O-CTP: Hybrid Opportunistic Collection Tree Protocol for Wireless Sensor Networks

Joakim Flathagen*‡¶, Erlend Larsen*, Paal E. Engelstad§¶ and Øivind Kure‡¶
*Norwegian Defence Research Establishment (FFI), ‡Q2S NTNU, ¶UNIK, §University of Oslo
Email: {jfi,erl}@ffi.no,{paale,okure}@unik.no

***Abstract*—Radio interference or deliberate jamming attacks can cause highly unpredictable communication in Wireless Sensor Networks (WSNs). Most prevalent WSN platforms consist of low-cost hardware with no effective measures against these threats. Most proposed countermeasures require a more advanced hardware design or radical changes to the 802.15.4 MAC protocol. These alternatives can be very difficult or even impossible to apply to existing WSN designs. In this paper we do not attempt to change the hardware or the MAC protocol. Instead we investigate how WSN routing protocols behave when the network is affected by interference. The paper proposes enhancements of CTP, the de-facto tree-based routing protocol for WSN, using opportunistic routing. We compare our approach with a wide range of protocols: CTP, TYMO, MultihopLQI, broadcast and geographic opportunistic routing in a real-life TelosB testbed subjected to different interference levels. The results show that our hybrid protocol, O-CTP, both improves the data delivery rate and reduces the cost when compared to standard routing protocols.**

***Index Terms*—Interference, Jamming, Opportunistic routing, Wireless Sensor Networks**

I. Introduction

Wireless Sensor Networks (WSNs) often suffer from highly unpredictable wireless communication conditions. The quality of the communication depends on several factors such as the deployed environment, the frequency spectrum and modulation schemes utilized, and the communication devices themselves [1]. The multi-hop nature of WSNs further increases the problem. Results on deployed networks and testbeds show that typical delivery ratios are between 70 and 99% [2]–[4], but could even go as low as 20-40% [5]–[7]. One reason for the unpredictable packet delivery rate is that the wireless channel fluctuates significantly with time. People or vehicles entering the sensed area, or even rain and wind, give unreliable RF propagation. Interference in the chosen frequency band adds further weight to the problem. For an IEEE 802.15.4 equipped sensing node operating at 2.4GHz, possible sources of interference include other radio transmitters operating in the same frequency band (e.g., 802.11, Bluetooth or video transmitters), harmonic interference from other bands, microwave ovens and military radars. An opponent may also use interference intentionally to disrupt communications (i.e., radio jamming) [8].

Much work has been dedicated to create effective measures against interference and jamming in WSNs. The most effective methods involve changes to the physical layer, e.g., moving from the standard Direct-sequence spread spectrum (DSSS) in 802.15.4 to Frequency-hopping spread spectrum (FHSS) or using directional antennas. Some methods focus on changing the MAC protocol [6]. Few of these countermeasures can, however, be effectively applied to the prevalent WSN platforms today (i.e., TelosB, Mica and IRIS), without redesigning the platform. The focus in this article is therefore to study how the delivery rate can be maximized even in interfered environments, simply by choosing the routing protocol cleverly.

Traditional routing protocols for WSNs deal with dynamics in the underlying network structure by using various metrics, e.g., the number of hops [9], radio link quality [10] or Expected Transmission Count (ETX) [4]. Despite these attempts, the metric calculations have difficulties in coping with the rapid changes in the unreliable wireless medium, making it difficult to choose the optimal next hop node. This observation has led to the development of opportunistic routing [11]–[13]. Opportunistic routing is proven to be very effective in error-prone wireless networks, since it allows *any* node that is closer to the destination to participate in packet forwarding. The overhead that comes with opportunistic routing is, however, a difficult problem to tackle. Our experiments show that opportunistic routing is most relevant when the network is subjected to high and unpredictable interference and traditional routing thus performs badly.

The main contributions in this paper are:

- A presentation of a new hybrid opportunistic protocol (O-CTP), which uses traditional routing when the network is stable and has reasonably little packet loss, but switches to opportunistic forwarding when the network is subjected to interference or jamming.
- An empirical comparison of six routing protocols in an interfered environment using a testbed of 20 TelosB sensing nodes. We employ four different interference patterns and show that O-CTP gives the overall best balance between packet delivery ratio and overhead.

The rest of this article is organized as follows. Section II reviews related work. Section III describes O-CTP in detail. The test and experiment setup is described in section IV. Section V and VI offer experimental results. Finally, in section VII we conclude the article.

II. Related work and background

In this section, we review the prior research addressing the issues of routing in WSNs. We focus primarily on protocols that are implemented and tested in real-world environments.

First, we discuss traditional routing protocols and then we explain different opportunistic alternatives. Finally, we explain why there is room for improvement in WSN routing.

A. Traditional routing

TYMO [9] and NST-AODV [14] both originate from the ideas behind DYMO and AODV, which are protocols tailored to mobile ad-hoc networks. There are three basic problems that arise with these protocols in WSNs. 1) The hop count metric does not provide good performance since it treats all hops as equal. 2) Routes are based on the end-to-end principle, meaning that they are costly both to establish and to maintain in a lossy environment. 3) The protocols do not exploit the fact that most traffic is destined to one node (i.e., the sink).

Convergecast routing protocols are proposed to address the above issues. In convergecast protocols, such as MultihopLQI [10] and Collection Tree Protocol (CTP) [4], all traffic is assumed destined to a single sink node. The sink node constitutes the root in the routing tree. Each node uses a gradient minimization approach to determine the next hop (i.e., its parent). MultihopLQI uses the Link Quality Indicator (LQI) from the physical layer to additively obtain the gradient towards the sink. LQI is proven to be more stable in selecting the best paths than using hop-count [15]. Beaconing (with fixed interval) is used by all nodes to measure LQI and to support changes in the topology. CTP builds on MultihopLQI but distinguishes from it on two central features: 1) It uses the Expected number of transmissions (ETX) as its routing metric as opposed to LQI: Starting with an ETX of 0 at the sink, each node calculates its own ETX as the ETX reported by the parent plus the ETX of its own link to the parent. 2) CTP uses adaptive beaconing by extending the Trickle algorithm [16] to reduce the route repair latency and send fewer beacons when the network is stable. To adapt quickly to topology changes, the trickle timer interval is reset whenever a routing loop is detected or the routing cost decreases significantly.

It is worth noting that NST-AODV, TYMO, MultihopLQI and CTP are implemented in TinyOS and tested in several real WSNs [4], [9], [10], [14].

B. Opportunistic routing

Traditional routing protocols aim to find the optimal paths through a network by daisy-chaining the links with the presumed best qualities. This approach stems from protocols found in fixed infrastructure and is ideal when there are minimal network dynamics. The metric calculations, however, have difficulties coping with the rapid fluctuations in the wireless domain. Consequently, the routing decisions may be based on historic and outdated metrics. Opportunistic routing differs from traditional routing since it exploits, rather than attempting to hide, the broadcast nature of the wireless medium [11]–[13], [17], [18]. In opportunistic routing, a node does not preselect a preferred forwarder according to a set of (possibly outdated) metrics. Instead, opportunistic routing exploits the fact that there might be many potential forwarders in a node‘s vicinity able to receive the broadcast packet. The designated forwarding nodes may differ from one packet to the next. Hence, channel fluctuations are implicitly taken into account since the forwarding decision is carried out while the packet moves through the network.

Various opportunistic routing protocols differ mainly in the way the relay nodes decide on which node should retransmit the packet. In the seminal opportunistic routing protocol ExOR [11], the sender chooses a candidate subset of all its neighboring nodes that could bring the packet closer to its destination. This list is prioritized according to distance and put in the packet header. Each recipient delays a certain time depending on its position in the list before forwarding the packet. LAOR [17] and GeRaF [19] take a similar approach. Other protocols, such as TORP [13] use ETX to choose the candidate subset. MORE [12] relaxes the need to coordinate the forwarding, since the approach combines opportunistic routing with network coding. ORW [20] is a promising opportunistic routing scheme tailored directly to duty-cycled networks and can supplement our work in a future version.

C. Towards a hybrid protocol

Although there are numerous papers that study opportunistic routing analytically or via simulations [11], [12], [17], there are few papers that investigate real-world implementations. The works by Carnley et al. [13], Joe et al. [18] and Landsiedel et al. [20] are rare exceptions. There are also few papers that specifically analyze the trade-off between traditional routing and opportunistic routing. Shah et al. [21] use simulations to conclude that opportunistic routing is superior to geographical routing when the channel quality is low. Carnley et al. [13] show that TORP improves throughput and lowers the overhead compared to CTP in some scenarios.

To the best of our knowledge, this is the first paper that analyzes the trade-off between traditional routing and opportunistic routing in interfered environments. Further, we are the first to provide a routing solution that is based on a hybrid approach.

III. O-CTP: A hybrid Opportunistic Collection Tree Protocol

The hybrid protocol presented in this paper is called Opportunistic Collection Tree Protocol (O-CTP). O-CTP consists of three fundamental parts:

1) The traditional routing part, which is largely based on CTP.
2) An opportunistic routing part, which is employed when traditional routing is no longer effective.
3) A set of *triggers*, which enables switching between traditional routing and opportunistic routing.

Before digging into the protocol specification, it is worth discussing the intuition underlying our protocol design.

A. Why opportunistic routing is a trade-off

It is helpful to consider the simple network presented in Fig. 1. In the network example there are three possible routes from source s to the destination d. The three alternative routes

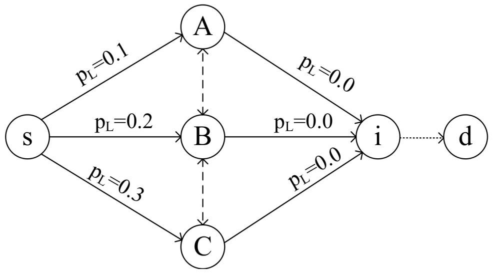

Fig. 1. Opportunistic routing exploits the broadcast nature of wireless networks. Node s does not preselect a preferred forwarder but exploits the fact that there might be many potential forwarders in a node‘s vicinity able to receive the broadcast packet

go via either of the nodes A,B or C to i. The three possible links from s are all subjected to some degree of packet loss varying from 10% to 30%. For the remaining path we assume no packet loss. In the following discussion, we use CTP as an example of a traditional routing protocol. CTP will choose A as the preferred forwarder for s, since choosing A minimizes the overall ETX from s to d. Hence, a packet loss of 10% can be expected for the first hop. *Opportunistic routing* on the other hand, takes a different approach, since it exploits the fact that all transmissions are broadcast. Hence, it does not preselect a single forwarder, but assumes that at least one of the neighbors receives and forwards the packet. In the case in Fig. 1, all the nodes A, B, C are able to receive a broadcast packet from s. The combined packet loss probability for the first hop is now reduced to $0.1 \times 0.2 \times 0.3 = 0.006$, which is a tremendous improvement over the CTP protocol. The performance of CTP is, however, not as depressive as it might first seem, since CTP employs retransmissions (up to 31 times as default). Consequently, the overall delivery rate can therefore be expected to be very close to 100%. Taking in account the retransmissions, the expected cost (transmissions per packet) to reach i using CTP is about 2.11 ($\frac{1}{1-0.1}$ for the first hop and 1.0 for the second).

The basic problem that arises with opportunistic routing is that the forwarding nodes are not necessarily able to hear each other. In our example, B will overhear all retransmissions performed by A or C, and since it is wasteful for B to forward those packets it effectively suppresses duplicate forwarding. But since A can not hear C and vice versa, they will both forward the same packet. Such duplicates are not only wasteful in terms of energy. They also increase the collision probability. Despite much research in reducing duplicates, there is no effective mechanism to eliminate such duplicates entirely [22]. Assume now that each of the nodes A,B,C has a probability of $P_{FA} = P_{FB} = P_{FC} = \frac{1}{3}$ to be the first forwarder and that the opportunistic routing protocol performs retransmissions. The expected cost can be calculated as the sum of the expected number of transmissions for each hop. For the fist hop, the expected number of transmissions is $\frac{1}{1-0.006}$, while the second hop gives $2P_{FA}+P_{FB}+2P_{FC}$. This gives a total cost of 2.67, which exceeds the CTP cost. Since duplicates will occur on the second hop when OR is used, CTP is the most effective protocol in this example.

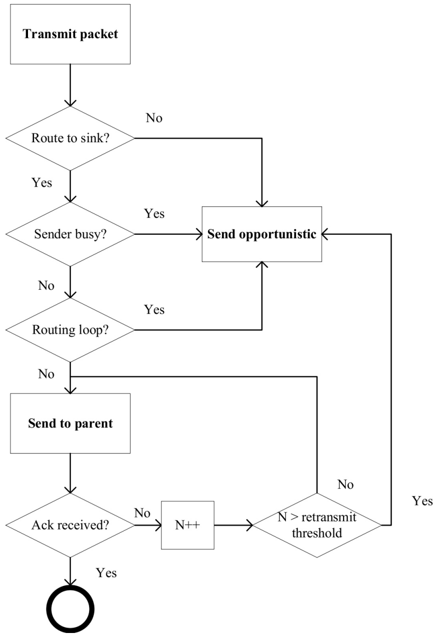

Fig. 2. The basic operation of O-CTP

As previously discussed, the link loss is never stable as in the above example, but fluctuates with time. Imagine now that the packet loss probability on the link $s \rightarrow A$ suddenly increases to 90% due to some interference. CTP will still choose A as its preferred forwarder for some time. The overall cost on the route s to i via A now increases to 11 ($\frac{1}{1-0.9} + 1$). The high number of retransmissions required to achieve 100% delivery rate quickly translates to a huge waste of energy. For opportunistic routing, the situation is practically unchanged, since the packet loss on the first hop is now $0.9 \times 0.2 \times 0.3 = 0.054$ resulting in a total cost of 2.72. Hence, the cost does not increase significantly from the previous situation. In this example, the opportunistic protocol outperforms CTP.

We have now illustrated why traditional routing performs best when the network conditions are fairly good and predictable, while opportunistic routing performs best when the network conditions are poor and unpredictable. Our hypothesis is that a hybrid protocol, which is able to change its operation based on the current network dynamics, could benefit from both of these worlds and give an overall improved performance.

B. When to switch from traditional routing to opportunism?

We decided to build our hybrid protocol based on CTP, since this is the de-facto collection protocol for real-world deployed WSNs and has shown high delivery ratio in previous studies. The basic idea of O-CTP is to switch from CTP operation to opportunism whenever the network is subjected to interference. A best-of-both-worlds protocol is very difficult to construct, since there is no fail-free trigger that allows the protocol to switch to opportunistic routing at the optimal moment. The central component of O-CTP is therefore the triggering part.

The trigger could be built as dependent on cross-layer communication. However, since CTP is built to be independent of layer 1 and layer 2, we decided not to break this hardware-independency by introducing cross-layering. There are, however, some possibilities to monitor the underlying network status directly from the forwarding engine in CTP. We have used these to trigger opportunistic forwarding. This is a distributed decision, and all nodes can decide the forwarding method for its current packet transmission. A switch to opportunistic sending is performed if one of the following situations occur within the CTP routing protocol:

1) *There is no route to the sink (i.e., no parent).* Even if CTP is in a no-route state, there might be many possible routes available that could be used immediately by the opportunistic protocol.
2) *Sender is busy.* Normally, in CTP, the forwarding engine denies packet forwarding if the forwarding layer is busy. However, in this state, packets can still be forwarded opportunistically.
3) *Routing loop detected.* Even if standard CTP has mechanisms to deal with loops, we observed that loops occur very frequently in interfered networks. Since the detection of a loop means that there is a problem somewhere in the routing tree, O-CTP is implemented such that when a loop is detected, the packet is forwarded opportunistically.
4) *The retransmit threshold has expired.* In standard CTP, the forwarding engine gives up packet forwarding when the retransmit threshold expires. In O-CTP, the packet is forwarded opportunistically instead.

Either of the above circumstances indicate that there is a problem with the packet forwarding, which means that opportunism is beneficial. These trigger mechanisms are evaluated empirically in section V. The decision on whether to forward a packet opportunistically or not is memory-less (cf. Fig. 2) and it is not necessary to use a trigger to switch back from opportunistic forwarding to traditional forwarding. In other words, a packet following a previous packet that was forwarded with opportunistic routing, may be forwarded with opportunistic routing or traditional routing depending on the current state of the forwarding engine.

C. The opportunistic part of O-CTP

There are several previous routing protocols that shares salient opportunistic routing features, e.g., ExOR [11], LAOR [17], BRL [23], GeRaF [19] and IGF [24]. Many of the protocols in this category can serve the purpose as the opportunistic routing part of O-CTP. Since none of these opportunistic protocols are publicly available for TinyOS, we implemented our own protocol to validate the hybrid routing approach in O-CTP. Our protocol is a geographic-opportunistic routing protocol (GEOPP) that covers the basic opportunistic principles presented in previous research.

The key difference between various opportunistic protocols is how the forwarding decision is performed. For example, IGF, BRL, and GeRaF, employ RTS/CTS handshaking between the source and the possible forwarders before transmitting the data packet. The motivation behind the RTS/CTS approach is to pre-elect one single forwarder and in this way limit the number of possible duplicates. However, the drawback is that even after a successful RTS/CTS exchange, the probability of successfully receiving a larger data message might be very low [25]. Another method, used by LAOR [17] and ExOR [11], is to specify a list of forwarding nodes in the packet header. The list is sorted in decreasing order of progress towards the sink, and hence, represents the priority of the forwarders. The shortcoming of this approach is that all potential forwarders can not possibly be added to the list since the header size is limited. This limitation can leave some long-progress paths underutilized. Considering the example in Fig. 1, there could be a small possibility that a transmission from s might reach i directly. This opportunity will be left unused if only A, B, C is stated in the forwarding list. Further, if any of the links $s \rightarrow A, B, C$ are downstream unidirectional, they will be left unused since s has no knowledge of them.

Due to our interest in making a working system, we had to trade off some advanced protocol ideas presented in previous research for simpler ones. In GEOPP, there is no RTS/CTS scheme. Neither is there any forwarding list in the packet header. Hence, there can be many possible forwarders receiving the same packet. To make sure that a minimum number of these neighbors forward the packet, each neighbor computes a dynamic forwarding delay (DFD) as in ExOR, depending on its position relative to the sink. The node with a small progress towards the sink computes a higher delay than a node with a large progress. Assuming that all nodes know their own location and the sink location, the DFD is simple to calculate. The node that computed the smallest DFD (i.e., the node which is closest to the sink) forwards first. The other forwarders overhearing this retransmission, stops their DFD-timer and deletes the packet from their forwarding queue. In addition, the node transmitting the packet uses the overheard retransmission as an implicit acknowledge indicating that the packet is undergoing a positive progress towards the sink. If no such implicit acknowledge is heard, the node may choose to retransmit the packet (still opportunistically) up to a predefined number of times. Notice that the problem with most geographical routing protocols is that packets can be routed to a dead-end, where there is no neighbor closer to the destination. The aim of this paper has not been to attempt to solve this problem, and GEOPP therefore lacks a solution for the dead-end problem. Although this issue should be investigated, we do not consider it as a big problem here since GEOPP is a fallback solution used only when CTP fails.

Since the forwarding area in GEOPP covers all nodes with a positive progress towards the sink, GEOPP can expect a high delivery ratio but also a relatively high cost compared to some of the other opportunistic routing protocols due to more duplicated packets. Finally, even if GEOPP is presented here as an integral part of O-CTP, it is, as shown in the empirical analysis later in the paper, possible to run the protocol stand-

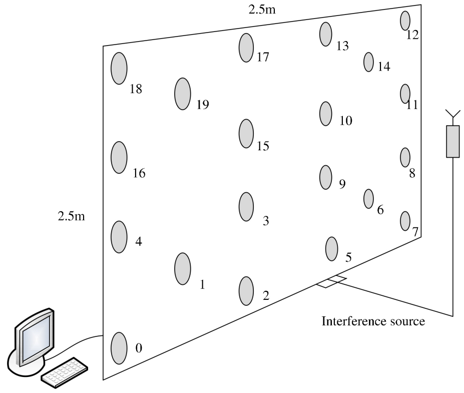

Fig. 3. The testbed consists of 20 TelosB sensing nodes and a 2.4GHz software controlled interference source. Node 18 is the sink collecting all information.

alone as a pure opportunistic protocol.

IV. Materials and methods

A. testbed

To evaluate the performance of different routing protocols in a realistic setting, we implemented a real testbed (cf. Fig. 3). The testbed consisted of 20 TelosB sensing nodes [26] covering an area of $2.5 \times 2.5m^2$. The TelosB has a 4 MHz MSP430 processor, 10 KB of RAM and 48KB program memory. TelosB uses the Chipcon CC2420 radio in the 2.4GHz band, an IEEE 802.15.4 compatible radio with O-QPSK modulation with DSSS at 250kbps. The output power was set to -25dBm, which gave a multihop network with an average node degree of 6. The nodes were connected to a standard laptop using a combination of USB cables and hubs. This USB backbone was used for reprogramming and debugging. Node 18 was the designated sink, forwarding packets to the computer over USB.

TABLE I
The four different interference patterns employed in the experiments and the resulting average packet loss

	Interference pattern			
	No	Low	Medium	High
Dutycycle (T_{on},T_{off})	0,∞	10s,60s	10s,30s	20s,30s
Avg packetloss	2%	13%	23%	33%

Network interference can come from various sources. To allow interference in a controlled fashion, we used an ATT Q30 2.4GHz signal jammer, which was placed 1m from the testbed surface (cf. Fig. 3). Our goal was to introduce realistic interference and not complete jamming, and the jammer-antennas were therefore equipped with 20dB damping. Since most interference sources (be it radar, video links or 802.11) are transient, we used duty cycling of the signal jammer controlled from software for the experiments. This approach enabled both realistic and reproducible results. By employing different interference patterns, from continuously off to increasingly more aggressive interference, we could manipulate the packet loss in the network in a predictable manner. Typical packet losses for communication from the sensing nodes towards the sink for the different interference patterns are presented in Table I.

B. Protocols

For the purpose of the experiments in this paper, O-CTP was implemented for TinyOS 2.x. In our empirical study, we compare O-CTP with the most prevalent routing protocols for WSNs: CTP [4], MultihopLQI [10] and TYMO [9]. We use the default parameter setting for all three protocols. We also compare with the pure opportunistic protocol GEOPP, and with naïve broadcast (BCAST). Our BCAST implementation works as follows: Message originators send broadcast packets. A node hearing a BCAST transmission, records the sequence number and the originator (to avoid duplicate retransmissions) and retransmits the packet. Eventually, the packet reaches its destination (i.e., the sink). BCAST can be seen as the simplest routing protocol available. Since it also can be categorized as opportunistic (it uses multiple forwarding nodes), it serves well as a baseline for comparison in our study.

V. Analyzing O-CTP triggers

To obtain valuable understanding of O-CTP, we first investigate the triggers initiating opportunistic forwarding. Table II shows the relationship between the traffic sent with opportunistic routing and the traffic sent with traditional routing when the network is exposed to different interference patterns. Further, the table shows the fraction of the opportunistic routing traffic directly traced to each trigger. In this experiment, the retransmit threshold was set to 3. For each of the interference settings, we ran 10 experiments lasting one hour each. As shown, the share of the opportunistic data traffic increases with increasing interference. Another observation is that the expiration of the retransmit threshold contributes to most of the opportunistic data traffic. The other incidents (i.e., no parent, sender is busy, routing loop) do not occur very often. In practice, the retransmit threshold is the critical parameter in optimizing the performance of O-CTP and manipulating this threshold is the logical next step in the investigation.

TABLE II
The amount ot traffic transmitted opportunistically (for each trigger) and using traditional routing

Opportunistic	Interference pattern			
trigger	No	Low	Medium	High
No parent	5.5%	6.9%	6.8%	8.2%
Sender busy	0%	1.1%	0.4%	0%
Loop	0%	1.6%	1.7%	3.4%
RTX expired	2.9%	11.8%	18.7%	27.5%
None (traditional routing)	91.6%	78.6%	72.4%	60.9%

Fig. 4 shows the effect of manipulating the retransmit threshold on the delivery ratio. We ran one one-hour experiment for each retransmit threshold between 1-40 for each interference setting - a total of 160 experiments. The astute reader can notice some small irregularities in the results in Fig. 4. They are natural, since we ran only one experiment per data point. Despite this fact, the trends are clear. When

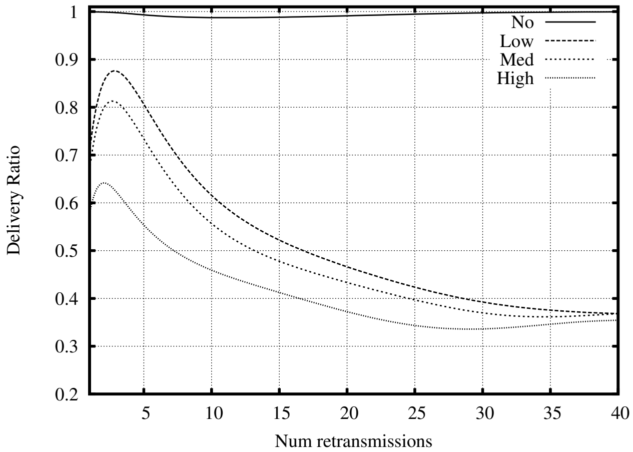

Fig. 4. Delivery ratio for O-CTP with different retransmit thresholds

there is minimal interference, the retransmit threshold setting is not crucial. The default setting in CTP is rather high (31). This is reasonable, since in a sink-routed tree, the next packet in the queue has the same destination as the current packet (i.e., the sink). Consequently, the outcome of transmitting the next packet in the queue will be the same as the current one [4]. For O-CTP, however, a high retransmit threshold for the traditional routing part is not beneficial for two reasons. *First*, a high number of retransmissions indicate that there is a problem with interference, meaning that the packet delivery could have been improved by switching to opportunism at an earlier stage. *Second*, retransmitting a packet several times puts a high load on the network. This can influence other on-going transmissions, which again increases contention and collisions. We also experienced that the probability for creating routing loops increased with increased retransmit threshold. A late switch to opportunism in a saturated and interfered network (with possible loops) gives no improvement for packet delivery. Based on the results shown in Fig. 4 the retransmit limit for traditional routing was set to 3 (triggering opportunistic forwarding) in the subsequent experiments. For GEOPP, we remember that the retransmission function is based on listening to implicit acknowledgements. Since these acknowledgements are unreliable (requiring symmetric links), incrementing the retransmission threshold therefore increases the cost as well. Retransmissions also contribute to more duplicate packets in the network. We observed that a high retransmit threshold setting for the opportunistic routing protocol indeed improves packet delivery, but the cost of bringing the delivery rate close to 100% could be extremely high in an interfered network. For the subsequent experiments, the retransmit threshold for opportunistic routing was set to 2 to balance reliability and cost.

VI. Routing protocol comparison

In this section we evaluate O-CTP using two empirical experiments. The first experiment investigates how the three protocols O-CTP, CTP and pure opportunism (GEOPP) react to interference. In the second experiment we study O-CTP against five routing protocols in various interference scenarios.

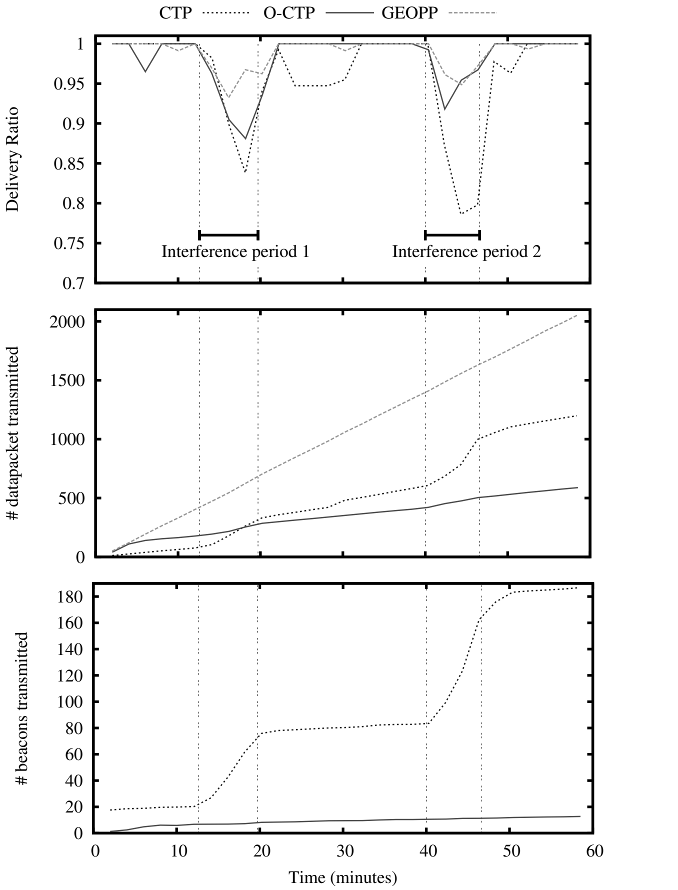

Fig. 5. The delivery ratio and overhead of CTP, O-CTP and GEOPP when the network is under medium interference.

In comparing the protocols, three key performance metrics are evaluated. 1) *Packet delivery ratio* – which is defined as the number of packets received (duplicates not included) divided by the number of application packets transmitted, 2) *the number of data packets transmitted* – which gives a picture on the number of retransmissions and duplicates created by the protocol, 3) *the number of beacon messages transmitted* – which is the overhead of maintaining the routing protocol tables.

A. O-CTP related to CTP and pure opportunism

First, we perform an experiment with mixed interference. For the experiment, we have used the testbed setup explained previously. We ran CTP, O-CTP and GEOPP (isolated) on the testbed for one hour. The packet rate was fixed at one packet per node per 20s, which represents a typical medium duty cycle sensor network. Between 12-19 and 40-46 minutes, we ran the signal jammer with the medium interference pattern. The rest of the test period elapsed without any interference.

Fig. 5 shows the delivery ratio averaged each 2 minutes. In the periods without interference, the delivery ratio is close to 100% for all three protocols. During interference, all protocols

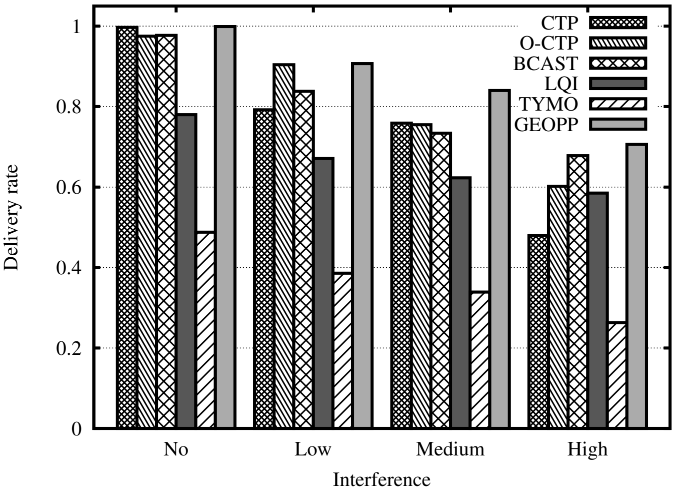

Fig. 6. The delivery ratio on different interference patterns

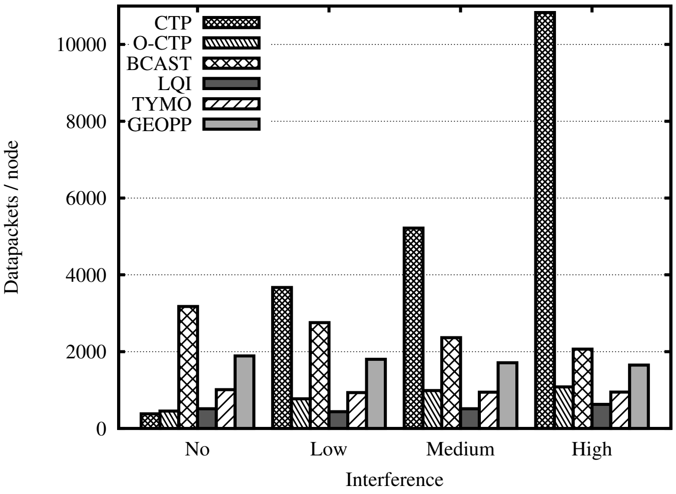

Fig. 7. Datapackets transmitted by each node on different interference patterns

are affected. CTP lose most packets, O-CTP is affected to a lesser extent and GEOPP loses the fewest packets. This is in compliance with our previous analysis. The same figure also shows the cumulative number of data packets and beacon packets transmitted per node. CTP increases both the number of beacon packets and the number of data packets during the interference period. One part of the data packet increase is traced to a rise in the number of retransmissions, and one part is caused by routing loops, which are inevitable when the parent change rate increases. The data packet rate of O-CTP changes slightly during interference since the number of opportunistic transmissions increases. For GEOPP, the data packet rate is stable during the test period. Notice that CTP transmits more data packets than O-CTP even during the non-interference time. Even though our jammer is turned off during this period, weak links in the network can occur, leading to packet retransmissions or loops. In such cases O-CTP performs better. It is important to note that it is possible to reduce the overhead of CTP significantly by altering the routing parameters. By increasing the minimum trickle interval from 64ms to 30000ms and reducing the number of retransmissions from 31 to 3, we were able to reduce the overhead to almost $\frac{1}{10}$ of the numbers presented in Fig. 5. However, the major disadvantage was that the delivery ratio was reduced with 15-20%, so this setting can not be recommended.

In the comparison, O-CTP presents excellent packet delivery ratios (albeit lower than GEOPP) and it clearly has the lowest overhead. In the next section we measure the performance of O-CTP under a wider range of conditions, and compare with an extended set of routing protocols.

B. *Comparing six routing protocols*

The routing protocols we consider here are CTP, O-CTP, BCAST, MultihopLQI (LQI), TYMO, and GEOPP. Each routing protocol is tested for one hour for each interference setting (i.e., ”no”, ”low”, ”medium”, and ”high”), repeated ten times and the results are averaged.

Fig. 6 shows the packet delivery ratio for each routing protocol and for each interference setting. Let us first focus on the situation without any external interference. We observe that the packet deliveries for CTP, O-CTP, BCAST and GEOPP are very similar. Compared with these, LQI loses about 20% more packets and TYMO about 50% more packets. When increasing the interference from ”no” to ”low”, CTP and BCAST loses 10-15% more packets than O-CTP. By increasing the interference further, BCAST and GEOPP (pure opportunistic routing) show the best performance, while CTP seems to be very sensitive to high interference. This observation is in compliance with our previous analysis. In all cases LQI and TYMO are outperformed by O-CTP, BCAST and GEOPP.

Fig. 7 shows the average number of data packets transmitted per node during the test. The first observation is that CTP is very effective when there is no interference. This shows that the ETX routing works excellent as long as the links are stable. However, even with low interference, CTP has a vast overhead, which increases tremendously when the interference increases. The rise is caused by CTP’s quick reaction to topology changes, which increases the parent change rate and again increases the probability for routing loops. Interestingly, BCAST is more efficient than CTP in interfered environments. Our hybrid protocol, O-CTP, shows higher overhead than CTP in the ”no interference”-setting. This is due to the fact that a fraction of the traffic is sent opportunistically (see table II), with unavoidable duplicates. When there is much interference, the hybrid protocol sends an even larger part of the traffic opportunistically, and this is also reflected by the overhead. Nevertheless, the overhead with standard CTP is higher with one order of magnitude. The hybrid approach also reduces the overhead with 50-80% compared to pure opportunism. In all cases, LQI demonstrates much lower data packet load than the other protocols; however, it comes at a price, since the delivery ratio is significantly reduced (cf. Fig. 6).

For CTP, the number of beacon messages increases tremendously even with little interference (cf. Fig. 8). The problem

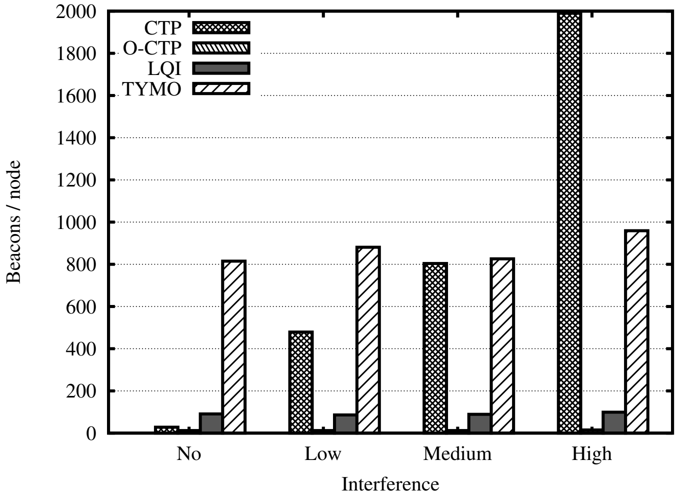

Fig. 8. Beacons transmitted by each node on different interference patterns

worsens when adding more interference. This phenomenon is mainly caused by the trickle timer controlling the beacon interval, which is reset to a (default) 64ms interval whenever a parent is lost or a neighbor node detects a topology problem. TYMO shares the high overhead problem, albeit its cause is different. One reason is that TYMO floods the entire network in order to find the route to the sink; a process that is performed very often. Another reason is that TYMO is not capable of constructing routes over asymmetric links. Compared with Fig. 7, we see that the number of beacon packets and data packets combined for TYMO, surpasses the number of data packets for BCAST. Although we have only tested one testbed size, there is no reason to believe that TYMO is better than BCAST for larger networks. O-CTP shows stable beacon results regardless of the network environment. Obviously, for BCAST and GEOPP there is no routing traffic, since both protocols are beaconless.

C. Discussion of the results

It is worth discussing our results compared to other studies on real WSNs. TYMO performed badly in all our experiments, which complies well with results from other recent studies [7], [27]. Nevertheless, we believe that there might be room for improvement by taking advantage of some more advanced AODV-features. CTP and MultihopLQI have been studied numerous of times recently [2], [4], [28]. Most studies conform to our conclusion that CTP has overall better packet delivery than MultihopLQI. The work by Gnawali et al. [28] is the only one studying CTP under interference. However, in our setup, CTP showed much higher overhead than the results presented in their paper. Carnley et al. [13] and Landsiedel et al. [20] support our finding that opportunism can indeed outperform CTP.

VII. Conclusions and future work

Radio interference or deliberate jamming attacks can cause highly unpredictable communication in WSNs. While advancements in hardware design and MAC protocols can improve packet delivery, we have investigated a simpler approach using hybrid opportunistic techniques on the routing layer. Our hybrid protocol (O-CTP) is designed by combining the high packet delivery ratio of opportunistic routing in error-prone wireless networks, and the energy efficiency of traditional routing in stable networks. In the paper we used a real testbed and showed that O-CTP improves both packet delivery and system lifetime in an interfered network compared to five other protocols.

There is still a huge potential for improvement of O-CTP. Future works include improvements in the trigger (e.g., using cross-layering) making the protocol react faster to interference, and techniques to reduce the number of duplicate packets. Further, the protocol should incorporate the challenges posed with duty-cycled sensing nodes.

References

[1] J. Zhao and R. Govindan, “Understanding packet delivery performance in dense wireless sensor networks,” in *Sensys ’03*. ACM, 2003, pp. 1–13.

[2] S. Lin, G. Zhou, K. Whitehouse, Y. Wu, J. Stankovic, and T. He, “Towards stable network performance in wireless sensor networks,” in *RTSS’09*. IEEE, 2009, pp. 227–237.

[3] T. Liu, A. Kamthe, L. Jiang, and A. Cerpa, “Performance Evaluation of Link Quality Estimation Metrics for Static Multihop Wireless Sensor Networks,” in *IEEE SECON ’09*, june 2009, pp. 1 –9.

[4] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol,” in *SenSys ’09*. ACM, 2009, pp. 1–14.

[5] V. Shnayder, B. Chen, K. Lorincz, T. Fulford-Jones, and M. Welsh, “Sensor networks for medical care,” in *SenSys 05*, 2005, pp. 314–314.

[6] A. Wood, J. Stankovic, and G. Zhou, “DEEJAM: Defeating energy-efficient jamming in IEEE 802.15. 4-based wireless networks,” in *SECON’07*. IEEE, 2007, pp. 60–69.

[7] 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 *2010 IEEE SUTC*, 2010, pp. 189–196.

[8] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, “A survey on jamming attacks and countermeasures in WSNs,” *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 4, pp. 42–56, 2009.

[9] R. Thouvenin. TYMO source code. [Online]. Available: http://www.tinyos.net/tinyos-2.x/tos/lib/net/tymo/

[10] G. Tolle. MultihopLQI source code. [Online]. Available: http://www.tinyos.net/tinyos-2.x/tos/lib/net/lqi/

[11] S. Biswas and R. Morris, “Opportunistic routing in multi-hop wireless networks,” *ACM SIGCOMM’04*, vol. 34, no. 1, pp. 69–74, 2004.

[12] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, “Trading structure for randomness in wireless opportunistic routing,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 169–180, 2007.

[13] J. Carnley, B. Sun, and S. Makki, “TORP: TinyOS Opportunistic Routing Protocol for Wireless Sensor Networks,” in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*. IEEE, pp. 111–115.

[14] C. Gomez, P. Salvatella, O. Alonso, and J. Paradells, “Adapting AODV for IEEE 802.15.4 Mesh Sensor Networks: Theoretical Discussion and Performance Evaluation in a Real Environment,” in *WOWMOM ’06*, 2006, pp. 159–170.

[15] P. Swieskowski and G. Werner-Allen, “Improving the Performance of a Data Collection Protocol,” *Division of Engineering and Applied Sciences, Harvard University*, 2005.

[16] P. Levis, N. Patel, D. Culler, and S. Shenker, “Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks,” in *Proc. of the USENIX NSDI Conf.*, 2004, pp. 2–2.

[17] X. Yang, J. Yin, and S. Yuan, “Location-aided opportunistic routing for mobile ad hoc networks,” in *WiCom’09*. IEEE, 2009, pp. 1–5.

[18] I. Joe and D. Kim, “An opportunistic routing protocol for underground wireless sensor networks,” in *SNPD’09*. IEEE, 2009, pp. 602–605.

[19] M. Zorzi and R. Rao, “Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance,” *IEEE transactions on Mobile Computing*, pp. 337–348, 2003.

[20] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, "Low power, low delay: opportunistic routing meets duty cycling," in *Proceedings of the 11th international conference on Information Processing in Sensor Networks*. ACM, 2012, pp. 185–196.

[21] R. Shah, S. Wietholter, A. Wolisz, and J. Rabaey, "When does opportunistic routing make sense?" in *PerCom 2005*. IEEE, 2005, pp. 350–356.

[22] A. Triviño-Cabrera, "Survey on Opportunistic Routing in Multihop Wireless Networks," *IJCNIS*, vol. 3, no. 2, 2011.

[23] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wälchli, "BLR: beacon-less routing algorithm for mobile ad hoc networks," *Computer communications*, vol. 27, no. 11, pp. 1076–1086, 2004.

[24] B. Blum, T. He, S. Son, and J. Stankovic, "IGF: A state-free robust communication protocol for wireless sensor networks," *Tech Rep. Dept Comp. Sci, Univ of VA*, 2003.

[25] J. Sanchez, P. Ruiz, and R. Marin-Perez, "Beacon-less geographic routing made practical: challenges, design guidelines, and protocols," *Communications Magazine, IEEE*, vol. 47, no. 8, pp. 85–91, 2009.

[26] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *IPSN'05*. IEEE Press, 2005, pp. 48–es.

[27] J. Lee, B. Kusy, T. Azim, B. Shihada, and P. Levis, "Whirlpool routing for mobility," in *MobiHoc'10*. ACM, 2010, pp. 131–140.

[28] O. Gnawali, L. Guibas, and P. Levis, "A case for evaluating sensor network protocols concurrently," in *WINTECH'10*. ACM, 2010, pp. 47–54.