] are routing agnostic and do not include energy efficient routing metrics nor tailor link estimation to anycast routing.

Adaptive and low-power routing in WSNs proposes dynamic change of parents in routing. DSF [15] selects the next hop of a packet based on the sleep schedule of neighboring nodes and different metrics such delay, reliability, and



(a) **Duty Cycle**, Indriya (b) **Delay**, Indriya (c) **Reliability**, Indriya (d) **Duty Cycle over Delay**, Indriya

(e) **Duty Cycle**, Twist (f) **Delay**, Twist (g) **Reliability**, Twist (h) **Duty Cycle over Delay**, Twist

Figure 13: **Choice of Wakeup Intervals:** Performance of ORW and CTP on Indriya and Twist (with TX power of $0dBm$) for wakeup intervals between 0.25 and 16 seconds: ORW can operate at much lower wakeup intervals than CTP while achieving low delays and high reliability. Some data points for A-MAC are omitted due to inconsistent results.

energy consumption. Similar to ORW, DSF shows strong improvements over unicast routing in these metrics. However, it focuses on synchronized networks. Furthermore, it requires iterative message exchanges to stabilize the forwarding schedules of all nodes, leading to control traffic overhead in the presence of dynamic links.

The Backpressure Routing Protocol, BRP [23], forwards packets to the neighbor with the lowest queue level. This improves throughput when compared to traditional unicast routing, while increasing delay. However, BRP can only be applied when the overall system is saturated, i.e., nodes always have packets to forward. This is rare in WSN deployment scenarios as these commonly show low traffic rates. BRE [1] reduces hop counts by exploiting link dynamics: when a far ranging link of intermediate quality becomes temporary available, BRE uses it to short-cut in the routing tree. In duty-cycled environments, BRE shows two key limitations: (1) its short-cuts are only stable for a couple of milliseconds, making it difficult to exploit them in low traffic scenarios. (2) In BRE, nodes overhear data traffic to determine possible short-cuts. This is not practical when nodes are asleep most of the time.

6. CONCLUSIONS

This paper introduces opportunistic routing for Wireless Sensor Networks (ORW). It targets applications with low duty cycles. A packet in ORW is forwarded by the first awoken node that successfully receives it and offers routing progress. This provides two key benefits: By utilizing all neighbors as possible next hops, ORW reduces delay and energy consumption significantly when compared to unicast routing. Additionally, it improves the resilience to link dynamics and churn.

Overall, ORW tailors the concept of opportunistic routing to the specific demands of WSNs. It integrates duty-cycled nodes and relies on energy and delay as key routing metrics. To enable opportunistic routing in WSNs, this paper introduces (1) a new anycast routing metric to reflect the multi-path nature of opportunistic routing, (2) mechanisms for selecting unique forwarders in duty-cycled opportunistic routing and (3) coarse-grained, long-term link estimation.

Our results show that ORW doubles energy efficiency in dense networks. It reduces duty cycles on average by 50% and delays by 30% to 90% while achieving reliability and transmission counts similar to the state of the art.

Acknowledgments

We would like to thank our shepherd, Chieh-Jan Mike Liang, and the anonymous reviewers for their feedback and insightful comments. We thank Adam Dunkels and Fredrik Österlind for providing feedback throughout this work and Philip Levis for motivating us to look into opportunistic routing. The first author gratefully thanks M. Hamad Alizai, Klaus Wehrle, et al. for the joint work on BRE [1], showing that highly adaptive approaches can be practical. This work was funded in part by SSF, the Swedish Foundation for Strategic Research, through the Promos project.

7. REFERENCES

[1] M. H. Alizai, O. Landsiedel, J. A. Bitsch Link, S. Götz, and K. Wehrle. Bursty Traffic Over Bursty Links. In *SenSys: Proc. of the ACM Int. Conference on Embedded Networked Sensor Systems*, 2009.

[2] F. Ashref, R. H. Kravets, and N. H. Vaidya. Exploiting Routing Redundancy using MAC Layer Anycast to Improve Delay in WSN. *SIGMOBILE Mob. Comput. Commun. Rev.*, 14, 2010.

[3] M. Autenrieth and H. Frey. PaderMAC: A Low-Power, Low-Latency MAC Layer with Opportunistic Forwarding Support for Wireless Sensor Networks. In *ADHOC-NOW'11: Proc. of the 10th Int. Conference on Ad-Hoc, Mobile, and Wireless Networks*, 2011.

[4] P. Basu and C.-K. Chau. Opportunistic Forwarding in Wireless Networks with Duty Cycling. In *CHANTS: Proc. of the ACM Workshop on Challenged Networks*, 2008.

[5] S. Biswas and R. Morris. ExOR: Opportunistic Multi-Hop Routing for Wireless Networks. In *SigComm: Proc. of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2005.

[6] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-MAC: a Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. In *SenSys: Proc. of the ACM Int. Conference on Embedded Networked Sensor Systems*, 2006.

[7] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading Structure for Randomness in Wireless Opportunistic Routing. In *SigComm: Proc. of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2007.

[8] R. R. Choudhury and N. H. Vaidya. MAC-Layer Anycasting in Ad Hoc Networks. *SIGCOMM Comput. Commun. Rev.*, 34(1), 2004.

[9] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. In *MobiCom: Proc. of the ACM Int. Conference on Mobile Computing and Networking*, 2003.

[10] M. Doddavenkatappa, M. C. Chan, and A. Ananda. Indriya: A Low-Cost, 3D Wireless Sensor Network Testbed. In *TridentCom: Proc. of the Int. ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, 2011.

[11] H. Dubois-Ferriè andre, M. Grossglauser, and M. Vetterli. Valuable Detours: Least-Cost Anypath Routing. *IEEE/ACM Trans. Netw.*, 19(2), 2011.

[12] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis. Design and Evaluation of a Versatile and Efficient Receiver-Initiated Link Layer for Low-Power Wireless. In *SenSys: Proc. of the ACM Int. Conference on Embedded Networked Sensor Systems*, 2010.

[13] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis. Four Bit Wireless Link Estimation. In *HotNets: Proc. of the Workshop on Hot Topics in Networks*, 2007.

[14] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection Tree Protocol. In *SenSys: Proc. of the ACM Int. Conference on Embedded Networked Sensor Systems*, 2009.

[15] Y. Gu and T. He. Data Forwarding in Extremely Low Duty-Cycle Sensor Networks with Unreliable Communication Links. In *SenSys: Proc. of the ACM Int. Conference on Embedded Networked Sensor Systems*, 2007.

[16] V. Handziski, A. Köpke, A. Willig, and A. Wolisz. TWIST: a Scalable and Reconfigurable Testbed for Wireless Indoor Experiments with Sensor Networks. In *Proc. of the Int. Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality*, 2006.

[17] J. Kim, X. Lin, and N. B. Shroff. Optimal Anycast Technique for Delay-Sensitive Energy-Constrained Asynchronous Sensor Networks. In *InfoCom: Proc. of the IEEE Int. Conference on Computer Communications*, 2009.

[18] J. Kim, X. Lin, N. B. Shroff, and P. Sinha. Minimizing Delay and Maximizing Lifetime for Wireless Sensor Networks with Anycast. *IEEE/ACM Trans. Netw.*, 18, 2010.

[19] P. Larsson. Selection Diversity Forwarding in a Multihop Packet Rradio Network with Fading Channel and Capture. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5, 2001.

[20] S. Liu, K.-W. Fan, and P. Sinha. CMAC: An Energy-Efficient MAC Layer Protocol using Convergent Packet forwarding for Wireless Sensor Networks. *ACM Trans. Sen. Netw.*, 5, 2009.

[21] S. Liu, M. Sha, and L. Huang. ORAS: Opportunistic Routing with Asynchronous Sleep in Wireless Sensor Networks. In *ICFCC'10: Proc. of the 2nd Int. Conference on Future Computer and Communication*, 2010.

[22] X. Mao, X.-Y. Li, W.-Z. Song, P. Xu, and K. Moaveni-Nejad. Energy Efficient Opportunistic Routing in Wireless Networks. In *MSWiM: Proc. of the 12th ACM Int. Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2009.

[23] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali. Routing without Routes: the Backpressure Collection Protocol. In *IPSN: Proc. of the ACM/IEEE Int. Conference on Information Processing in Sensor Networks*, 2010.

[24] B. Pavković, F. Theoleyre, and A. Duda. Multipath Opportunistic RPL Routing over IEEE 802.15.4. In *MSWiM: Proc. of the ACM Int. Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2011.

[25] J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *SenSys: Proc. of the ACM Int. Conference on Embedded Networked Sensor Systems*, 2004.

[26] G. Schaefer, F. Ingelrest, and M. Vetterli. Potentials of Opportunistic Routing in Energy-Constrained Wireless Sensor Networks. In *EWSN: Proc. of the European Conference on Wireless Sensor Networks*, 2009.

[27] K. Srinivasan, M. Jain, J. I. Choi, T. Azim, E. S. Kim, P. Levis, and B. Krishnamachari. The κ Factor: Inferring Protocol Performance using Inter-Link Reception Correlation. In *MobiCom: Proc. of the ACM Int. Conference on Mobile Computing and Networking*, 2010.

[28] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis. The β Factor: Measuring Wireless Link Burstiness. In *SenSys: Proc. of the ACM Int. Conference on Embedded Networked Sensor Systems*, 2008.

[29] S. Unterschütz, C. Renner, and V. Turau. Opportunistic, Receiver-Initiated Data-Collection Protocol. In *EWSN: Proc. of the European Conference on Wireless Sensor Networks*, 2012.

[30] J. Vanhie-Van Gerwen, E. De Poorter, B. Latré, I. Moerman, and P. Demeester. Real-Life Performance of Protocol Combinations for Wireless Sensor Networks. In *Proc. of the IEEE Int. Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2010.

[31] T. Winter (Ed.), P. Thubert (Ed.), and RPL Author Team. RPL: IPv6 Routing Protocol for Low power and Lossy Networks. Internet Draft draft-ietf-roll-rpl-19, work in progress.

[32] A. Woo, T. Tong, and D. Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In *SenSys: Proc. of the ACM Int. Conference on Embedded Networked Sensor Systems*, 2003.

[33] W. Ye, F. Silva, and J. Heidemann. Ultra-Low Duty Cycle MAC with Scheduled Channel Polling. In *SenSys: Proc. of the ACM Int. Conference on Embedded Networked Sensor Systems*, 2006.

[34] M. Zorzi and R. R. Rao. Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance. *IEEE Trans. on Mobile Computing*, 2, 2003.